

Design of a White Lab

Component Vending Machine

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

October 17, 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(s) of other people has been attributed, and has been cited and referenced.
3. This report is my, Baden David Morgan, 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:
October 17, 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

Abstract

Contents

| | | |
|----------|---|----------|
| 1 | Introduction | 1 |
| 1.1 | Subject and motivation for the Research | 1 |
| 1.2 | Background to the Research | 1 |
| 1.3 | Objectives of this Research | 1 |
| 1.3.1 | Problems to be investigated | 1 |
| 1.3.2 | Purpose of this study | 1 |
| 1.4 | Scope and Limitations of the Research | 1 |
| 1.5 | Plan of Development | 1 |
| 1.5.1 | Note For Reader | 1 |
| 2 | Literature Review | 2 |
| 2.1 | Similar Work | 2 |
| 2.1.1 | Portland State University Vending Machine | 2 |
| 2.1.2 | Medication Dispenser, University of Tasmania | 3 |
| 2.1.3 | Medication Dispenser, Narcotic Rehabilitation | 4 |
| 2.1.4 | 3D Adjustable cavity Medication Dispenser | 5 |
| 2.2 | Communication between devices | 6 |
| 2.2.1 | I2C | 6 |
| 2.2.2 | RS-485/TIA-485 | 7 |
| 2.3 | RFID Reader | 8 |
| 2.3.1 | System Specification | 8 |
| 2.3.2 | UCT's RFID Solution | 8 |
| 2.4 | Website, Hosting and Server | 9 |
| 2.4.1 | The Web server: Apache vs. Nginx | 9 |
| 2.4.2 | The Scripting Language: PHP | 10 |
| 2.4.3 | The Database: MySQL | 10 |
| 2.5 | ATX Specification | 10 |

| | |
|---|-----------|
| 3 System Specifications | 12 |
| 3.1 Mechanical Specifications | 12 |
| 3.1.1 DIP Specifications | 12 |
| 3.1.2 Delivery Mechanism | 13 |
| 3.1.3 Enclosure | 13 |
| 3.2 PCB Specifications | 13 |
| 3.2.1 Raspberry Pi | 14 |
| 3.2.2 MCU Modules | 14 |
| 3.3 Software Specifications | 14 |
| 3.3.1 Master Program for the Raspberry Pi | 14 |
| 3.3.2 MCU Module | 14 |
| 3.3.3 Website | 15 |
| 4 Design and Prototyping Methodology and Procedure | 16 |
| 4.1 Design | 16 |
| 4.1.1 Mechanical Design Methodology | 16 |
| 4.1.2 Circuit Design Methodology | 16 |
| 4.1.3 Software Design Methodology | 17 |
| 4.2 Prototyping Methodology and Procedure | 17 |
| 5 Component and Material Selection | 18 |
| 5.1 Communications bus | 18 |
| 5.2 Servo Motor | 18 |
| 5.3 Stepper motor and Driver | 18 |
| 5.4 Power Supply | 19 |
| 5.5 Microcontroller | 19 |
| 5.6 Delivery Mechanism Housing | 19 |
| 5.7 Enclosure Frame | 19 |
| 5.8 Enclosure Casing | 19 |
| 5.9 Enclosure Shelving | 20 |

| | |
|---|-----------|
| 6 System Design and Prototyping | 21 |
| 6.1 Guide For Component Tubes | 21 |
| 6.1.1 IC Guide Version 1 | 21 |
| 6.1.2 IC Guide Version 2 | 22 |
| 6.1.3 IC Guide final | 22 |
| 6.2 Delivery Mechanism | 23 |
| 6.2.1 Vertical Roller Version 1 | 23 |
| 6.2.2 Prototyping Vertical Roller Version 1 | 24 |
| 6.2.3 Vertical Roller Version 2 | 25 |
| 6.2.4 Prototyping Vertical Roller Version 2 | 25 |
| 6.2.5 Horizontal Roller Version 1 | 26 |
| 6.2.6 Prototyping Horizontal Roller Version 1 | 27 |
| 6.2.7 Horizontal Roller Final Version | 28 |
| 6.2.8 Untested design: Feed Up | 30 |
| 6.3 Enclosure | 30 |
| 6.4 Enclosure Redesign | 32 |
| 6.5 RFID Coil Cover | 33 |
| 6.6 PCB Design | 33 |
| 6.6.1 Raspberry Pi HAT | 33 |
| 6.6.2 MCU Module | 34 |
| 6.6.3 LCD Breakout Board | 35 |
| 6.7 Software Design | 36 |
| 6.7.1 Software overview | 36 |
| 6.7.2 API | 37 |
| 6.7.3 Web design | 38 |
| 6.7.4 Raspberry Pi Master Program | 39 |
| 6.7.5 Microntroller Software | 40 |
| 7 System Assembly | 43 |
| 7.1 Software Assembly: Server | 43 |

| | | |
|-----------|---|-----------|
| 7.2 | Enclosure Assembly | 43 |
| 7.3 | PCB Assembly | 44 |
| 7.3.1 | Pi HAT | 45 |
| 7.3.2 | MCU Module | 46 |
| 7.3.3 | LCD Breakout Board | 48 |
| 7.4 | Delivery Mechanism Assembly | 48 |
| 8 | Results and Discussion | 50 |
| 9 | Conclusion | 51 |
| 10 | Recommendations | 52 |
| A | Research Mind Map | 55 |
| B | Detailed Drawings | 56 |
| B.1 | IC Guide DIP Components | 56 |
| B.2 | IC Guide DIP Holder | 57 |
| B.3 | Vertical Roller & Housing Version 1 & 2 | 58 |
| B.4 | Horizontal Roller & Housing Version 1 | 59 |
| B.5 | Horizontal Roller & Housing Final Version | 60 |
| B.6 | Horizontal Roller Final Version | 61 |
| B.7 | RFID Coil Cover | 62 |
| C | PCB Diagrams | 63 |
| C.1 | PI HAT Diagram | 63 |
| C.2 | MCU Module Diagram | 64 |
| C.3 | LCD Breakout Diagram | 65 |
| D | Drawing Templates | 66 |
| D.1 | Material Request Forms | 66 |
| E | Website | 73 |
| E.1 | Components Store Page | 73 |

| | | |
|----------|--|-----------|
| E.2 | Contacts Page | 75 |
| E.3 | Laser Cutting Templates | 76 |
| E.3.1 | DIP 8 Template | 76 |
| E.3.2 | DIP 14 or 16 Template | 77 |
| E.3.3 | DIP 20 Template | 78 |
| E.4 | Drawings for Assembly: Enclosure | 79 |
| F | Manuals and Guides | 80 |
| F.1 | Programming Guide | 80 |
| G | Terminal Commands | 81 |
| H | Ethics Forms | 83 |

List of Figures

| | | |
|------|--|----|
| 2.1 | The Portlands State Universitys' component Vending Machine and how components are presented [2]. | 2 |
| 2.2 | Proposed design of medication dispenser from the group at UT [3]. | 3 |
| 2.3 | Dispensing mechanism for Medication dispensor for narcotic rehabilitation with cover removed [4]. | 4 |
| 2.4 | Picture of an (a) isometric and (b) cross sectional view of the tablet bucket and its different compartments. [5]. | 5 |
| 2.5 | Picture of the 3DAC with the rods (a) unextended and (b) extended. [5]. | 5 |
| 2.6 | An illustration of the dispensary action of the 3DAC. [5]. | 6 |
| 2.7 | External Low Pass Noise Filter Circuit[6] | 7 |
| 2.8 | When idle one can see the line is held high, this is seen before data is sent and after [8]. | 7 |
| 2.9 | The RDM 6300 Module with an external antena connected directly to the module [13]. | 9 |
| 2.10 | Graph showing active websites and their backend web server being used [17]. | 10 |
| 2.11 | ATX 20 pin connection layout [22]. | 11 |
| 3.1 | Mind map used to help assist the definition of the topic, system specifications and design of the Vending Machine. | 12 |
| 3.2 | Two tubes of components, the top full of DIP IC Holders and the bottom one being full of DIP 8 Components. | 13 |
| 6.1 | IC Guide (a) detailed diagram and (b) isometric view. | 21 |
| 6.2 | Comparison between Version 1 and Version 2 of the IC Guide. | 22 |
| 6.3 | Comparison between Version 1 and Version 2 of the IC Guide. | 22 |
| 6.4 | Rectangular IC guide for DIP IC holder tubes. | 23 |
| 6.5 | Vertical Roller and Housing Delivery mechanism. | 23 |
| 6.6 | Vertical Roller (a) cross section (b) isometric view (c) view to help explain loading. . | 24 |
| 6.7 | Prototype of Delivery mechanism using the Vertical roller. | 25 |
| 6.8 | Common jam action that occurs with vertical roller. | 25 |
| 6.9 | Changes made to the Vertical Roller for version 2. | 26 |
| 6.10 | Horizontal Roller and Housing Delivery mechanism. | 26 |
| 6.11 | Labeled isometric view of the Horizontal roller. | 27 |
| 6.12 | Picture of the prototype Delivery mechanism with the horizontal roller. | 28 |

| | | |
|------|---|----|
| 6.13 | Final design of the horizontal roller and housing. | 28 |
| 6.14 | Comparison between version 1 and final version of horizontal roller | 29 |
| 6.15 | All rollers to dispense DIP 8, 14 or 16 and 20 for comparison. | 29 |
| 6.16 | Feed up design with cutaway and top view to illustrate how it operates. | 30 |
| 6.17 | Frame of the enclosure illustrating the upper and lower compartments. | 31 |
| 6.18 | A detailed drawing of the sloped plate. | 31 |
| 6.19 | Isometric and top view of the enclosure with the case on interior. | 32 |
| 6.20 | Isometric view of the redesign. | 32 |
| 6.21 | Isometric back and front views of the RFID coil cover. | 33 |
| 6.22 | PCB layout of the Raspberry Pi HAT, top and bottom layer. | 34 |
| 6.23 | PCB layout of the MCU Module, top and bottom layer. | 35 |
| 6.24 | LCD breakout board PCB layout, Bottom layer with silkscreen of top layer overlayed. | 35 |
| 6.25 | LCD breakout board PCB layout, Bottom layer with silkscreen of top layer overlayed. | 36 |
| 6.26 | LCD breakout board PCB layout, Bottom layer with silkscreen of top layer overlayed. | 39 |
| 6.27 | State Machine Diagram for MCU software. | 41 |
| 6.28 | A simplified logic diagram of the task manager and how it will function. | 42 |
| 7.1 | Pictures of the enclosure frame with the walls bolted on. | 44 |
| 7.2 | Pictures of the enclosure frame with the walls bolted on. | 45 |
| 7.3 | Pictures of the PiHAT PCB by itself, on the left, on the Raspberry Pi, on the right. | 46 |
| 7.4 | Pictures of the MCU Module PCB, with the LCD plugin on the left, without the pcb plugin on the right. | 47 |
| 7.5 | Pictures from the etching process to make the LCD breakout baord. | 48 |
| 7.6 | Picture of the 3D printed IC guide and roller after assembly. | 49 |
| 7.7 | Picture the housing structure for the delivery mechanism. | 49 |

List of Tables

| | | |
|-----|---|----|
| 3.1 | DIP package dimension for varies components measured in millimeter. | 13 |
| 6.1 | API used to communicate from master to slave. | 37 |
| 6.2 | API used to communicate from slave to master. | 38 |
| 7.1 | Bill of materials for the Server. | 43 |
| 7.2 | Bill of materials for the enclosure. | 44 |
| 7.3 | Bill of materials for the Pi HAT. | 46 |
| 7.4 | Bill of materials for the MCU module. | 47 |
| 7.5 | Bill of materials for the LCD breakout board. | 48 |
| 7.6 | Estimated Bill of Materials for the Delivery mechanism. | 50 |

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

1.2 Background to the Research

1.3 Objectives of this Research

1.3.1 Problems to be investigated

1.3.2 Purpose of this study

1.4 Scope and Limitations of the Research

1.5 Plan of Development

1.5.1 Note For Reader

For convenience to the reader, if this document is being read on a computer or digital device that support 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. Also the author would like to thank you personally for taking the time to read this report and hopes you will enjoy it.

2 Literature Review

Investigations into similar work to the topic of this project, dispensers of small objects, was done in order to gain more insight into the project and better understand how a design of this caliber should operate. The reviews were done so that similar mistakes could be mitigated and 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 needs of this project effectively.

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 is dispensed at a time. This drew parallels to the White Lab 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 retro-fitted 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 package these bags with varous components. Below is a picture of what is inside the vending machine [1, 2].



Figure 2.1: The Portlands State University's component Vending Machine and how components are presented [2].

A major advantage of Portland State University's solution is that it is not component specific. Since a vending machine can have multiple racks with multiple rails on each rack, the capacity of components

can be very large. Although this is a good solution to the problem and up scaling an old Vending Machine is a good use of resources, this solution has many drawbacks.

The loading solution is to have multiple volunteers put packages together and manually load each rail of the vending machine. This requires multiple individuals input to reload a rail which could delay restocking the machine. Since packages are pre-packed students don't have a choice to smaller quantities and it being a vending machine, although it is based on a payed model, the access is not regulated so one person could empty a rail in one day if they choose. Overall This is a good solution but a more targeted solution is needed for the UCT Vending Machine.

2.1.2 Medication Dispenser, University of Tasmania

A group from University of Tasmania [3] proposed a solution for medication dispensers that would be installed at a patients house. The device is to be connected to the internet and the dispensing action would be initiated by the physician overseeing the patient whose home the device is in. Below is a picture of how the device would operate mechanically.

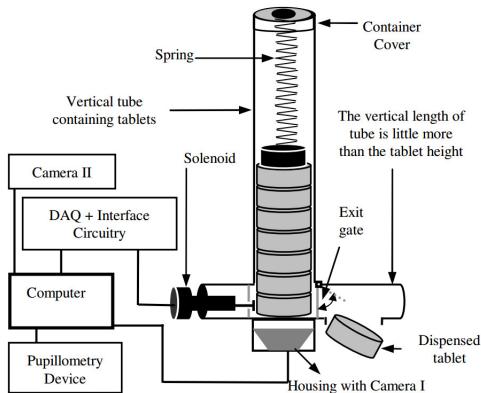


Figure 2.2: Proposed design of medication dispenser from the group at UT [3].

The dispenser is fed by a gravity, spring assisted, magazine of medication, which is labeled "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 hinders this capability. The spring would limit the number of tablets to the length of compression, and so 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 effect the reliability of the magazine feed and determine how many tablets would be able to fit into a single load of the magazine.

To dispense the tablets a solenoid is placed at the bottom of the vertical tube and is actuated to dispense one tablet. The solenoid is activated via an "interface circuitry" which in turn is controlled by a DAQ receiving commands via an internet connection. This is similar to the design requirements to the White Lab Vending Machine. A physician controls when a tablet is dispensed. One issue with the device is that is doesn't have any sensors depicted that would detect a jam or empty magazine. Although a camera is shown this is to get visuals for the physician of the patient the tablet is being dispensed to.

Although the device doesn't have any sensors for jamming or empty magazine detection having it connected to the internet allows for logging if the device is reliable. The physician can keep track and moderate dosages as they see fit. This seems ideal in a medical application however would not work for a Component Dispenser. A useful idea from this medication dispenser is using a gravity fed system allowing for a great load of components reducing the time between every reload of the device.

Having it connected to the internet is also as is in the medication dispenser.

2.1.3 Medication Dispenser, Narcotic Rehabilitation

The second medication dispenser reviewed was one intended for narcotic rehabilitation [4]. It has many features similar the those needed from the White Lab Vending Machine Including single tablet dispensing and control via the internet.

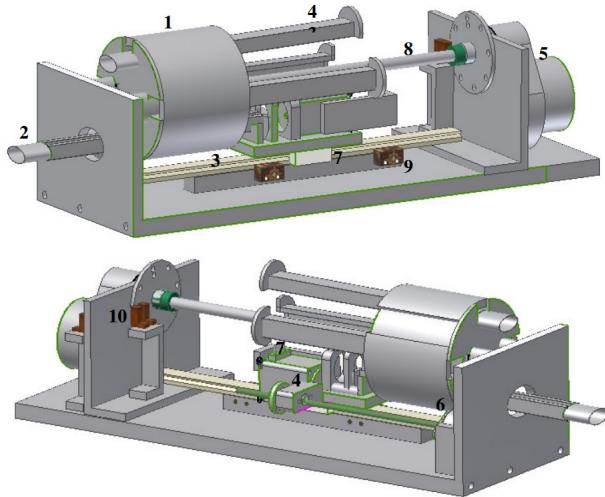


Figure 2.3: Dispensing mechanism for Medication dispensor for narcotic rehabilitation with cover removed [4].

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 Vending Machine. Depicted in Figure 2.3, the mechanism is 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), are used to determine the position of the bottom rod an 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 is blocked and re-appears. The mechanism would then be situated in a strong housing to prevent tamper, and if tamper did occur it would be obvious due to damage needed to open the housing. This dispensary mechanism is controlled by a microcontroller communicating with a computer, which in turn, is connected to the internet.

The cylinder provides a reliable delivery mechanism but limits the amount of stock of the tablets that can be loaded in the dispenser. This may be a design choice by the people who designed it as patients being rehabilitated shouldn't have access to medication in large amounts even if it is protected. This reliable method of using a cylinder or a wheel can be advantageous as the design rotates 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 is avoided by using sensors to detect the extension of the rod. This thinking of preventing the device from causing damage to itself by effectively using sensors and programming should be implemented in the Vending Machine.

Another interesting approach used for the programming is the compartmentalization of tasks. The microcontroller acts on its own with all the programming to do a complete dispensary action if needed but, receives commands to do so by a computer connected to the internet. This method makes each individual device act reliably, like the microcontroller and the computer, and enables the system as a whole to be more reliable. This kind of programming style should be used for the Vending Machine to

make it work effectively.

2.1.4 3D Adjustable cavity Medication Dispenser

The 3D adjustable cavity Medication Dispenser aimed to solve a few problems observed in a previous attempts by other parties. These being 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 [5]. The 3DAC system was also to a home dispensary system meaning it had to be compact for a home space.

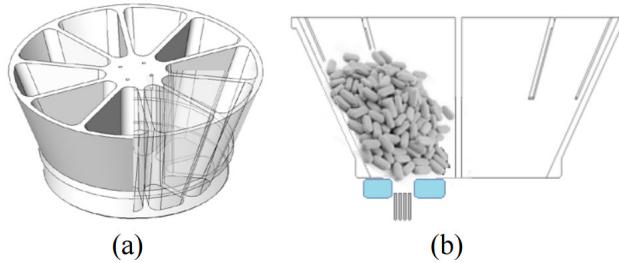


Figure 2.4: Picture of an (a) isometric and (b) cross sectional view of the tablet bucket and its different compartments. [5].

The problem of density of tablets and variety was solved with a bucket called the "Hopper" which can 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 are also sized to fit a large number of tablets solving the problem of density of tablets. The hopper would rotate with a opening on the bottom for a tablet to fall through into the 3DAC.

In order to solve the problem of having an adaptable delivery mechanism able to dispense multiple shapes of tablets the 3DAC was theorized. 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.

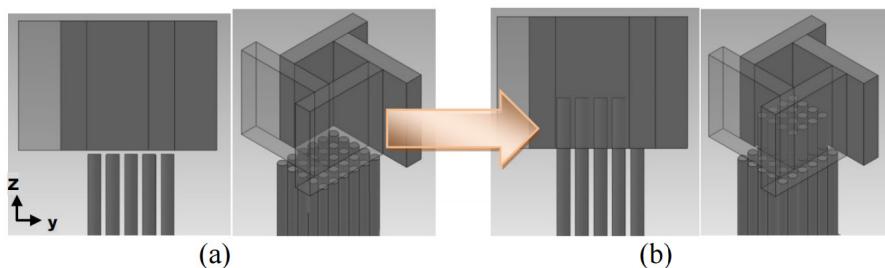


Figure 2.5: Picture of the 3DAC with the rods (a) unextended and (b) extended. [5].

The 3DAC can be illustrated in Figure 2.5 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. 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 shape of the tablet as well. This made it possible for the 3DAC to dispense a wide range of tablet shapes without having to manually change the delivery mechanism, 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.

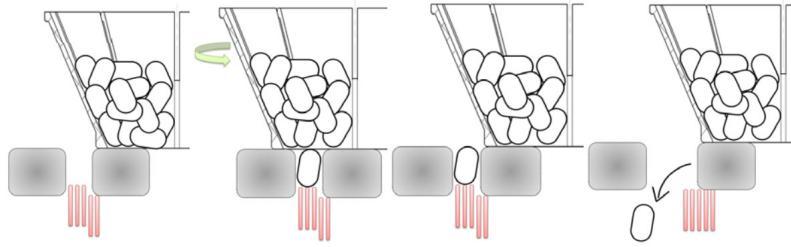


Figure 2.6: An illustration of the dispensary action of the 3DAC. [5].

The dispensary action of the 3DAC is illustrated in Figure 2.6. As one can see 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. In testing this method was proven to be effective at expelling a tablet 95% [5] of the time giving evidence that this was a potentially successful design.

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

2.2 Communication between devices

Because 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 is a communications protocol that is easy to use with most micro-controllers having build 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 [6] and signal generators like the ones present in UCT's White Lab.

In order to improve reliability and noise immunity one way 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 is 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 is an image of a recommended filter from the White Paper with a good balance between loading and filtering. The values set at $R_{pullup} = 1800\Omega$, $R_{s1} = 130\Omega$, $R_{s2} = 51\Omega$, $C_f = 180Pf$ [6].

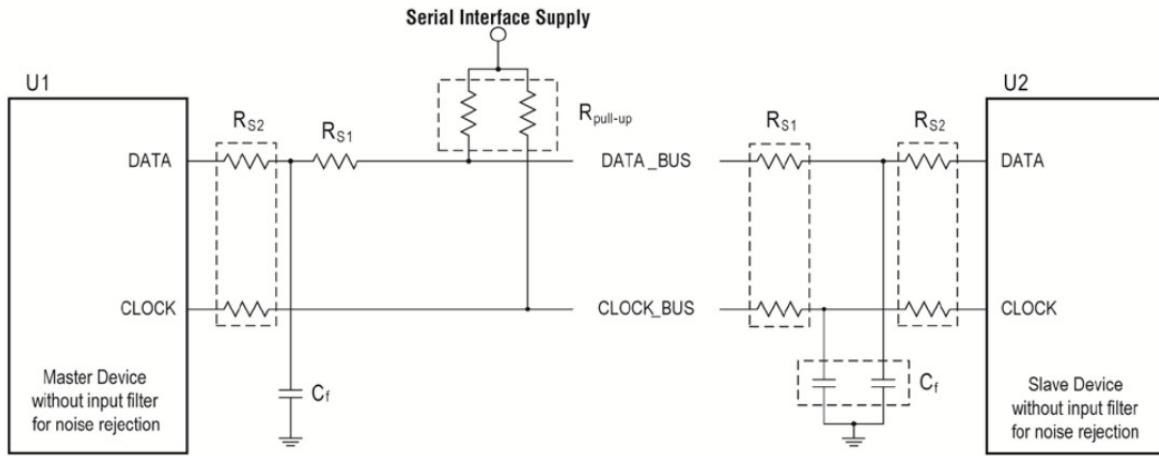


Figure 2.7: External Low Pass Noise Filter Circuit[6]

2.2.2 RS-485/TIA-485

RS485 was considered as it is an industry standard as it can operate over long distances (up to 120 meters at 100kbps [7]), and in noisy environments. This is ideal for the vending machine as it will most likely be placed in White Lab at UCT where there are multiple devices capable of interfering with the communications bus and potentially corrupting the data as discussed in the above section.

The network topology is similar to that of I2C were each device is "daisy chained" to one another making what is called a bus for communications. It is common to use a 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 [7]. UART can not 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 makes it impossible for 3 or more devices to be connected on the same lines for UART as corruption will occur.

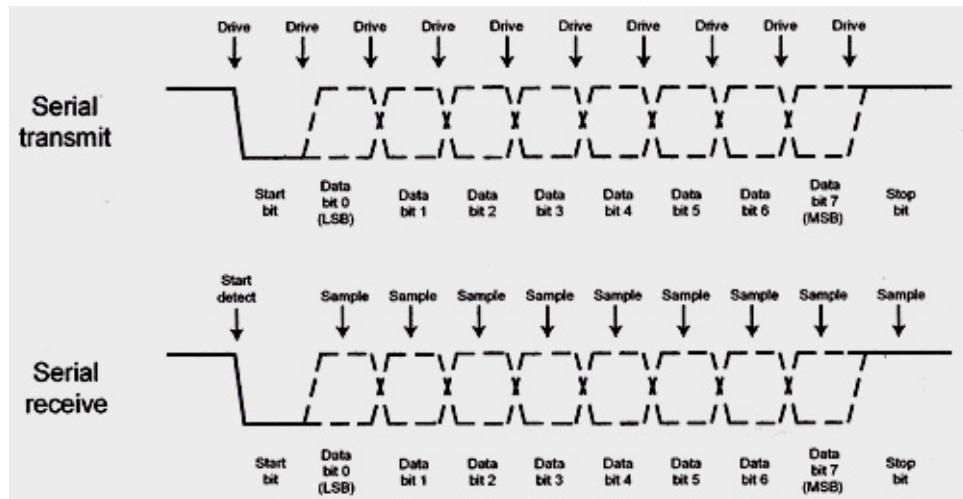


Figure 2.8: When idle one can see the line is held high, this is seen before data is sent and after [8].

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

Noise in a system can be split into common-mode and single-ended noise of which RS-485 is immune

to common-mode noise unlike standard UART or I2C. This makes single-ended noise which comes from improper transmission line termination from mismatched resistance on the output, transmission line and input. 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 are recommended making the bus less susceptible to interference [9].

2.3 RFID Reader

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

2.3.1 System Specification

There are 3 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 are powered (active, passive and battery assisted passive) [10]. The system that will be used for the Vending Machine will be a low frequency system with an RFID Reader and Passive tag. In RFID tags are referred to as the item to be tracked or the identifier in this case the student card. The tag has an on board antenna and a "tag-chip" which contains an ID that can either be factory set, programmable or write once. The antenna is used to power the tag, by receiving power from the reader when in range, and transmit data to the reader. The reader also has an antenna which is used to transmit power to the tag and receive the data being transmitted by the tag [11].

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 is a solution put together in house at UCT using the existing RDM 6300 module and MCP2200 FTDI Chip [12].

The RDM 6300 is a 125KHz low frequency card reader for 125KHz compatible tags like the ones used by UCT for student cards. It supports an external antenna with a range of about 50mm [13]. This is a fairly popular module in the maker community as it is relatively cheap, at about \$12.50, compared to commercial equivalents. It is also popular because of the many resources associated with the maker community and its widespread adoption meaning any problems will 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.

The MCP2200 is a USB2.0 to UART Protocol Converter when paired with the RDM module allows the RFID Reader to communicate with any computer with a USB port [14]. This allows the RFID reader to send data over the USB cable to the computer whenever a tag is 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.

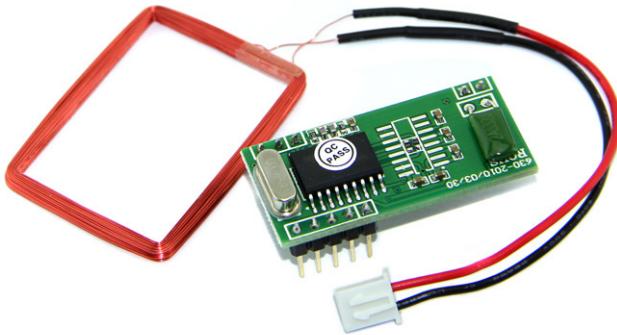


Figure 2.9: The RDM 6300 Module with an external antena connected directly to the module [13].

2.4 Website, Hosting and Server

In order to prove that the Vending Machine could operate with a website an interactive and functional website would be needed. The model used for the website will be a LAMP archetypal structure, or Stack, as this is 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 are open source [15].

2.4.1 The Web server: Apache vs. Nginx

In order to host a functional website a web server must be used. Although a LAMP stack was originally favored, other web servers were considered. The major alternative was Nginx as it has some benefits over Apache. Its main advantage and selling point is that it can handle more connections concurrently than Apache so for a heavy duty web site Nginx is necessary [16]. Apache was chosen over Nginx however, as Apache is a more popular web server with widespread popularity and because the website required will not have a high client base. Netcraft do a survey every month posting the results at the end of each month. Lately they have been getting responses from 1 billion sites giving credibility to the survey. According to the survey, as of the end of August, Apache has 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 has shown a steady decline in market share since 2011 it is still the leader in web servers [17]. This means support and knowledge base will be abundant making using Apache easier to troubleshoot over Nginx the primary contender. Below is a graph showing the market shares of each web server program over the past 8 years.

What Apache does is translate a url that is attached to an IP address and fetch files related to that IP address, returning them to the browser or program that the IP request came from. This can also be a program, where the server will execute the program requested and return the output. This is all done through a protocol called HTTP (Hypertext Transfer Protocol) which allows the browser to make a request in a manner the server will understand it, similar to a protocol used to communicate between micro-controllers. More specifically for Apache these are files stored on the computer the LAMP stack is installed on and in a directory Apache is directed to [18, 19].

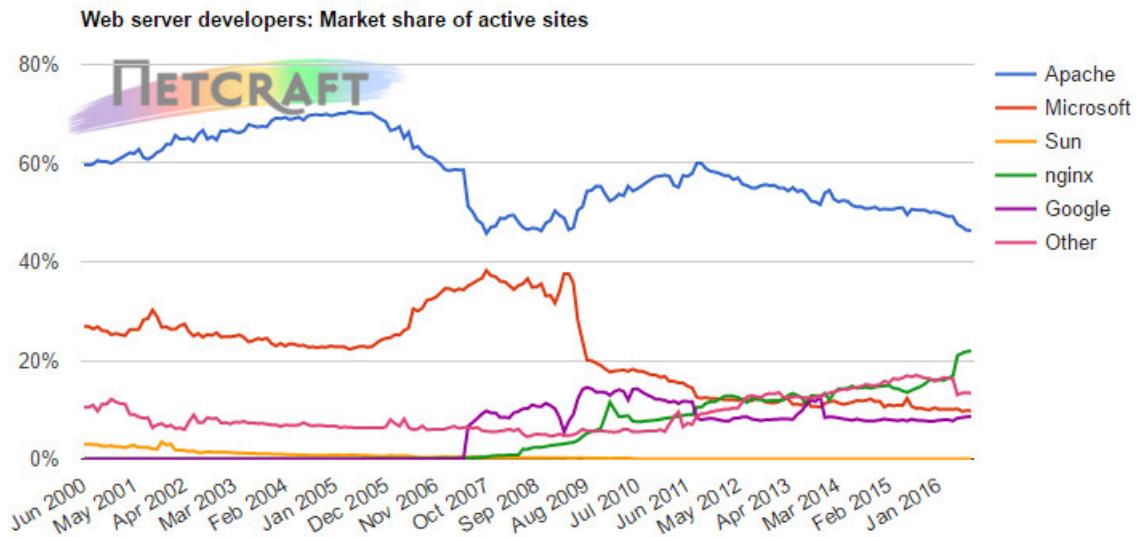


Figure 2.10: Graph showing active websites and their backend web server being used [17].

2.4.2 The Scripting Language: PHP

If the site is directed to a PHP file or a file it sees contains PHP syntax it will pass it onto the PHP interpreter. The interpreter will then execute the code on the server side and a result will be returned with a static page like HTML. Although a static page could be served using just HTML, PHP has the advantage of enabling dynamic content based on server side variables; such as content from a database. Another advantage of PHP is it can be embedded into HTML code. This makes 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 is its ability to interface with multiple database servers. The one to be focused on in the report will be MySQL [20].

2.4.3 The Database: MySQL

MySQL is a database platform built to run independently. It manages data by storing in databases with separate tables each with its own rows and columns. It is 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 helps protect the databases with proper protection giving access to users defined during configuration, making it a secure way to store sensitive data. MySQL uses the ANSI/ISO standard SQL (Structured Query Language) which enables easy access when permitted. The SQL standard and access to MySQL works on a query basis, ie; when an SQL statement is constructed and sent to the MySQL. Once the query has been interpreted the relevant data pertaining to the request is returned [21].

2.5 ATX Specification

In order to power the Vending machine a computer power supply is used. In order to understand how to use this power supply connection and how to use properly the standard used was investigated. The standard used for computer power supplies is well documented and open source, it is called the ATX Specification and is made by Intel [22]. 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 is an updated standard adding addition pins for newer motherboards.

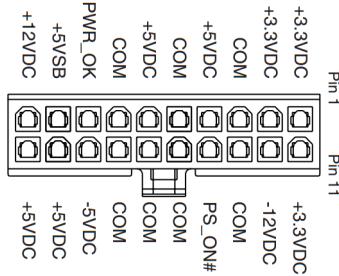


Figure 2.11: ATX 20 pin connection layout [22].

The ATX standard complies to the specifications set out by Intel to operate their processors. This is useful as an ATX compliant power supply has 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 need to be made and certain power rails must be used. To power up the PSU a kettle plug must be connected for AC power and the AC switch must be on, the the "PS_ON#" must be grounded to supply power over the 12V, 5V and 3V3 rails. If left open the voltage rails will be shut off . A "PWR_OK" output is provided from the power supply that indicates if the voltage rails are supplying the correct voltages with a high signal [22]. This is 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 is connected to a motherboard. This can be done by connecting a 5W load to the 5V rail [23].

3 System Specifications

In order to begin designing the Vending Machine the specifications needed to be defined in detail in order to know what the designs should focus on. In the design and discussion of this report there will be three major sections. 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, 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 each and how they interact.

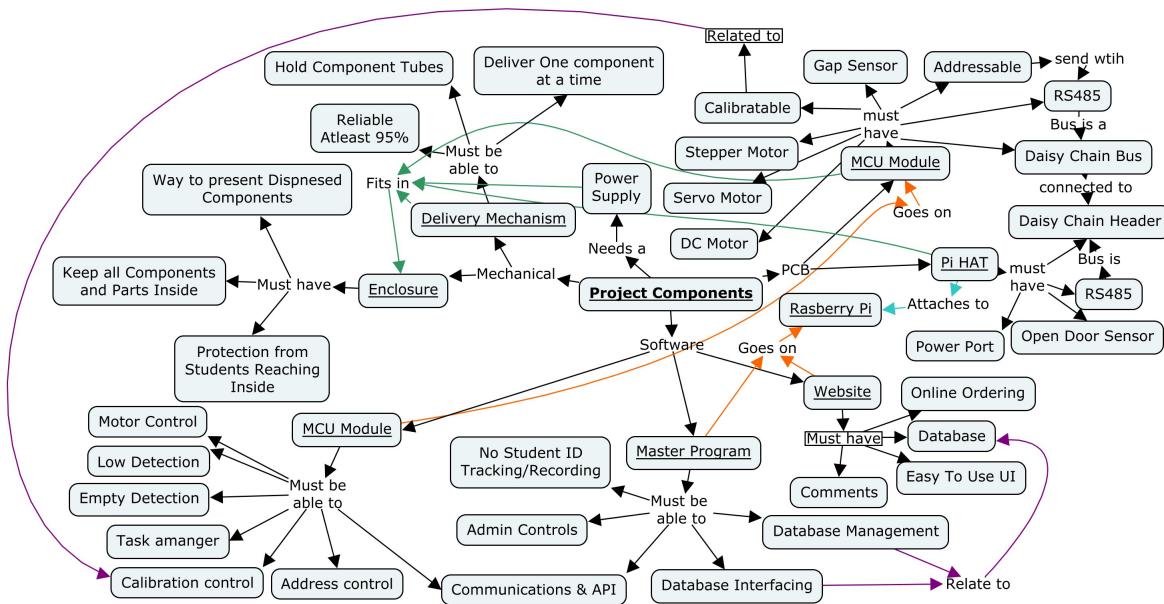


Figure 3.1: Mind map used to help assist the definition of the topic, system specifications and design of the Vending Machine.

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

In order to determine the design requirements for the dispensers the dimensions of varies DIP components where measured. The method of investigation was searching a well known vendors site[24] for datasheets on the varies DIP components. Datasheets were check 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 varies 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 Vending Machine was to have. This autonomy was another specification as the Vending Machine was to operate at potentially late hours of the night when no authority to amend a failure would be around. To improve autonomy Component tubes, pictured below, would need to be held in the machine and fed into the delivery mechanism to give the Vending Machine a large capacity.

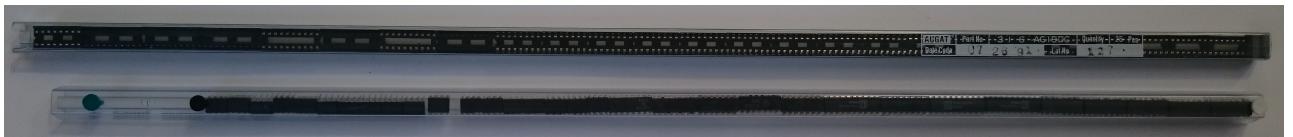


Figure 3.2: Two tubes of components, the top full of DIP IC Holders and the bottom one being full of DIP 8 Components.

3.1.3 Enclosure

The enclosures priority specification is to be able to hold all the modules of the Vending Machine. The enclosure also needs to be able to have space for expansion for future needs 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 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 a PCB to interface with the Raspberry Pi is to be able to power the Pi and create a starting point for communications bus. This module should also be able to sense if the door of the enclosure is open. This is to prevent the machine dispensing components or moving parts when someone could potentially have their finger in the mechanism to try and fix a fault.

3.2.2 MCU Modules

The MCU module has many small specifications all coming together to serve the the main purpose of being modular and interchangeable for different types of delivery systems that may be needed. It would need to operate a DC motor, stepper motor, and servo motor either all at the same time or at once. Gap sensors to be able to detect an empty load for delivery or a jam, or a low cartridge or tube. It was later decided that the MCU module should be able to operate an LCD, This was so messages could be relayed to someone using the Vending Machine i.e. to communicate a problem with the machine. A micro-controller capable of handling the specifications of the MCU Module with a crystal designed to the micro-controllers specifications. A communications and power bus that can be daisy chained to addition modules adding expansion for more than one module on the same bus. Calibration controls to adjust and fine tune the operation of the device. This will allow small imperfections in manufacturing and building to be worked around 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 must be to recognize student cards and the student number attached to them and communicate with the MCU modules. This communication will allow the Master program to control the actions of each MCU module when needed. The master program must also manage the data base and make sure the orders are not invalid. Another feature the Master program must have is admin control in order to be able to induce addition admin features on the each MCU module for maintenance purposes. To do all this the Master program must be able to interface with a database but it must not record any user ID's or information taken from a student card.

3.3.2 MCU Module

The specification for the program were set out to make sure the program ran reliably. To ensure this one of the first requirements were that the MCU would use very few delays and instead use a task manager. The task manager must enable to MCU module to operate without delays when dispensing components so the system will not be held up by any one task. In addition to the task manager, motor control for the 3 kinds of motors must be present. Detection for empty load and low cartridge/tube using optical gap sensors. An address storage system must be in place so the device will now when it is being commanded over the communications bus. Finally the MCU module must have a task to calibrate the device and save the newly set values. A summary of the I/O needed has been listed below:

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

3.3.3 Website

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

4 Design and Prototyping

Methodology and Procedure

In order to begin the design process 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 and more. 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 prototyping:

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.

In order to limit the complexity a simple design approach was used where simplicity and the method forward thinking of "how would it fail" were always the first and ongoing design considerations. Once the 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. were functionality took priority. To reduce complexity the number of moving parts would be kept at a minimum in order to prevent failure of functionality and structure.

Design for the delivery mechanism started out on paper as sketches with basic ideas until a practical idea was ready. Once ready the idea was designed in SolidWorks with the above mentioned goals. Once the Model was fully defined in SolidWorks, the model was printed on a 3D printer to prototype and test the effectiveness of the design. 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 more 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 the delivery mechanism however there was no prototyping 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 in mind more so than functionality, although not to say functionality didn't play a part in decision making. For the enclosure first the frame was designed then the internal housing to hold all the delivery mechanism, Raspberry pi and power supply were designed. Next the shell was designed along with the slide for the components to fall down and the front door. Once design was finished 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 have it be versatile and able to adapt to the needs of the project by adding in features to allow for multiple configurations of mechanical delivery needs.

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. 3 configurations were considered as the mechanical system was to need a motor of some kind, so the design was to be able to handle a stepper motor, servo motor and simple DC motor, 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 needs.

4.1.3 Software Design Methodology

The software for the machine is one of the most important parts of consideration as it will impact each part of the design and how they 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 Vending Machine. It was designed to be easy to understand for the user and operate as the UI for the interaction for the students who would eventually use the 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 designed similarly and in conjunction at times 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 3 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 a test method that could be used for all test cases. Also a recording structure was made 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 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 which 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 has many advantages over many 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 meets the requirements of the communications needs, also this means only 2 wires will be needed for communication plus a common ground. To manage the half duplex communications each transceiver has data enable and receive enable triggers to allow it communicate and receive data only when needed allowing the device in control to manage its communications better, i.e. when tasks are being performed and no communication is expected or needed it can turn off its bus connection. This will allow for more reliable task execution.

5.2 Servo Motor

In order to drive the roller version of the delivery mechanism a 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 is inexpensive at €2,42 [25]. 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 is more expensive at €7,61 [25] it is 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 caters for a stepper motor for future development. The stepper motor the PCB was designed for is the NEMA 17. This stepper motor was chosen as it is a fairly commonly used stepper motor and popular in the 3D printing community, RepRap project [26], meaning trouble shooting potential problems would be easier than using another motor.

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

5.4 Power Supply

The power supply chosen is a PSU from a computer. The computer PSU was chosen as it can supply 3.3v, 5v and 12v simultaneously and provide 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 came 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 whereas the STM MCU costs around R30 [27].

The STM MCU features a 48MHz max system clock with 32Kbytes of flash and 8Kbytes of SRAM. It has 11 timers in total which will assist with the PWM channel and task manager. It has 27 available I/O channels. Of these 27 channels 10 can be mapped to the ADC, 6 can be mapped to the timers and used as PWM channels. Additionally the STM MCU has 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 perspex would be able to handle the small forces applied to the mechanism. It was also chosen as perspex is easy to cut given a laser cutter is used, making for accurate assembly of the design. Lastly clear perspex was specifically used so the housing wouldn't impede viewing of the delivery mechanism for 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 is one of the easier metals to weld making it ideal as the facilities are common for welding steel.

5.8 Enclosure Casing

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

5.9 Enclosure Shelving

Shelving for the delivery mechanisms and the housings was chosen to be wood as it was the most cost effective solution over using steel, aluminium or perspex. Wooden planks 12mm thick would be used.

6 System Design and Prototyping

The design of the Vending Machine will be split up into three section 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 were relevant and not in its own section to maintain its context with the design it is related to. All 3D models, unless specified otherwise, were made using Solidworks.

6.1 Guide For Component Tubes

To help make restocking the Vending Machine easier the delivery mechanism would be fed from the same component tubes the components are stored in. The idea was to also make the feed 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 tube was needed to keep them in place and make restocking simple for the user, this was called the IC guide. Although the Guide did not change much through each version, changes were made after each problem was found. Although the Guide form part of the delivery mechanism it deserved a review of updated by itself and so these 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.

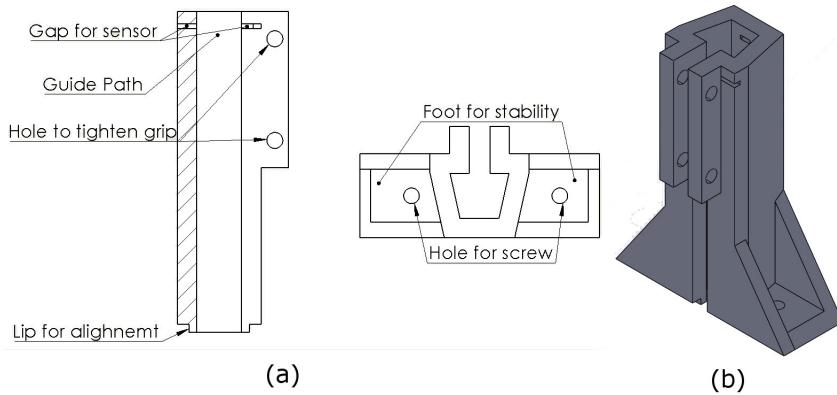


Figure 6.1: IC Guide (a) detailed diagram and (b) isometric view.

In Figure 6.1 the IC guide is depicted in (a) detailed diagram showing features and (b) an isometric view for a real life rendering. The component tube would fit in the center 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 giving 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 at different heights to distribute the load of the screws reducing pressure applied at each point 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 different 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

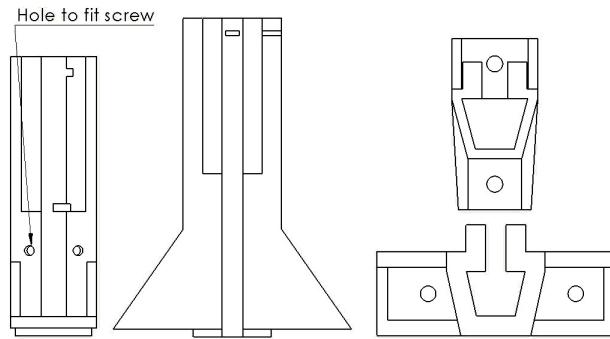


Figure 6.2: Comparison between Version 1 and Version 2 of the IC Guide.

Version two of the IC guide improved on the design of the first and fixed 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. Additional holes were made so screws could fit to press up against the tube and further fix it in place.

6.1.3 IC Guide final

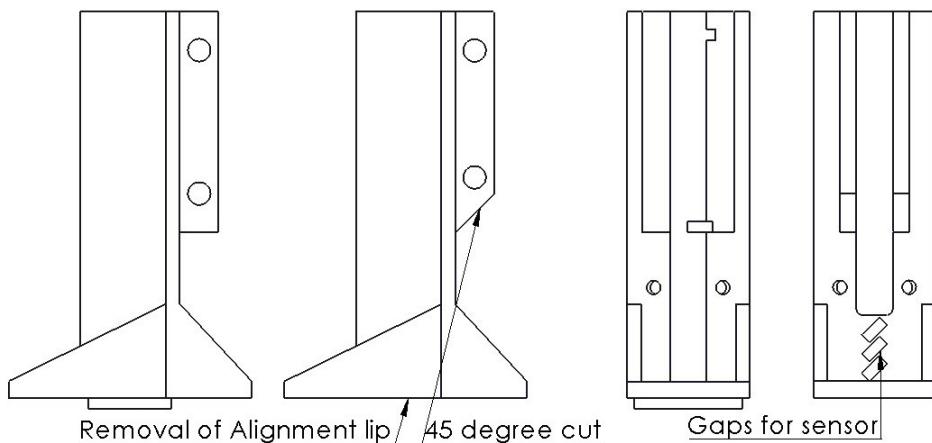


Figure 6.3: Comparison between Version 1 and Version 2 of the IC Guide.

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 that 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 which is rectangular. The Guide hole was simply changed to fit this shape. This is illustrated in Figure 6.4.

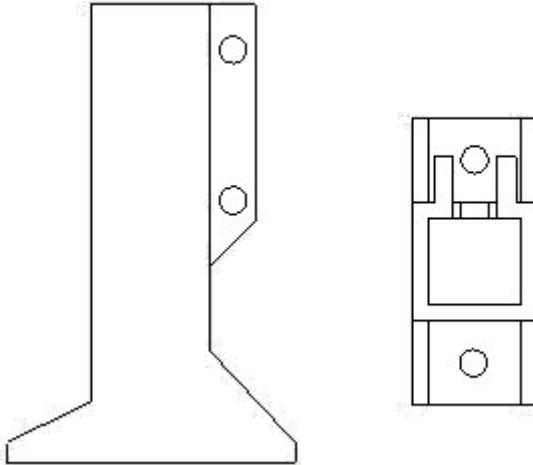


Figure 6.4: Rectangular IC guide for DIP IC holder tubes.

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 to be done by a servo motor. This Roller was to be designed to be able to be 3D printed using PLA or ABS.

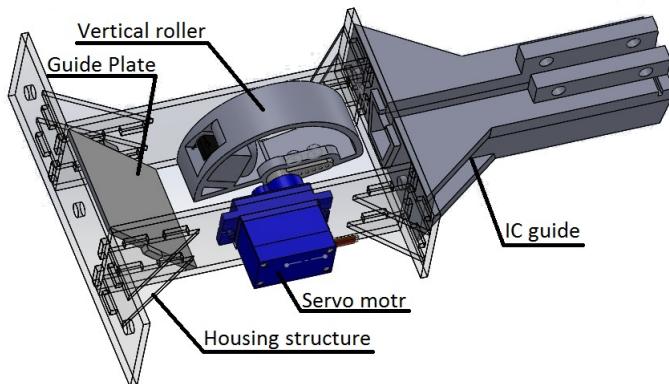


Figure 6.5: Vertical Roller and Housing Delivery mechanism.

The "Vertical Roller" is labeled in the Figure 6.5 situated in the "Housing Structure", made of 2mm perspex, that would enable it to pivot on the vertical axis with the assist 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 so 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 is depicted in the Figure 6.5. A plate labeled "Guide plate" was designed to force components to fall away from the housing if they fell straight down. The components are guided into the bucket by the "IC Guide", the design of the IC guide are 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.

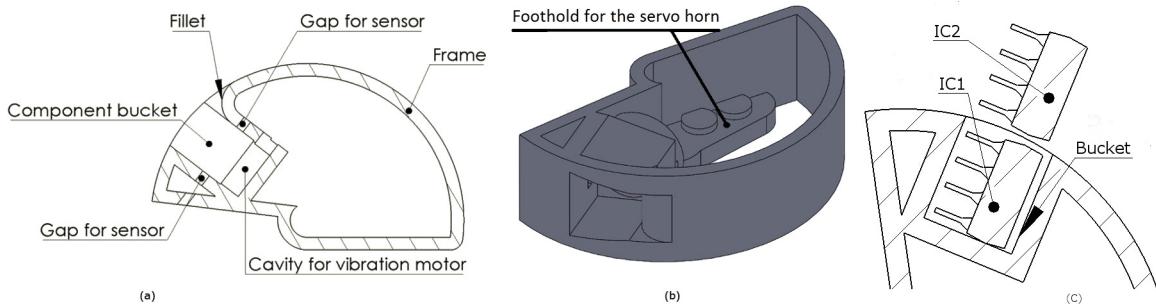


Figure 6.6: Vertical Roller (a) cross section (b) isometric view (c) view to help explain loading.

The Figure 6.6, shows the Vertical Roller in a cross sectional (a) and isometric view (b). The cross sectional view shows features of the part. The cavity labeled "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 to catch on the wall of the bucket. The slot labeled "Gap for sensor" would house a 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 was jammed. A "Frame" was to keep structural rigidity. The isometric view shows detail on how the design would look in real life. A "Foothold for servo horn" is labeled which is 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, labeled (c), IC1 will be the name given the component in the bucket. IC2, as shown in the figure above, will be the name given to component that will be loaded on 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 are guided into the bucket, one on top of another from a tube situated in the guide which is positioned at 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 is seen in the video [28, 29]. The parts for the housing were laser cut, scrap perspex was used to reduce prototyping costs. The Vertical Roller and IC guide were 3D printed using PLA.

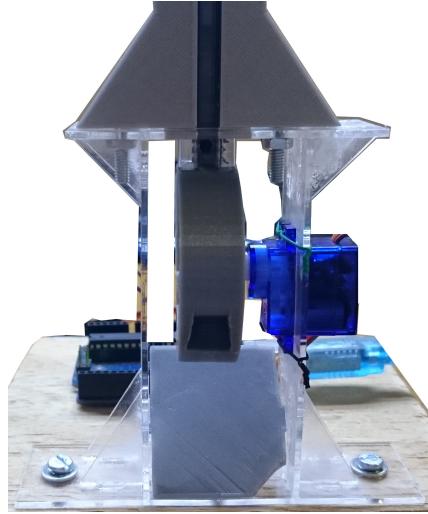


Figure 6.7: Prototype of Delivery mechanism using the Vertical roller.

In testing the prototype had a reliability factor of just under 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.

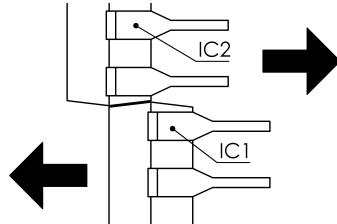


Figure 6.8: Common jam action that occurs with vertical roller.

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 cause by jamming of the IC1 on IC2 could be mitigated through changing the why the mechanism operated through programming. Additionally a center hole was added to the servo foothold to make it possible to align the center 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 is seen in the video [30]. 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 where an IC being caught on the lip of the bucket was 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.

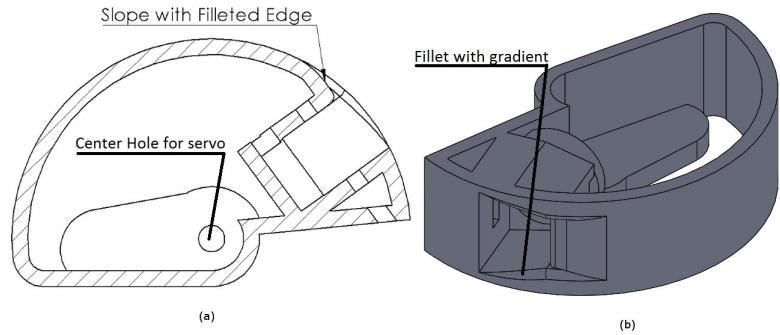


Figure 6.9: Changes made to the Vertical Roller for version 2.

6.2.5 Horizontal Roller Version 1

The Horizontal roller just like the vertical roller is named after the axis it pivots 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.

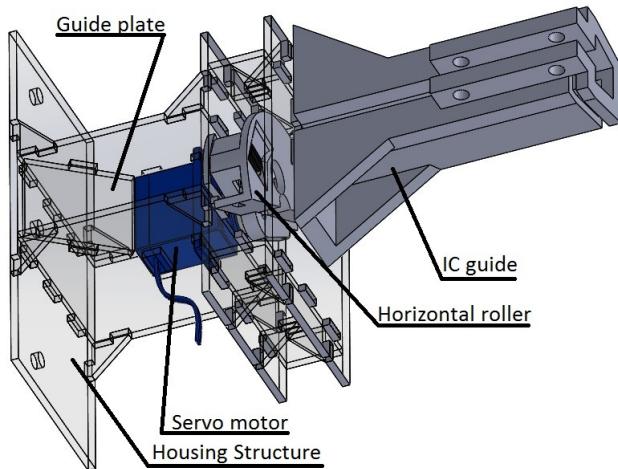


Figure 6.10: Horizontal Roller and Housing Delivery mechanism.

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" labeled in Figure 6.10 would be situated in a "Housing structure" made of 2mm perspex. The housing structure had the same span as the vertical roller. The design was also 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 is that the wall of the servo and the housing made 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 properly. A more detailed diagram of the Housing and roller can be seen in Appendix B.4 Horizontal Roller & Housing Version 1.

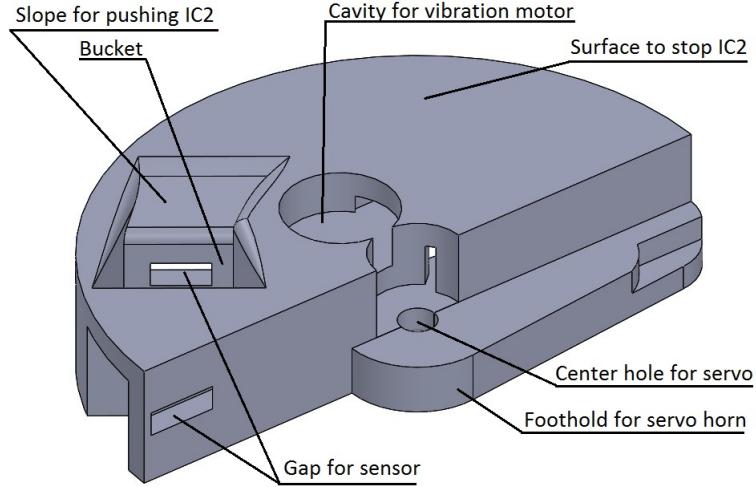


Figure 6.11: Labeled isometric view of the Horizontal roller.

The horizontal roller is illustrated in Figure 6.11. The feature labeled "Gap for sensor" is for the IR LED and IR phototransistor that will detect if a component is 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 gradually pushing IC2 up and the "Surface to stop IC2" will hold it up while IC1 is being dispensed. The feature labeled "Foothold for servo horn" and the "Center hole for servo" are to attach the horizontal roller to the servo horn and align it with the center of rotation for a balanced rotation.

This design works by first loading a component in the bucket, IC1. Then the roller will rotate about 90 degree. 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 where IC1 being caught in the bucket, by using a horizontal roller that used a different method to expel the component this would be eliminated. Secondly, IC2 jamming up against IC1 as illustrated in Figure 6.8, by rotating IC1 instead of moving tangentially on the circumference of the roller away from IC2 this would eliminate IC2 from jamming against IC1. This is because the force on IC1 by IC2 will 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 like in 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 perspex, to keep prototyping costs low, and the roller was 3D printed using PLA. A video of a portion of the testing is seen in the video [31].

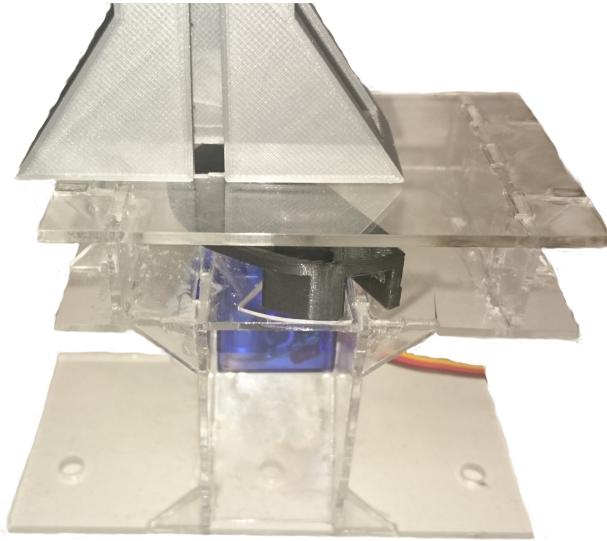


Figure 6.12: Picture of the prototype Delivery mechanism with the horizontal roller.

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 it was still present. The vibration motor was to help with this issue. As 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 to help with this issue as well.

6.2.7 Horizontal Roller Final Version

I final version of the roller and its housing were design. 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 supports would be reduced, 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 vending machine.

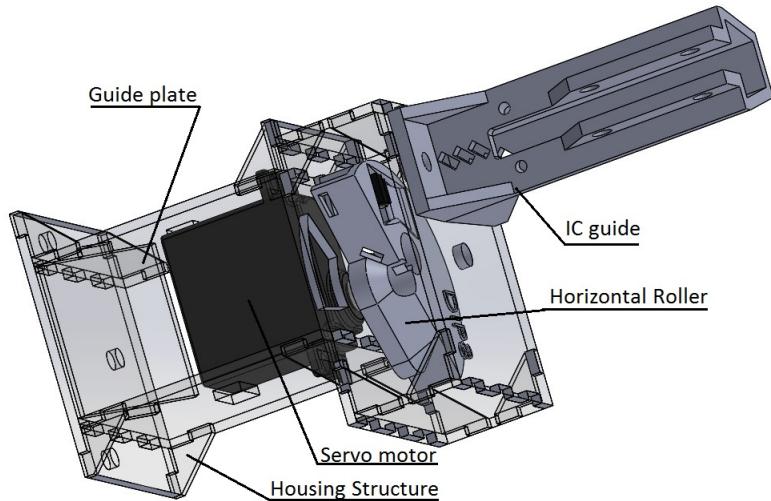


Figure 6.13: Final design of the horizontal roller and housing.

The servo motor was changed to a stronger servo 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

lead to threading. Another reason for using the stronger servo was that it has a better build quality and responds 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 more of them were readily available than the 9g servo. The new "Housing Structure" now measured with a width of 65.4 mm meaning on a 1 m span, 15 delivery mechanisms could be placed, this is an increase of 6 from the previous design.

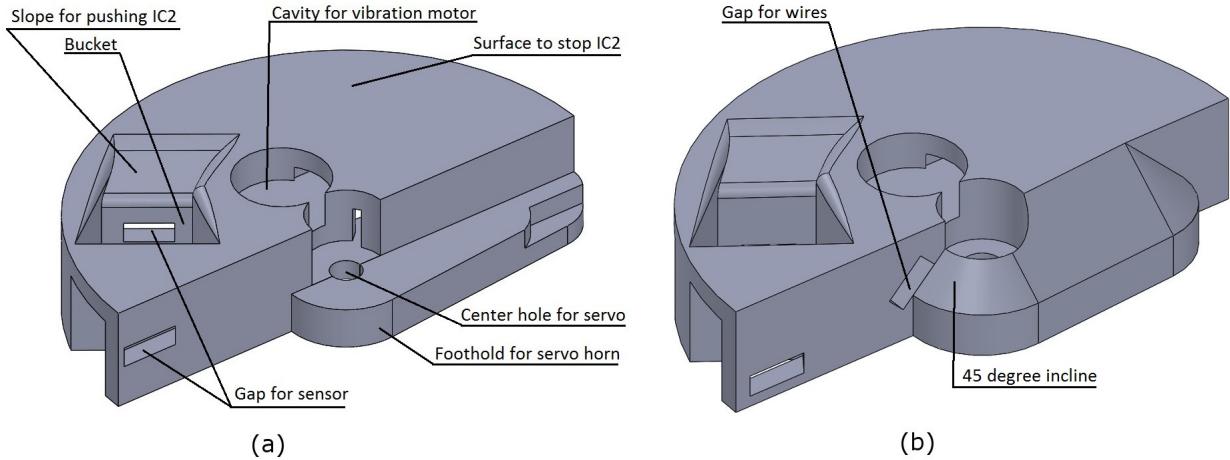


Figure 6.14: Comparison between version 1 and final version of horizontal roller

In addition to 45 degree 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 so 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.

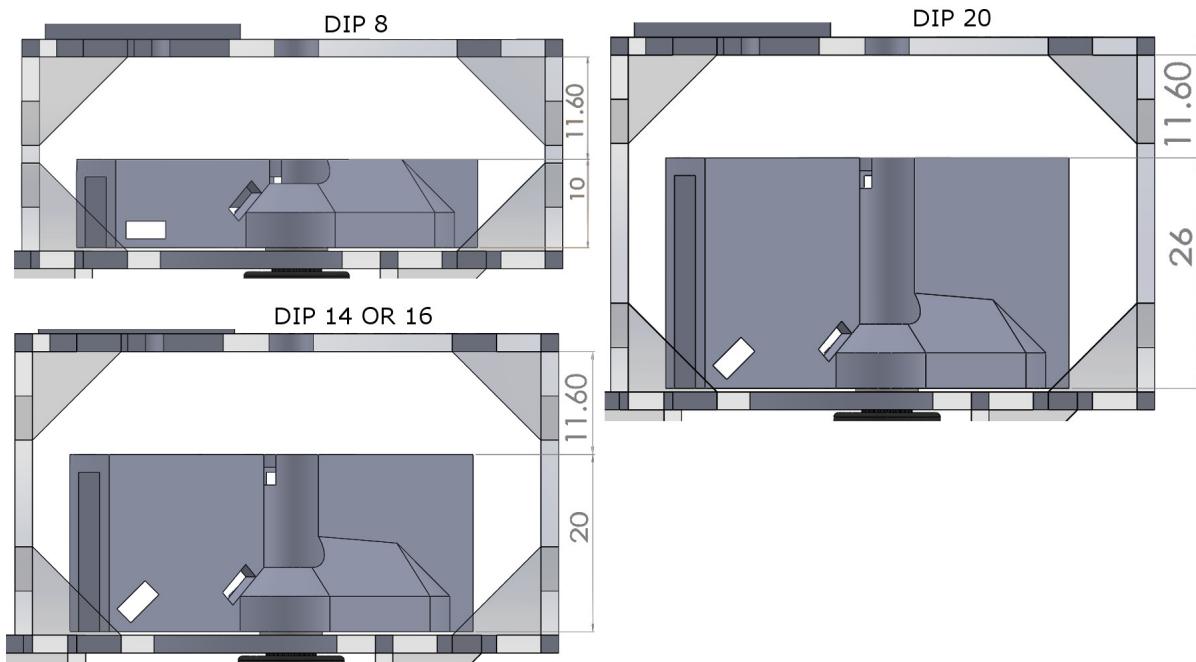


Figure 6.15: All rollers to dispense DIP 8, 14 or 16 and 20 for comparison.

The heights of the rollers wheels were decided by using the average of the component length obtained in Table 3.1 and rounding to the nearest integer. Designs were made 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 with different height 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 designed. It was not tested or built as the Horizontal roller worked well enough that an improved design was not needed. The Feed up method used a very different delivery method. Instead of using a gravity fed system the components would be pushed up 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.

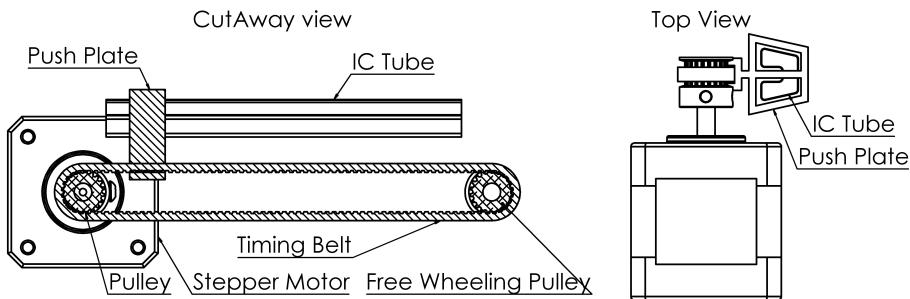


Figure 6.16: Feed up design with cutaway and top view to illustrate how it operates.

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 a IC tube. The solution as illustrated in Figure 6.16 is to use a timing belt with a small plate like attachment that would gradually push the components in the component tube up as the timing belt progresses. 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 favored was because of its complexity and the fact that a modified component tube would be needed. This modified component tube would mean it would be fixed and the restocking process would entail manually putting 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 is 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 all the delivery mechanism and electronics that made up the 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 and to a compartment for the user to collect their order. The upper section would be the compartment housing the interior parts of the Vending Machine.

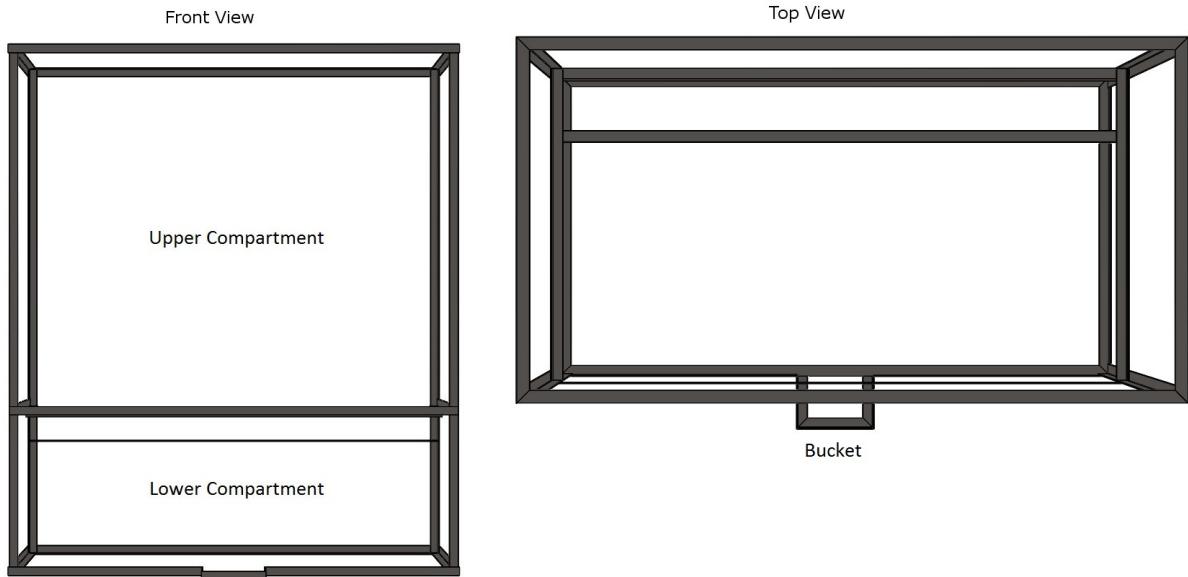


Figure 6.17: Frame of the enclosure illustrating the upper and lower compartments.

The "Upper Compartment" as illustrated in Figure 6.17, of the frame 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 and 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 in at about 600 mm with an addition 150 mm for potentially longer tubes, 10 mm for error, leading to a upper compartment height of 860 mm. The width of the enclosure was determined by the space it would take to fit 3 rows of dispensers in the Vending Machine leading to a width that was rounded to 600 mm. The "Lower Compartment" was designed to fit a sloped piece of aluminium that would guide the dispensed components to a "Bucket", labeled in the top view, accessible to the user to collect the component. This sloped piece of aluminium is illustrated in Figure 6.18.

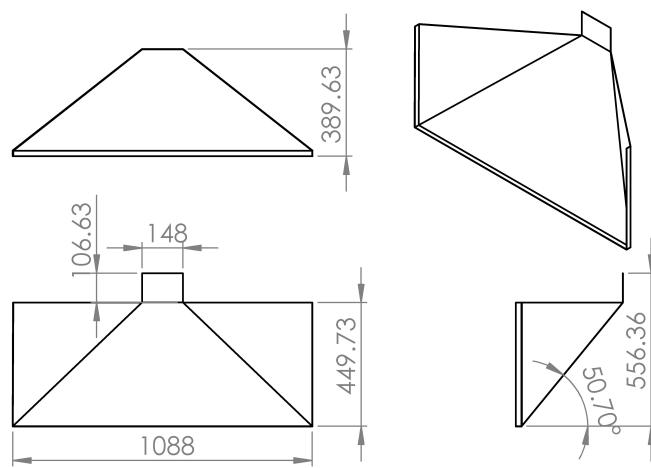


Figure 6.18: A detailed drawing of the sloped plate.

This sloped plate was designed to have an angle of incline above 45 degrees to ensure that components wouldn't be caught on the slope and fail to reach the bucket. For this reason 50 degrees 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.

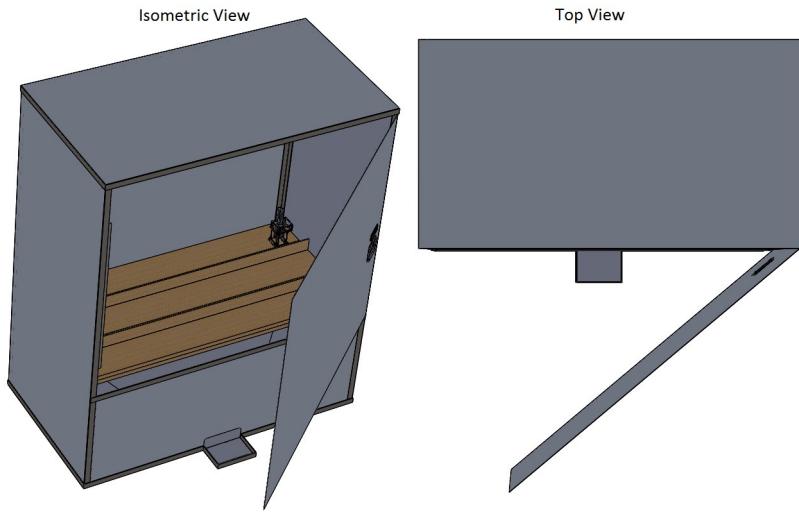


Figure 6.19: Isometric and top view of the enclosure with the case on interior.

The sheets that cover the enclosure can be seen in Figure 6.19. Inside one can see 3 planks of wood as the shelving. The wood plank furthest back has a delivery mechanism situated on it to illustrate how the mechanism would be places and a reference for size. The first plank, being the rear most plank, will be indented to be installed for the vending machine and the other two are to represent the means for expansion when the first plank has run out of space for more delivery mechanisms. Drawings of the parts that make up the enclosure and have been provided in Appendix E.4 Drawings for Assembly: Enclosure and Appendix D.1 Material Request Forms, these were used to assist assembling the enclosure and will be discussed more in later section 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 needed was not feasible and so a small redesign of the enclosure was done.

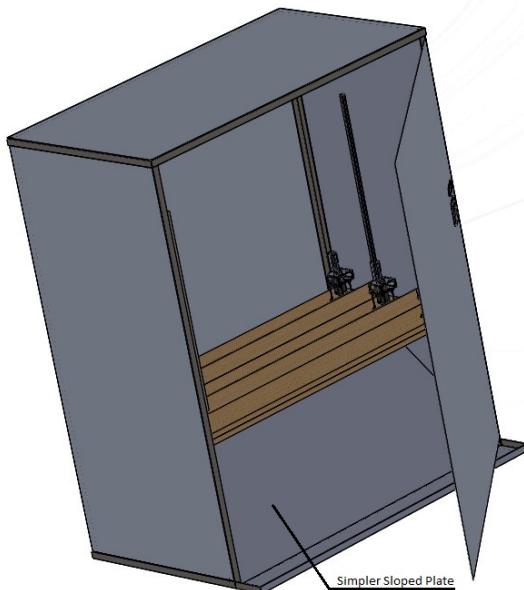


Figure 6.20: Isometric view of the redesign.

As can be seen in Figure 6.20 the previous sloped plate is replaced with the "Simpler Sloped Plate". This simpler plate would be feasible to bend 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 material were ordered however did not need additional material, the material ordered would be enough to complete the redesign.

6.5 RFID Coil Cover

A cover for the RFID reader was needed to protect the coils from damage from 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.

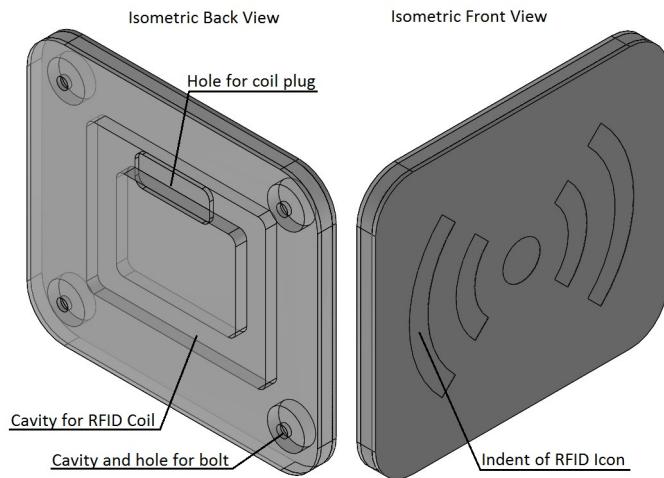


Figure 6.21: Isometric back and front views of the RFID coil cover.

The cover can be seen in Figure 6.21 from the back 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 easy for the user to know where to swipe their card illustrated by the item labeled "Indent of RFID Icon". The cavity was designed to fit the coil with a bit of room for unpredictable coil measurements, this is labeled "Cavity for RFID Coil" in the figure. "Cavity and hole for bolt" labeled above illustrated 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 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 present the designs that made the electronics of the vending machine. The PCB 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 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.

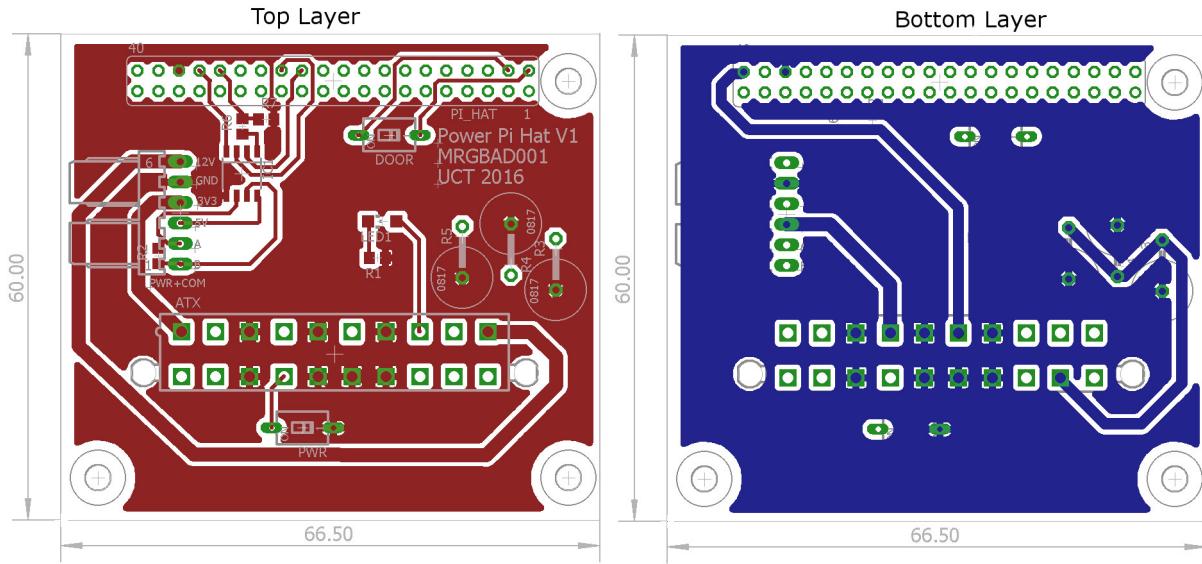


Figure 6.22: PCB layout of the Raspberry Pi HAT, top and bottom layer.

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, and have a way to connect 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 a molex socket was used that met the standards of the ATX plug, this connector is labeled "ATX" in the Figure 6.22. The differential bus transceiver discussed in subsection 5.1 is used to facilitate the RS485 communication, labeled 'IC1'. The receive signal is divided by a voltage divider, labeled "R6" and "R7", as the Raspberry Pis inputs are 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, labeled "PWR". A switch for the door was also added so the status of the door could be detected, labeled "DOOR". Finally a bank of resistors was added to stabilize the computer PSU by drawing 5W, tricking the PSU into thinking it is connected to a motherboard, these are labeled "R5" to "R3".

6.6.2 MCU Module

The MCU Module was originally designed to sit flat behind the delivery mechanism it was to control. It was later decided that it would be propped up vertically to help increase the density of 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 are illustrated in Figure 6.23. The MCU external oscillator was placed first then the DIP switch, "SW1", and SN75176B placed to facilitated 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 are placed at "MODE" and "TRIM" respectively. In addition 3 LED indicators are used in the bottom left corner. Green for power, red for MCU on and yellow for status. Holes were added to provide the ability to mount the board on standoff bolts.

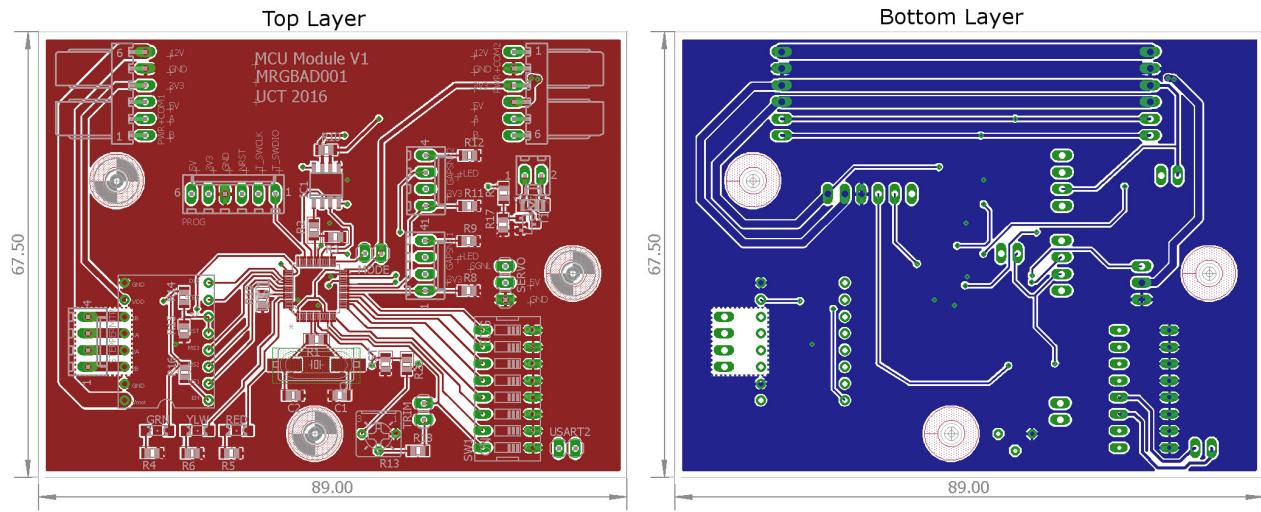


Figure 6.23: PCB layout of the MCU Module, top and bottom layer.

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 wasn't designed to fit. 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.

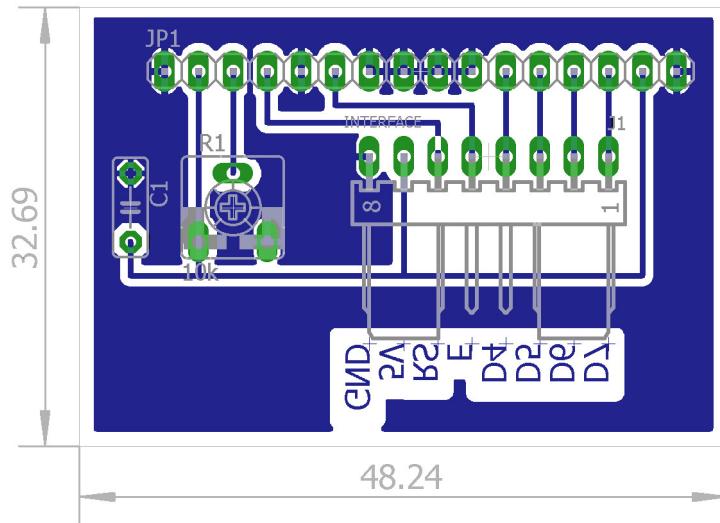


Figure 6.24: LCD breakout board PCB layout, Bottom layer with silkscreen of top layer overlayed.

The PCB layout of the LCD breakout board is illustrated in Figure 6.24. Unlike the previous two PCBs the board was designed to fit on one layer, for this reason 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 will be attached via the 16 way female header at the top of the board, labeled "JP1". The back light is wired via track on the PCB. A trimpot, at "R1", is attached to adjust the contrast ratio of the printed text on the LCD. A molex header is used, at "J1" to interface with the LCD with only the relevant pins being given a header pin with each labeled on the tracks. This was done so no silkscreen was needed to reduce manufacturing costs and simplifying

assembly. This simplification was designed to make it possible to make a home made 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 give an idea of how each program will interact with one another then the API will be discussed before covering the actual software.

6.7.1 Software overview

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

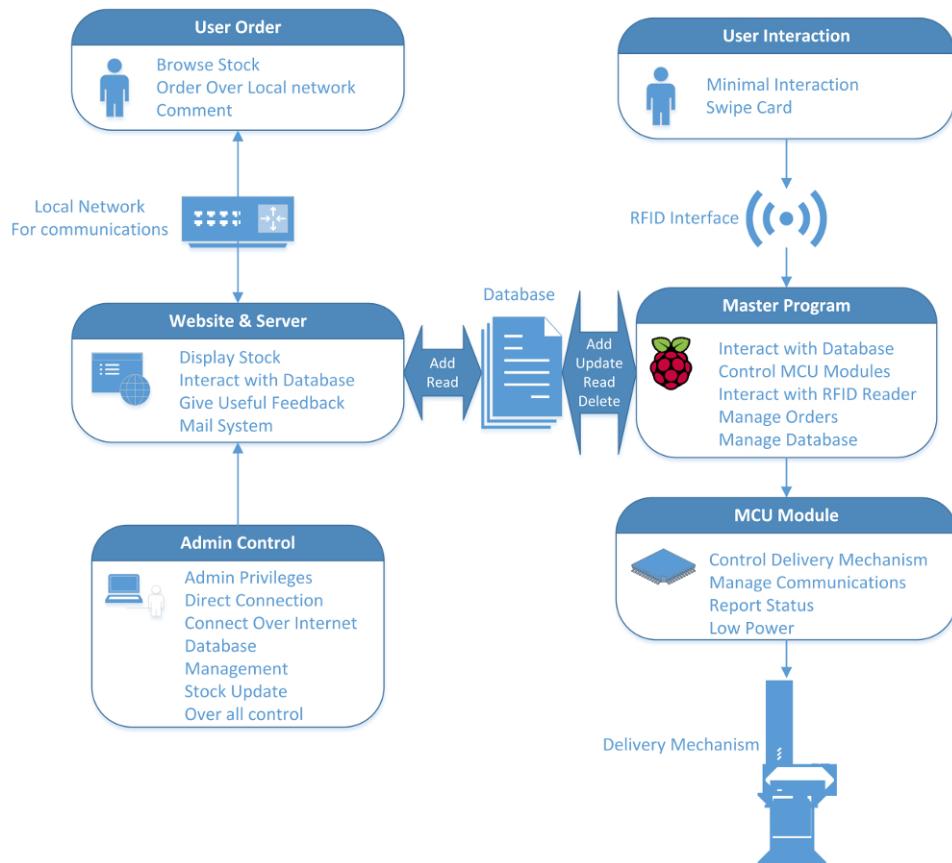


Figure 6.25: LCD breakout board PCB layout, Bottom layer with silkscreen of top layer overlayed.

The Figure 6.25 illustrates the different modules and how they will interact for the purposes of this version of the Vending Machine. Users will order online using the website designed for the purpose of verifying the viability of an internet connected Vending Machine. This website will be hosted on a LAMP web server, hosted on the Raspberry Pi, allowing the website to interact with a database. The website will only have the ability to add or read entries in the database giving it limited control. In addition a mail server will be used to give admins status updates like; Empty rails. Admins will also have access to the website with elevated privileges allowing them complete control of the web server,

website and database. This will allow admins to edit who is notified by the email system, update and manage the databases on the web server, website updates and edits and master program updates.

The Master program will also be connected to the database. This way there is no direct interaction between the website and the Master program, everything is run through the database. This is a safer way of managing orders as the database tool has protection for invalid entries itself. On top of this the Master program will have database management tools with elevated privileges allowing the Master program to add, update, read and delete entries in the database. The Master program must also interact with the RFID reader and MCU Modules. The routines on the Master Program will however only be invoked when there is user interaction through the RFID reader. Once invoked the Master program will communicate with the MCU Module which will control the delivery mechanism. This Module will be a low power solution consuming very little power when idle. The MCU Module will 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 and 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 will send LCD print data. The table below documents the API that is 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 will 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.

| 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.

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. This is inplace ontop of the differential buffer as an added layer of security for the communications. 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 framework researched in subsection 2.4, that 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 made to be user friendly and able to work 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 first designed. An image of basic components and the UCT logo were edited together to make a cover photo for the website and 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 staple on each page of the website. The Website consists of two pages; the component store page and the contacts page. The components store page begins with basic instructs then a design to fit 3 columns, 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 requires a student number, with a submit button below. When the submit button is pressed the website is directed to a PHP script that checks the order to make sure it abides by the rules and corrects it if needed. The script then enters the data in the database and gives the user a small notification, in the form of a pop-up box, that his order was processed and a breakdown of what his order contained. Then the user will be redirected back to the store page. The Contacts page is simply to allow users to contact the administrator with problems encountered with the Vending Machine or comments about improvements and additional features that could be implemented. It contains a introductory message with a entry form below that requires; a name, student number, email address and message. A send button is at the bottom of the form to submit the form. Once submitted the user will be redirected to a script that will send the relevant information in the form of an email to the administrator. Then the user will 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 all most tasked 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 progresses in the future the software can be easily updated to adapt to future requirements.

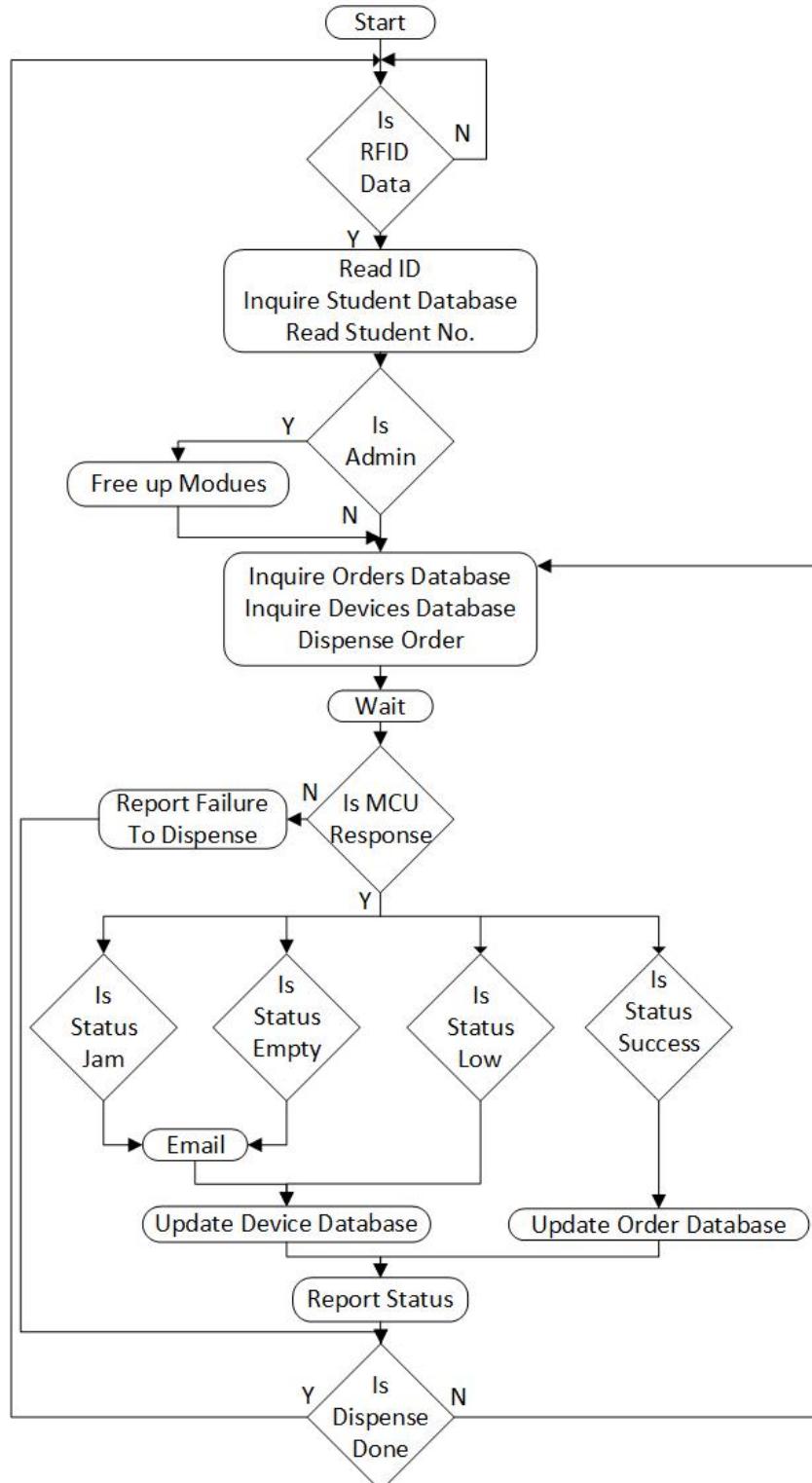


Figure 6.26: LCD breakout board PCB layout, Bottom layer with silkscreen of top layer overlayed.

Before talking about the program the database's should be explained. Two tables were used in the

database, one called "Orders", for the orders that are to be placed on the Component Store page and another table called "Components", for a record of the dispensary mechanisms in the Vending Machine, the component they dispense and the address they are using.

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 begins by checking the RFID reader for incoming data. The database is then queried with the RFID data to get the student number. Extra checks are done to check if an admin is present, if so a free up command will be performed resetting the empty and jam flags in the database and relevant modules. The orders relating to the student number will then be acquired from the database and dispensary requests will be sent to each relevant module one by one. Responses from each module will be received after each module has finished dispensing and the master program will act accordingly.

The responses expected are; Jam, Empty, Low and Success. A Jam and Empty report will warrant the master program to email an admin advising them to attend to the Vending Machine and why. A Low and Success report will just update the relevant tables but no email will be sent. All cases will give a report in the log, which will be discussed later.

This is repeated until all orders, relating to the student/staff number requesting a dispense, are done. The program will then return to wait for a student card to read in and receive RFID data which will then begin this process again.

In addition to the above loop addition user feedback is provided in the program. Firstly a log is kept of all the activities of the program and whether they are errors or successful operations. Secondly LCD messages are transmitted to the first MCU module which should have an LCD connected to give the user swiping his/her student card.

6.7.5 Microcontroller Software

The MCU program logic diagram can be seen in Figure 6.27. Although the logic flow for the MCU module is fairly complicated it can 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, smaller functions will be discussed briefly later. The State machine diagram for the MCU module and its software can be seen in Figure 6.27. To gain a better understanding of the diagram key features will be highlighted, those not highlighted should be explained by the diagram itself.

The MCU module program starts off with initialisation. The initialisation carried out entails setting up registers for the required inputs and outputs including PWM and ADC channels. Also the Timers and UART channels are setup. Lastly the address of the module is saved from the DIP switch meaning the address is only recorded once, if the address is changed the module would need a reset in order for it to take effect. Once initialisation is finished the integrity checks begin. The first check is the "Mode" check, which determines whether the module must run the dispensary or calibration task. If the calibration mode is selected then the module will run through different angles on the servo motor for loading and dropping a component. The module will hold at each angle for a set period of time and the angle on the servo can be adjusted to match the delivery mechanism using the potentiometer. If in delivery mode the module will run the dispensary task when requested via the UART communications over the RS485 bus. The module will idle in a low power state until serial data is received, this low power state releases 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 are done next which will have been set if a previous attempt at dispensing occurs in a jam or the tube on the delivery mechanism is empty. The only way to reset this flag is by sending a free command from the master. There are two other major checks that need explaining. The first being "Is IC In Bucket", which is the sampling of the IR sensor on the bucket to determine if there a component in the bucket. Secondly the "Is IC In Tube", which is the sampling

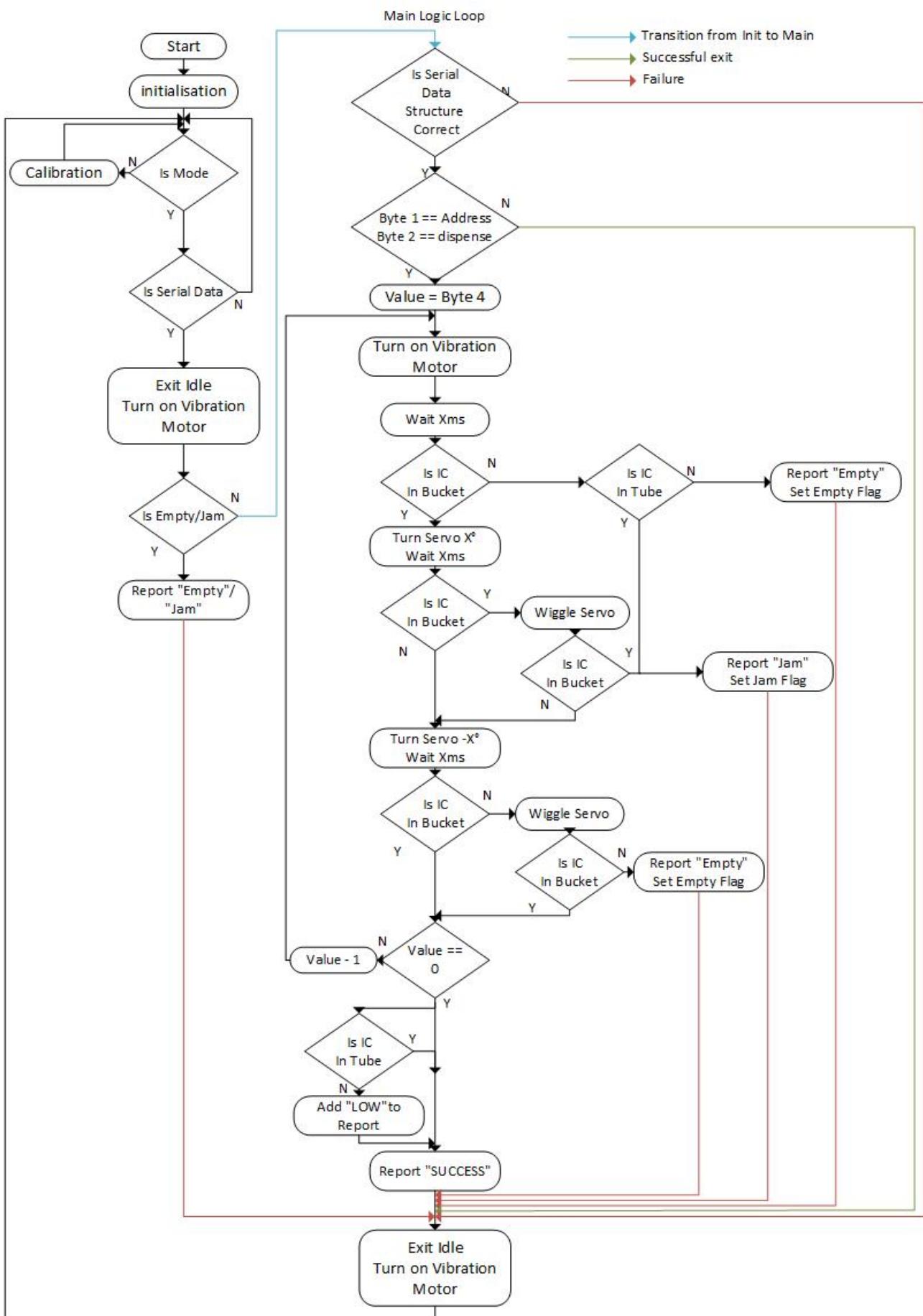


Figure 6.27: State Machine Diagram for MCU software.

of the IR sensor on the IC Guide to determine if there are any components left in the tube. The last feature to be highlighted is the reporting and where "Add" is used. This is because multiple reports can be sent simultaneously. This is done using bit logic, where there are four report codes, the last four bits of the report code byte represents one of each report code.

Additional features, not covered in the diagram, are also available for the module. These come in effect when a different command matches those in the API. The first is a "Call" command which prompts the module to send a blank "Success" report back. Next the "Free" command will prompt the module to set the Jam flag back to default allowing the module to resume normal operation. The last being the "LCD" command, this command only applies to the module with the address 1 as it is intended that this module have a LCD connected to it. If an "LCD" command is received, bytes 4 to the 3rd last byte will be the data that should be displayed on the LCD. The data received should be in ASCII format. For all the above the normal integrity checks are applied to ensure no corruption has occurred.

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

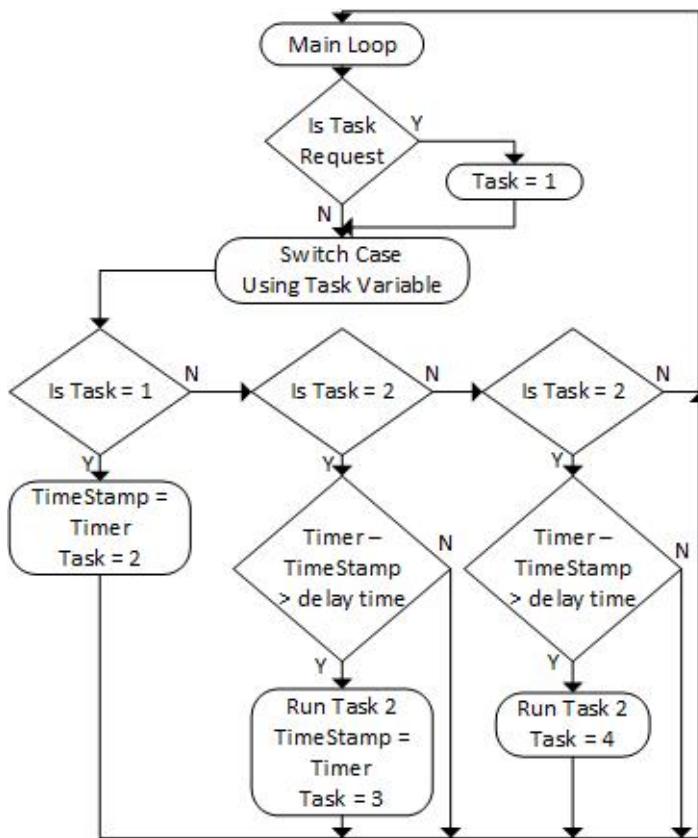


Figure 6.28: A simplified logic diagram of the task manager and how it will function.

One of the other features discussed in previous sections and specifications was the minimal use of delays and a task manager to facilitate this. This was designed using a timer that was set to count up once every 1 millisecond. The design was 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 sampled and the servo was moved X°. After these actions were taken, a short delay was needed before the next would occur. To achieve this kind of task manager, a method was theorized and is illustrated in Figure 6.28. This method can be implemented for any number of tasks and is implemented in the MCU module programming.

7 System Assembly

This section will cover the assembly of the Vending Machine from software, which hasn't already been covered, to putting everything together. The Software assembly is 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. Were prices are quoted in the bill of materials, the prices were acquired from either RS components [27] or from the receipt from purchasing the material unless stated otherwise.

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. The installation process is documented below.

In order to get the Raspberry Pi ready for the LAMP server an operating system was needed to handle the software. The latest version of the Raspbian OS image was acquired from the official Raspberry Pi site [32]. The latest version of Win32DiskImager was then acquired, [33], and used to mount the Raspbian image onto a Micro SD card. Once the process was finished 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.

Now that the Pi was powered and running installation of the LAMP server could begin. A detailed log of commands used to make this possible has been included in Appendix G. 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. Next the plugins for the Master program were installed along with registering the master program to run on boot

The bill of materials for the servo 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

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 accomodate the PSU, Ethernet cable, LCD and RFID cover. Next the drop plate was bent using a bending machine. 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 was also welded onto these pieces, which would hold up the shelves, 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 aluminum were drilled and the hinges were secured with countersunk bolts and nuts. Next the holes for the lock were drilled and the lock was also attached with countersunk bolts and nuts. Finally the frame was brushed and painted to give it a nice finish. 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.

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

* Obtained from UCT ** Obtained from a hardware store

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 are documented below. The PiHAT and MCU Module PCB were manufacturer



Figure 7.2: Pictures of the enclosure frame with the walls bolted on.

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 PiHAT 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 is 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

To begin assembly the STM microcontroller was first soldered. 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 code was uploaded to test proper operation. After successful tests were run the remainder of the components were soldered to complete the PCB. in total 6 board were ordered. Initially 3 were soldered for testing purposes and further debugging of the code. The idea was to test the code with more than one module on the system.

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 of the PCB and a two way terminal for the STEP and DIR through hole of the PCB. Those through holes mentioned relate to the stepper module for future expansion and yet to be designed delivery mechanisms. A picture of both version build for this report can be seen in Figure 7.4. The remaining 3 PCBs were intended to be soldered for the purpose of the final build but were not complete as of compiling this report. A complete bill of materials can be seen in Table 7.4.

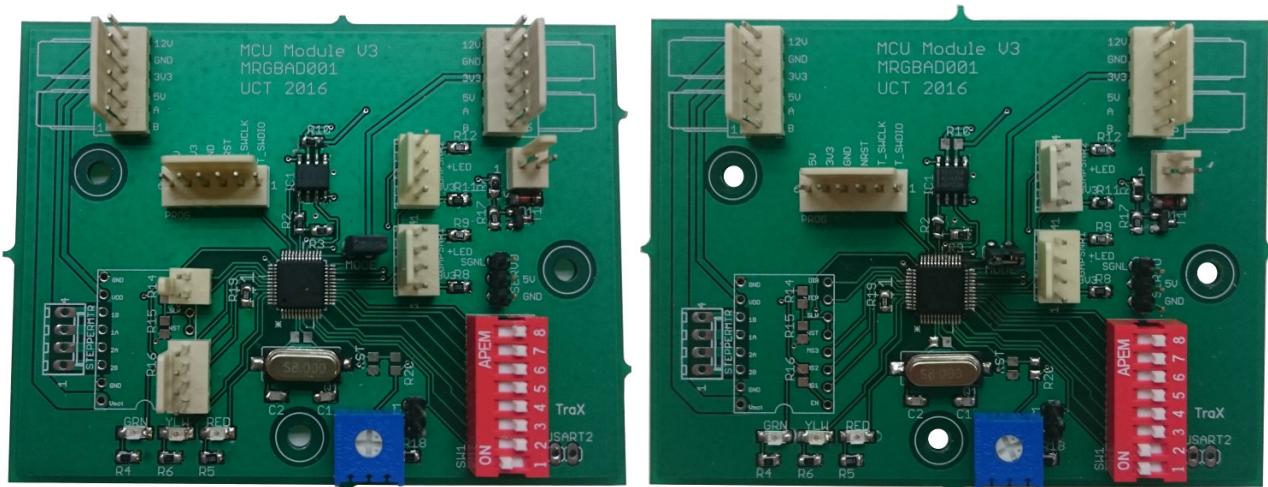


Figure 7.4: Pictures of the MCU Module PCB, with the LCD plugin on the left, without the pcb plugin on the right.

| 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 | R 3.76 (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 Potentiometerv | 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 requires very little resources and can be done without professional equipment reducing the manufacturing cost of PCB making.

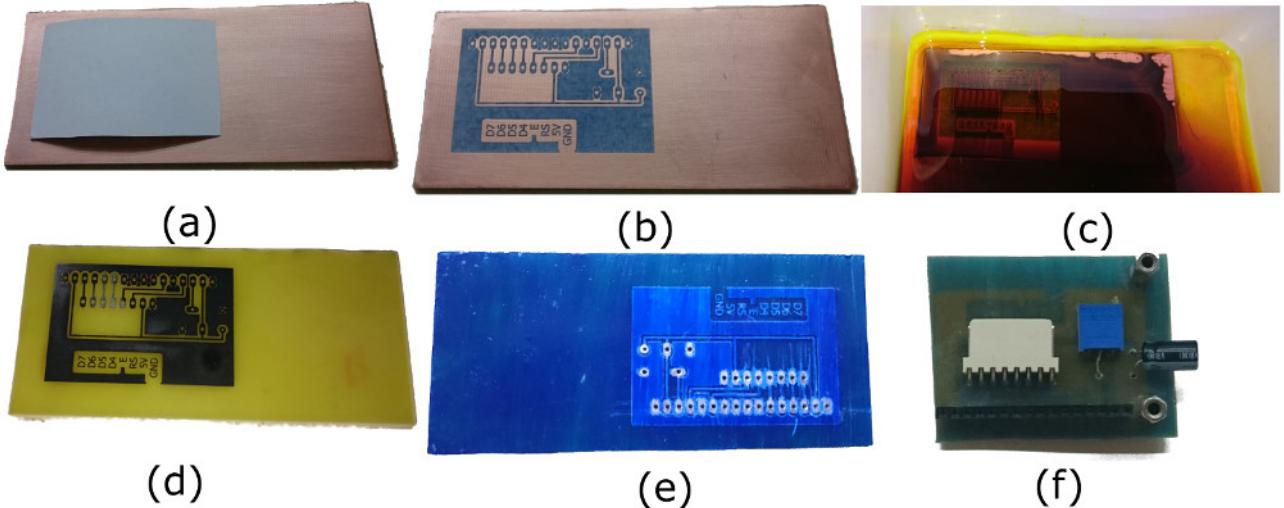


Figure 7.5: Pictures from the etching process to make the LCD breakout baord.

The Figure 7.5 illustrates the assembly process at varies stages. To start the process the PCB layout was printed on photo quality paper with 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 end result is shown 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 is dissolved and the result is shown in (d). The toner is then removed using 1000 grit sandpaper and hole are drilled where needed. The board was then painted with UV treated solder mask (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

To begin assembly of a delivery mechanism the perspex was ordered and cut as per of one of the templates in Appendix E.3 Laser Cutting Templates. The Roller and IC guide were 3D printed using PLA. Once all the parts had been acquired assembly could begin.

Once the 3D printed IC guide were done printing the bolts and washers were placed in the designated holes. Next the two IR LEDs and two phototransistors had wires soldered to the leads. A pair of

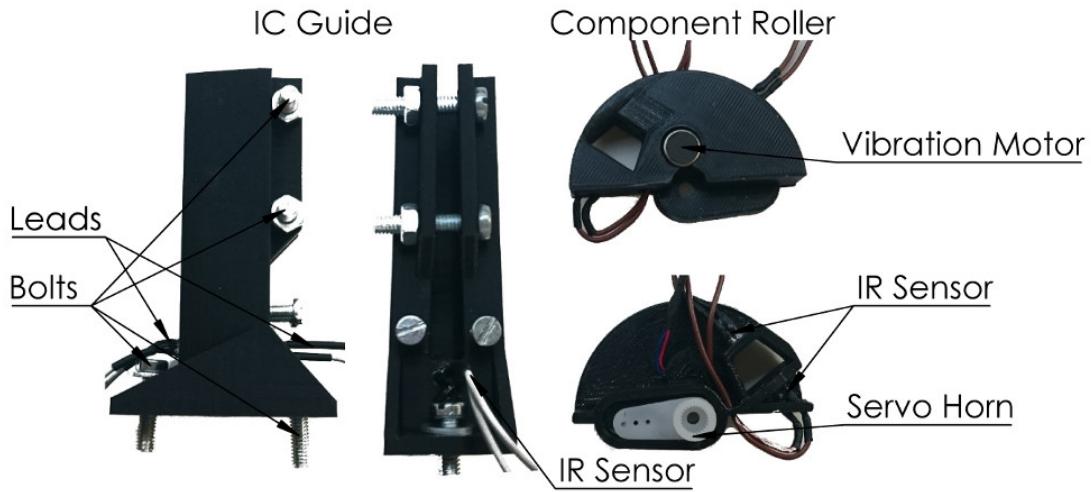


Figure 7.6: Picture of the 3D printed IC guide and roller after assembly.

IR LED and phototransistor were glued into the designated slots making sure to line them up with the intended component to minimize false readings. The Vibration motor was then placed inside the cavity made for it and leads soldered to it. Finally the servo worn was lined up with the foothold and a small amount of glue was used to secure it. The finished product for one of the IC guides and rollers can be seen in Figure 7.6.

Next the housing structure was assembled. The structure was assembled according to the 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 where assembled and described above where then fixed to the housing structure with bolts, nuts and a screw for the servo horn. The final product of this assembly can be seen in Figure 7.7.

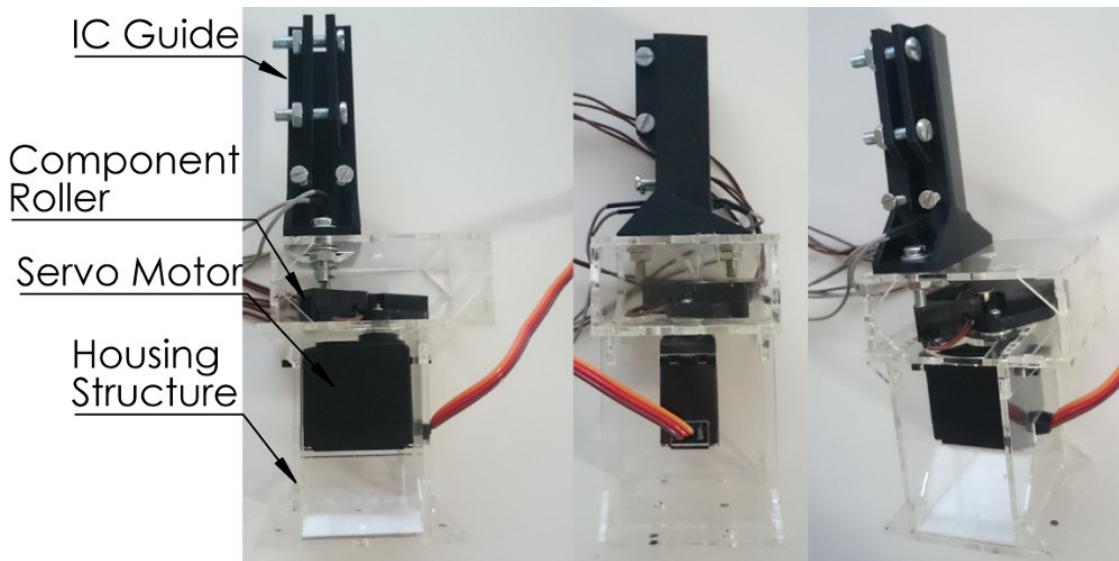


Figure 7.7: Picture the housing structure for the delivery mechanism.

| Part Name | Cost | Quantity |
|---------------------------------|-------------|-----------------|
| Perspex 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.

8 Results and Discussion

9 Conclusion

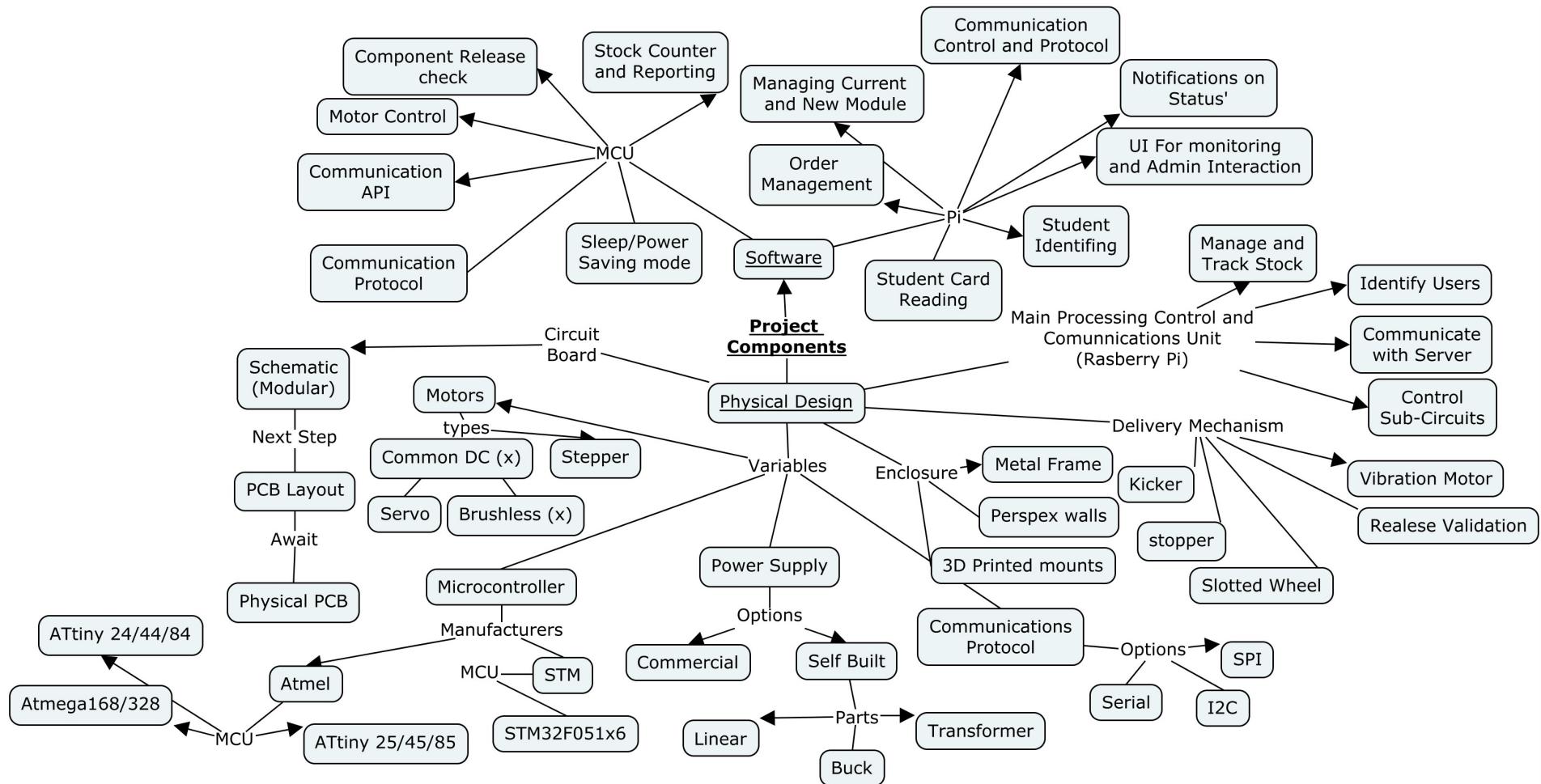
10 Recommendations

References

- [1] Portland State University, Maseh College of Engineering & Computer Science: Electrical & Computer Engineering. Engineering parts vending machine. [Accessed: Sept. 2016]. [Online]. Available: <https://www.pdx.edu/ece/engineering-parts-vending-machine>
- [2] M. G. Gusso. (August 4th, 2014) Late-night component cravings? [Accessed: Sept. 2016]. [Online]. Available: http://www.latticesemi.com/~media/LatticeSemi/Documents/WhitePapers/HM/ImprovingNoiseImmunityforSerialInterface.pdf?document_id=50728
- [3] 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.
- [4] V. Q. Ho, T. J. Gale, 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.
- [5] C. K. Toh, S. L. J. Ng, 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.
- [6] Lattice Semiconductor, "White paper: Improving noise immunity for serial interface," 2014, pp. 2.
- [7] T. K. Texas Instruments, "Application report: The rs-485 design guide," 2008, pp. 1 - 4.
- [8] A. Minor. (2007) Implementing your mcu-based system's serial uart in software. [Accessed: Sept. 2016]. [Online]. Available: <http://www.embedded.com/design/other/4025995/Implementing-your-MCU-based-system-s-serial-UART-in-software>
- [9] J. G. Texas Instruments, "Literature report: Low-voltage differential signaling yields megatransfers per second with milliwatts of power," 1996, pp. 3.
- [10] IMPINJ. (2016) The different types of rfid systems. [Accessed: Sept. 2016]. [Online]. Available: <http://www.impinj.com//resources/about-rfid/the-different-types-of-rfid-systems/>
- [11] IMPINJ. (2016) How do rfid systems work? <http://www.impinj.com/resources/about-rfid/how-do-rfid-systems-work/>. [Accessed: Sept. 2016].
- [12] J. Pead. (Sept. 2016). Principal Technical Officer [Online]. Available e-mail: justin.pead@uct.ac.za, Message: RFID Documentation.
- [13] Seedwiki. (2016) 125khz rfid module - uart. http://wiki.seedstudio.com/wiki/125Khz_RFID-module_-_UART. [Accessed: Sept. 2016].
- [14] Microchip. (2011) Mcp2200 datasheet. [Accessed: Sept. 2016]. [Online]. Available: <http://ww1.microchip.com/downloads/en/DeviceDoc/22228B.pdf>
- [15] J. V. Jay Versluis, *LAMP Stack for Humans: How to turn a laptop into a web server on your local network*. Jay Versluis(Self Published), 2015.
- [16] O. Garret. (October 9, 2015) Nginx vs. apache: Our view of a decade-old question. [Accessed: Sept. 2016]. [Online]. Available: <https://www.nginx.com/blog/nginx-vs-apache-our-view/>
- [17] Netcraft. (2016) August 2016 web server survey. [Accessed: Sept. 2016]. [Online]. Available: <https://news.netcraft.com/archives/2016/08/24/august-2016-web-server-survey.html>

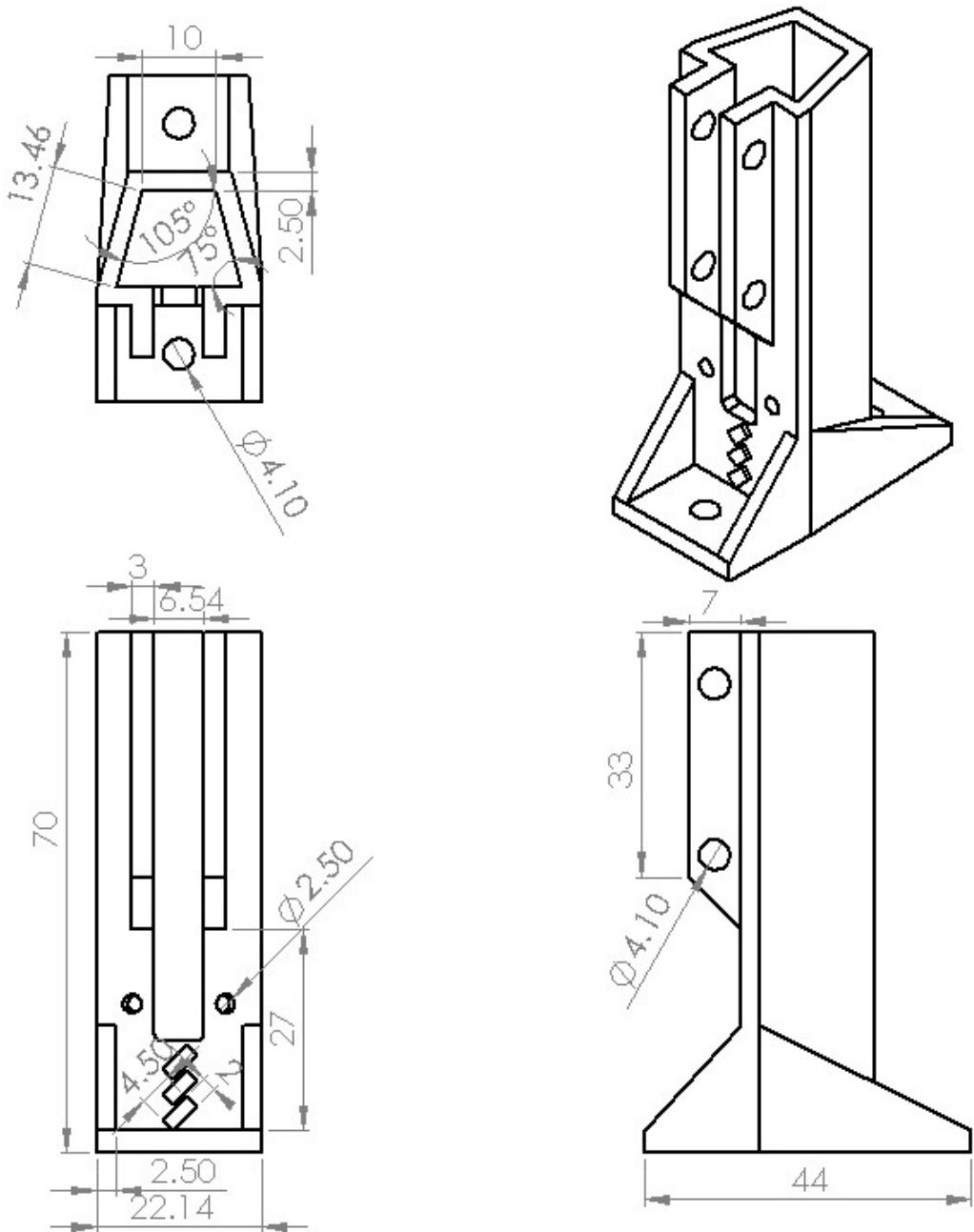
- [18] S. S. Ben Laurie, Peter Laurie, *"Apache: The Definitive Guide"* 3rd Edition. O'Reilly Media, Inc., 2002. pp. 1-5.
- [19] M. M. Antum Peicevic, *Apache HTTP Server introduction*. Geek University Press, 2016. pp. 4-6.
- [20] R. Nixon, *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5 (Learning Php, Mysql, Javascript, Css & Html5) 4th edition*. O'Reilly Media, Inc., 2015. pp 35-37.
- [21] O. C. and/or its affiliates. What is mysql? [Accessed: Sept. 2016]. [Online]. Available: <https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>
- [22] I. Corporation, “Atx specification version 2.1,” June 2002.
- [23] 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>
- [24] Mantech. (September 2016) Mantech electronics. [Online]. Available: <http://www.mantech.co.za/>
- [25] H. King. (September 2016) Hobbyking. [Online]. Available: <http://www.hobbyking.com/hobbyking/store/index.asp>
- [26] A. Bowyer. (September 2016) Reprap. [Online]. Available: <http://reprap.org/>
- [27] R. C. (SA). (September 2016) Rs-online. [Online]. Available: <http://za.rs-online.com/>
- [28] B. Morgan. (September 2016) Vertical roller v 1 test part 1. [Online]. Available: <https://youtu.be/nNe5YlRPoKM>
- [29] B. Morgan. (September 2016) Vertical roller v 1 test part 2. [Online]. Available: <https://youtu.be/4hIJvzhvt88>
- [30] B. Morgan. (September 2016) Vertical roller v 2 test. [Online]. Available: <https://youtu.be/DLGEwFN1RbM>
- [31] B. Morgan. (September 2016) New horizontal roller test. [Online]. Available: https://youtu.be/_Ol1D7sp83Q
- [32] R. P. Foundation. (october 2016) Raspbian download. [Online]. Available: <https://www.raspberrypi.org/downloads/raspbian/>
- [33] J. D. Tobin Davis. (october 2016) Win32 disk imager. [Online]. Available: <https://sourceforge.net/projects/win32diskimager/>

Appendix A Research Mind Map

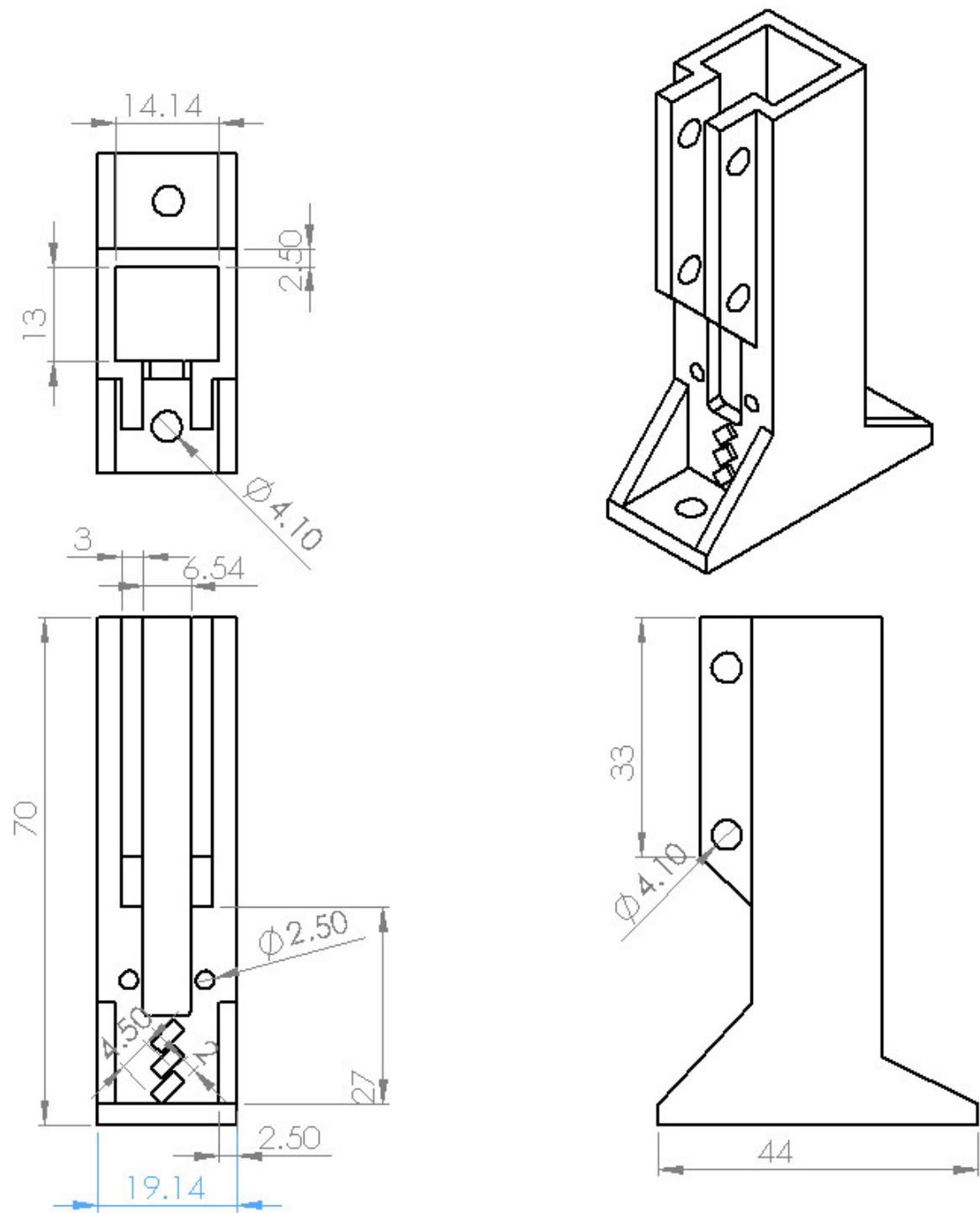


Appendix B Detailed Drawings

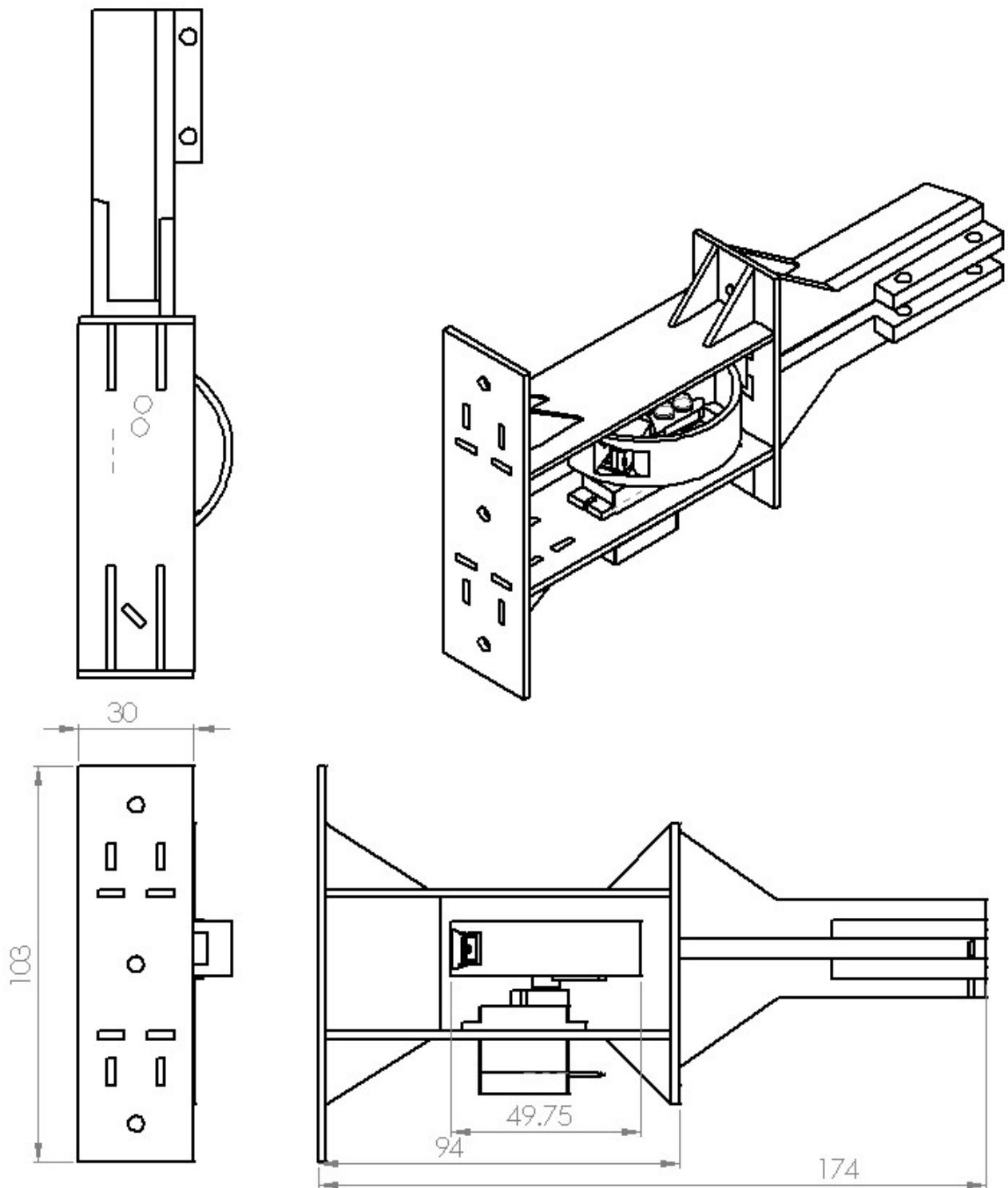
B.1 IC Guide DIP Components



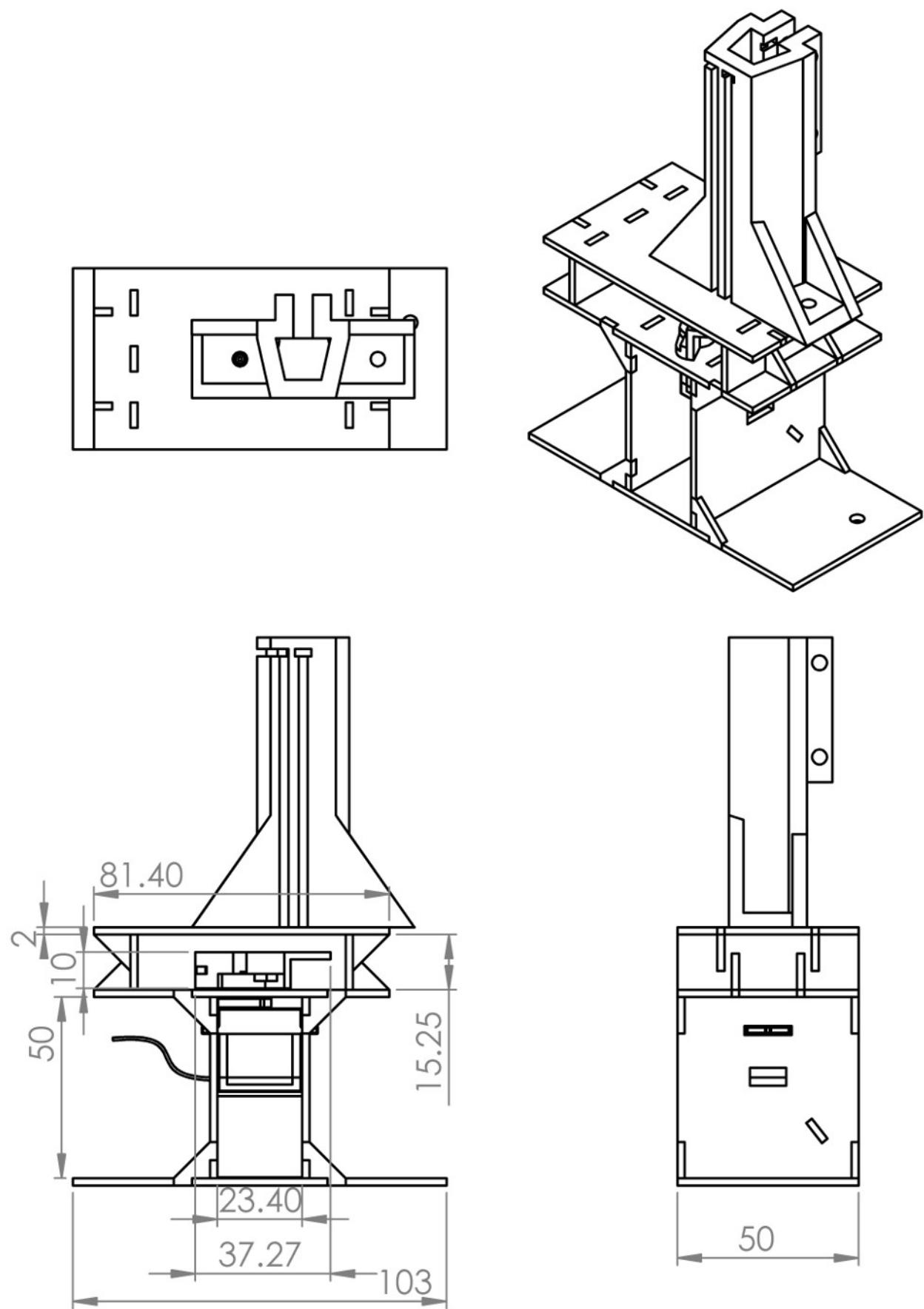
B.2 IC Guide DIP Holder



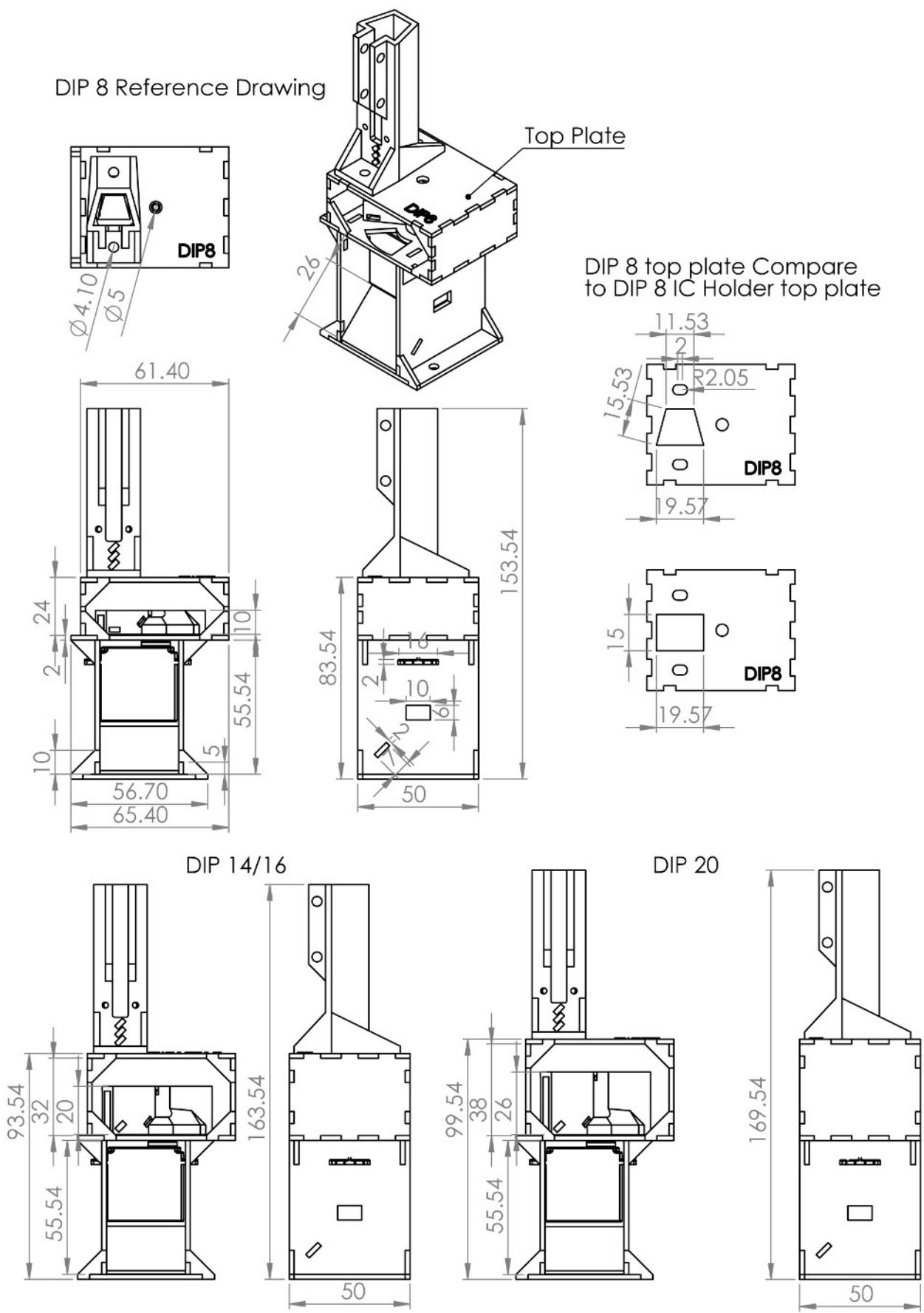
B.3 Vertical Roller & Housing Version 1 & 2



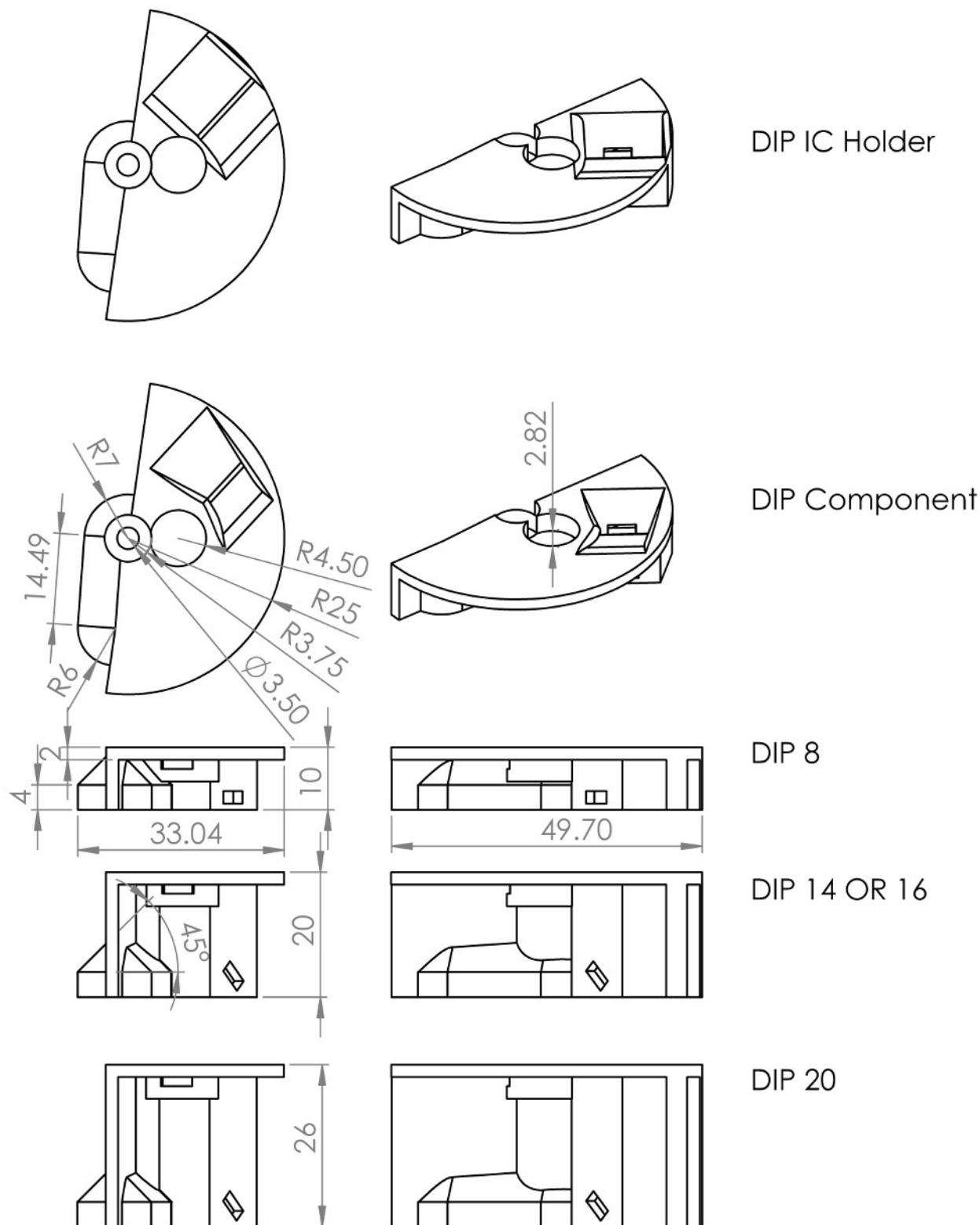
B.4 Horizontal Roller & Housing Version 1



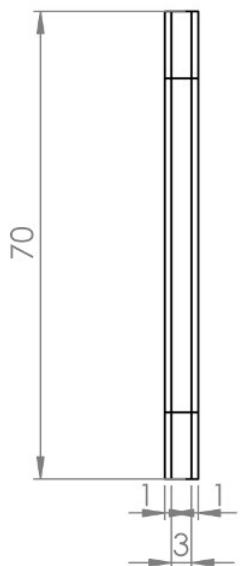
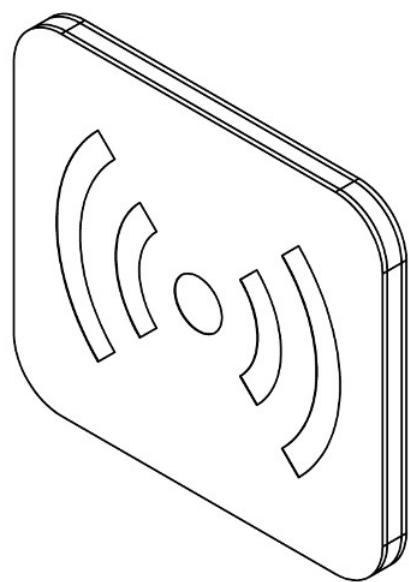
B.5 Horizontal Roller & Housing Final Version



B.6 Horizontal Roller Final Version

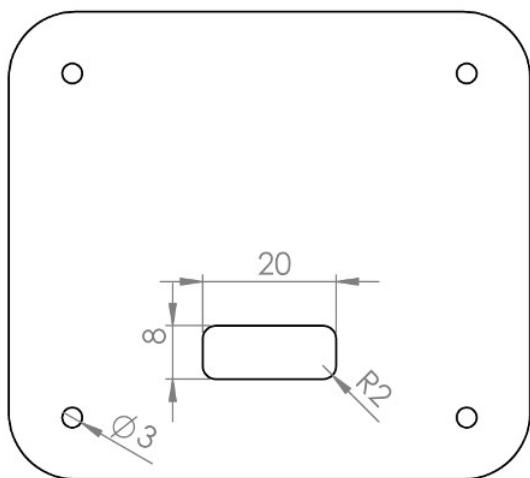
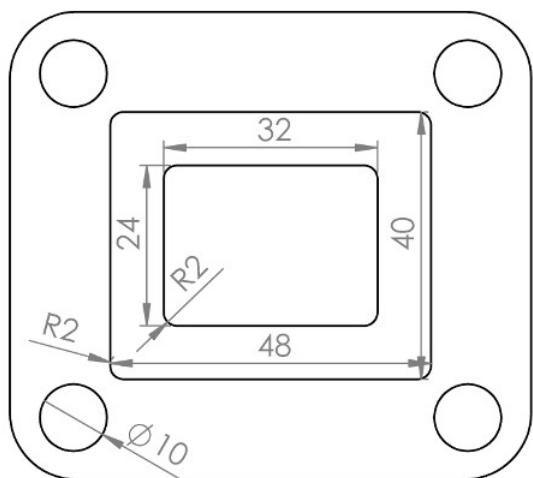


B.7 RFID Coil Cover



Middle Plate

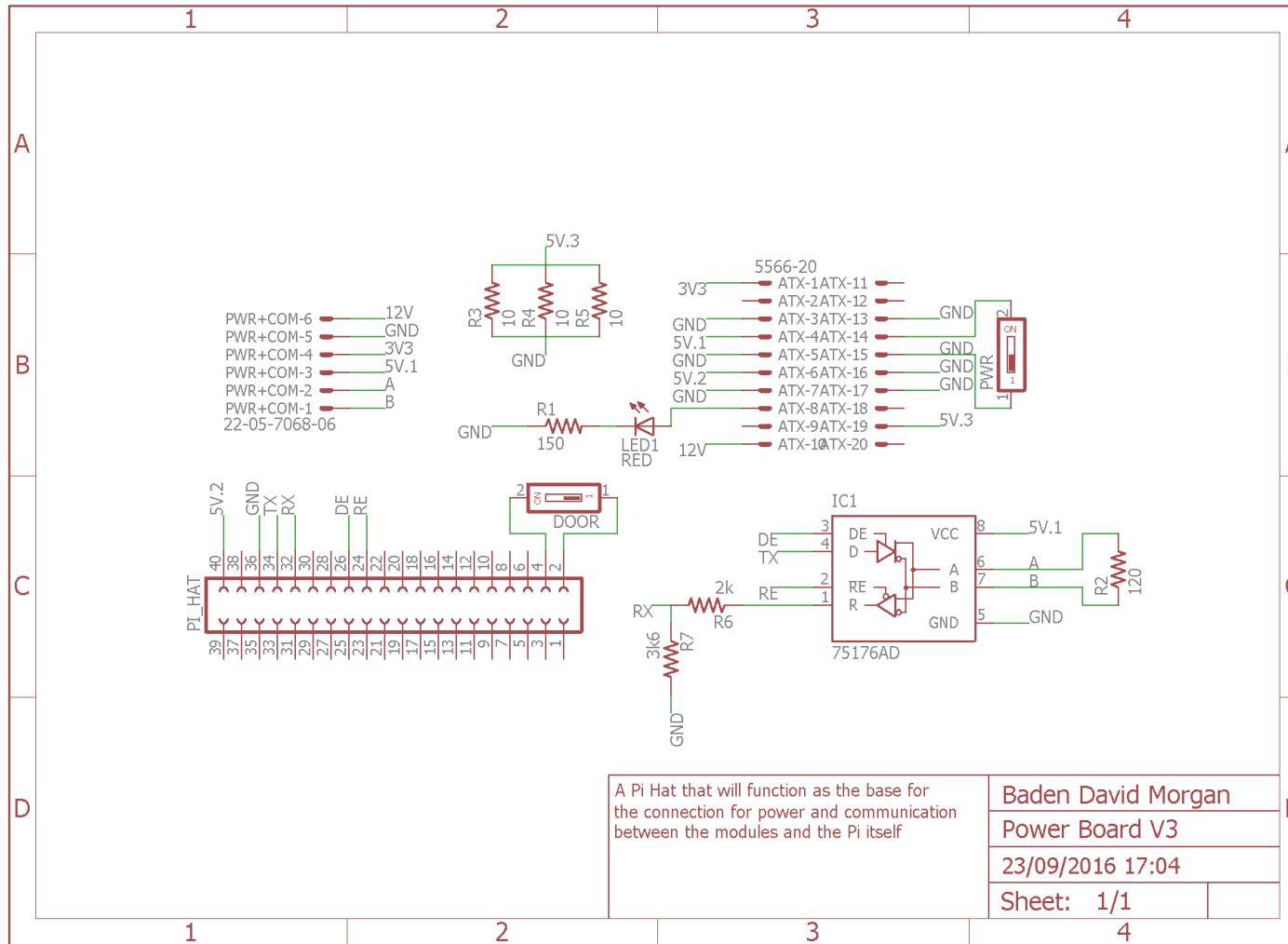
Back Plate



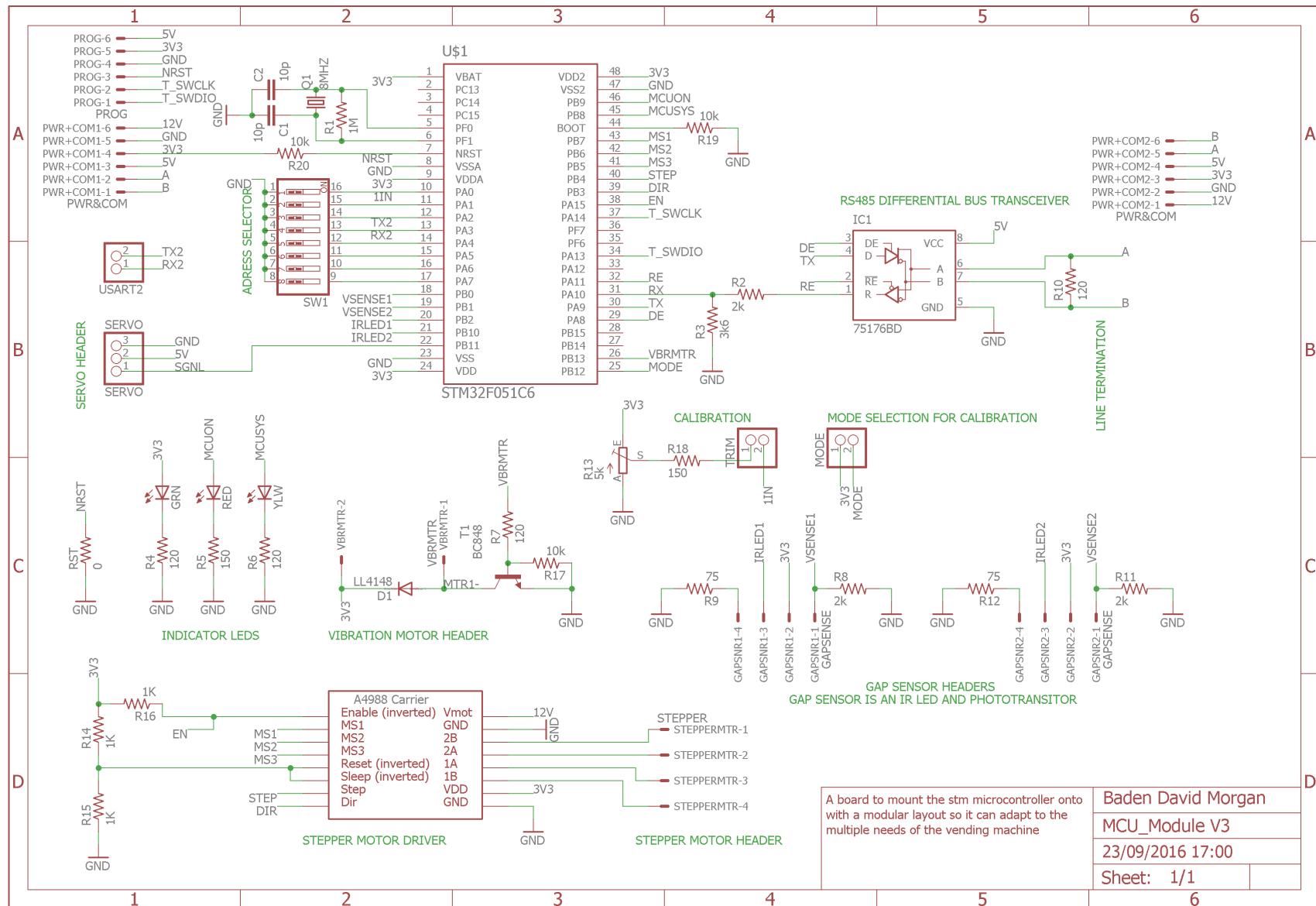
Appendix C PCB Diagrams

C.1 PI HAT Diagram

E9

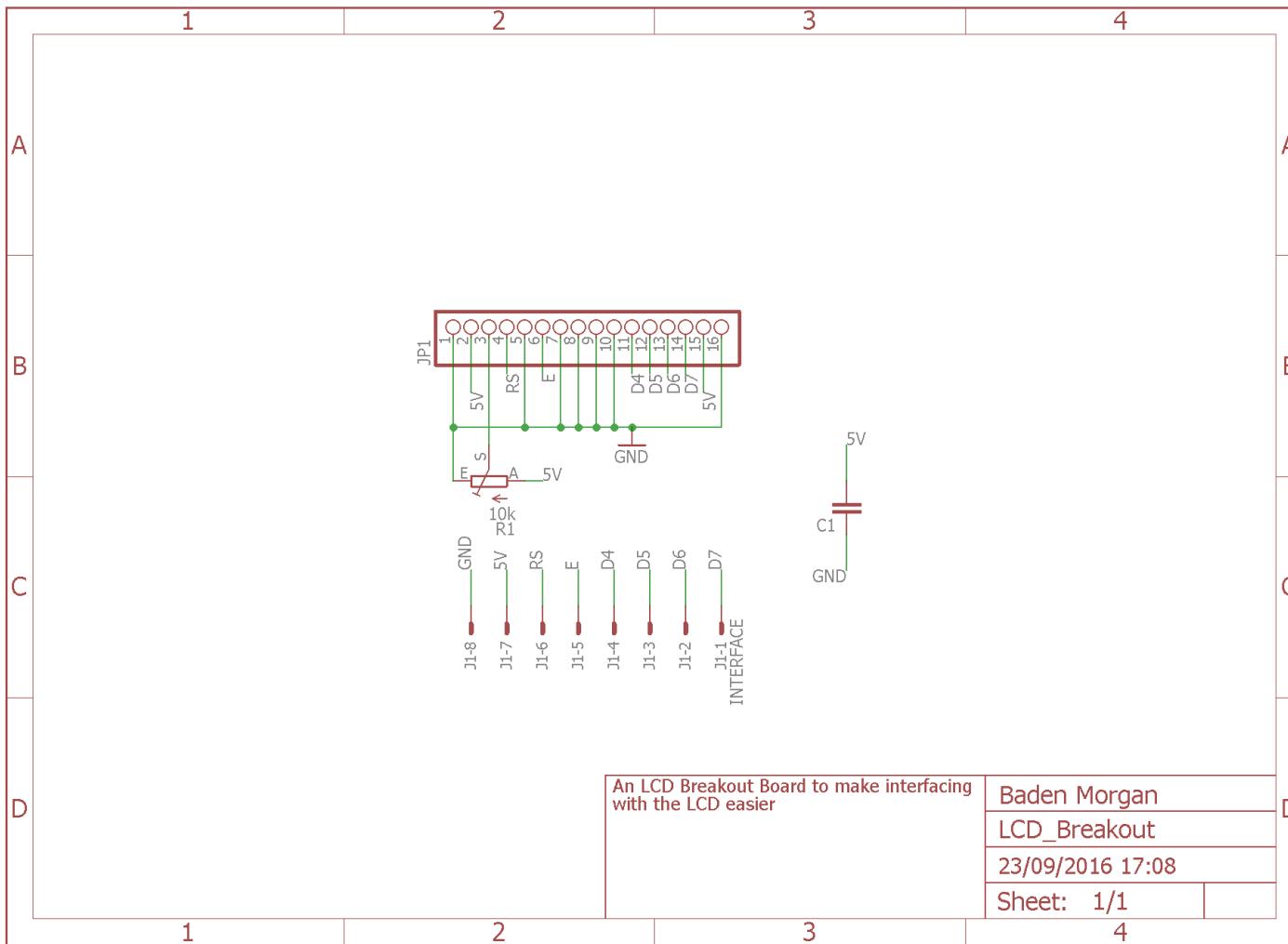


C.2 MCU Module Diagram



C.3 LCD Breakout Diagram

29



Appendix D Drawing Templates

D.1 Material Request Forms

Aluminium Sheets with thickness 1mm

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

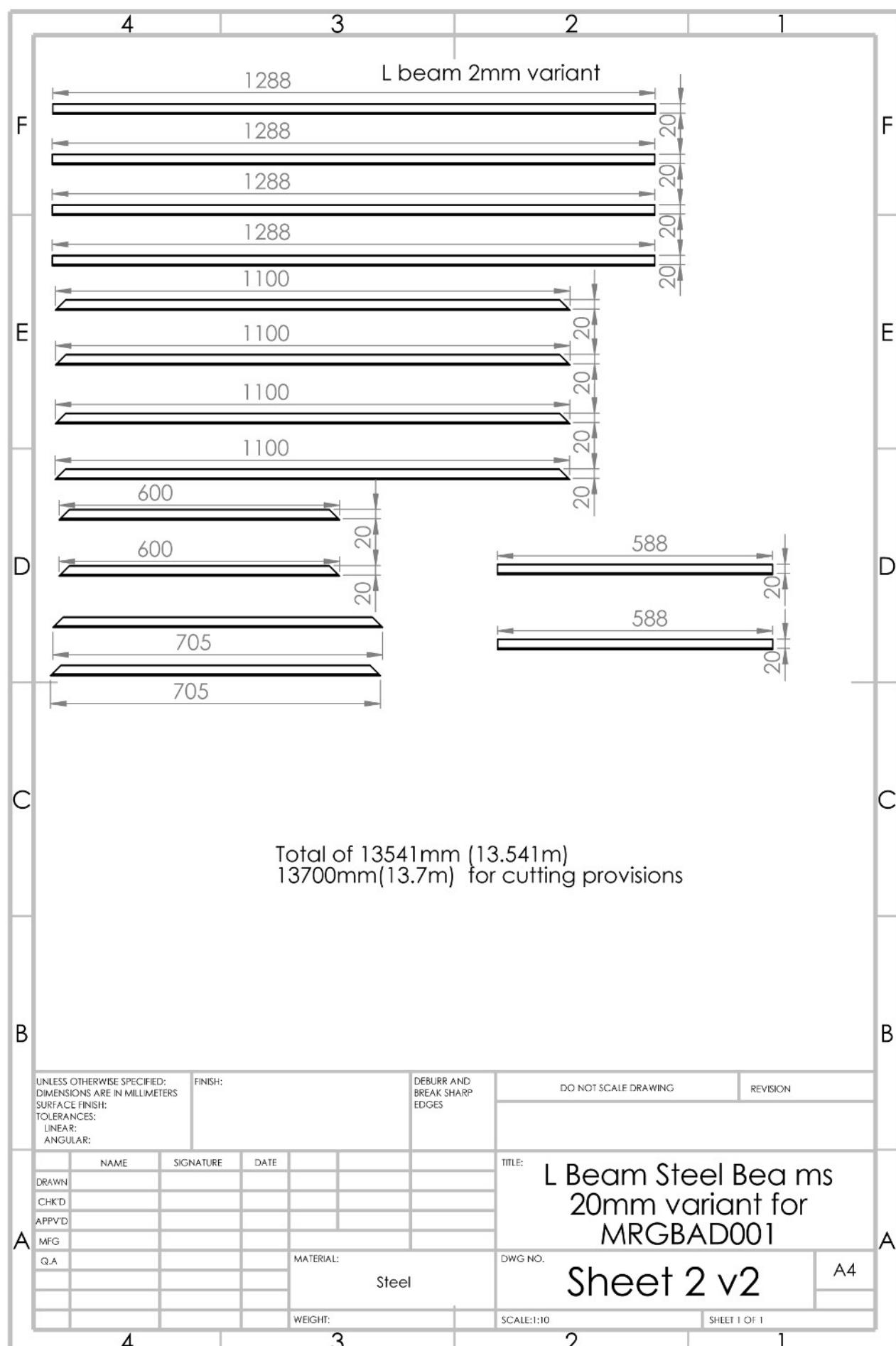
| | | | |
|--------|------|-----------|------|
| DRAWN | NAME | SIGNATURE | DATE |
| CHK'D | | | |
| APPV'D | | | |
| MFG | | | |
| Q.A. | | | |

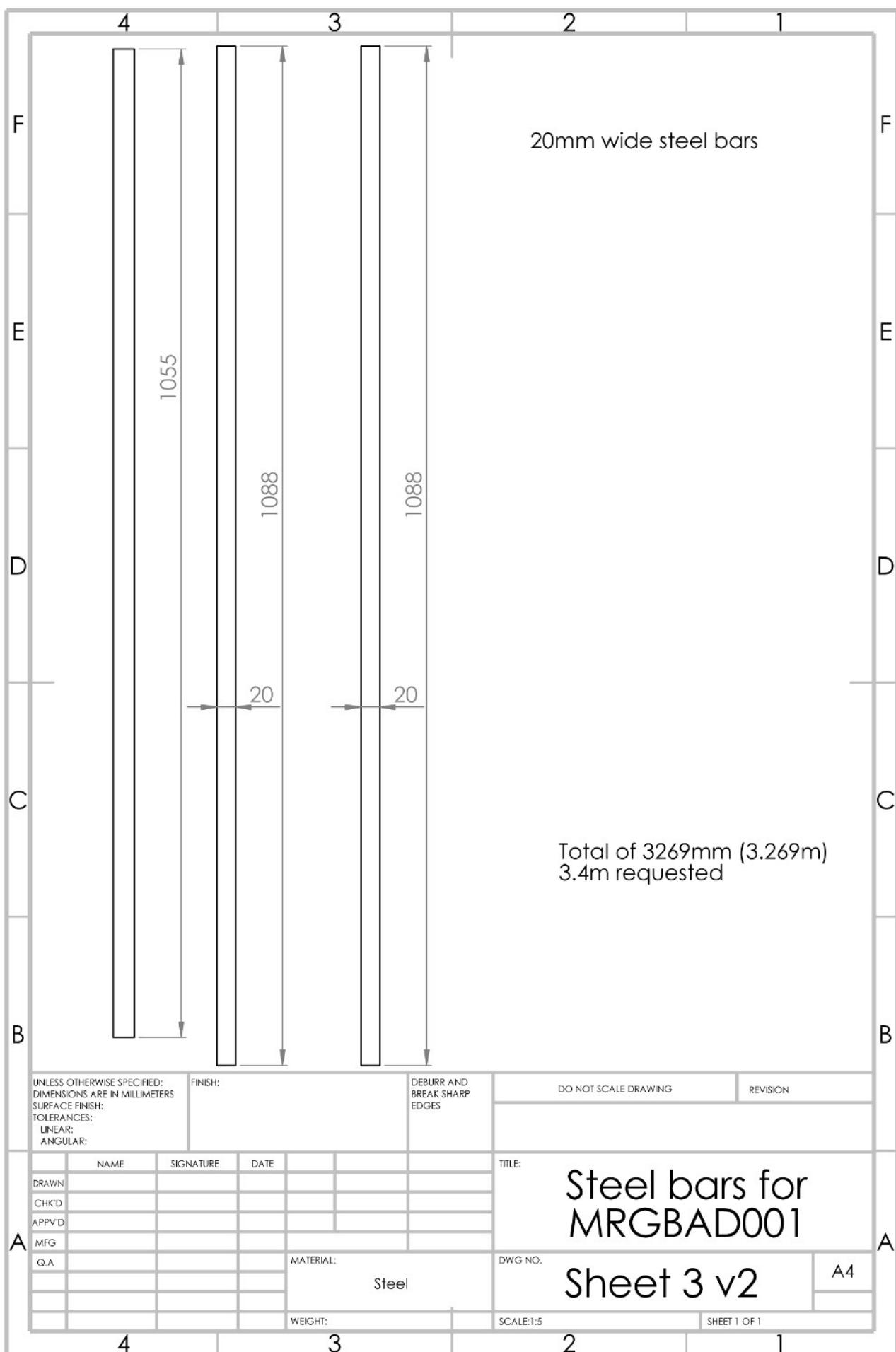
TITLE: Aluminium Sheets For MRGBAD001

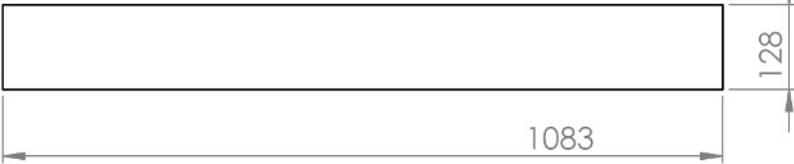
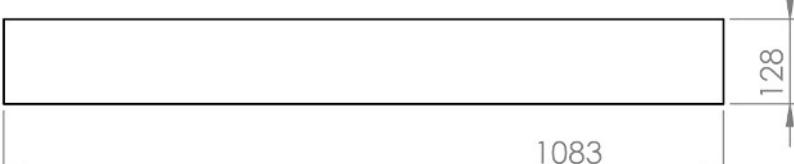
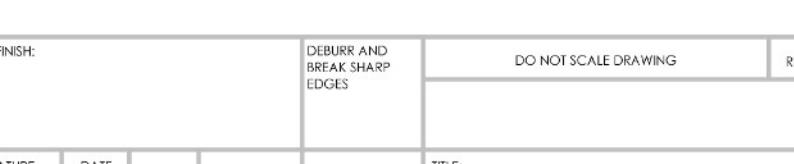
MATERIAL: Aluminium

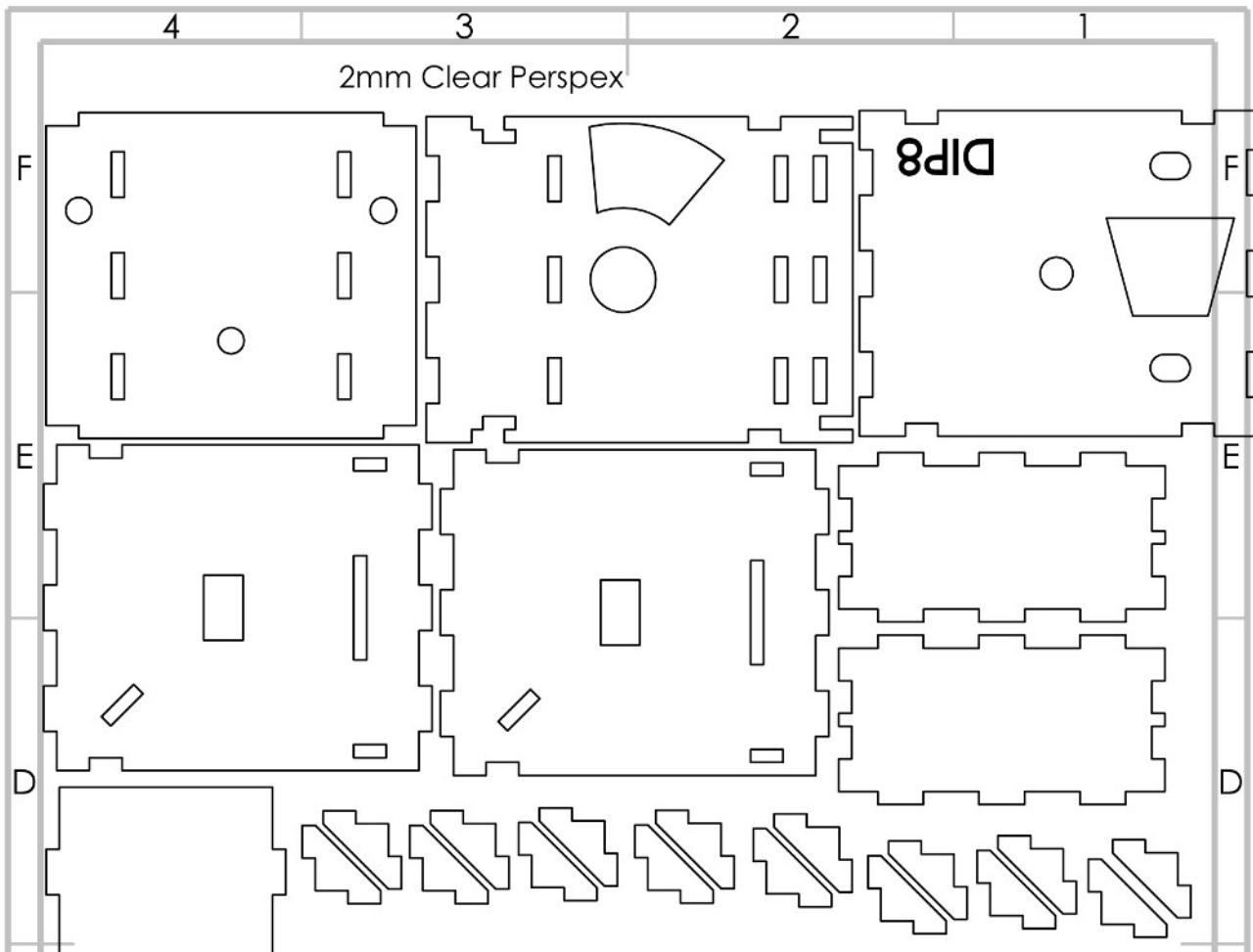
DWG NO. 1:20 SHEET 1 OF 1

A4





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



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 | |

1mm White Perspex

70

78

D

C

B

A

F

E

F

E

D

C

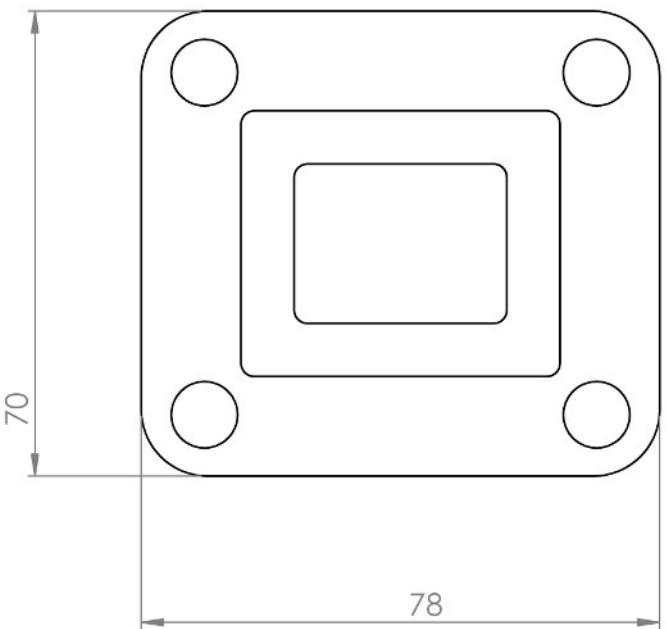
B

A

1mm non transparent white perspex 200 x 200

| | | | | | | | | |
|---|------|-----------|------|--|---|--|--|----------|
| 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 | | | | | | | | |
| QA | | | | | | | | |
| | | | | | MATERIAL: Non Transparent 1mm White Perspex | TITLE: 1mm White Perspex for MRGBAD001 | | DWG NO. |
| | | | | | WEIGHT: | sheet 6 | | A4 |
| | | | | | SCALE:1:1 | SHEET 1 OF 1 | | |

4 3 2 1

| | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|-----------|------------------------------|-------|------|-----------|------|-------|--|--|--|-------|--|--|--|-----|--|--|--|------|--|--|--|--|--|--|--|
| 4 | 3 | 2 | 1 | | | | | | | | | | | | | | | | | | | | | | | | |
| F | | | F | | | | | | | | | | | | | | | | | | | | | | | | |
| E | 3mm Non Transparent White Perspex | | | | | | | | | | | | | | | | | | | | | | | | | | |
| D |  | | | | | | | | | | | | | | | | | | | | | | | | | | |
| C | 3mm sheet of Non transparent White perspex 90x90 | | | | | | | | | | | | | | | | | | | | | | | | | | |
| B | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| A | UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR: <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <tr> <td style="width: 25%;">DRAWN</td> <td style="width: 25%;">NAME</td> <td style="width: 25%;">SIGNATURE</td> <td style="width: 25%;">DATE</td> </tr> <tr> <td>CHK'D</td> <td></td> <td></td> <td></td> </tr> <tr> <td>APP'D</td> <td></td> <td></td> <td></td> </tr> <tr> <td>MFG</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Q.A.</td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </table> | | | DRAWN | NAME | SIGNATURE | DATE | CHK'D | | | | APP'D | | | | MFG | | | | Q.A. | | | | | | | |
| DRAWN | NAME | SIGNATURE | DATE | | | | | | | | | | | | | | | | | | | | | | | | |
| CHK'D | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| APP'D | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MFG | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Q.A. | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR: | | FINISH: | DEBURR AND BREAK SHARP EDGES | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | DO NOT SCALE DRAWING | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | REVISION | | | | | | | | | | | | | | | | | | | | | | | | |
| TITLE: 3mm White Perspex for MRGBAD001 MATERIAL: 3mm Non Transparent White Perspex DWG NO. Sheet 7 WEIGHT: SCALE:1:1 SHEET 1 OF 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | 3 | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | 2 | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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, transistors, and integrated circuits. A navigation bar at the bottom left includes links for "Home", "Contact", and "UCT".

UCT White Lab Vending Machine

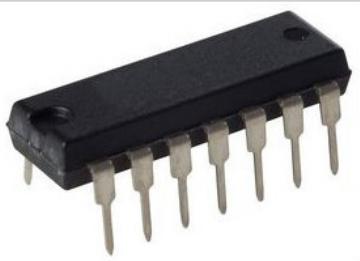
How to use: Simply use the drop down boxes to select the quantity of components you want of each type(maximim of 6 ie the first 6 you pick, the rest will be ignored). Once you have selected your components enter your student number in the field at the bottom of the page and hit submit(Double check it, 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:

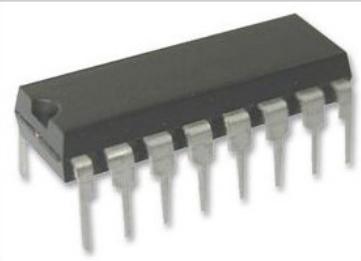
| | | |
|---|--|--|
| 555 Timer Datasheet quantity: 0 | LM311 Datasheet Out of Stock | LM393 Datasheet Out of Stock |
| LM741 Datasheet Out of Stock | LM358 Datasheet Out of Stock | LM339 Datasheet quantity: 0 |

LM324



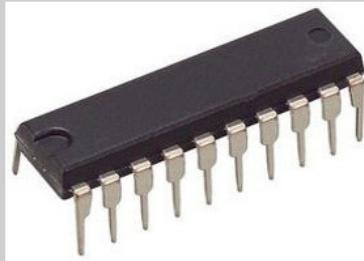
[Datasheet](#) | Out of Stock

SN7475



[Datasheet](#) | Out of Stock

T74LS241B1



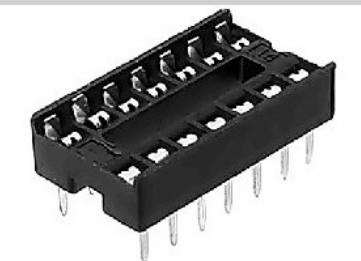
[Datasheet](#) | Out of Stock

DIP 8 IC Holder



quantity:

DIP 14 IC Holder



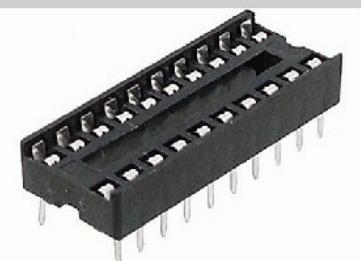
Out of Stock

DIP 16 IC Holder



Out of Stock

DIP 20 IC Holder



quantity:



*Student Number

submit

E.2 Contacts Page



The image shows a collection of electronic components like resistors, capacitors, and integrated circuits, all arranged around a central circular logo of the University of Cape Town. The logo features a shield with a book, a quill, and an anchor, with the text "SPES BONA" and "UCT". Below the shield is the university's name in Afrikaans and English.

Home Contact UCT

Contacts Page

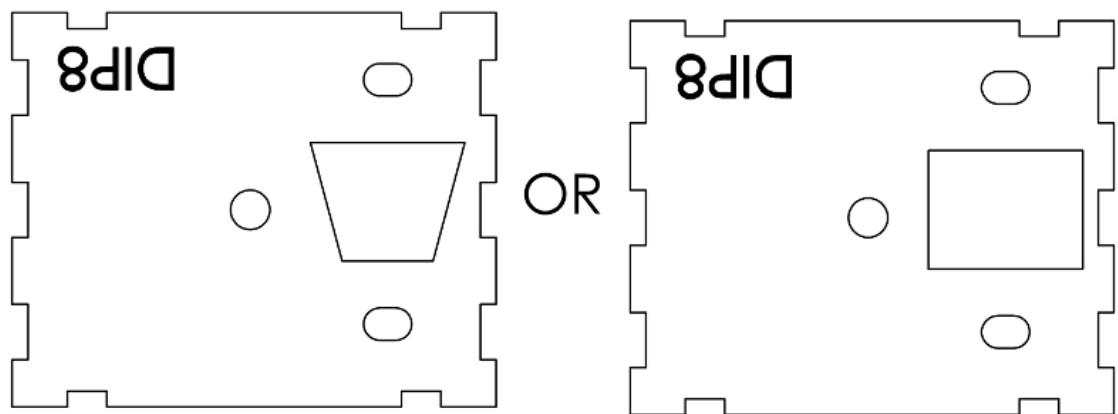
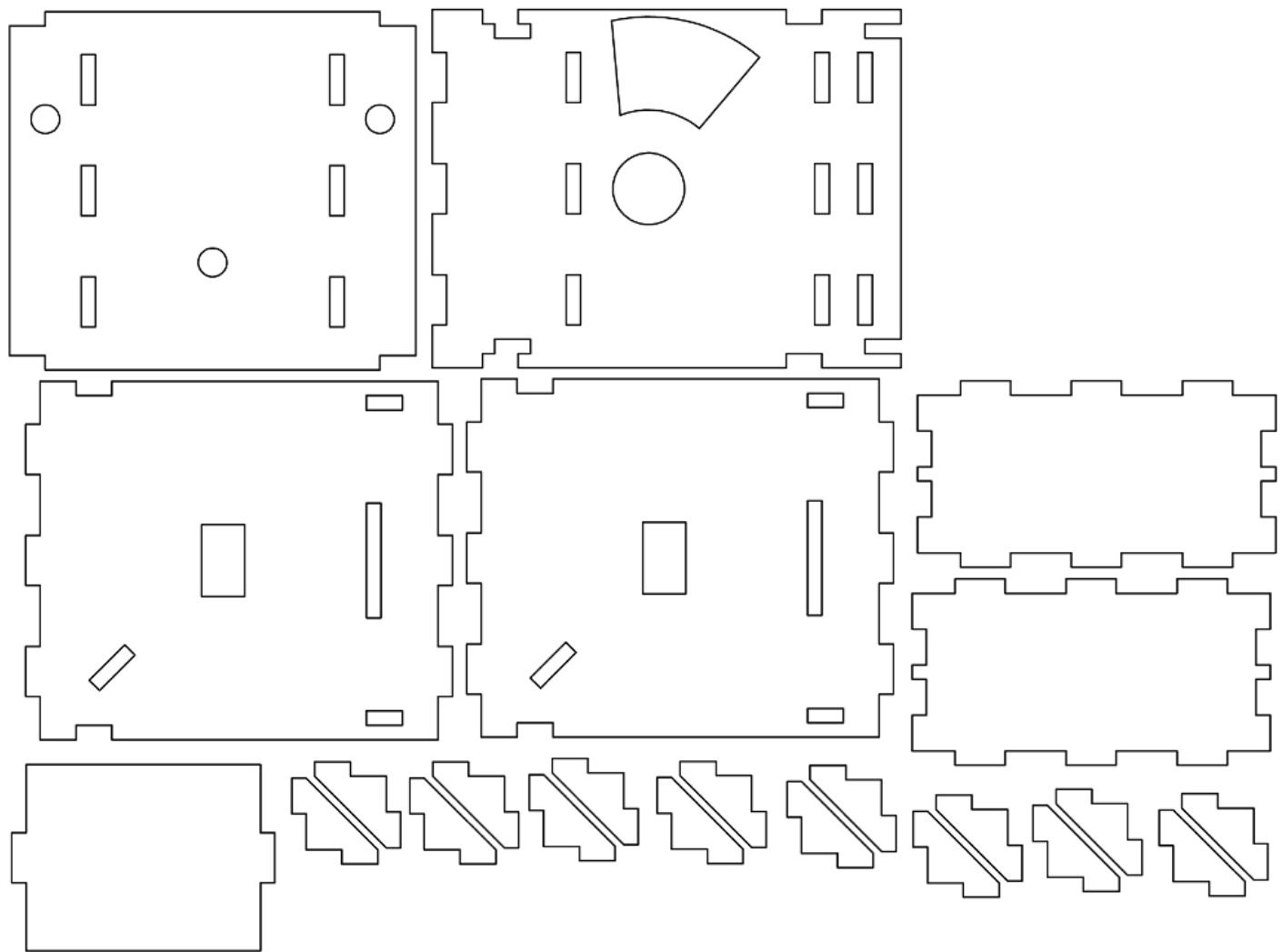
Contact us for a query, problems with the vending machine, suggestions or to just say hello.

Please fill in all the details below

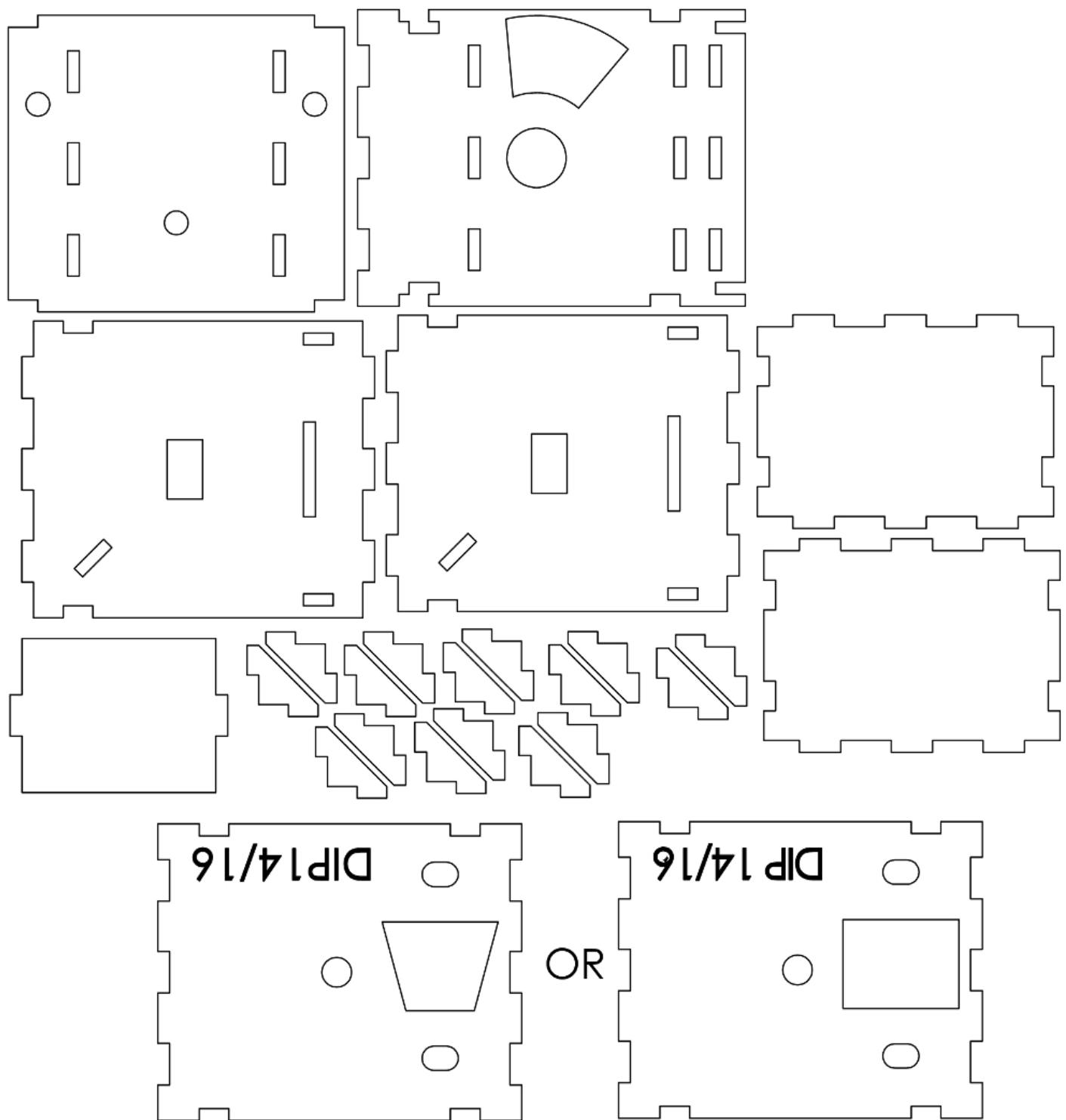
*Name
 *Student Number
 *Email
 *Message

E.3 Laser Cutting Templates

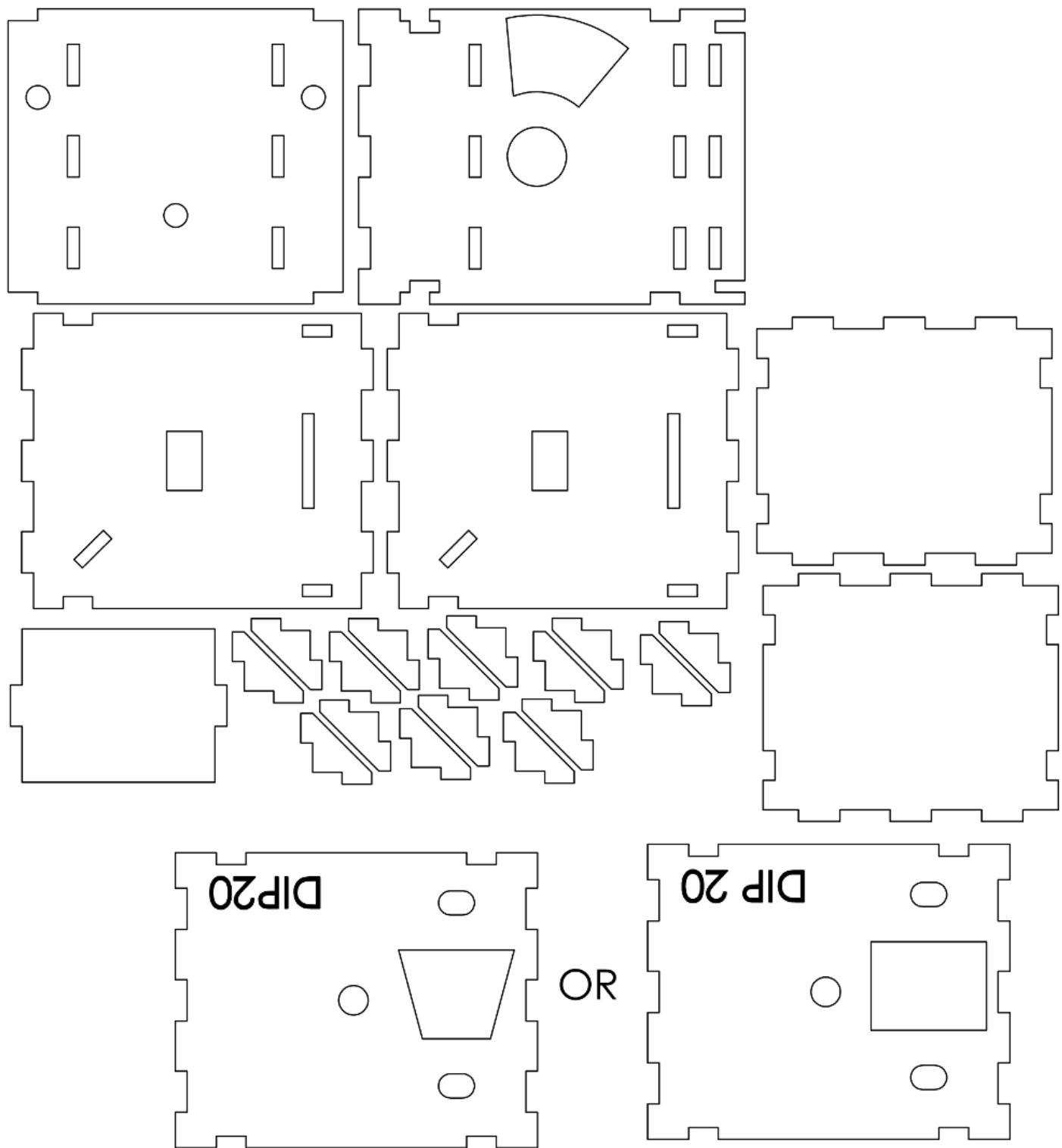
E.3.1 DIP 8 Template



E.3.2 DIP 14 or 16 Template



E.3.3 DIP 20 Template



E.4 Drawings for Assembly: Enclosure

Appendix F Manuals and Guides

F.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 level 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) use the Calibration potentiometer calibrate the update the values for dispensary actions.
 - (c) Play with this action to see what values change to learn what the action does. It is fairly self explanatory once observed.
 - (d) 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".
 - (e) after 15s calibration for the pickup value will begin.
 - (f) Record the value on the serial monitor and enter it as the new value for "pickupINIT" in definitions.h on line 31.
 - (g) 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.
 - (h) Place terminal short back on "Mode" selection
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.
12. If everything went well the board is ready to be installed in the vending machine.
13. If there is an issue it is more than likely the board has not been assembled correctly, go back and check the tracks and components again before trying to upload anymore code.

Appendix G Terminal Commands

This Appendix server to present the commands used to create a functional LAMP server to run the website on, the commands are listed in the order they should be executed:

Update software repository:

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

Install Apache web server:

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

Type in IP address of Pi into browser to test the Apache server. One should see a Apache information page. 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 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, 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 specific 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:

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

Add the following to the bottom of this configuration file:

```
21 <IfModule mod_dir.c>
22     DirectoryIndex HomePage.php
23 </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. Now to install the mail server to enable admin notifications via email:

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

Follow the on screen instructions. First select "mail sent by smarthost; received via SMTP or fetch-mail". 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:

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

Add the following to the bottom of the file:

```
27 gmail-smtp.l.google.com:uctvendingmachine@gmail.com
      :uctvm2016
28 *.google.com:uctvendingmachine@gmail.com:uctvm2016
29 smtp.gmail.com:uctvendingmachine@gmail.com:
      uctvm2016
```

Ctrl+x then Y and enter to save and exit. Send a test email on the contacts page of the website.

Appendix H Ethics Forms