Design of a keyless storage system using a raspberry pi to solve the security problem in a supermarket.

**S.I. Mkhonza**

Group F

A project submitted in compliance with the

Requirements for the subject Design Project 3

Partial fulfillment of the requirements for the qualification

National Diploma: Engineering: Computer System to **the**

**Department: Electronic Engineering**

**Vaal University of Technology**

Vanderbijlpark

South Africa

First semester 2018

Mr. M Viljoen

## Declaration

I, Sizwe-se-Afrika declare that this project is my own, unaided work. It is being submitted for the National Diploma: Engineering: Computer Systems to the Department of Electronic Engineering at the Vaal University of Technology, Vanderbijlpark. It has not been submitted before for any assessment to any educational institution.



Sizwe-se-Afrika Mkhonza

Signed: 25-02-2018

## Acknowledgments

*Many thanks to all those who helped with this project, including the many students in the National Diploma: Engineering: Computer Systems and to the Department of Electronic Engineering at the Vaal University of Technology, Vanderbijlpark, who always help me see complex concepts through beginner’s eyes,* *and we thank two “anonymous” Logic design lab assent’s, and Mr. M. Viljoen for reviewing our work and for their so-called insights. Lastly I would like to thank my three group members, for their contribution towards the research.*

## Dedication

In memory of my Mom, your guidance and love shall not be forgotten...I Love you’ mama, and thank you for everything.

## Table of Contents

[Design of a keyless storage system using a raspberry pi to solve the security problem in a supermarket. 1](#_Toc513660917)

[Declaration 2](#_Toc513660918)

[Acknowledgments 3](#_Toc513660919)

[Dedication 4](#_Toc513660920)

[Table of Contents 5](#_Toc513660921)

[List of Figures 7](#_Toc513660922)

[List of Tables 8](#_Toc513660923)

[List of Abbreviations 9](#_Toc513660924)

[Chapter 1 10](#_Toc513660925)

[Introduction 10](#_Toc513660926)

[Background 11](#_Toc513660927)

[Objectives & Methodology 14](#_Toc513660928)

[Importance of the research 22](#_Toc513660929)

[Delimitations 23](#_Toc513660930)

[Budget forecast 24](#_Toc513660931)

[Proposed Timetable 25](#_Toc513660932)

[Summary 26](#_Toc513660933)

[Chapter 2 27](#_Toc513660934)

[Design of Keyless Storage System 27](#_Toc513660935)

[2.1 Introduction 27](#_Toc513660936)

[2.2 Literature Review 28](#_Toc513660937)

[2.3 Simulation Model 41](#_Toc513660938)

[2.5 Comparison and Discussion of Results 46](#_Toc513660939)

[2.6 Project Outcome 51](#_Toc513660940)

[2.7 Recommendation 53](#_Toc513660941)

[Bibliography 56](#_Toc513660942)

[Annexure 57](#_Toc513660943)

[User Interface 57](#_Toc513660944)

[Database 58](#_Toc513660945)

[Serial Communication 59](#_Toc513660946)

[Transmutation Code, (From Rpi3 to uC and the uC Responses). 59](#_Toc513660947)

[Electronic Components Source: 59](#_Toc513660948)

[Notes 60](#_Toc513660949)

## List of Figures

[Figure 1: Keyless entry using a mobile application. 10](#_Toc513660838)

[Figure 2: Keyless Mail and Parcel Lockers for Office Building Courier Delivery and Pick Up 13](#_Toc513660839)

[Figure 3: Component: Diagram of the Keyless Storage System. 14](#_Toc513660840)

[Figure 4: Shows the steps taken by the system to allow the shopper to store a parcel in the keyless storage counter. 15](#_Toc513660841)

[Figure 5: Shows the continued steps taken by the shopper at the storage unit to allow the shopper to store a parcel in the keyless storage counter. 16](#_Toc513660842)

[Figure 6: Retrieval Process 19](file:///J:\Design%20Projects\Unit%203\Chapter%201%20Introduction.docx#_Toc513660843)

[Figure 7: Shows the continued steps taken by the shopper at the storage unit to allow the shopper to retrieve a parcel 20](#_Toc513660844)

[Figure 8: Shows Shoprite’s entrance, parcel counter [Parcels] and customer carrying a plastic bag. “Customers enter a Shoprite Holdings Ltd supermarket in Alexandra district of Johannesburg South Africa on Monday” (Swiegers). 28](#_Toc513660845)

[Figure 9: Raspberry Pi 3 Model B+ 31](#_Toc513660846)

[Figure 10: Shows the user interface for the design 33](#_Toc513660847)

[Figure 11: 8051 Microcontroller - AT89S8253 - DIP 42 34](#_Toc513660848)

[Figure 12: Basic connection diagram of the HD44780 4x20 LCD display and Pin connection to Raspberry Pi 3 35](#_Toc513660849)

[Figure 13: 4x20 HD44780 LCD display and Simulation of LCD display and its program code 36](#_Toc513660850)

[Figure 14: The 4x4 Matrix keypad layout, Showing Pin connections 37](#_Toc513660851)

[Figure 15: Load Cell ALX-50KG 38](#_Toc513660852)

[Figure 16: ADC0834-N 4-Channel MUX SOIC/PDIP 39](#_Toc513660853)

[Figure 17: Shows the simulation of microcontroller AT89C51 and the control devices attached to it. 41](#_Toc513660854)

[Figure 18: Shows the simulation outcome for the design 46](#_Toc513660855)

[Figure 19: The figure show the user interface for storing or retrieval of parcels displayed from the laptop 46](#_Toc513660856)

[Figure 20: Serial communication 48](#_Toc513660857)

[Figure 21: Shows the practical model. 54](#_Toc513660858)

[Figure 22: Shows the practical mode implemented on bread boards. 55](#_Toc513660859)

[Figure 23: About us, user interface screen 57](#_Toc513660860)

[Figure 24: Shows Main UI Screen for the design 57](#_Toc513660861)

[Figure 25: Shows the SQLite3 database; Database table name, and its scheme. 58](#_Toc513660862)

[Figure 26: All the electronic components used in the design 59](#_Toc513660863)

## List of Tables

[Table 1: Budget forecast 24](#_Toc513660864)

[Table 2: Proposed Timetable 25](#_Toc513660865)

[Table 3: Features of Raspberry Pi 3 32](#_Toc513660866)

[Table 4: Load Cell Specifications 38](#_Toc513660867)

## List of Abbreviations

* **SEI**  – Storage Entry Information
* **CPI** –   Consumer Price Index
* **IO** – Input Output
* **GPIO** – General-purpose input/output
* **ASCII** – American Standard Code for Information Interchange
* **UART** – Universal Asynchronous Receive and Transmit
* **UI** – User Interface
* **UX**  – User Experience
* **LCD** – Liquid Crystal Display
* **DIP** – Dual Inline Package
* **Rpi3** – Raspberry Pi 3
* **µC** – Microcontroller (the 8051 microcontroller)
* **ADC** – Analog to Digital Converter
* **MUX** – Multiplexer
* **mW** - Milli-Waltz
* **μs** - Micro-Seconds
* **LSB** - Lower Significant Bit
* **Oz** - Ounce

# Chapter 1

## Introduction



Figure 1: Keyless entry using a mobile application.

*“Locks are the central element of your home or business security. Traditional locks, based on inventions from 1800s, involved a lock cylinder and simple key, but now there are so many more options and technologically advanced designs available that forego the need to hand out too many copies of those keys.*

*Keyless locks allow for so many advantages. Access can be give without risk of copying or unapproved duplicating, because codes or credentials can be change or deleted without physically changing locks and keys. They can be used for temporary access, for instance for a contractor. They allow for immediate security when employees leave or in unfortunate situations where previously trusted friends, family, colleagues or clients betray your confidence. And, do not forget, unlike when you or someone else loses a key, you can change a code or disable a credential, and you are immediately secure again.*

*You often see keyless systems in the commercial sector but you do not have to have a full-featured, high-end access control system for all cases. For a home or small business, there are a several options available, with a variety of features.”* (Keyless Entry, 2018)

## Background

The growth of customers in supermarkets throughout urban areas has drastically increased throughout the years; Stats SA reviles that *"In January the CPI for goods increased by 3.7% year-on-year (down from 4.1% in December), and the CPI for services increased by 5.1% year-on-year (down from 5.3% in December)."* (Consumer Price Index - Key Findings, 2018), consequently with falling prices this creates the need for satisfactory and secure parcel storage space in supermarkets; known as parcel counters - that will be helpful to the customers and the supermarket.

A persistent problem exists throughout the supermarkets around communities related to the insufficient security around the parcel storage system. Most supermarkets use a number of open view locker as their parcel storage space, the procedure for storing parcels is as follow, customers that place their parcels in the supermarkets’ Parcel Counter, are given tags that are used to identify their storage space, the tags are numbered; used as a proof that the customer has actually kept their belongings in at the parcel counter kiosk. Hence these storage spaces do not have a locking system but instead there’s an employee placed to handle and secure the parcels.

The parcel counter kiosk space is prone to:

* *“Theft which may come from an employee and other shoppers.”* (penmof, 2017)

International Perspective: What Deters Shoplifters? Gill, Billy, and Turbin (1999:29-39) Interviewed 38 experienced shoplifters in England to assess the effectiveness of anti-shoplifting strategies. Their research is summarized here: Retail staff: while nine out of 38 respondents suggested that retail staff would sometimes deter them, 24 answered *“never”*, “They argued that salespeople were too busy or uninterested in shoplifting. (Purpura, 2002)

* Tags that are used to identify the parcel(s) get lost, and they are not printed well to identify the customer’s parcel.
* Mishandling of parcels may lead to damage of items that the retail must replenish.

*“For each retailer, if the prosecution of shoplifters becomes expensive and time consuming, repeat offenders are not deterred, cases are not successful, and brand is harmed, then anti-shoplifting strategies must be changed.”* from (Security And Loss Prevention 4th Edition, 2002) .Presently some supermarkets are trying to solve the problem of parcel storage, by using the locking storage system that uses a physical key, which is handed to individuals who place their items in the parcel storage, but this implementation does not proved sufficient security to the parcel counter, because this keys given to the individuals may be misplaced, lost , broken, or copied, thereby giving other individuals access and increasing the level of insecurity of the storage unit.

The keyless storage system utilizing raspberry pi will be a storage system that will be designed to store customers’ parcel(s) in a more secure way in supermarket as it will be automatically locking without a physical key and would not open unless the parcel’s owner enters a password in the user-friendly interface(s). This system will create a more secured parcel storage space in supermarkets, thereby decreasing the level of theft and damages of the shoppers’ parcels.

Research Aim



Figure 2: Keyless Mail and Parcel Lockers for Office Building Courier Delivery and Pick Up

The aim of this project is to research and design a keyless storage system utilizing raspberry pi to solve the security problem of parcel counter; in a supermarket, in order to eliminate the incident of theft and parcel mishandling.

This project will make use of Raspberry pi, 8051 microcontroller, storage unit boxes, display screen, input keypad, weight sensor, magnetic lock, display LEDs. The raspberry pi will handle the SEI database and the generation of storage unit password(s); it is also responsible for interfacing the function of locking and unlocking the storage unit via the 8051 microcontroller.

The raspberry pi uses the SEI captured information from the interface, and sends the instruction to the 8051 microcontroller’s IO ports, (Port 2) which is connected to the magnetic lock(s) on storage units is triggered; to lock or unlock the storage unit. Also the signal is sent from Raspberry Pi to the LEDs to indicate that the door is unlocked. The weight sensor will send the signal back to the Raspberry Pi to indicate if the storage unit is available or on duty.

The aim is to reduce the level of missing, and the mishandling of parcels, at the parcel counter kiosk of supermarkets around South Africa, and to being about confidence to shoppers, “that when they are shopping inside the market, their parcels are in a secure space”.

## Objectives & Methodology

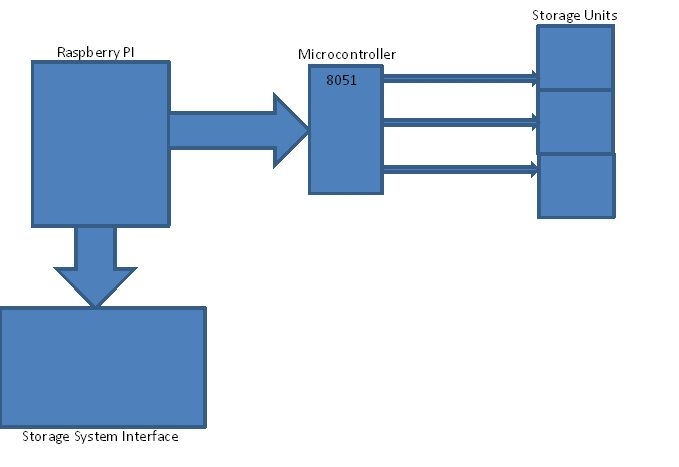


Figure 3: Component: Diagram of the Keyless Storage System.

Figure 3 shows the component diagram of the Keyless Storage System, raspberry pi control the whole process of storing and retrieving of parcels, while the microcontroller delegates the functions of locking and unlocking of the storage units (via the serial communication line between the raspberry and the microcontroller), and its response LED indicates(via the microcontroller to storage unit). The storage system interface handles the Input Output Interactions between the shopper and the Keyless Storage System (via the raspberry pi to storage system interface).

**Storage Process:**

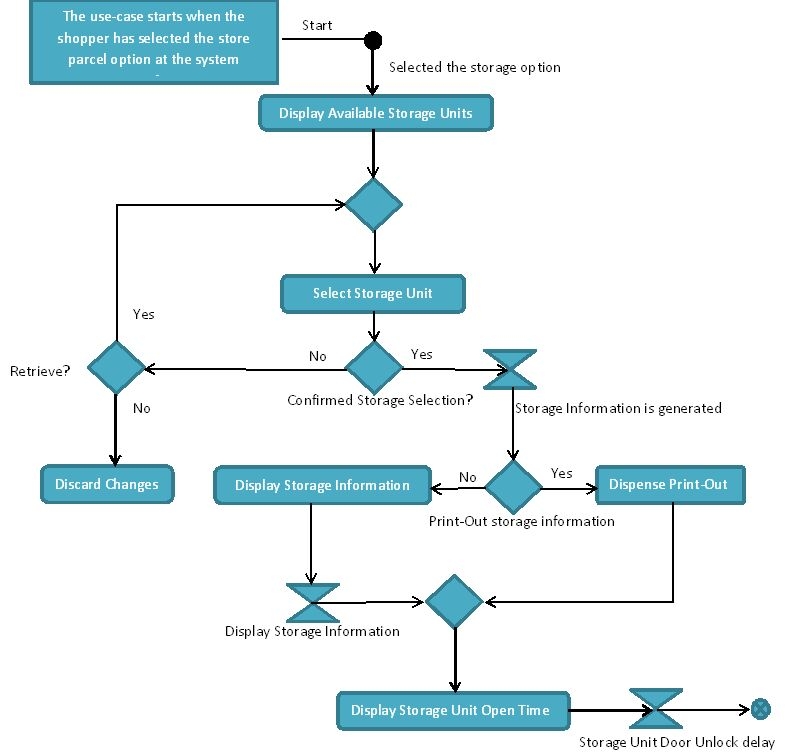


Figure 4: Shows the steps taken by the system to allow the shopper to store a parcel in the keyless storage counter.

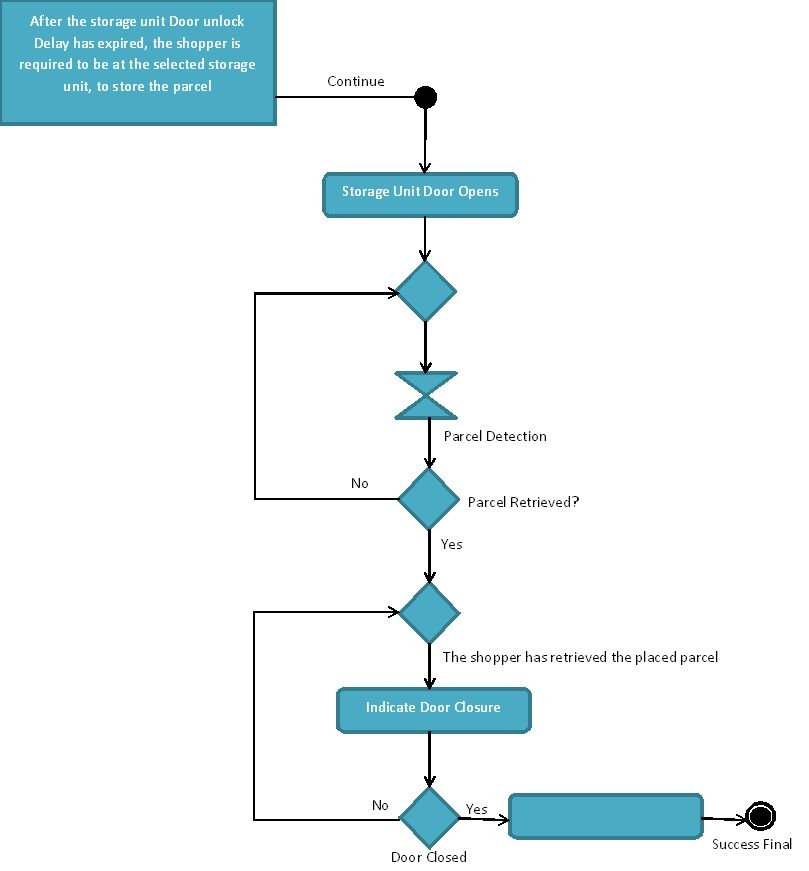


Figure 5: Shows the continued steps taken by the shopper at the storage unit to allow the shopper to store a parcel in the keyless storage counter.

**Overview of the Store Parcel Activity Diagram:**

1. At the system interface - The shopper must select storage option:

1.1. The system will indicate the available storage boxes (storage unit); where to place the parcel: shoppers are given a single slot per storage entry.

1.2. The shopper must select a storage unit:

1.2.1. The Shopper must indicate where the parcel should be stored, and confirm this selection.

1.3. After the shopper has given confirmation, the shopper is given a choice to print-out or display the storage entry information (SEI) - (which consist of the storage unit the shopper choose, a system generated password and the time when the selected storage unit will be open).

1.3.1. The system will generate a password for the selected storage unit, and will either display or print out the SEI.

1.3.1.1. If the shopper choose to display the SEI; this information will be display for a set duration and then disappear; (the shopper will be advised to remain at the terminal until this event has happened).

1.4. The shopper is directed to go, to the storage unit section and wait for the selected storage unit to open.

2. At the storage unit section - system will indicate to the shopper that the storage unit is open. A persistent indicator will start to indicate, to the shopper to close the door.

2.1. The shopper must securely place the parcel, and close to storage unit door.

2.1.1. After the door is closed the storage unit will lock, and the persistent indicator will stop indicating.

3. The shopper has successfully secured the parcel, and can go shopping.

**In the event where the shopper has done all steps but:**

Placing a parcel in the storage unit; the weight sensor cannot detect a parcel, after the door is closed; the system will discard the SEI, in this case the shopper may have or not place a parcel but has a SEI.

* This event is mentioned at, the delimitation part of the document.
* System will dim that this storage unit is vacant.

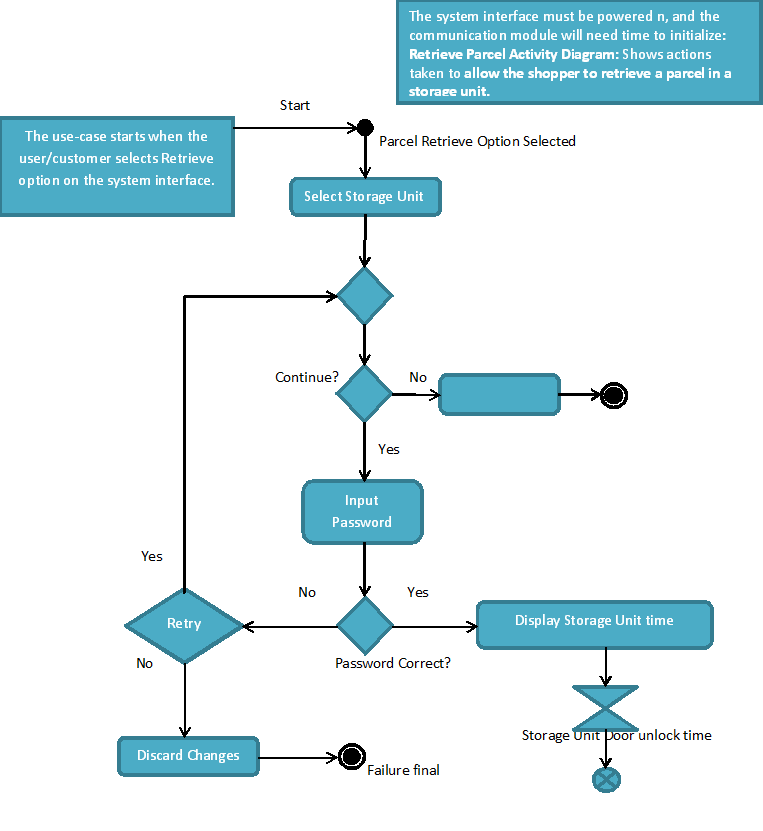
****

Figure : Retrieval Process

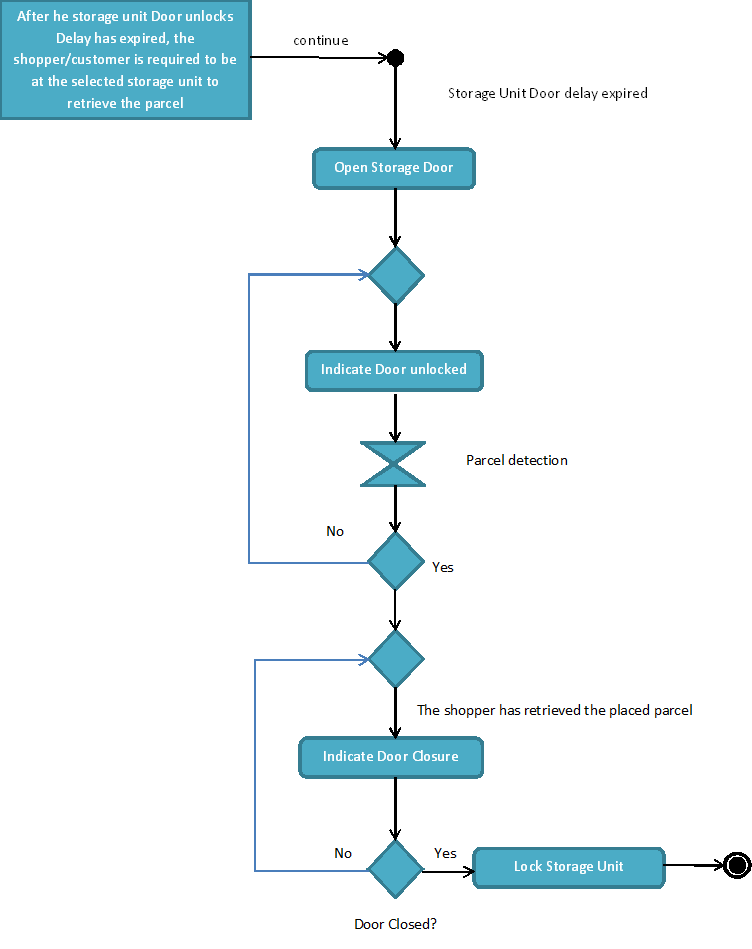


Figure : Shows the continued steps taken by the shopper at the storage unit to allow the shopper to retrieve a parcel

In the keyless storage counter.

**Overview of the Retrieve Parcel Activity Diagram:**

1. At the system interface - The shopper must select the parcel retrieval option:

1.1. The system will ask the shopper, whether he or she wants to continue to retrieve their parcel or not.

1.2. The system will ask the shopper to type the password.

1.3. The system will check the password, if it is correct.

1.3.1. If the password is correct, then the system will display the time the storage unit will be unlocked.

1.3.2. If the password is incorrect the system will let the shopper a chance to retry again.

1.3.2.1. If the shopper wishes to opt-out, the system will discard the changes done by the shopper, and the process ends.

1.4. After the storage door delay has expired, the shopper is required to be at the selected storage unit, to retrieve the parcel.

1.4.1. The door unlocks, the shopper will be prompted by the LED to open the storage unit door.

1.4.2. The weight sensor will detect the absences of the placed parcel.

1.4.2.1. If the shopper has removed the placed parcel, the weight sensor will indicate to the system to display a door closure LED flash.

1.4.2.1.1. Then the SEI will be discarded from the database, and the storage unit will be available for the next user.

1.4.2.1.2. The system will lock the storage unit door, the process end.

1.4.2.2. If the shopper did not remove the parcel, the weight sensor will indicate to the system that the parcel is still present.

1.4.2.2.1. The system will flash both LEDS to indicate that the shopper must remove the placed parcel; this will go on indefinitely until the parcel is removed.

## Importance of the research

This research will benefit companies like Shoprite where a high level of lost parcel has been reported by individuals, hence the keyless secured storage system will help by decreased the level of parcel theft and clean the company imagine to the society.

Individuals who often place their belongings in the supermarkets’ parcel counters will also benefit from this in such a way that they would not have to worry about the safety of their storage units anymore since they will be the only ones who will be able to access it.

Supermarkets like Pick n Pay, Spar, Checkers and others by moving them from the usage of modern parcel counters to more secured Keyless secured storage system, thereby increasing the security in the area.

Fitness center may as well profit from this project since this storage system can also be used as secured fitness lockers in this area.

## Delimitations

* Research will not include the lifespan and structure of the microcontroller and integrated circuits, information related to such may be observed on an individual manufacturer's datasheet.
* The research will not include the addition for more the three storage units due to: this observation that the microcontroller would restrict the addition of more storage units, due to microcontroller IO pin count. Information related to such may be observed on an individual manufacturer's datasheet.
* The research will not include the storage unit: access time duration and space allocations per shopper, due the supermarket-parcel counter operational constraints.
* The research will not include the calibration of weight sensor due to an issue: That the weight of a parcel placed in a storage unit, may throw off the weight sensor detection; this include small items weighing under the sensors detection range; the sensor may perceive that there is no object is placed on the storage unit, this is due to manufacturer's calibration range.
* The research will not include the development of a 12V power supply and the power outage plan.

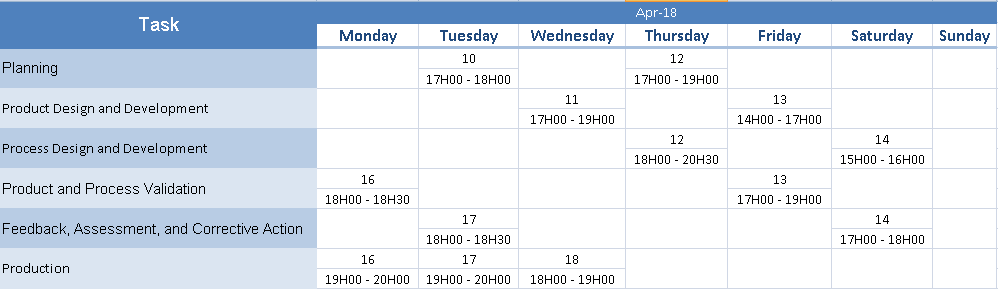
## Budget forecast

Table 1: Budget forecast

|  |  |
| --- | --- |
| **Activity** | **Cost** |
| Registration for EADES3B | R2 710 |
| Language and printing costs | R 350 |
| Literature study (obtain journal articles) | R 550 |
| Labor and Manufacturing | R3 500 |
| Instruments for Testing Purposes | R 350 |
| Electronic Components Needed | R2 000 |
| Total | R9 460 |

## Proposed Timetable

Table 2: Proposed Timetable

****

## Summary

As we have seen the issue of supermarkets parcel storage insecurity has not been addressed properly by the various companies, therefore to solve this problem a keyless system utilizing raspberry pi, will automate the process of locking without a physical and would not open unless the parcel owner enters a password the user-friendly touch screen interface, this will be of great benefit to supermarkets as well fitness centers, and will create a much secure storage space in these areas.

# Chapter 2

### Design of Keyless Storage System

### 2.1 Introduction

In this chapter, information will be gathered to successfully implement a working simulation of a keyless storage system. This will benefit the reader to briefly understand in depth about inner workings of the keyless storage system as it will involve sketches and design concerning the practical model.

#### 2.1.1 Overview

Importance will be placed on the whole project, which will include the schematic of the project, operating principles, and the programming part that needs to be applied for the operation of the keyless storage system. Chapter 2 presents the literature review section which will have descriptions on existing devices that can be suitable to fulfill the current project, simulation and practical models. The scope of review contains Literature review, the simulation model, practical model, comparison and discussion of results, projects outcome and recommendations.

#### 2.1.2 Summary

The Literature review section consists of substantive finding, theoretical and methodological contributions to this project, the simulation model presents all possible details of this project with the aid of sketches and pictures of practical software simulation, and it is presented in sequential logic. In terms of the practical model, physical and environmental aspect has been taken to consideration in order to produce accurate outcome for this project. Simulation and practical model will be discussed, and compared. The obtained result will also be presented accordingly. Lastly the project outcome and recommendations will be given to people who contributed their knowledge for this project.

### 2.2 Literature Review



Figure : Shows Shoprite’s entrance, parcel counter [Parcels] and customer carrying a plastic bag. “Customers enter a Shoprite Holdings Ltd supermarket in Alexandra district of Johannesburg South Africa on Monday” (Swiegers).

The invention or innovation of shopping bags being used at the retail market has changed over the past decades.

Starting from baskets, paper bags, plastic bags, and now to recyclable plastic bags, the trend seems not to deviate from the norms, the Times Live said that, “South Africans will soon be able to dispose of their plastic bags along with the rest of their waste without having to worry about any possible negative effects on the environment.” (Collins Farren, 2017). People need means of transporting or carrying, their goods from one shop to the next, in search for great deals or offers.

May come about when shoppers conceal the retail’s inventory in their bags/parcel. A parcel counter at the retail market(s) was intended to reduce theft, shoplifting or incidence such as this is ones.

The other concept that may have brought about this, was when shopper came to the retail with their bags that had the same inventory as the retail’s; this caused confusion when a shopper did not produce evidence that they are items were theirs and not the retail’s, in such a case as this one the shopper has committed shoplifting or just replaced their items with retail inventory; (this case is the same thing as shoplifting), the are many situations that may not have been analyzed. Till now, the focus was about the retail being able to secure its inventory; it was never discussed and viewed from the shopper’s viewpoint, one shopper might suggest that it came about because of shopper convenience and the other might say something else; whatever the result, that brought the conception of the current parcel counter system, as times and situations change, an innovation for such a system will be needed.

*“Every day millions of people decide to grant their Smartphone a bit more control over their lives or try a new and more effective antidepressant drug. In pursuit of health, happiness, and power, humans will gradually change first one of their features and then another, and another, until they will no longer be human.”* (Yuval Noah Harari, 2015) As shoppers demand more security over their belongings in retail markets, they constantly find it inconvenience for a stranger to be a person that looks after their belongings. A demand for security; at and around the parcel counter kiosk would be an essential necessity for shoppers to have. From Harari’s quote, *“if people grant more and more functionality to their Smartphone device(s), why not for this device?* “An innovation to the current parcel counter system would ensure that shoppers have a secure and convenient storage space. This project intent is to bring about such for shoppers and the retail market.

This keyless storage system would bring about a drastic change in the way that current parcel counter kiosk, conduct their day to day operation(s), beginning with the security vulnerabilities of the currently employed system, as forth mentioned in chapter one. It would benefit the system if the control devices portray such behaviour and functionality as of the currently employed system.

The use of control devices to bring about the desired behaviour for the keyless parcel counter system is mentioned next.

#### 2.2.1 Control Devices

The following comprises of the control devices that the keyless storage system uses,

* Raspberry Pi 3
* 8051 Microcontroller (AT89S8253)
* HD44780 4x20 LCD display
* Keypad 4x4 Matrix
* Load Cell (ALX-50KG)
* ADC Multiplexer

Some of the control devices can be referenced on figure 3 of chapter 1; [figure 3 shows Component: Diagram of the Keyless Storage System](#_Objectives_&_Methodology) and how they connected.

##### The Raspberry Pi 3

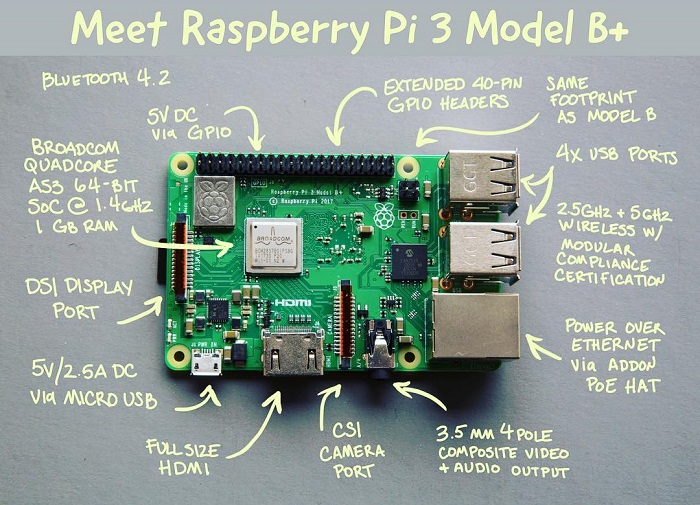


Figure : Raspberry Pi 3 Model B+

The figure above shows Raspberry Pi 3 Model B+, and short a comparison to its’ predecessor the Raspberry Pi model B. The table below shows some of the Raspberry Pi 3 model B+ features.

Table : Features of Raspberry Pi 3

|  |  |
| --- | --- |
| CPU | Quad-core 64-bit ARM Cortex A53 clocked at 1.4 GHz |
| GPU | 400MHz Video Core IV multimedia |
| Memory | 1GB LPDDR2-900 SDRAM (i.e. 900MHz) |
| USB ports | 4 |
| Video outputs: | HDMI, composite video (PAL and NTSC) |
| Audio | 3.5 mm jack |
| Network | 10/100Mbps Ethernet and 802.11n Wireless LAN |
| Peripherals | 17 GPIO plus specific functions, and HAT ID bus |
| Bluetooth | 4.1 |
| Power source | 5V via Micro USB or GPIO header |
| Dimension | 85.60 × 56.5 mm |
| Weight | 45g (1.6 oz) |

###### Tasks (done by the Rpi3):

* 1. Mitigate the UI task(s); to be performed (e.g. Store Parcel Button pressed; this action is delegated).
  2. Monitors the storage units; for availability, the Rpi3 will send a control ASCII character (‘M’) to the 8051, the monitor each storage unit ([for more info, refer to the Tasks (done by the µC), point a](#_Tasks_(done_by_1)).
  3. It grants shoppers access; to store and retrieve their parcel (or small belongs); [for more info, refer to the Tasks (done by the µC), point b](#_Tasks_(done_by_1).
  4. Temporally archives the shopper’s SEI information; of the
  5. It delegates the user interface communication between the above mentioned control behavior, through the use of the system SEI database. Shopper and the system; this is handled by the, 4x4 keypad, and the 4x20 LCD display, shown below on figure 11 for reference.

The Raspberry Pi 3 will be used to perform the main sequential logic of the project. Control behavior such as the ones mentioned at points A and B, can be achieved through the use of UART communication between the raspberry pi and the 8051 microcontroller; an instruction or acknowledgement comprises of, an ASCII byte code being sent back and forth these devices..

###### The User Interface:

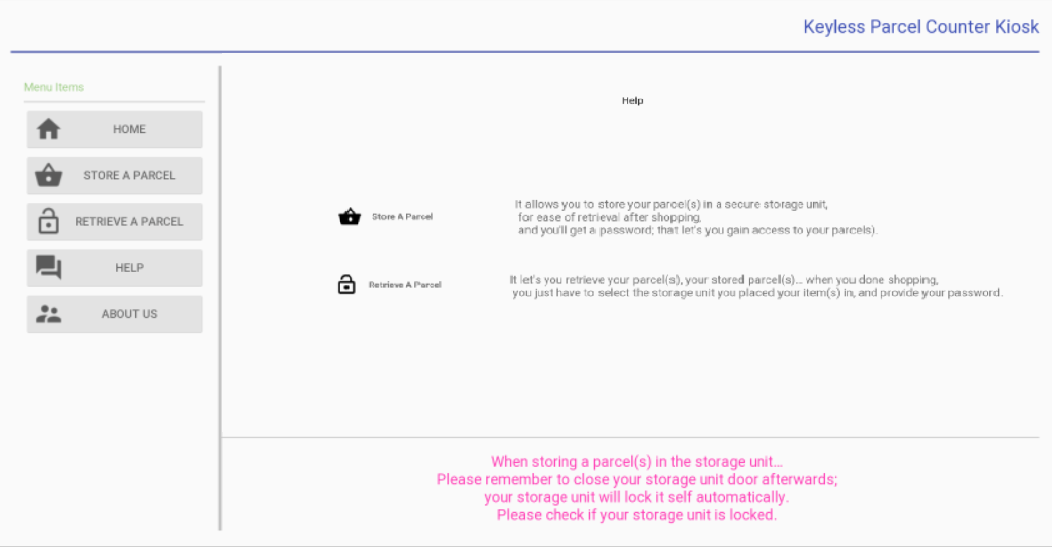


Figure : Shows the user interface for the design

The design of the user interfaces; used the android things platform, to achieve a user friendly interface. [Android things](https://developer.android.com/things/), is a platform that utilizes the Android mobile app (framework), a product that is implemented on the android things design; has many advantage over other devices. Android things offers both support for the android framework and Linux kernel updates, their also have a [System on Module](https://developer.android.com/things/console/create) support, these gives developer freedom to enter the market with their [Android things devices](https://developer.android.com/things/hardware/), but the entry level to this platform is quiet high. Most of what is discussed here can be found on the Android developer website and an easy explanation to it can be view on [YouTube](https://www.youtube.com/watch?v=zpR_jQXc1fs).

With the growth of Android devices on the market, the shoppers may be familiar with; android things framework, because it uses the same user interface as its mobile apps (Applications) which anyone with an android mobile device is used too, so this design aims not to complicate things, for the shopper.

The Raspberry Pi will display the UI via the HDMI, or the 3.5 mm 4 pole composite video + audio output terminal.

##### The 8051 Microcontroller (AT89S8253):

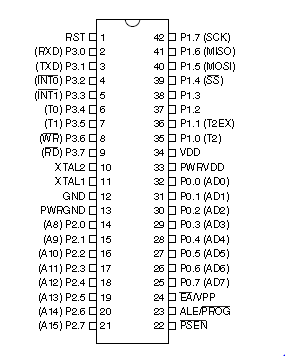
**

Figure : 8051 Microcontroller - AT89S8253 - DIP 42

The AT89S8253 is a low-power, high-performance CMOS 8-bit microcontroller with 12K bytes of In-System Programmable (ISP) Flash program memory and 2K bytes of EEPROM data memory. The device is manufactured using Atmel’s high-density non-volatile memory technology and is compatible with the industry-standard MCS-51 instruction set and pin out.

###### Features

* 12K Bytes of In-System Programmable (ISP) Flash Program Memory
  + SPI Serial Interface for Program Downloading
  + Endurance: 10,000 Write/Erase Cycles
* 2K Bytes EEPROM Data Memory
* Endurance: 100,000 Write/Erase Cycles
* 2.7V to 5.5V Operating Range
* 256 x 8-bit Internal RAM
* 32 Programmable I/O Lines
* Enhanced UART Serial Port with Framing Error Detection and Automatic Address Recognition

###### Tasks (done by the µC):

1. **Monitor the storage units,** check each storage units’ weight sensor; the weight sensor will give a positive numerical data reading when these a parcel placed in a storage unit, otherwise a 0 to negative numerical reading will be obtained from the weight sensor.

The µC will send the monitor data to the Rpi3, for processing.

1. **Access Control to the storage unit(s)**; the microcontroller will grant, and secure access to each storage unit; based on the task the system needs to perform.

##### The Storage unit LCD Display (HD44780 LCD display)

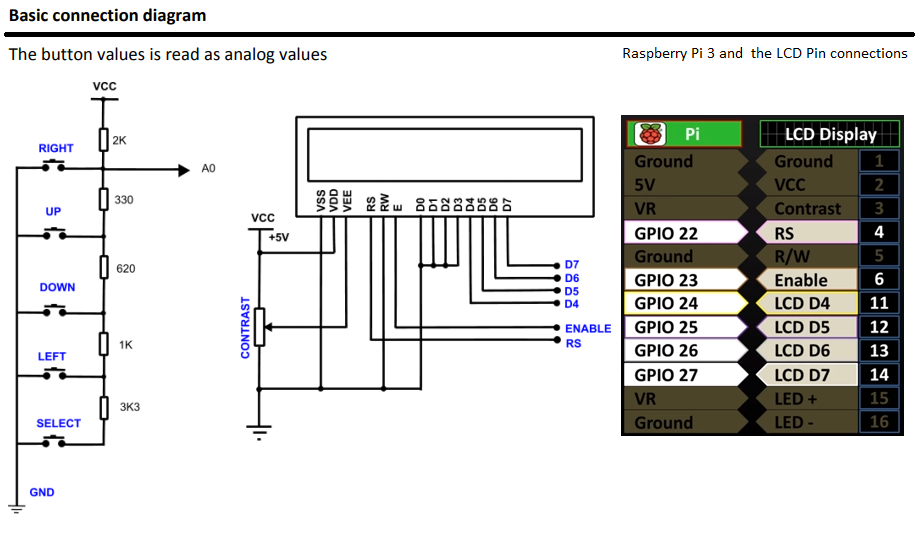


Figure : Basic connection diagram of the HD44780 4x20 LCD display and Pin connection to Raspberry Pi 3

Why choose to use 4x20 LCD display, for this project? “LCD character displays are a simple and a cost-effective way to display a text. Thanks to the HD44780 controller, the control of the modules has become very simple. However, one must occupy many GPIOs for it. An alternative is the I2C data bus, which means that only two GPIOs are used.” (Control a HD44780 LCD display via I2C with the Raspberry Pi, 2017). Figure 10 show the basic connection diagram of the HD44780 LCD 4x20 display, more information can be obtained for this [HD44780LCD4x20display](http://www.mantech.co.za/Datasheets/Products/SHD036.pdf).

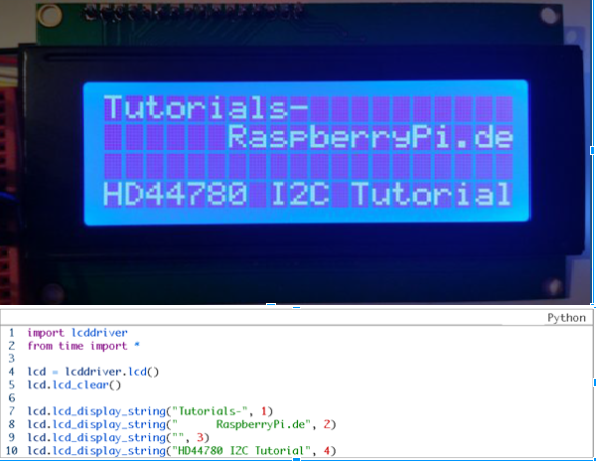
1

Figure : 4x20 HD44780 LCD display and Simulation of LCD display and its program code

The above figure shows a simulation results to display text to the LCD display, followed by the Raspberry Pi python code; this simulation is available at (Control a HD44780 LCD display via I2C with the Raspberry Pi, 2017) as tutorial.

###### Tasks (done by the Storage Unit LCD display):

* This control device will be one part of the user interface (UI) of the keyless storage system. The 4x20 means 4 rows and 20 columns of ASCII alphanumeric character can be displayed on an LCD at one time. The other types of this product are available like the 2x16 LM016L LCD display.

##### The Keypad

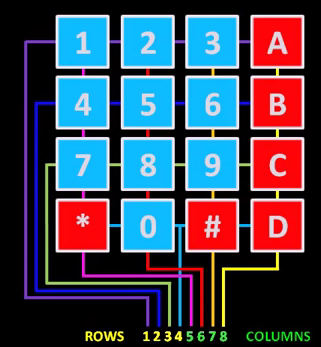


Figure : The 4x4 Matrix keypad layout, Showing Pin connections

The 4x4 Matrix keypad is amongst the widely used control device for entering data, it is comprised of 16 keys/buttons connected via rows and columns: has no other internal circuitry, only the wires connecting the keys; seen on the figure above.

###### Tasks (done by the keypad):

* The shopper will use the keypad, when entering the password.

###### Connection Approach

1. Attach matrix 8-pin interface to the 8 GPIO pins.
2. Setup column pins as outputs and set to high.
3. Setup row pins as inputs with pull-up resistors (inputs are set high).
4. Set outputs low, ones at a time.
5. When a key is pressed, input becomes low indicating which key has been pressed (Since we now know which column is set low; one column is low at a time).
6. Implement a short 2µs time delay in between key presses; this avoids the double key press problem (within the program code).

##### The Load Cell ALX-50KG



Figure : Load Cell ALX-50KG

The figure above shows the load cell, it is one of the control device(s) that is used for sensing of objects weight.

Table : Load Cell Specifications

|  |  |
| --- | --- |
| Capacity | 50kg |
| Sensitivity | 1.0mV/V |
| Zero OUT | 0.3mV/V |
| Resistance (I & O) | 1kOhm |
| Excitation | 10VDC |
| Overload limit | 150kg |

###### Applications

* Personal scale
* Other scales

They may be used in two’s or fours to make scale of100 and 200 kg.

###### Tasks (done by the load cell):

* Sense the presence of parcel(s) in each storage unit.

###### Connection Approach

1. Connect the power connections.
2. Connect the data wire to the ADC0834-N Multiplexer

##### Serial I/O A-D Converters ADC0834-N 8-Bit - 4 Channel Multiplexer

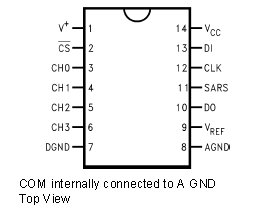


Figure : ADC0834-N 4-Channel MUX SOIC/PDIP

The figure above shows the ADC0834-N which is of the ADC0831 series, and is 8-bit successive approximation A/D converters with a serial I/O and configurable input multiplexer with up to 8 channels. The serial I/O is configured to comply with the TI MCROWIRE serial data exchange standard for easy interface to COPS family of processors, and can interface with standard shift registers or µPs.

###### Features

* Easy Interface to all microprocessors, or operates “Stand-Alone”.
* No zero or full-scale adjust required.
* 2-, 4- or 8-channel multiplexer options with address logic.
* Shut Regulator allows operation with high voltage supplies.
* 0V to 5V input range with single 5V power supply.
* Remote operation with serial digital data link.
* 0.3 in. standard width, 8-, 14- or 20 –Pin PDIP Package.

###### Key Specifications

* Resolution: 8 Bits
* Total Unadjusted Error: ±½ LSB and ±1 LSB
* Single Supply: 5 VDC.
* Low Power: 15 mW.
* Conversion Time: 32 μs.

###### Tasks (done by the ADC Multiplexer):

* Converts the load cell weight reading to digital format, then the data reading is processed.

### 2.3 Simulation Model

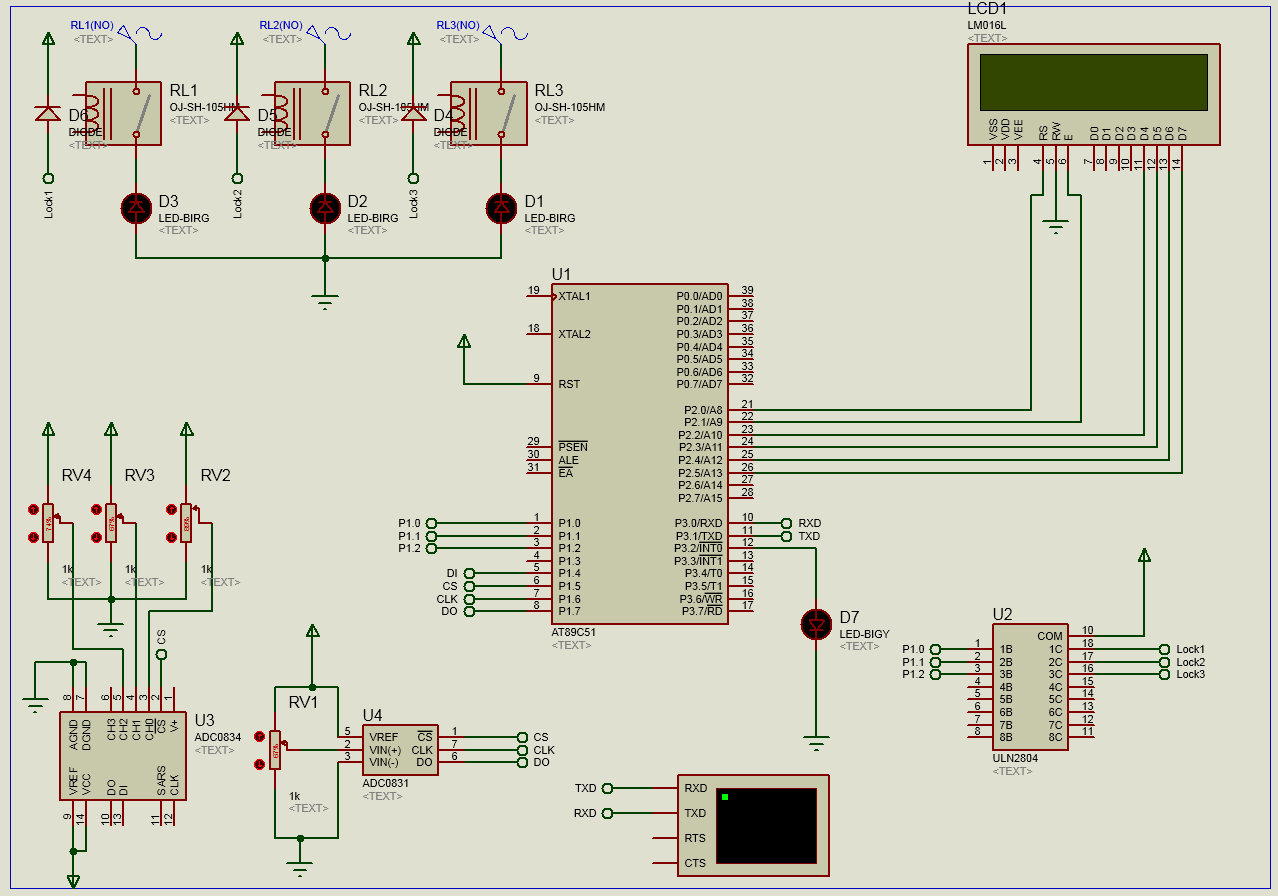
****

Figure : Shows the simulation of microcontroller AT89C51 and the control devices attached to it.

The Application used is Proteus 8(Professional). Proteus is one of the best simulation software due to its flexibility and enabling us to work with all the devices that we wish to put in our design. It include devices like led, keypad, relays, and also serial communication for our communication with the raspberry pi3.

There are also important setting that should be taken into consideration like setting the baud rate for serial communication, selecting the microcontroller to be used, selecting the file of storage for the code to be imported.

Microcontroller communicates with the raspberry pi3, storage units, LCD, and the weight sensors of each storage, on serial communication when the raspberry pi 3 sends the letter a/A to the microcontroller it request the parcel status on the storage unit one using the sensor inside the unit, with d/D the lock status is requested, with b/B the parcel status is requested for storage unit 2 and with e/E its lock status is requested. With c/C the parcel status is requested from storage unit three and with f/F its lock status is requested.

If the status is requested the microcontroller reads the status of port 1.7(DO) to check if the status is in logic high or Low, if is high it sends the logic high to the raspberry pi indicating that the sensor is reading a value/weight of the parcel inside the storage unit, and it also displays massage on the LCD written “Storage Unit Occupied”. Else if the sensor of the storage unit does not read any value, the P1.7(DO) reads the logic Low and sends it to the raspberry pi indicating storage is available and also displaying to the LCD ”Storage Unit Available”.

After the raspberry pi has finished communicating with the use on the interface and the password is generated, and selection of the storage unit is made by the user/customer. The microcontroller connects raspberry pi and relays. Function of the relay is to supply the required amount of voltage to the magnetic locker so it opens or closes, each relay of the storage unit is connected to its magnetic locker. When the raspberry pi send a logic High to the microcontroller it request the selected storage unit to the opened, The microcontroller also sends the logic High to the relay of the selected storage unit, the relay generates the required voltage by the magnetic locker and it triggers the locker to open the storage unit.

When the user wants to retrieve the parcels, firstly communicates with the interface which is connected to the raspberry pi. The raspberry pi checks if the password entered is correct, if the password is correct is sends a logic Low to the microcontroller, the microcontroller also sends the logic low to the relay of the storage unit a user wants to retrieve parcels from. The relay disconnect and the low is sent to the magnetic locker of the storage unit. The magnetic locker loses its magnetic strength and the storage unit opens for the user. But if the password entered is incorrect the logic High is kept on the relay of the locker to keep the locker locked.

2.3.2 Walkthrough the Microcontroller C/C++ code

Serial communication code from the microcontroller to the raspberry pi

void RaspberryPiComs() iv IVT\_ADDR\_ES ilevel 0 ics ICS\_AUTO {

char cmd;

if(ri\_bit==1)//Raspberry Pi is transmitting

{

    P1\_6\_bit =1;

    P1\_7\_bit =0;

    current\_cmd = sbuf;

    switch(sbuf)

    {

     ///locks

     case 'a':

     case 'b':

     case 'c':

     ///weight

     case 'd':

     case 'e':

     case 'f':

     previous\_cmd = current\_cmd;

     break;

    }

    //locking and unlocking; lock0  Raspberry Pi sends 'a',followed by 'l' unlocks and 'L' to lock

    //                    lock1 Raspberry Pi sends 'b',followed by 'l' unlocks and 'L' to lock

    //                    lock1 Raspberry Pi sends 'c',followed by 'l' unlocks and 'L' to lock

    switch(previous\_cmd)

    {

      case 'a':

      if(current\_cmd== 'l')setLock(0,0);

      else if(current\_cmd=='L')setLock(0,1);

      break;

      case 'b':

      if(current\_cmd== 'l')setLock(1,0);

      else if(current\_cmd=='L')setLock(1,1);

      break;

      case 'c':

      if(current\_cmd== 'l')setLock(2,0);

      else if(current\_cmd=='L')setLock(2,1);

      break;

    }

    //monitor wieght sensor; parcel0  Raspberry Pi sends 'd', micro controller transmits back a '0' meaning these no parcel and '1' these a parcel

    //                    parcel1 Raspberry Pi sends 'e', micro controller transmits back a '0' meaning these no parcel and '1' these a parcel

    //                    parcel2 Raspberry Pi sends 'f', micro controller transmits back a '0' meaning these no parcel and '1' these a parcel

    switch(previous\_cmd)

    {

      case 'd':

      ri\_bit =0;

      sbuf = (storageHasWeight(0)== 0) ? '0': '1';

      break;

      case 'e':

      ri\_bit =0;

      sbuf = (storageHasWeight(1)== 0) ? '0': '1';

      break;

      case 'f':

      ri\_bit =0;

      sbuf = (storageHasWeight(2)== 0) ? '0': '1';

      break;

    }

    //monitor door; door0  Raspberry Pi sends 'g', micro controller transmits back a '0' meaning the door is open and '1' door is closed

    //           door1 Raspberry Pi sends 'h', micro controller transmits back a '0' meaning the door is open and '1' door is closed

    //           door2 Raspberry Pi sends 'i', micro controller transmits back a '0' meaning the door is open and '1' door is closed

    switch(current\_cmd)

    {

        case 'g':

        ri\_bit=0;

        sbuf = (getDoorStatusAt(0) == 0) ? '1': '0';

        break;

        case 'h':

        ri\_bit=0;

        sbuf = (getDoorStatusAt(1) == 0) ? '1': '0';

        break;

        case 'i':

        ri\_bit=0;

        sbuf = (getDoorStatusAt(2) == 0) ? '1': '0';

        break;

    }

    ri\_bit =0;        //acknowledge Rpi3 transition

}

if(ti==1)   //uC transmitting

{

 P1\_6\_bit =0;

 P1\_7\_bit =1;

 ti\_bit =0; //acknowledge Rpi3 reception

}

}

### 2.5 Comparison and Discussion of Results

#### 2.5.1 Practical and Simulation outcome

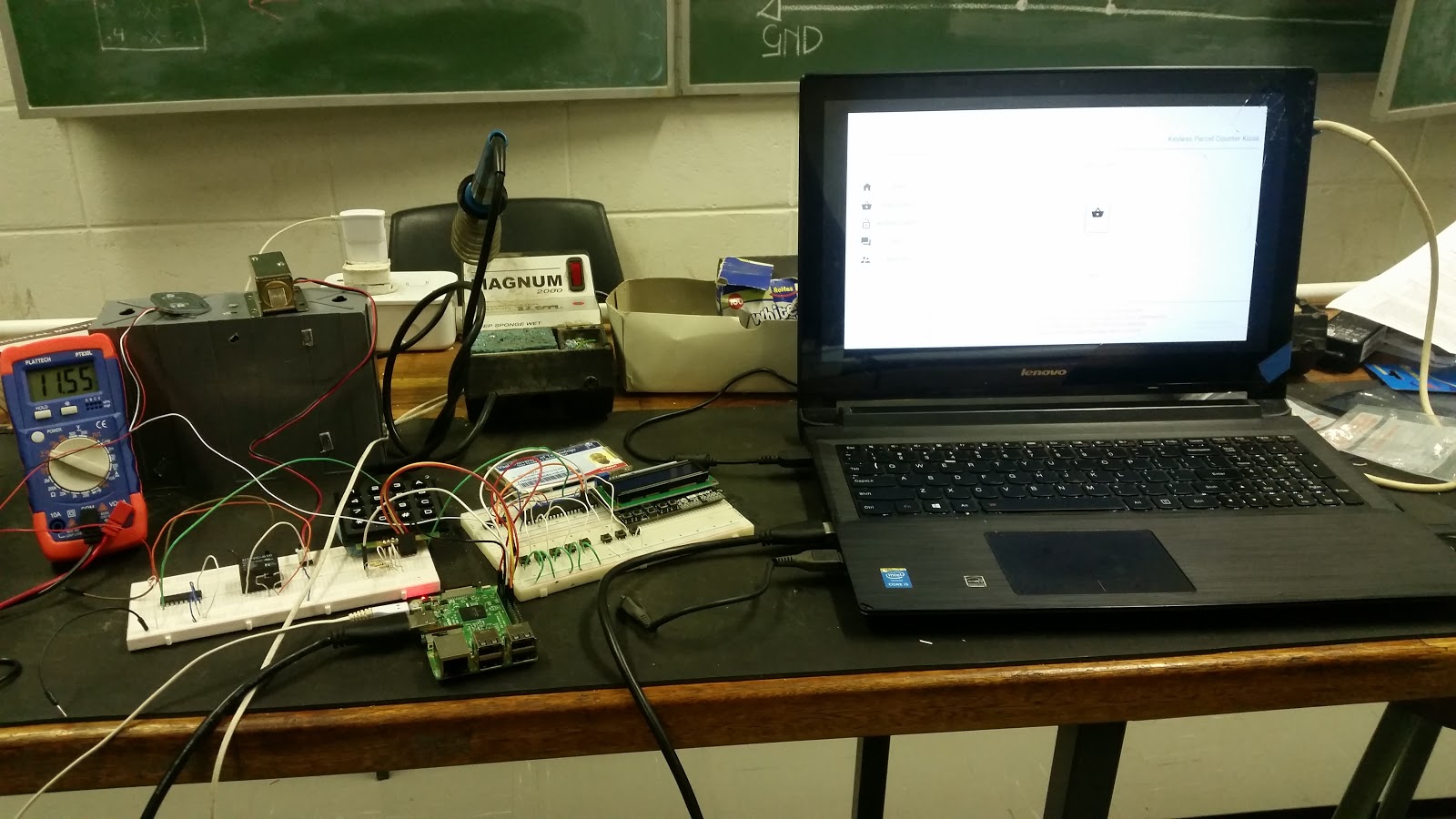


Figure : Shows the simulation outcome for the design

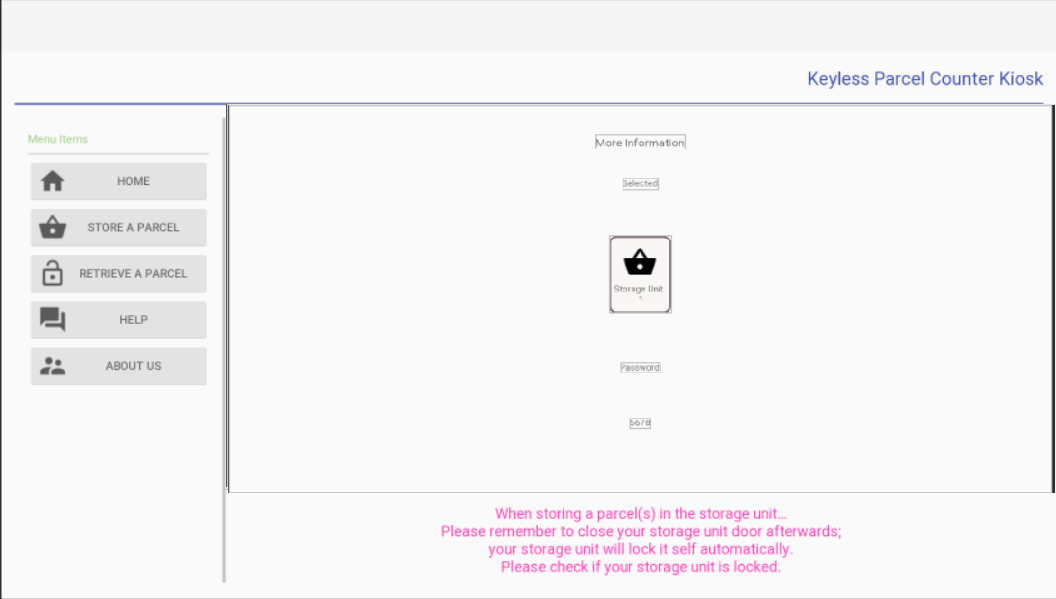
****

Figure : The figure show the user interface for storing or retrieval of parcels displayed from the laptop

Both figure 18 and 19, show the desired design outcome of the schematic and the user interface that will be used to trigger the control behaviour events.

Firstly the screen display “Keyless parcel counter kiosk”, which is the identifier of this system. Also the interface displays options: Home, Store a Parcel, Retrieve A Parcel, Help, About Us, this are the options that the user will choose from when using system interface. And the software part which is discussed earlier will handle the behaviour of each and every option that the user selects. The events on the screen are operating on the raspberry pi that is connected using HDMI to the laptop screen.

The connection of keypad, Microcontroller, relay, ADC, buttons and Dalen Pair Interface are all mounted on the breadboards. The magnetic lock is placed is a sense that we can observe it mechanism when the microcontroller triggers it. We placed and removed a certain weight on top on top of the weight sensor to observe the reading on LCD, and is sending connect values to the raspberry pi.

The power supply was placed to supply the voltage of 11.55V (12V) to the locks and the raspberry pi supplied the microcontroller with 5V. Also a raspberry pi is powered using a phone charger so it won't cause any damages.

Finally if the user selects Store a Parcel, password is generated and the magnet lock is trigger to open a storage if empty. And if Retrieve a Parcel is selected and the entered password is correct using keypad, then the lock unlocks.

#### 2.5.2 Analysis of results

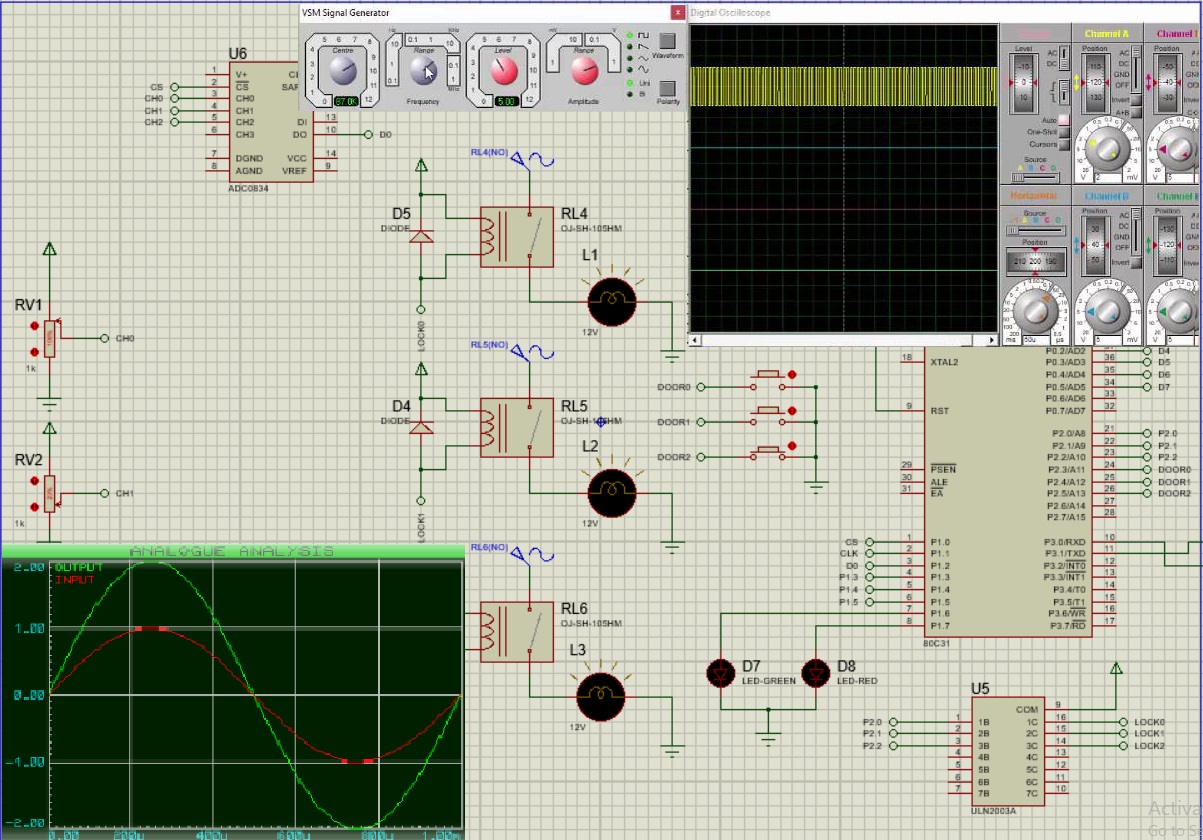


Figure : Serial communication

The figure above shows the successful serial communication between the microcontroller and the serial communication. The oscilloscope shows the communication from the raspberry pi with the frequency of 1.0582 microseconds and the baud rate between the microcontroller and raspberry pi is set to 9600 so that the communication is successful.

The graph and the bottom shows the two signals of the TXD and RXD between the two devices, when raspberry pi sends the information serially to the microcontroller using TXD, it also receives the feedback of results using RXD. Or it might be the instruction to the microcontroller to perform. This also proves that other devices connected to the microcontroller are cooperating and functioning as expected.

With this results it is evident that the relay connected to locks and the sensor with its interface are all responding to the microcontroller, as the microcontroller is able to respond back to the raspberry pi using RXD.

The LED’s on the schematic are turned on, they are monitoring the functionality between the Microcontroller and the relay that connects to the magnetic locks. as they are turned on, this proves that the serial communication is success and is fully in control of the locking or unlocking  mechanism of the storage units using the raspberry pi so that the user and store or retrieve the parcels.

Also the pull down switch will monitor the storage units, if they are pressed the lights goes off. This will be used in case of user opening the storage door, and they will be physically mounted on the storage door. This analysis show success operation of storing or retrieving mechanism of our system.

#### 2.5.3 Comparisons

### 2.6 Project Outcome

It is evident that the current parcel counter systems have not been working effectively enough to satisfy customers’ needs. As mentioned in chapter one that the main issues were based on problems like, customers belongings being misplaced, human error, and theft from the parcel attendant, availability and time consumption.

As these issues were proven in most of the retails, then the implementation of this system will eliminate many of the human errors and also advance in more functionality that will make shopping more convenient for our customers. The backbone controlling the system includes raspberry pi 3 model B and 8051 Microcontroller (AT89S8253).

This the most useful feature to make a concrete security and efficient communication between the system and the customers. Hence we agree this is a good solution to theft, misplacement and confusion of customer’s items.

This also come to another point that was raised in chapter one, which is the issue of availability and time consumption. It is clear that the storage system will be available anytime a customer wishes to store or retrieve the parcels. And the time that a parcel attendant takes to store the parcel of each and every person, it is time consuming for most customers. hence we solve the problem by making the storage boxes reachable by any customer at any time, where customer can store their parcel/items simultaneously and this will be possible because raspberry pi 3 provides Bluetooth and WIFI connectivity for a Website we created for the customer to store or retrieve their parcels using Web Browser on their cell phones, also the user interface will increase the process of storing/retrieving more faster.

The main interaction happens between the microcontroller and the raspberry pi. A raspberry pi carries the main functionality since is the one interacting with the user, but it is also depending on the micro controller so that the system can perform its function completely.

With serial communicating, a raspberry pi is depending on microcontroller for transmitting and receiving data such as sensor and locker mechanism. For doors to close, a microcontroller is depending on the relay to supply enough voltage to the magnetic locks, so this mean lock are depending on the relay to perform their function. Then the microcontroller can send back the status of the locks. This also applies to the sensors.

With the sensor mechanism, a microcontroller has to get the parcel status using ADC0834 sensor interface, since is the interface between the actual sensor and the and the microcontroller.ADC0834 is depending on sensor and microcontroller is depending on it, this leaves a sensor being independent.

### 2.7 Recommendation

#### 2.7.1 How can the project be improved:

##### These are some recommendations stating the improvement of this project:

* Build a power supply that will supply sufficient voltage to power the raspberry pi circuit.
* Implement an emergency button for opening all the storage units, in case of an emergency.
* Strong and reliable Wi-Fi connection for communication with a web-server, backup route is needed in case where the primary route is down, secondary route will take over.
* As the UX of shopper grows or develop in navigating the systems UI, it is recommended to conduct a usage poll to determine whether, the web interface usage is favored more than the main interface usage. If the web interface is favored, this then works as an advantage; due to less use of electricity to power the main interface’s display components and its printer resources, and its processes.
* Install cameras to monitor the keyless storage system; this is for cases of feudalization of property.

#### 2.7.2 Practical model

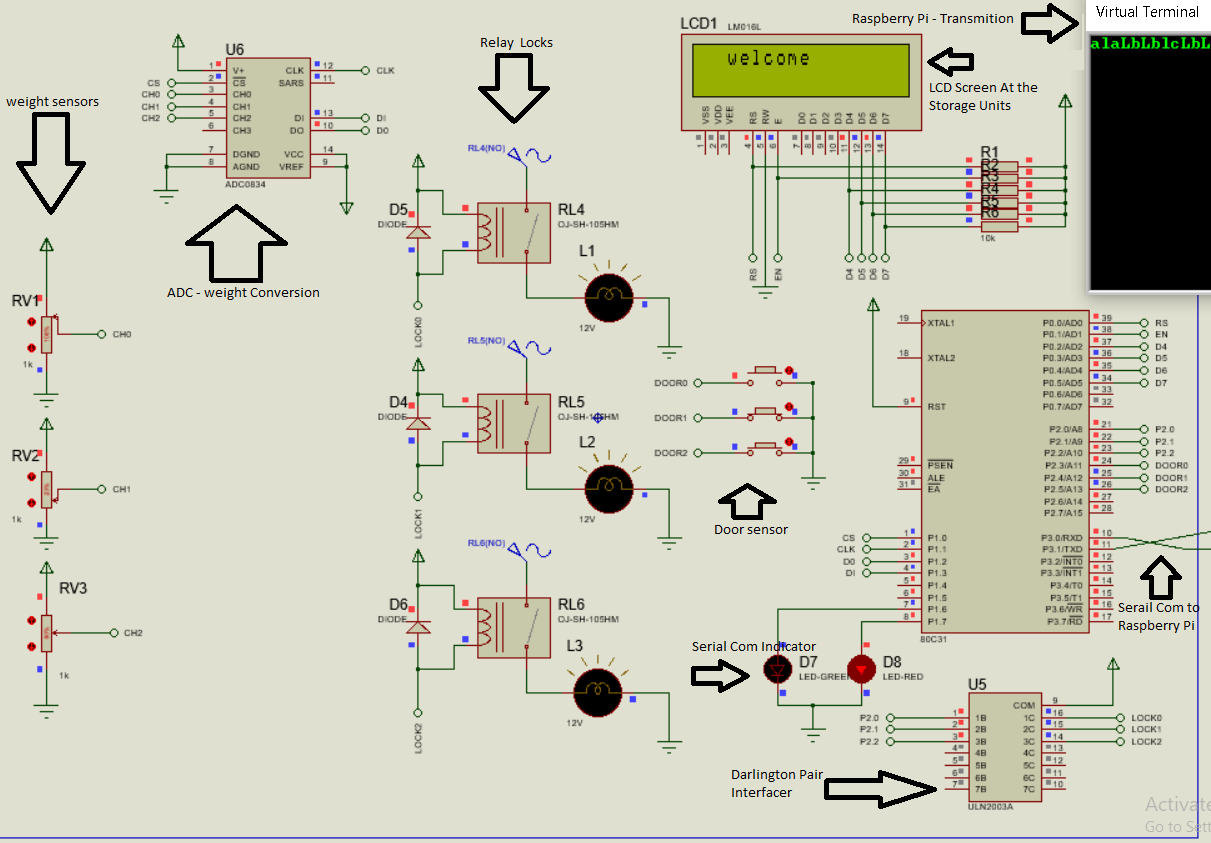


Figure : Shows the practical model.

The figure above show the practical model for the project design, the control devices discussed earlier are soon with arrowheads, the figure is not clear due to image compression, the read is advised to zoom at the figure.

The microcontroller circuit on the figure above; shows how it is able to relay the given commands by the Rpi3; this can be seen at the top right corner of this figure. This is achieved using a serial communication between the major devices, their baud rate (the speed of serial communication measured in bits per second (bps)) is much high; at 9600 bps rate and a greater baud rate can be used. As this design is dependent on the serial communication to relay the control behaviour, this hugely benefit the design, due to gpio pin limitations of the Rpi3, the design implemented the physical security elements on the uC side; the Rpi3 handles the access control to the security elements.

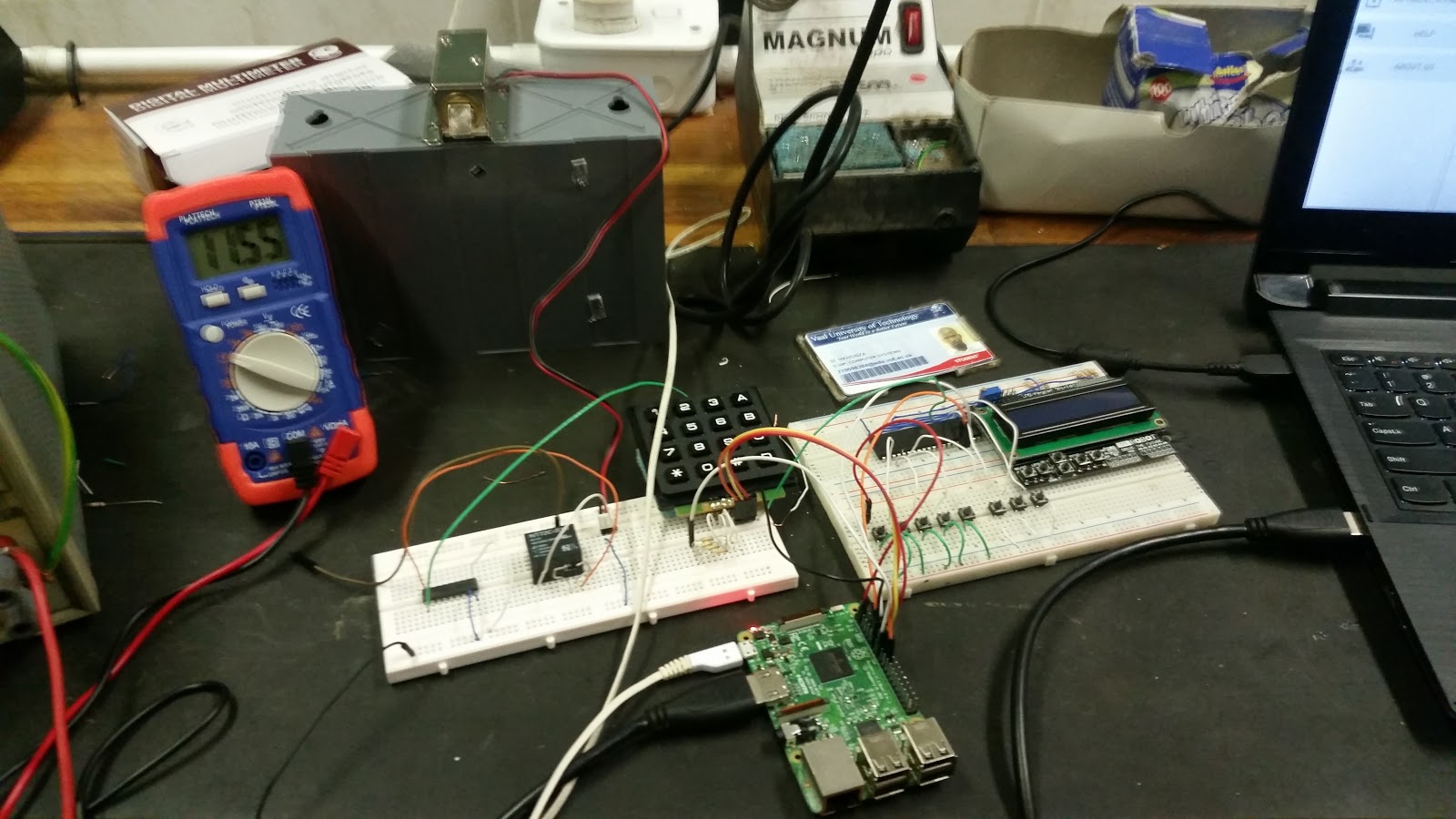


Figure : Shows the practical mode implemented on bread boards.

The implementation of the Practical model was design on bread boards, so to easily implement the design functionality, the major control devices as seen on the figure above.

# Bibliography

Collins Farren. (2017, 07 17). *Sci Tech*. Retrieved from Time Live: https://www.timeslive.co.za/news/sci-tech/2017-07-17-biodegradable-plastic-bags-developed-in-sa/

*Consumer Price Index - Key Findings*. (2018, 01 15). Retrieved 03 15, 2018, from Statistic South Africa: http://www.statssa.gov.za/publications/P0141/P0141January2018.pdf

*Control a HD44780 LCD display via I2C with the Raspberry Pi*. (2017, 09 14). Retrieved 04 11, 2018, from Tutorials Raspberry Pi: https://tutorials-raspberrypi.com

*Keyless Entry*. (2018, 03 12). Retrieved 03 12 , 2018, from Anytime Lock & Safe : http://anytimelocksafe.com/keyless-entry/

penmof. (2017, 09 27). *bag-stolen-from-baggage-counter-with-accountability-2353198*. Retrieved 03 16, 2018, from hellopeter: https://www.hellopeter.com/shoprite/reviews/bag-stolen-from-baggage-counter-with-accountability-2353198

Purpura, P. P. (2002). Security And Loss Prevention 4th Edition. In P. P. Purpura, *Security And Loss Preventation, 4th Edition* (p. 554). Oxford Butterworth-Heinemann.

Swiegers, W. (n.d.). Shoprite Holdings Ltd supermarket. *Bloomberg via Getty Images.* Getty Images, Alexandra district, Johannesburg, South Africa.

Yuval Noah Harari, K. Z. (2015). The Gods of Planet Earth. In K. Z. Yuval Noah Harari, *Homo Deus - A Brief History of Tomorrow* (p. 32). Israel: HarperCollins Publishers.

# Annexure

On this annexure the reader should expect discussions not mentioned on the document above.

## User Interface

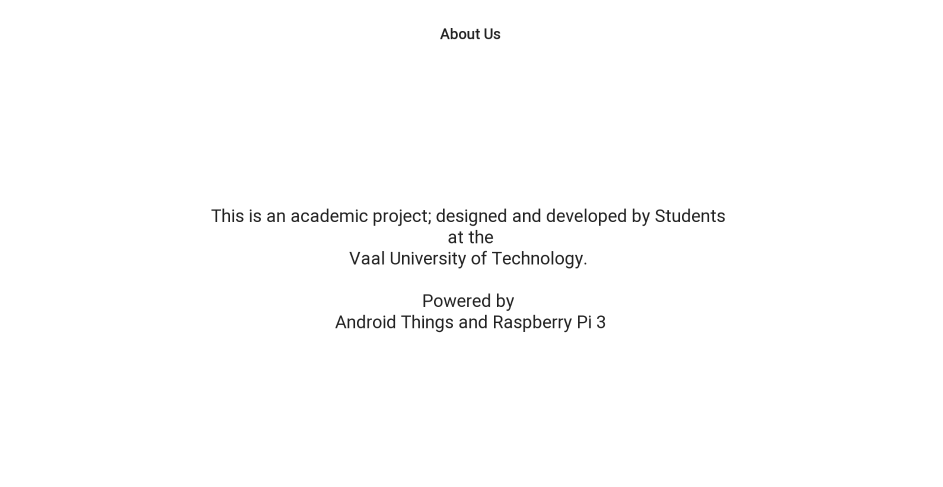


Figure : About us, user interface screen

The User Interface (UI) used the Android Things/Android Development Platform, please refer to Control Devices under the Raspberry Pi: The User Interface for more information.

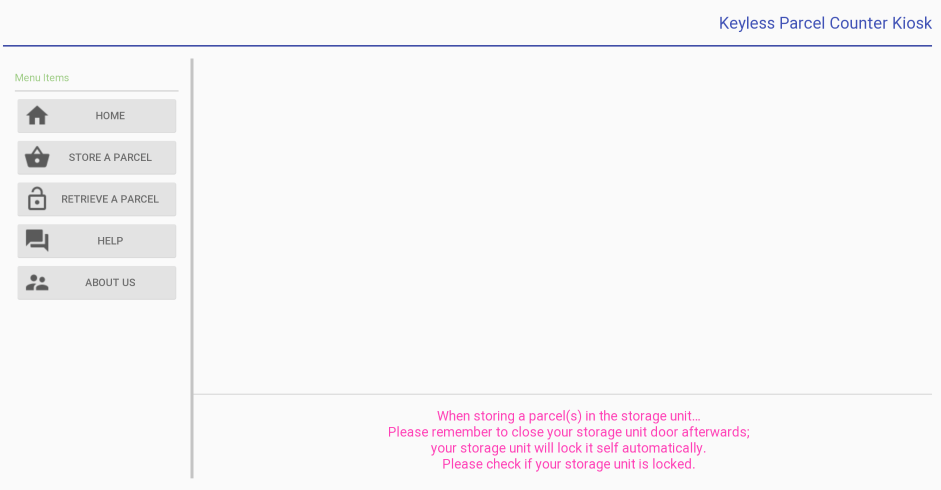


Figure : Shows Main UI Screen for the design

## Database

The database used to store the shopper’s token, was designed using SQLite3 supported by the Android App Development and also the Android Things Development Platform.

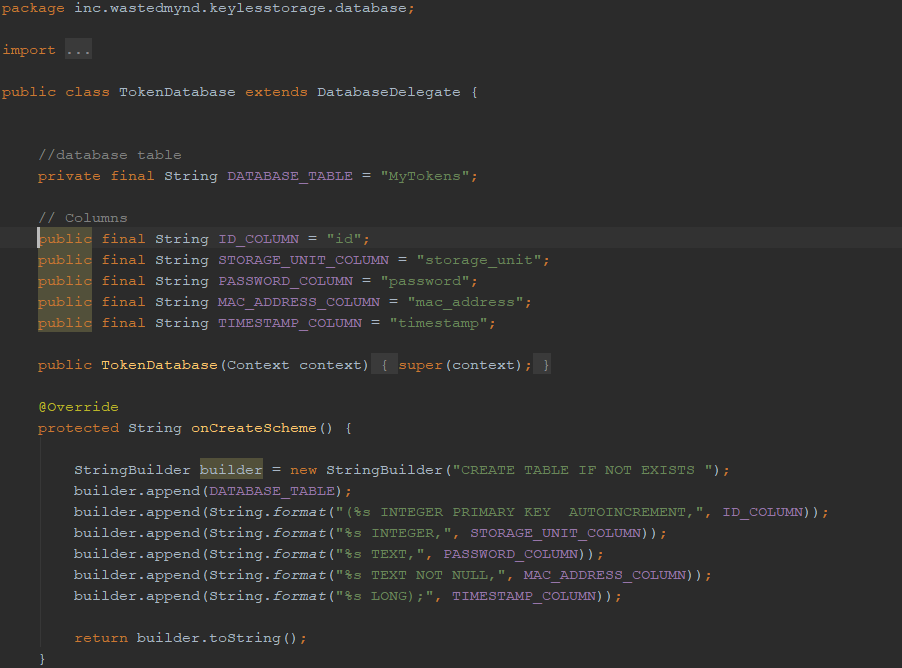


Figure : Shows the SQLite3 database; Database table name, and its scheme.

The figure above shows the TokenDatebase class that inherits its functionality from the DatabaseDelegate class; all the design code can be found on Github repository including the one above and the MikroC code (<https://github.com/wastedMynd/KeylesStorageSystem>).

## Serial Communication

Baud rate = 9600 bps >= 115200 bps

### Transmutation Code, (From Rpi3 to uC and the uC Responses).

**To monitor Parcel(s) Placed in a storage unit**. For the first storage unit: Raspberry Pi sends ‘d’, to the micro controller transmits back a 'p' meaning these no parcel and 'P' these a parcel. For the next storage unit an ‘e’, and an ‘f’; these sort of command might be change as more storage unit get incorporated to the design.

**To monitor storage unit door(s).** For the first storage unit: Raspberry Pi sends ‘g’, the micro controller transmits back a ‘d’ meaning that the storage unit door is open, and a ‘D’ door is closed. For the next storage units doors an ‘h’, and an ‘i’ are used for the monitoring of their doors respectively; these sort of command might be change as more storage unit get incorporated to the design.

**To Access the locking component on each storage unit**. For the first storage unit: Rpi3 sends ‘a’, followed by a ‘l’ (lower case L), for an intent to unlock a storage unit; the first command is used to locate the storage unit; an upper case ‘L’ is used for the intent to lock the storage unit. For the next storage unit a letter ‘b’ is sent to the microcontroller to locate the Second storage unit, and third storage unit the letter ‘c’; these sort of command might be change as more storage unit get incorporated to the design.

## Electronic Components Source:

ManTech-Electronics 

Figure : All the electronic components used in the design

# Notes