ASHESI

DESIGNING A COMPUTER NUMERICAL CONTROL FARMING SYSTEM FOR LOCAL HOMES

CAPSTONE PROJECT

BSc Computer Engineering

JOSHUA EDEM GASU

2020

ASHESI UNIVERSITY COLLEGE

DESIGNING A COMPUTER NUMERICAL CONTROL FARMING SYSTEM FOR LOCAL HOMES

CAPSTONE PROJECT

Capstone Project submitted to the Department of Engineering, Ashesi University College in partial fulfilment of the requirements for the award of Bachelor of Science degree in Computer Engineering.

Joshua Gasu

2020

DECLARATION

I hereby declare that this capstone is the result of my original work and that no part of it has been presented before for another degree in this university or elsewhere.

Candidates Signature

…………………………………JEG……………………………………………..

Candidate's Name

…………………JOSHUA EDEM GASU………………………………….

Date:

………………………………………………………………………………

I hereby declare that the preparation and presentation of this capstone were supervised in accordance with the guidelines on supervision of capstone laid down by Ashesi University.

Supervisor's Signature:

……………………………………………………………………………….

Supervisor's Name:

……………………………………………………………………………….

Date: ………………………………………………………………………

Acknowledgements

I dedicate this paper to God Almighty and my parents for their support and encouragement throughout the process.

To my supervisor, Mr. Kofi Adu-Larbi whose interest, encouragement and academic advice helped me undertake this project.

I would like to thank Mr. Samuel Shamo Abbey for his immerse help with CAD models. I would also like to show my appreciation to Mr. Micheal Danquah for his guidance in this project.

ABSTRACT

This project explores the idea of using the principle of Computer Numerical Control(CNC) to build a farm machine. The project can be divided into three facets.

The first involves the mechanical component of the machine. Building the farm and assembling and mounting the motors that would be used for movement.

The second part is heavily based on electrical work. Connecting sensors, and motors and fixing the communication bit for the machine.

The final part is the computerized part. This part involves communication of the machine with a server to provide the user with up to date information where ever they are.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

CHAPTER ONE

1.1.0 BACKGROUND

The gradual increase in population comes with a lot of advantages but also poses a major challenge in the area of agriculture. Researchers are anticipating the increasing population would affect agriculture heavily and this is a major concern [1].

Thus, in this project we explore a solution to this issue viewed from an engineering perspective.

1.1.1 PROBLEM STATEMENT

The general theme concerning this project is the concern surrounding the supply of food as the world population increases. The increase in demand of agriculture produce and an unequal supply of demand, requires a solution to the situation.

1.1.2 OBJECTIVES OF THIS PROJECT

The main aim for this project is to ensure that homes can actively own farms by providing an automated smart farm system that is cost effective and minimize human interaction.This is to also ensure that homes can produce what they eat and generally contribute as individuals to the nation.

1.1.3 EXPECTED OUTCOMES

1. An automated farming system that can achieve basic functionalities like planting, and watering.

2. A system that gives constant updates to keep the farm owner abreast with the current data on farm.

1.1.4 MOTIVATION/JUSTIFICATION FOR THIS PROJECT

The project seeks to engage people who enjoy farming. Also, by creating this machine, it could be scaled up to help larger farms and increase productivity.

This project helps me apply all that I have learned for the past three and a half years to make something beneficial to many.

CHAPTER 2

2.0.0 LITERATURE RESEARCH

To bring a farm in every home, the FarmBot is a CNC farming system that can perform most of the cultural practices before harvesting. That is, it can sow seeds, water plants and can also remove weeds [2]. It can plant about 30 different crops on a 2.9 meter by 1.4 meter area and can be planted both indoor and outdoor [2].

The FarmBot has a universal tool, on which different devices can be mounted on to perform a specific function [2]. Tools like the weeder are inserted on the universal tool to remove weeds, the same for the watering tool, planting tool for planting, and the camera for detecting weeds. Each of these tools must be attached to the universal tool by human intervention. However, the FarmBot has great functionality, using belts on both the x and y-axis to cause the Farmbot to move in the vertical direction. To plant in the y-direction, the Farmbot uses timing belts and uses motors. Mounted on the vertical structure, which moves in the y-axis is a structure that moves in the x-axis. Movement of the structure in the z-axis is to allow the up and down movement of the universal tool for planting, watering, and removing weeds. Attached to the FarmBot structure is the control panel, which uses an Arduino as the brain of the system. The FarmBot also allows for interaction between the user and the system. The brain of the machine communicates with an API, which then sends the data gathered onto a cloud that can be accessed by anyone using the mobile app or web application. The mobile application allows a user to get updated on the farm regardless of their location and also notifies the user of any maintenance that has been made on the farm.

FarmBot is by far the most significant model designed to tackle this issue. However, many others have contributed to this area as well. A self-sustaining farming system designed for a home in Pakistan [3] also implements the principle of Computer Numerical Control in farming to plant and water. In this article, the authors explain their approach to tackle this problem, like Farm Bot. However, the Self- Sustained Farming System (SFS) is useful in such terrains where conventional farming is difficult [3]. The design of the SFS makes it capable of performing multiple tasks, such as weeding, seed injection, and soil moisture measurement. Moreover, the SFS changes the daily watering pattern for different plants concerning the weather conditions of the area. Furthermore, the cloud interface of the SFS provides the feature of continuous updated remote assessment of the SFS system parameters. The SFS has another advantage to work both in grid-connected and off-grid (solar-powered) mode [3].

Most of these systems use threaded rods for movements on both the x, y, and z-axis. The threaded rod is a good material; however, it is too slow and takes a while to complete the task. Also, these systems will need human intervention at some point before harvesting, thereby defeating the purpose of making it autonomous. Also, some of the components increase the cost of the final product making it a little difficult for a larger market to purchase.

This is what my project is trying to solve. To reduce the human factor on a farm in a household setting and improve the quality and reduced costs of the machine by using different components. How can the machine be built, such that human intervention is not needed for a process to take place? Thus, I seek to solve this issue by further building extra components around the seed planter, which will achieve the same thing if there was human intervention. Reducing human intervention on the farm would make the system a lot more autonomous and more reliable. It would be economically purchasable by many.

CHAPTER 3

3.0.0 **Review of Existing Design**

From Chapter 2, an overview was given to some existing solutions. Regardless of the approach, the cost seems to be a limiting factor, and it shows from the statistics, according to Farm Bot. Comparing its deliveries to the continents, Africa had the least patronage. This was attributed to the cost of the product.

Thus, the proposed solution combines implementations from existing designs and using knowledge of APIs for webserver communication to produce a robust and, more importantly, cheap system. Figure 1 shows the existing design for FarmBot which is an existing CNC farming system.

FIG 1: An image of a Farm Bot CNC farming machine

3.0.1 How Does this Device work

The system is made up of two parts, hardware, and software. The hardware is the CNC farming system that is a self-automated farming system that performs every cultural practice before harvesting. It achieves this by moving motors that are attached to a 3D printed seed planter, which holds the seeds. The farm is treated as a matrix; thus, each quadrant in the matrix is treated as a solo seed planting land. Thus, at say, (1x1), a seed would be planted; after this process, the motor is powered again and then moves to the location(1x2). The same process occurs here. The process is repeated until each quadrant in the matrix is filled.

The same idea is applied when the farm is irrigated. It moves to each quadrant and waters it until the whole matrix is watered. Each process that is completed is communicated and transmitted unto a webpage that holds information on all processes completed. The webpage can then be accessed through a mobile app that would display any relevant information to the user.

3.0.2 Block Diagram Design

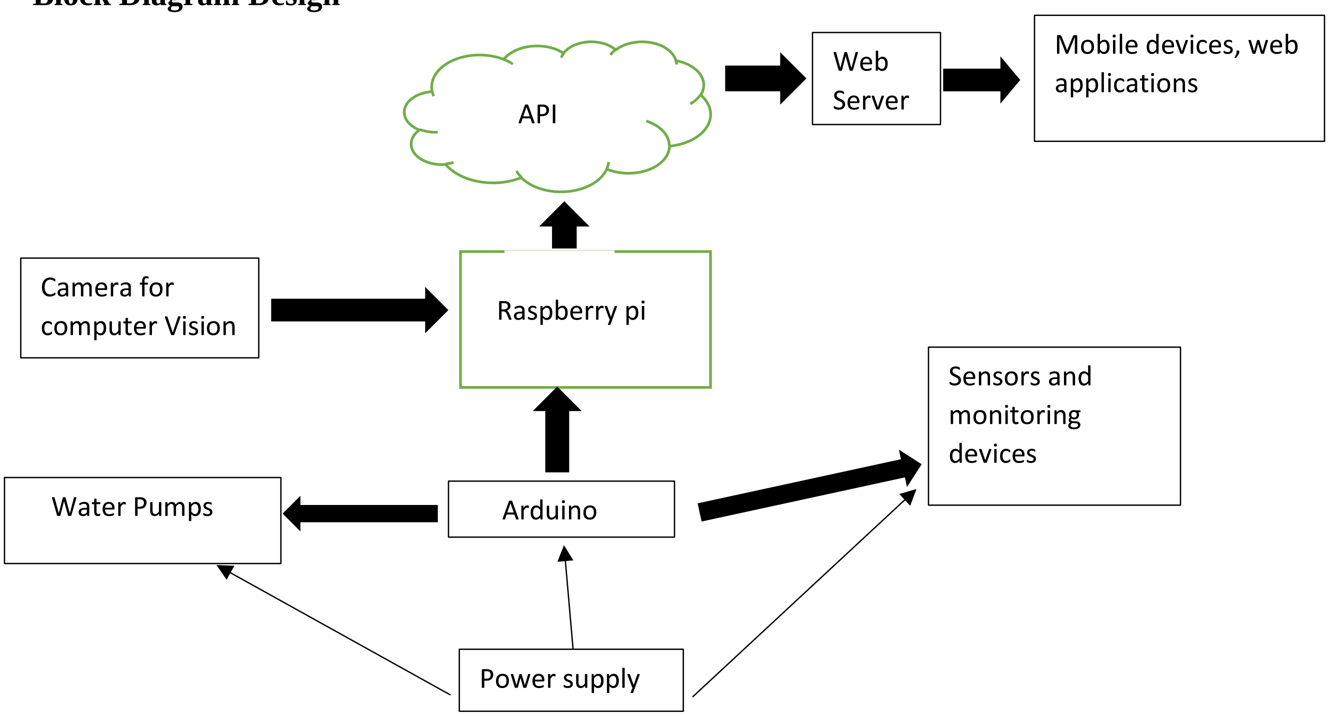


FIG 2: A block diagram of the system

The project is divided into three main parts - the mechanical (building the physical system). Electical(connecting sensors and electrical components) and the mobile application. Figure 2 is a block diagram that shows how both the hardware and software together to function as a unit.

3.0.3 Component Selection

To make the component selection, a Pugh matrix was used. A Pugh matrix is a tool that serves as a benchmark for the selection of components.

3.0.4 Working with a Pugh Matrix

A fundamental matrix consists of alternative proposals and a set of criteria. Start by writing the criteria in the left-hand column and the alternatives in the top row. Fill the matrix by indicating how each alternative is compared against the standard (in the form of pluses and minuses). Finally, look at the bottom of the table to make the best decision for your situation.

The Baseline represents the general component typically used to implement or solve the problem.

		Baseline	Weight	React Native	Node Js
Criteria					
1	Cost	0	1	0	0
2	Hot Restart	0	3	0	-3
3	Complexity	0	4	-4	-4
4	Documentation	0	3	+3	+3
5	Updates	0	2	-1	+2
Score				-2	-2

TABLE 1

The Baseline for this project is Flutter. Flutter is being compared with React Native as well as Node JS which are all software developing kits for making mobile apps. The option for the citeria were very specific. In mobile app development, Hot restart is a feature that allows the coder to quickly view what exactly he or she is doing without having to restart the whole application, thus, its name Hot restart. Cost was another criteria. How much was it going to cost to get the software, and how much was it going to cost to get tutorials to understand the language. Documentation is another important criteria. From the developers, documentation is an easy way for the coder to navigate his way through the process without going through a difficult process. One can liken this as an architecture plan in the hands of the architect. Makes a lot of researching easier. Finally updates, updates are an important feature in any software development kit. Most of these kits are not bug free. However, the more updates come, the more these bugs are fixed and more features are added to make the coders life easier.

Project Components:

1. **Aluminum Extrusions**: Extrusions are the hardware components that would be mounted on each other to build the frame.
2. **Valves and Pumps:** Valves and pumps would be used in conjunction with pipes to provide the irrigation functionality.
3. **3D Printed Holder:** This component would hold certain aluminum frames to allow for easy movement along the y-axis of the farm planter.
4. **Electronic kit:** Consists of wires, resistors, LEDs, transistors for electrical connections.
5. **Motors:** Motors are used with a GT2 timing belt for motion in both the x and y-axis.
6. **Servo Motor:** Used in the manufacture of the seed planter.

7. **Camera:** For image processing.
8. **Sensors:** DHT11(Temperature and Humidity Sensor), Soil Sensor.

3.0.5 DETAILED DESIGN

The project is divided into two major parts

- Hardware Design(Mechanical and Electrical)
- Software Design (Computer)

Hardware Design

The hardware design for the CNC farming system can also be divided into three facets:

- Movement
- Planting and Watering
- Weeding

MOVEMENT

The design for movement is solely to ensure easy movement for parts in the X, Y and Z axis.

In Figure 3, the base of the farming system is wood.

FIG 3: A *design for my CNC farming system*

The dimension for the base of the structure in Figure 3 is 90cm by 30cm. The type of wood used for the base is hardwood. There are a few advantages of using hardwood.

One advantage of hardwood is it is easy to maintain and easily remains clean.

They are highly resistant to water as compared to softwood; thus, it takes a long time for them to rot. Finally, hardwood is strong and can withstand a lot of situations and adverse weather conditions [4].

Therefore, hardwood is used as the base of the farming system. It is upon this hardwood base that other structure like the aluminuim extrusions are mounted.

Using aluminum extrusions, a similar structure in Figure 3 is built. The aluminum extrusions are equal in dimension to the dimension of the base structure for Figure 3, which is 90 cm by 30 cm. These extrusions are mounted on either side of the base (y-axis). This structure allows for movement on the y-axis, as shown in Figure 7. This structure allows movement in the y-axis.

The seed planter would be mounted on the extrusion in the x axis to ensure planting occurs in every row. The details about the seed planter would be discussed next.

Movement on the specified axis listed previously can occur because of the component designed to ensure smooth movement. Figure 4 details how it works. The structure is 3D printed and has a little extrusion to fit six bearings, which are (5mm by 19mm by 5mm). Three bearings are placed on the top and bottom as shown in Figure 4. Figure 5 shows how this is implemented in the system.

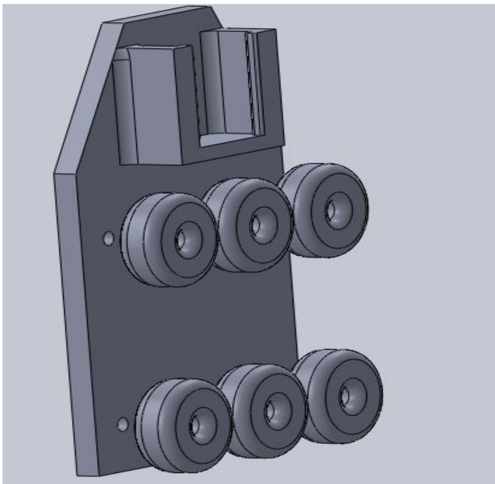


FIG 4: *A 3D printed component fitted with bearings replacing the threaded rod*

A motor is attached to FIG 4. Figure 5 is a figure to show how it looks like.

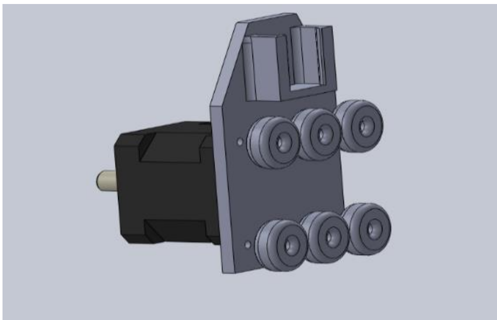


FIG 5: *A motor attached to the component for movement*

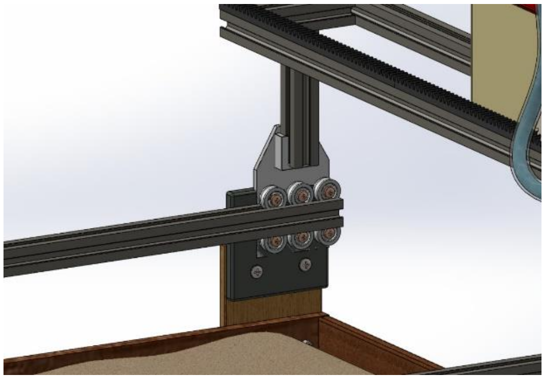


FIG 6

Figure 5 can easily be fit into the crevices of the aluminuim extrusions as shown in figure 6, thus anytime the motor is powered, it moves.

The motor is controlled with an Arduino microcontroller and the GRBL module. Arduino is paired with a GRBL module. GRBL is a free, open-source, high-performance software for controlling the motion of machines[5].

The motor is coupled with a GT2 timing belt. The motor will move on the GT2 timing belt, causing the structure to move in the y-axis. Figure 7 shows how it will be assembled.

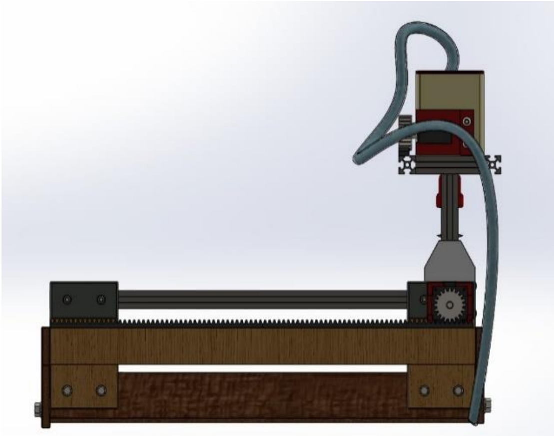


FIG 7: *Side view of the system*

SEED PLANTER

The seed planter is a 3D printed component and designed as shown in Figure 7.

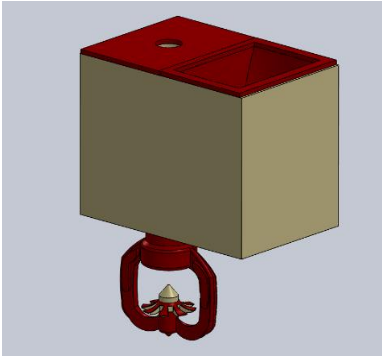


FIG 8: *Figure of the Seed Planter & Water Machine*

The seed planter is functioning as a seed planter and as a watering machine. Thus in this description, focus is placed on each aspect and how they operate. Figure 9 is an image of the top view of the seed planter,

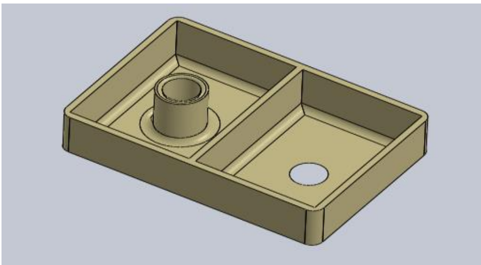


FIG 9: *Interior look into the seed planter*

The left side of Figure 9, details the design for the watering system. The other side would host the seed planting mechanism, which would be of a funnel-like shape, as shown in Figure 10. The left compartment in Figure 9, would house the watering mechanism. The cavity in that compartment would be connected to a water hose which supplies water needed for watering.

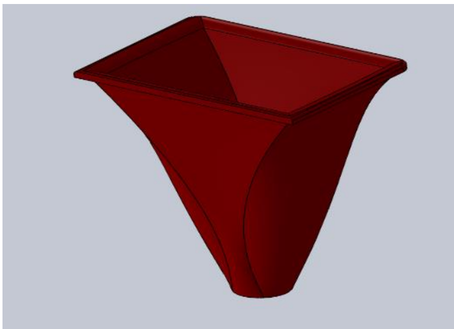


FIG 10: *The 3D printed funnel*

The base of the funnel-like structure is fitted with a servo method. The main function of the servo motor would be to turn to allow the seeds in the neck of the funnel-like structure to drop. Since the servo motor is a little slow, it is coupled with a slot that would allow a minimum drop of seeds.

Planting

As stated earlier, the seed planter is 3D printed mainly because of weight issues. By 3D printing, the weight is reduced to ensure that the motors can easily move the seed planter as shown in Figure 3 on the x-axis.

The seed planting section is designed in a funnel-like manner. The head of the funnel would contain the seeds, while the neck of the funnel is attached to a servo motor.

The servo motor is used to control the rate and when the seeds fall as well. This type of planting is known as drop-seed planting.

It mimics the traditional form of planting without human intervention.

Watering

For a relatively smaller project, this is the ideal approach to water. However, for larger systems, a smart pipe system would have to be laid on the farm to ensure watering is done.

For this system, though, the seed planter is designed to hold an elastic rubber hole that is connected to a well or water reservoir. A pump and valve are connected to the rubber hole, also controlled by the Arduino microcontroller. Thus, irrigation patterns are determined intelligently, and wastage of water is minimized.

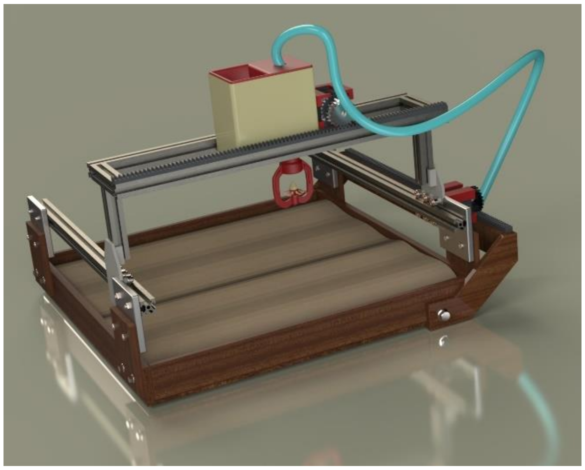


FIG 11: *Blue tubing carries water from the reservoir to seed planter*

Chapter 4

4.0.0 Software Design

As discussed earlier, this project designs a smart farming system for households. Software is an integral part of this build, In this chapter, a lot of focus is put on the software and how each build works together to inform the user of the progress made on the farm.

Every tool and software development kit used is explained and the rationale behind every choice. The main sections of the software of the project would be the

- The website connected to a database which receives information from the farm
- The mobile application

4.0.1 Website and Database design

The main frameworks used to build the website were Hyper-Text Markup language(HTML), Cascading Style Sheets(CSS), Canvas, JavaScript, and SQL.

The main idea behind making the website is to create a platform that can directly communicate with the farm and receive sensor information from the farm. This would be displayed through the mobile application giving the user updates in real-time.

HTML was used with CSS to create a stylish platform, while Canvas was used to develop the radial dials or speedometer look-alikes for effect as shown in Figure 12. The speedometer dial lies on the actual number from the database representing the actual sensor value.

FIG. 12 *The speedometer look-alike to display sensor values at the front end*

Figure 12 is a representation of what was built with HTML, CSS, and Canvas. The database is built using SQL, which receives the sensor data from the farm, which would be discussed later on in this chapter. The database gets sensor data for temperature, humidity, and soil moisture. These are the basic maintenance features added to update the user. Figure 13 shows a visual representation of the database or the schema of the database.

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
	1	**Date**	date			No	*None*			Change Drop More
	2	**Temperature**	int(11)			No	*None*			Change Drop More
	3	**Humidity**	int(255)			No	*None*			Change Drop More
	4	**SoilSensor**	varchar(255)	latin1_swedish_ci		No	*None*			Change Drop More
	5	**Water-Level Sensor**	int(11)			No	*None*			Change Drop More

FIG 13: *A database to receive sensor values*

The data from the farm is received into the database. However, to get the animation of moving the dial to actual sensor values received, the AJAX code displayed in Figure 14 is used. The snippet of code in Figure 14 receives the sensor value from the database, and according to the value, it moves the dials in Figure 12 to the actual value.

```
<script type="text/javascript">
    function setValue(){
            var xhttp = new XMLHttpRequest();
            xhttp.onreadystatechange = function() {
            if (this.readyState == 4 && this.status == 200) {
                var val = JSON.parse(this.responseText);
                $("temperature").attr("data-value", val.temperature);
                $("humidity").attr("data-value", val.humidity);
                $("soil").attr("data-value", val.soil);
                $("water").attr("data-value", val.water);
            }
            };
            xhttp.open("POST", "back.php", true);
            xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
            xhttp.send();
        }

    setInterval(setValue, 5000);
```

FIG 14: *Ajax code*

Next is to show how data is sent from the farm to the database. To send data (sensor data) from the farm to the database, a NodeMCU, which is a WIFI module that can send data to a database, is used. NodeMCU Dev Kit/board consist of an ESP8266 Wi-Fienabled chip [6].

The ESP8266 is a low-cost Wi-Fi chip developed by Espressif Systems with TCP/IP protocol.

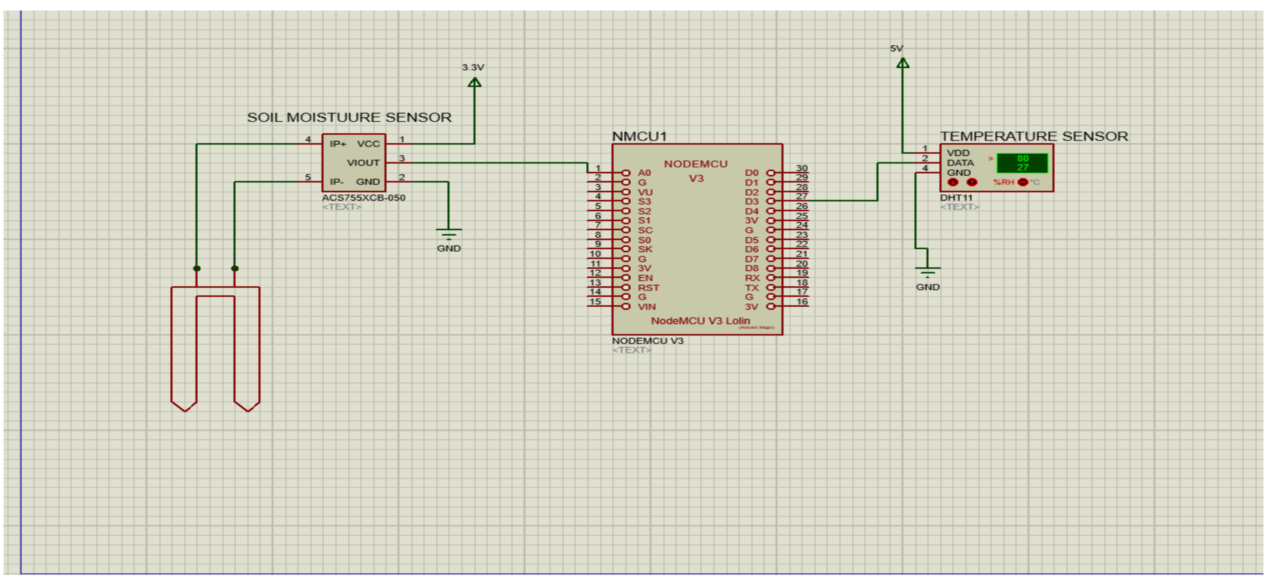


FIG 15: *A schematic of the connection of sensors to the NodeMCU*

The Soil sensor is connected to the NodeMCU, as shown in Figure 15 , as well as the DHT11 sensor, which reads both temperature and humidity values. The NodeMCU, as stated earlier on, is a Wi-Fi module, and this ability makes it the right board to send sensor information to the database.

The NodeMCU must be configured to send data from the farm to the database thus, Arduino is used as the development platform for any configuration for the NodeMCU. Figure 16 is a snippet of the code used to configure the NodeMCU.

```
      Serial.printf("[HTTP] GET... failed, error: %s\n", http.errorToString(httpCode).c_str());
    }

    http.end();
  } else {
    Serial.printf("[HTTP} Unable to connect\n");
  }
}

if (currentMillis - previousMillis2 >= interval2) {
  previousMillis2 = currentMillis;

  Serial.print("[HTTP] begin...\n");
  if (http.begin(client, "http://192.168.43.108/iot/iotlab3/insert.php?insert&SensorName=DHT11&SensorType=Temperature&CurrentReading="+String(temperature))) {


    Serial.print("[HTTP] GET...\n");
    // start connection and send HTTP header
    int httpCode = http.GET();

    // httpCode will be negative on error
    if (httpCode > 0) {
      // HTTP header has been send and Server response header has been handled
      Serial.printf("[HTTP] GET... code: %d\n", httpCode);

      // file found at server
      if (httpCode == HTTP_CODE_OK || httpCode == HTTP_CODE_MOVED_PERMANENTLY) {
```

FIG 16: *Code to configure the NodeMCU*

In Figure 16 is code that allows the NodeMCU to send temperature data to a database.A similar code is used to upload soil moisture data and humidity data as well.

4.0.2 Mobile Application

There is a multitude of software development kits (SDK), but Flutter developed by Google is used for this project. Flutter was a good option because [6];

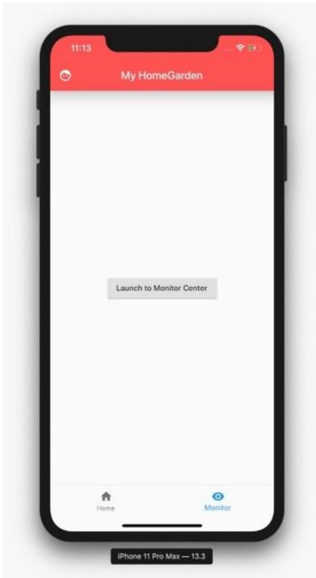
1. With one single source code, you can develop for both Android and iOS systems with minimum tweaks. It saves the developer a lot of time.
2. It has a hot reload functionality. A hot reload allows a developer to see changes he or she makes immediately.
3. It is also Open source and easy to work with.

To design the mobile app, the idea was to use the widgets provided in Flutter to create a simple user interface (UI) that would be appealing to the user and also offer the basic functionality the user needs.

Firebase database was used to keep track of all users using this application. Using the firebase database comes with many possibilities and advantages[7].

The mobile application has two main sections;

1. The page that updates the user of the weather. This was an extra addition.
2. The monitor page, which is a URL opener widget in Flutter which would open the aforementioned website in the app. Initially, the goal was to open the website in the mobile app; however, the widget in question to do this was not functioning as well as it ought to. Figures 17, 18, 19 and 20 are images of the functioning app in iOS



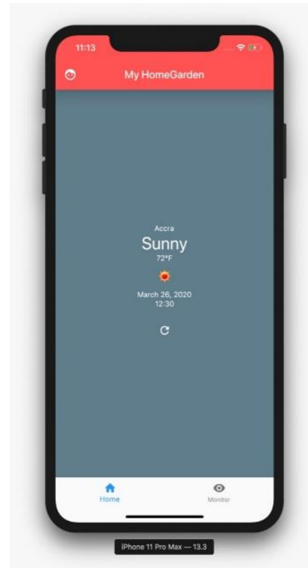
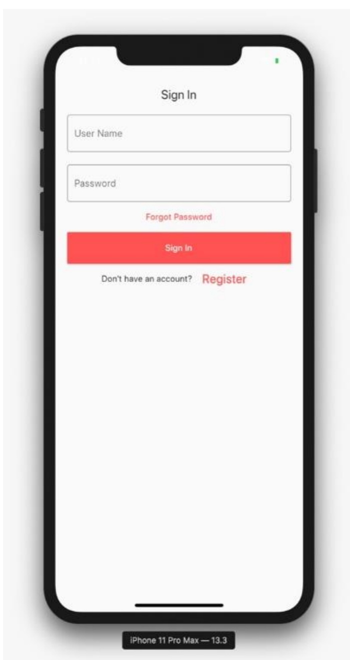


FIG 17: *Monitor Page*

FIG 18: *Weather App Page*



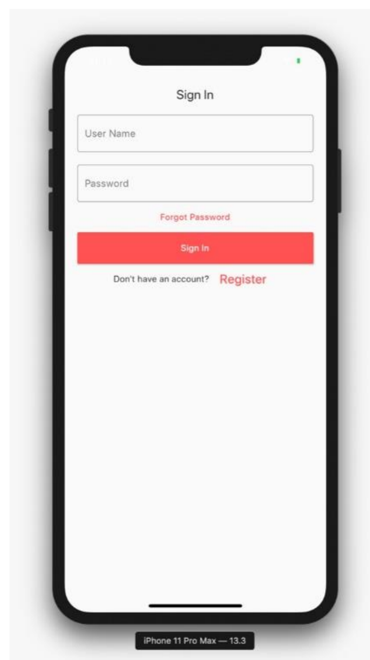


FIG 19: *Sign In Page*

FIG 20: *Sign Up Page*

Upon hitting the monitor page button, the user is directed to Figure 12 which provides the user with a live update of what exactly is happening on the farm.

Chapter 5

5.0.0 Implementation of Farming System

In this chapter, a step by step analysis is provided on how the machine works. A technical view of the system is provided, and specifications are listed here.

As stated earlier, this is a CNC farming machine coupled with a mobile application. A lot of focus would be placed on the hardware part since most of the mobile operations have been covered in the previous chapter.

The hardware is the farm, the central part of the project. The base of the farm is constructed with hardwood about 90 cm by 30 cm. Figure 3 is an overview of the look of the system; however, to make this more understandable to the reader, figure 21 would be used as a visual guide.

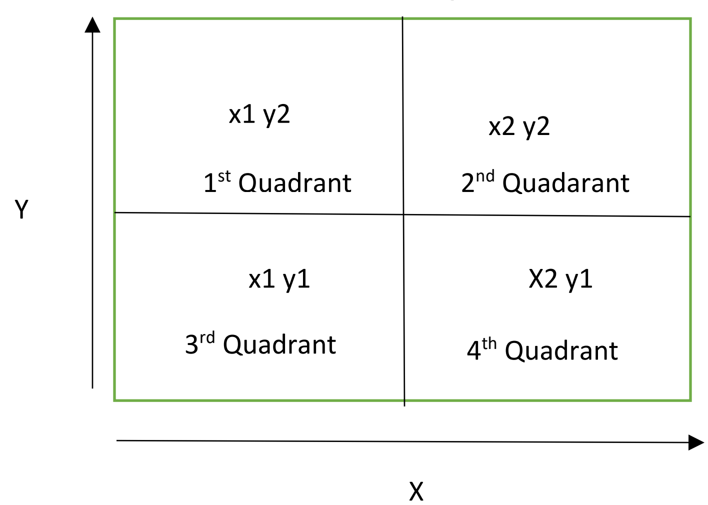


FIG 21: A diagram to illustrate the movement of the planting system

Let us assume that Figure 21 represents the dimensions of the farm. Kindly refer to Figure 3 since Figure 21 is a watered-down look of how the system ought to work. The basic movement on the farm is to move from x1y2 (which is our start) to x2y2. The next transition is to move completely from the first and second quadrant to the third and fourth quadrant. Essentially, movement is done across a row. When planting is completed across the row, it moves to the next column. This process can be applied on a larger scale as well.

On each side X, Y are placed motors, as shown in Figure 6. The details of Figure 6 are explained earlier on. A supporting structure is built on that side to allow a timing belt to run across that

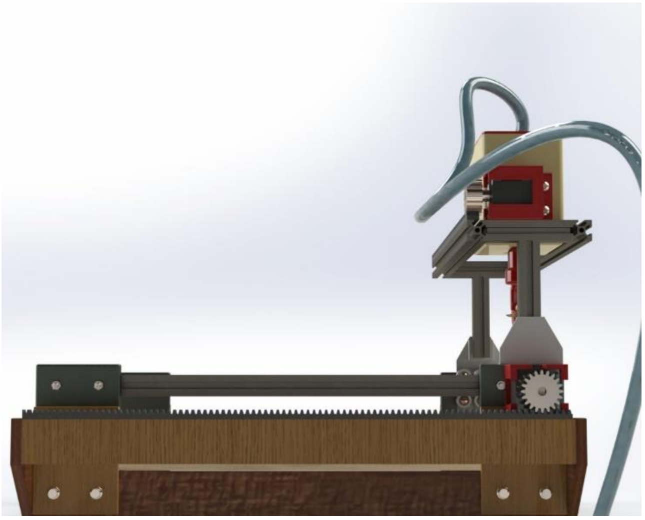


FIG 22: *Structures that aid in moving*

structure. This would then cause the whole system to move in the Y-direction. Similarly as shown in Figure 22, a motor is attached to the seed planter allowing it to move horizontally. This allows for movements from the first quadrant to the second quadrant. And the structure placed in the Y direction allows for the machine to move from the first and second quadrant to the third and fourth quadrant.

FIG 23

The labeled parts in Figure 23 on the system indicate the structures mentioned earlier on, which aid in movements. For the machine to know where it is on the farm, the GRBL module is used.

GRBL is a free, open-source, high-performance software for controlling the motion of machines that move, that make things, or that make things move, and will run on a straight Arduino. If the maker movement was an industry, GRBL would be the industry standard [5].

Most open-source 3D printers have GRBL in their hearts. It has been adapted for use in hundreds of projects, including laser cutters, automatic hand writers, hole drillers, graffiti painters, and oddball drawing machines. Due to its performance, simplicity, and frugal hardware requirements, GRBL has grown into a little open-source phenomenon[5].

With its vast applications and simplicity, the GRBL module is used in positioning the system. The GRBL works alongside Arduino. However, to gain a detailed position with a GRBL, the GRBL controller application is needed to set co-ordinates for the movement of the motors. By setting the GRBL, using the GRBL controller, the motors can be configured to run and stop at certain positions, say x1y1. This would be done through a process of trial and error, to get the right coordinates. By using the GRBL module, the motors can be configured as well as move them to the necessary positions.

After a few tests and observations, it can be observed the amount of time it takes to drop the seeds; thus, the GRBL module can be configured such that at position x1y1, it would stay in that same position and then move to the next position after a certain time has elapsed. The simple principle is used at higher levels for bigger CNC machines.

A simulation using a Matlab program was run to establish the amount of time it takes for the system to plant the crops in the whole system. To begin with, the motors controlling the x-axis were simulated to run and the time it takes to complete a full motion across the 30 cm breadth was taken. A simulation was run to generate the displacement against time graph. In the simulation, it took 10 seconds for the system to complete a row-wise planting. Fig 24 is a graph of displacement against time of the x-axis planting.

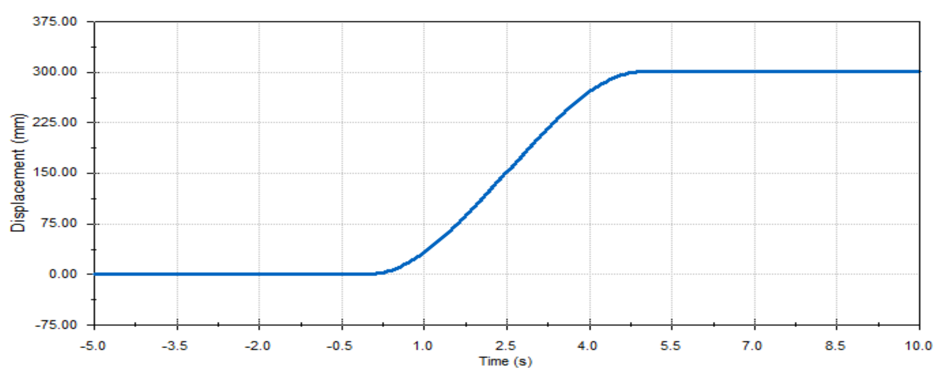


FIG 24: Graph for motion in the x-axis.

As stated in earlier chapters, sensors were mentioned. Specifically, temperature, humidity and soil moisture sensors. Their data was captured in the database for a period. Figure 25 shows the values collected over a period. Each of the sensor values were displayed on the radials as show in Figure 12.

1 ▾ > >> | Show all | Number of rows: 25 ▾ Filter rows: Search this table

+ Options

time	temperature	humidity	soil_moisture
2020-04-22 14:13:23	23	45	5
2020-04-22 14:22:44	25	61	35
2020-04-22 14:22:54	26	47	0
2020-04-22 14:22:59	24	64	55
2020-04-22 14:23:04	24	68	54
2020-04-22 14:23:09	26	48	23
2020-04-22 14:23:14	22	69	20
2020-04-22 14:23:19	26	66	28
2020-04-22 14:23:25	26	46	8
2020-04-22 14:23:29	26	51	32
2020-04-22 14:23:34	24	49	12
2020-04-22 14:23:39	25	45	43
2020-04-22 14:23:44	24	54	44
2020-04-22 14:27:40	26	40	29
2020-04-22 14:27:45	25	51	54
2020-04-22 14:27:50	23	44	52
2020-04-22 14:27:55	23	41	55
2020-04-22 14:28:00	24	44	43
2020-04-22 14:28:05	24	49	36

FIG 25: Database receiving sensor data

CHAPTER 6

6.0.0 Discussion

The goal of this project was to provide a cheaper farming CNC system that many could purchase at an affordable price and still have most of the benefits of the standard farming CNC machines. Another goal of the project was to provide a farming machine system that could plant, irrigate , maintain and send updated information to the user.

The following is a comparison between the bill of material of this project and that of FarmBot (An existing CNC farming machine).

Figure 26 is the current and accurate costing of FarmBot. An actual CNC farming machine on the market. To compare amongst the FarmBot models, the smallest model also known as the FarmBot Genesis which is 2.9 meter by 1.4 meter was used.

Category	Genesis Qty	Genesis Subtotal
Extrusions	8	$205.00
Plates and Brackets	23	$347.00
Plastic Parts	53	$246.00
Fasteners and Hardware	665	$252.20
Drivetrain	39	$346.00
Electronics and Wiring	67	$1,186.00
Tubing	26	$111.50
Supporting Infrastructure *	50	$200.00
GRAND TOTAL	**931**	**$2,893.70**

FIG 26: Bill of Material for FarmBot

TABLE 2

Item	Unit Cost	Total Cost
Extrusions	$50	$50
Electrical Components NodeMCU board DHT11 Sensor Soil Mosisture Sensor DC motors (2) GRBL module +Arduino UNO 3 Servo motor	 $10.40 $7.99 $3.25 $44.99 $10.00 $25.85	 $10.40 $7.99 $3.25 $89.98 $10.00 $25.85
Mechanical Components 3D printed material Wood(crafting base)	 $59.90 $70	 $59.90 $70
Tubing Water pump Waer hose	 $40 $22	 $40 $22
Screws, nails, bolts, nut	$10	$10
Mobile app cost	-	
		$399.70

6.0.1 Description of Table 2

Extrusions were alumunium frames on which components like Figure 5 moved on. The length of the extrusions were the length of the base which was stated earlier on. The extrusions were setup on the base of the farm structure.

The electrical components consists of NodeMCU board, the sensors i.e the DHT11, the soil moisture sensor and the GRBL module and motors as well

It should be stated that some of the cost in this project are not captured. For instance, the labour put into doing this project is not part of the costing. Also FarmBot still has some functionalities like image processing that this project currently doesn’t have.

To compare both costs, this project cost $399.00, while the smallest size of a FarmBot cost $2893.00. This project would save the user about $2494. A significant reduction in cost.

This expected outcome was to reduce the price of these CNC farm systems to make them purchasable by many. Comparing $399.00 to $2893.00, there is a significant decrease in cost.

6.0.2 Limitations

Some of the limitations faced during this project are:

1. Lack of components to complete the physical build. All work on the physical project had to be halted mainly due to the pandemic (Corona Virus pandemic). Although the software is complete, testing and trying out the software on the field was not a possibility.
2. This project was to be able to detect growing plants as well as weeds but again because of the lack of components and tests, this could not be carried out.

6.0.3 Future Works

Concerning future works, all unfinished tasks would be completed that is

1. Getting the physical components to complete the irrigation part of the project
2. Adding a few maintenance features
3. Make the seed planting mechanism plant variable seeds. For now it plants one kind of seed.
4. An image processing facet will be added.
5. This project would be tested out with a GSM module for areas with a relatively poorer connection or no access to Wi-Fi.

REFERENCE

[1] A. Z. Z. Karim, "Impact of a Growing Population in Agricultural Resource Management: Exploring the Global Situation with a Micro-level Example," vol. 1, no. 1, p. 9, 2013.

[2] M. Jawad, "Design of a Self-Sustained Farming System (SFS) for Pakistan," vol. 1, no. 1, pp. 6-12, 2019.

[3] R. Carino, "Farm.bot," FarmBot, 3 September 2019. [Online]. Available: https://farm.bot/. [Accessed 13 March 2020].

[4] I. Timber, "International Timber," International Timber, [Online]. Available: https://www.internationaltimber.com/4-benefits-of-hardwoods/. [Accessed 13 October 2019].

[5] endurancelasers.com, "Endurance," - - 2020. [Online]. Available: https://endurancelasers.com/an-important-things-you-need-to-know-about-grbl-firmware. [Accessed 6th March 2020].

[6] ElectonicWings.com, "ElectonicWings," 26 November 2019. [Online]. Available: https://www.electronicwings.com/nodemcu/introduction-to-nodemcu. [Accessed 6th March 2020].

[7] Hackernoon.com, "Reasons Why Flutter SDK Is Ideal For Mobile App Development In 2020," 21 January 2020. [Online]. Available: hackernoon.com/reasons-why-flutter-sdk-is-ideal-for-mobile-app-development-in-2020-mj2u3y9o. [Accessed 20 April 2020].

[8] marmelab.com, "Using Firebase for Backend-as-a-Service: Pros and Cons," 23 October 2019. [Online]. Available: https://marmelab.com/blog/2019/10/23/feedback-on-firebase-in-project-start-up.html. [Accessed 20 April 2020].