

Design of a White Lab Component Vending

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: this and that

Plagiarism Declaration

1. I, Baden David Morgan, know that plagiarism is wrong. Plagiarism is to use anothers work and pretend that it is ones 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
2	Literature Review	2
2.1	Communication between devices	2
2.1.1	I2C	2
2.1.2	RS-485/TIA-485	2
2.2	RFID Reader	3
2.2.1	System Specification	3
2.2.2	UCT's RFID Solution	4
2.3	Website, Hosting and Server	4
2.3.1	The Web server: Apache vs. Nginx	4
2.3.2	The Scripting Language: PHP	5
2.3.3	The Database: MySQL	6
3	System Specifications	7
4	Design and Prototyping Methodology and Procedure	8
4.1	Design	8
4.1.1	Mechanical Design	8
4.1.2	Circuit Design Methodology	8
4.1.3	Software Design Methodology	9
4.1.4	Prototyping Methodology and Procedure	9

5	System Design and Prototyping	10
6	System Assembly	11
7	Build Review, Results and Discussion	12
8	Conclusion	13
9	Recommendations	14
A	Research Mind Map	16

List of Figures

2.1	External Low Pass Noise Filter Circuit[1]	2
2.2	When idle one can see the line is held high, this is seen before data is sent and after [2].	3
2.3	The RDM 6300 Module with an external antenna connected directly to the module [3].	4
2.4	Graph showing active websites and their backend web server being used [4].	5
3.1	Mind map used to help assist the definition of the topic, system specifications and design of the Vending Machine.	7

List of Tables

Nomenclature

HAT	Hardware Attached onTop
HTTP	Hypertext Transfer Protocol
I2C	Inter-integrated Circuit
IP	Internet Protocol
PCB	Printed Circuit Board
PHP	Hypertext Preprocessor
RFID	Radio Frequency Identification
RX	Receiver Number x
TX	Transmitter number x
UART	Universal Asynchronous Receiver/Transmitter
URL	Uniform Resource Locator

1 Introduction

Well, and here begins my lovely article.

1.1 Subject and motivation for the Research

meh

1.2 Background to the Research

adding it here

1.3 Objectives of this Research

adding more

1.3.1 Problems to be investigated

a little here

1.3.2 Purpose of this study

a little here

1.4 Scope and Limitations of the Research

something something

1.5 Plan of Development

blah blah

2 Literature Review

introduction to chapter here

2.1 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 slaves was needed.

2.1.1 I2C

I2C is 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 [1] 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 a 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 $R_{pullup} = 1800\Omega$, $R_{s1} = 130\Omega$, $R_{s2} = 51\Omega$, $C_f = 180Pf$ [1].

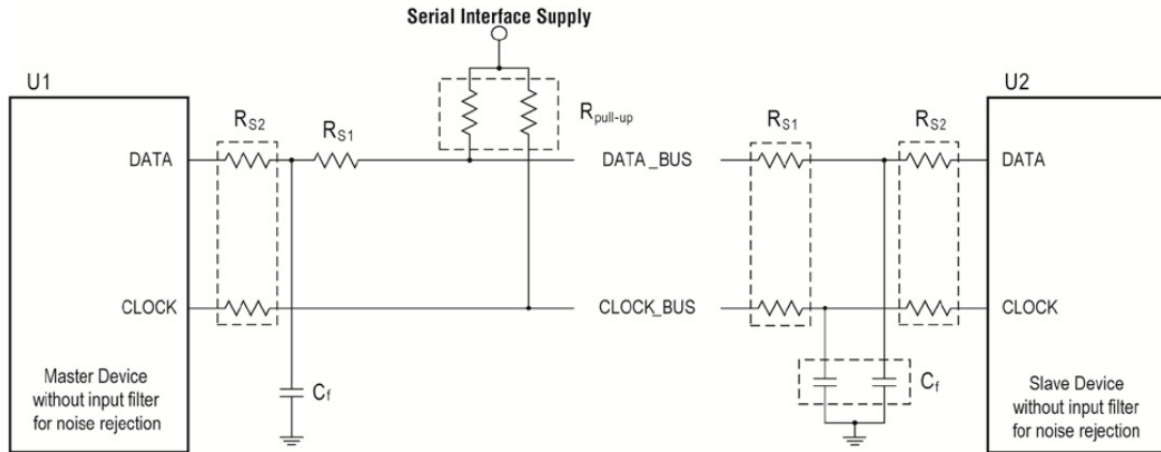


Figure 2.1: External Low Pass Noise Filter Circuit[1]

2.1.2 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 [5]), 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 where 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 [5]. UART can not be used alone even with shielded cables because both devices on the bus hold their TX lines high as shown in the picture below. This makes it impossible for a 3 device to connect and potentially corrupting the bus entirely.

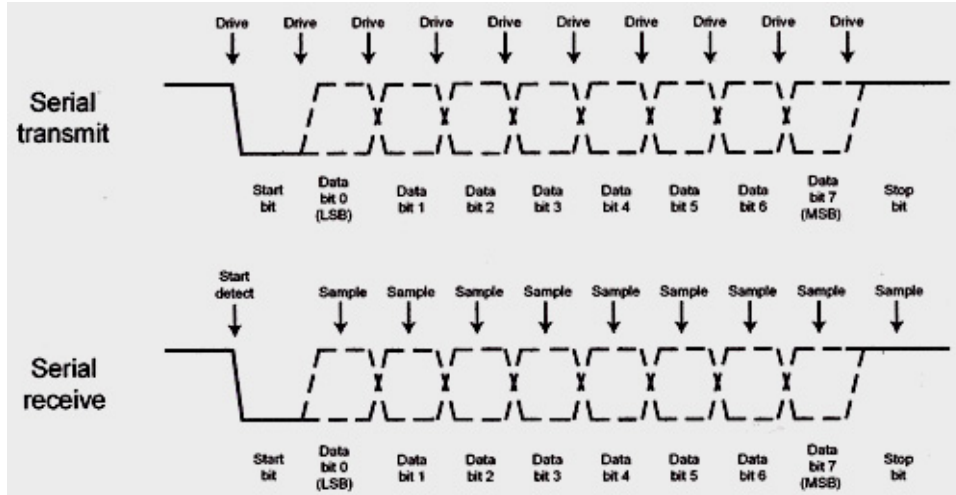


Figure 2.2: When idle one can see the line is held high, this is seen before data is sent and after [2].

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 [6].

2.2 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.

2.2.1 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) [7]. 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 refereed to as the item to be tracked or the identifier in this case the student card. The tag has on 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 he 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 [8].

2.2.2 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 [9].

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 [3]. 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. Below is a picture of the above mentioned RDM 6300 with an external antenna attached:

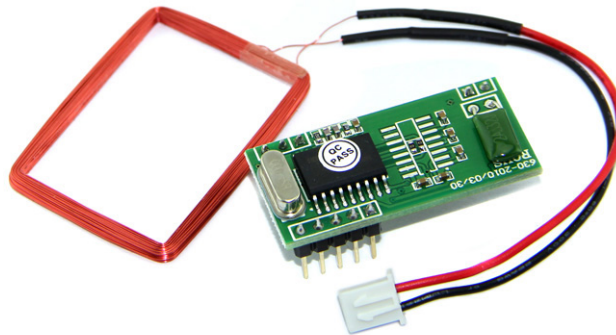


Figure 2.3: The RDM 6300 Module with an external antenna connected directly to the module [3].

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 [10].

2.3 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 [11].

2.3.1 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 [12]. 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. Although it has shown a steady decline in market share since 2011 it is still the leader in web servers [4]. 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.

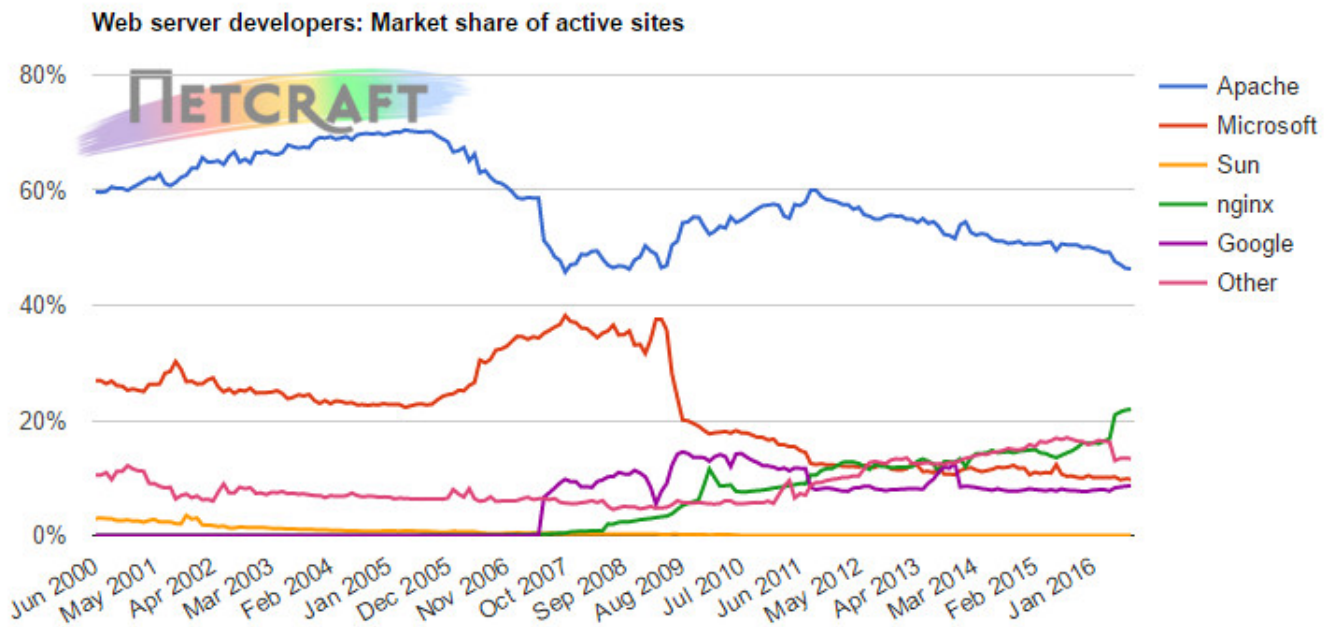


Figure 2.4: Graph showing active websites and their backend web server being used [4].

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 reliably. 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 [13, 14].

2.3.2 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.

2.3.3 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 use 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 base 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 [15].

3 System Specifications

In order to begin designing the Vending Machine the specifications needed to be more 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 and the Raspberry pi module plugin, The software for the master program i.e. the Raspberry Pi, the control boards and the website. A mind map was created to assist defining the specifications and can be seen below. It shows the relevant specifications for each three sections and each and how they interact and will also be discussed in detail below:

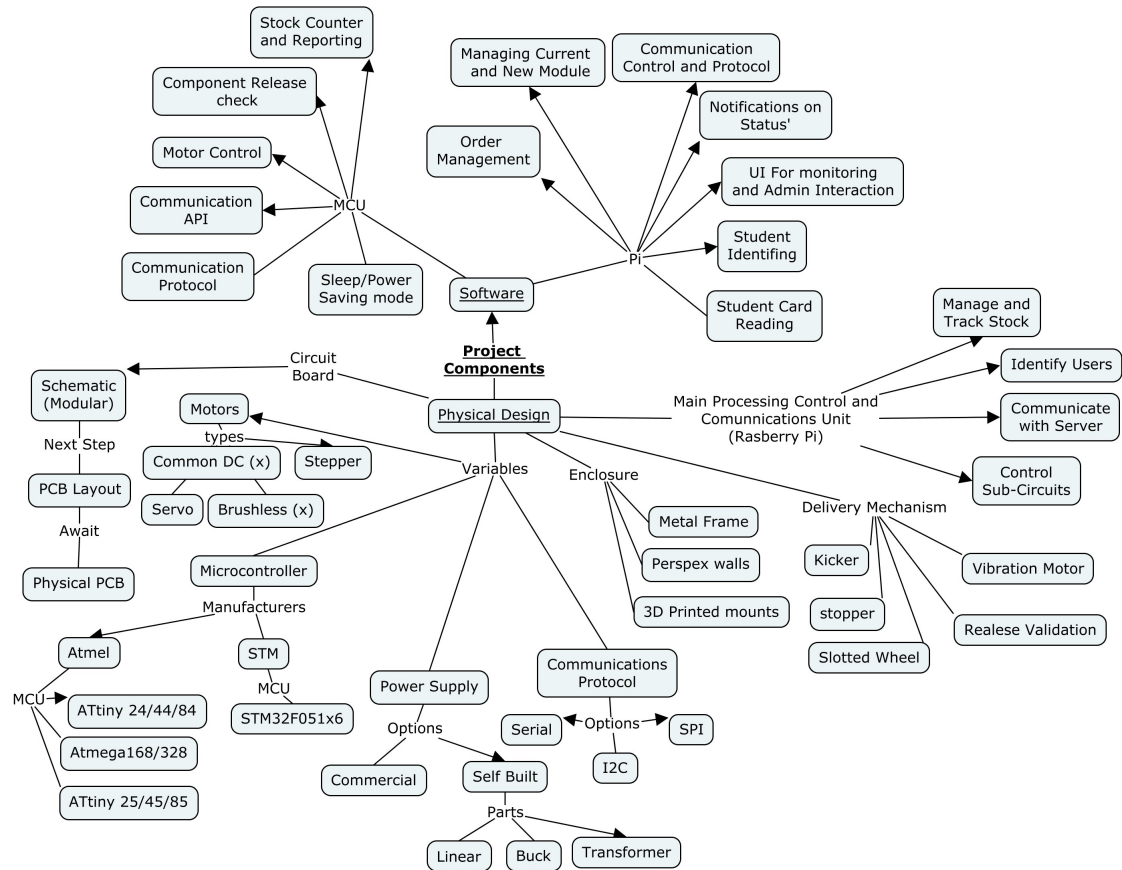


Figure 3.1: Mind map used to help assist the definition of the topic, system specifications and design of the Vending Machine.

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.

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 introductory.

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

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 forward thinking of "how would it fail" were always a 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. where 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 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 updates were needed until a working code base was achieved.

4.1.4 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 System Design and Prototyping

6 System Assembly

7 Build Review, Results and Discussion

8 Conclusion

9 Recommendations

Apache is losing market share which could mean its counter part may become the leader in the future. More time should be invested in researching the benefits of using Nginx over Apache

References

- [1] L. Semiconductor, “Improving noise immunity for serial interface,” http://www.latticesemi.com/~media/LatticeSemi/Documents/WhitePapers/HM/ImprovingNoiseImmunityforSerialInterface.pdf?document_id=50728, 2014, pp. 2, [Accessed: Sept. 2016].
- [2] A. Minor, “Implementing your mcu-based system’s serial uart in software,” <http://www.embedded.com/design/other/4025995/Implementing-your-MCU-based-system-s-serial-UART-in-software>, 2007, [Accessed: Sept. 2016].
- [3] Seedwiki. (2016) 125khz rfid module - uart. http://wiki.seeedstudio.com/wiki/125Khz_RFID_module-_UART. [Accessed: Sept. 2016].
- [4] Netcraft, “August 2016 web server survey,” <https://news.netcraft.com/archives/2016/08/24/august-2016-web-server-survey.html>, 2016, [Accessed: Sept. 2016].
- [5] T. Kugelstadt, “The rs-485 design guide,” <http://www.ti.com/lit/an/slla272b/slla272b.pdf>, 2008, [Accessed: Sept. 2016].
- [6] J. Goldie, “Low-voltage differential signaling yields megatransfers per second with milliwatts of power,” <http://www.ti.com/lit/an/snla156/snla156.pdf>, 1996, pp. 3, [Accessed: Sept. 2016].
- [7] IMPINJ, “The different types of rfid systems,” <http://www.impinj.com//resources/about-rfid/the-different-types-of-rfid-systems/>, 2016, [Accessed: Sept. 2016].
- [8] IMPINJ. (2016) How do rfid systems work? <http://www.impinj.com/resources/about-rfid/how-do-rfid-systems-work/>. [Accessed: Sept. 2016].
- [9] J. Pead. (Sept. 2016). Principal Technical Officer [Online]. Available e-mail: justin.pead@uct.ac.za, Message: RFID Documentation.
- [10] Microchip, “Mcp2200 datasheet,” <http://ww1.microchip.com/downloads/en/DeviceDoc/22228B.pdf>, 2011, [Accessed: Sept. 2016].
- [11] 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.
- [12] O. Garret, “Nginx vs. apache: Our view of a decade-old question,” <https://www.nginx.com/blog/nginx-vs-apache-our-view/>, October 9, 2015, [Accessed: Sept. 2016].
- [13] S. S. Ben Laurie, Peter Laurie, “*Apache: The Definitive Guide*” 3rd Edition. O’Reilly Media, Inc., 2002. pp. 1-5.
- [14] M. M. Antum Peicevic, *Apache HTTP Server introduction*. Geek University Press, 2016. pp. 4-6.
- [15] O. C. and/or its affiliates, “What is mysql?” <https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>, [Accessed: Sept. 2016].

Appendix A Research Mind Map

