Mokka – RFID Custom Coffee Machine

Daniel Betancourt, Guilherme Carvalho, Dylan Kirke, and Alessandro Vecchi

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — The project is to design a coffee vending machine that is able to brew and serve coffee to users. The coffee machine will be able to detect which user is accessing the machine using an RFID sensor. The user will have an RFID sticker attached to their cup. This sticker will contain a unique ID for that specific cup which was attached to the user's account before accessing the machine. The microcontroller will utilize the WINC1500 low power 2.4GHz Wi-Fi module to make a GET request to the backend of our application using the unique ID retrieved by the RFID module. The server will then respond with the user's default coffee preset settings and the machine will begin brewing the coffee. The coffee preferences are all defined by the user on the Mocha mobile application. On the application, users will be able to register cups, modify coffee setting presets, and modify payment information.**

I. Introduction

The driving force of a good product is the ability for it to solve a need. We were motivated to create this product because we observed the desire people had for a simple and fast way to obtain their everyday cup of coffee. We observed the long lines that people had to wait in at the on-campus coffee shops to simply obtain one cup of coffee. There was a need for a fast-premium cup of coffee.

To come up with the design we first had to ask ourselves a few questions. Why do people currently avoid using coffee vending machines to get coffee fast? How can we entice people to keep coming to our machine instead of going to a coffee shop? The answer that we came up with is a fast coffee vending machine that brews coffee on demand instead of storing pre-brewed coffee. The machine will also provide a free membership service powered by RFID stickers that the user will attach to their favorite cup. Whenever a person with an RFID cup sees any of our coffee vending machines, all they have to do is place their cup on the scanner and the machine will look up their desired coffee and payment information along with any coffee specifications.

The most efficient way we thought of to communicate with the machine was through a mobile application. In the application, the user would first register their RFID sticker using a QR code. Then they would enter their payment information and select their desired coffee grinds, cream amount, and sugar amount. They would then select their preferred coffee settings for their next visit to one of the machines. We were motivated to create a mobile application because it gave users the ability to change their coffee settings on the go without having to go to one of the machines to modify their coffee settings. This would also keep lines at the machines shorter without sacrificing functionality.

II. System Components

This section outlines the main components of the Mokka coffee machine. A brief description explains each component's role in the system along with other details and specifications. The main components are microcontroller, RFID, stepper motor and drivers, pumps, and the PID Controller.

A. Microcontroller

The main controller for the project is the ATSAMW25 low power Wi-Fi module. The ATSAMW25 contains three main blocks: SAMD21 ARM® Cortex®-M0+, WINC1500 low power 2.4GHz IEEE® 802.11 b/g/n Wi-Fi, and ECC508 CryptoAuthentication. The SAMD21 MCU provides processing speeds of up to 48 MHz, which is necessary for it to act as the brain of the system. The WINC1500 Wi-Fi module allows for speeds up to 72 Mbps to allow fast transfer of user data from the server. The module will need to be loaded with an Arduino bootloader to enable the use of the libraries available to Arduino devices.

B. Radio Frequency Identification

The RFID reader that our system will be utilizing is the Grove NFC 125kHz. The reader is used to retrieve user identities from the 125kHz tags that will be attached to the users' cups. Using this method of user identification reduces the amount of user interaction at the time of purchase to make the process more convenient for the user.

C. Stepper Motor and Drivers

Multiple "A4988 Stepper Motor Driver Carrier" and "12V 0.4A 36oz-in Unipolar Stepper Motor" have been incorporated into the system's design for the coffee ground dispenser. The two components in unison allow for a precise amount of coffee grounds to be dispensed with only a precision error of ±5% when the motor is in full step mode. More details on the use of these components as well as the coffee dispenser subsystem can be found in section IV Hardware Details.

D. Pump

Two different kinds of pumps will be used in our coffee machine to deliver a precise amount of water and cream to the coffee. The "Bayite 12V Fresh Water Diaphragm Pump" and the "Gikfun 12V DC Dosing Peristaltic Pump" were the pumps chosen based on the parameters needed for our system. These pumps will be discussed in further detail in the Hardware Overview section of this paper.

E. PID Controller

A PID controller adjusts for load changes by comparing the process variable against a reference point. This output informs the system how much adjustment is required to eliminate the difference between them. A PID controller consists of proportional, integral, and derivative parameters that can be adjusted to achieve a desired output. The proportional element of the controller simply changes the output in proportion to the error. If configured correctly, the proportional part of the controller will result in a stable output with a small offset compared to the set value. The integrator integrates voltage as a function of time. This allows the controller to reduce the offset from the proportional portion of the controller in integral time until the output matches the set point. The derivative element of the controller senses the change in error over a short time period and adds a portion of the error to the output. This is also called the disturbance action because it is the part of the controller that moves the output back to the set point in the case of an unexpected disturbance. The equation for a PID controller is shown below in the time and Laplace domains.

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt}$$

$$u(s) = \left(K_p + \frac{K_i}{s} + K_d s\right) \times e(s)$$

The gain terms K_P, K_I and K_D are provided by the op-amp configuration transfer functions as $R_p2/Rp1$, $1/R_iC_i$, and R_dC_d. e(s) is the Laplace transform of the error signal e(t) and u(s) is the control signal u(t) in the Laplace domain. This means that we can adjust the P, I, and D terms by changing each term's resistor values.

III. Design Overview

The following section contains a general overview of the system to give you a better idea of how the system is supposed to work before going into the details of each of the different subsystems.

A. Hardware Overview

Below in Figure 1 is a generalized overview of the system. This is to help give a better understanding of how all the different components in the system are connected. Most of the components in the system will be controlled by a single point, the microcontroller PCB, to ensure the linear execution pattern is followed to ensure a proper cup of coffee is made by the system.

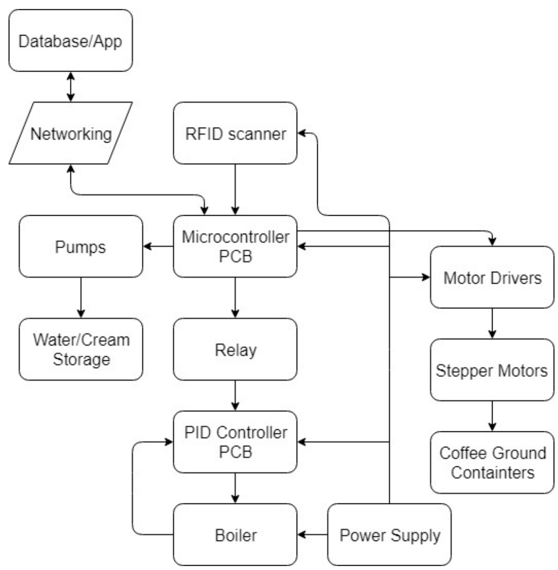

Figure 1 – Hardware Block Diagram

B. Software Overview

Seen in Figure 2 is a high-level flowchart of the software functionality of the system. The system will start off by waiting for the RFID reader to receive a valid user ID. Once it does this it will then retrieve the user's data and attempt to charge them. If all of this is successful then the system will then start to signal all of the subsystem to start brewing the cup of coffee. If there is a failure during any step of the process before payment confirmation the system will reset back to its original state of waiting to read an RFID tag.

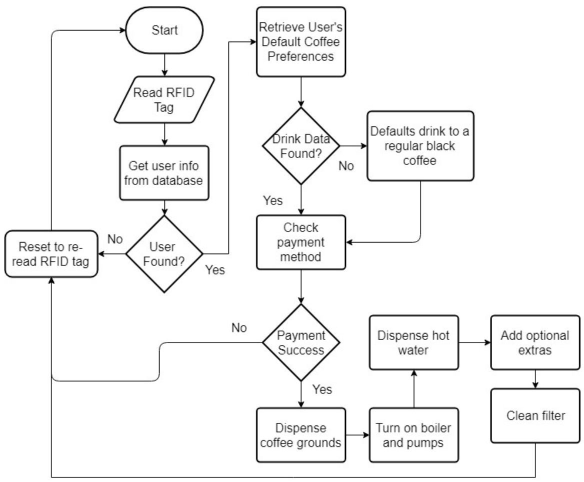

Figure 2 – High Level Software Flowchart

IV. Hardware Detail

A. Coffee Dispensers

This subsystem of Mokka's design is responsible for dispensing the coffee grounds that will be used to brew the user's cup of coffee. There will be one of these designs for each available type of coffee that Mokka will be available to brew. For our prototype design we will only be dealing with two different flavors of coffee, but the amount could be expanded based on the results of our initial prototype. There are three main components to this dispenser's design: Zevro KCH-06079 Indispensable 1/2-Pound Capacity Coffee Dispenser, and the aforementioned stepper motor and driver.

The repurposed Zevro coffee ground dispensers' base is a plastic container with a half-pound capacity. It has an airtight lid that will keep the coffee grounds fresh as well as make refilling the machine an easy process. The Zevro dispenser was originally designed to be a mechanical coffee dispenser that measures out one tablespoon of coffee grounds per each squeeze of the handle. After disassembling the device, it was discovered that with each pull of the trigger the device would turn a measurement wheel that holds the coffee grounds to be dispensed. The wheel can hold up to 8 tablespoons of grounds at a time. With our design we automated this process by attaching the stepper motor to the dispenser wheel.

The system's stepper motor is a professional high precision 12V stepper motor. The motor has a resolution of 1.8 degrees/step and a torque of 36 oz./inch and a step angle accuracy of ±5% on a full step when there is no load attached to it. The precision of the stepper motor is the main reason a stepper motor was chosen over a normal servo. These precise steps will allow the dispenser to release the same amount of grounds for each customer. A normal DC servo when used for continuous motion will eventually lose its timing and with this system that would lead to an inaccurate amount of grounds being dispensed. The motor's size and design, seen in Figure 3, allowed it to be attached to the wheel within our ground container. With a resolution of 1.8 degrees/step the motor will need to move 25 steps to dispense one tablespoon of coffee grounds.

.

The stepper motors will then be attached to a stepper motor driver which will control and power the motors. This system design will be using a A4988 stepper motor driver carrier. This motor driver operates from 8V to 35V and features current limiting, over-temperature protection, and can allow micro stepping down to 1/16 of a step, but this system will only be utilizing the driver's full step mode. This specific controller is listed as a bipolar motor controller, but it has been deemed capable to operate the system's unipolar stepper motors with the proper connections established. These stepper motors and drivers will then be connected to the microcontroller through the custom designed ports in the system's main printed circuit board. Below in Figure 4 is the wiring diagram for the system's motor drivers and connector pin for the stepper motors to be plugged into.

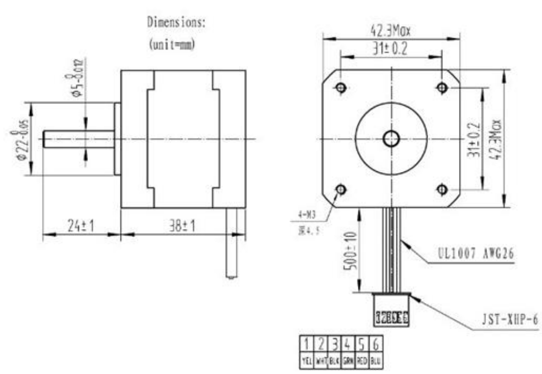

Figure 3 – Stepper Motor Schematic

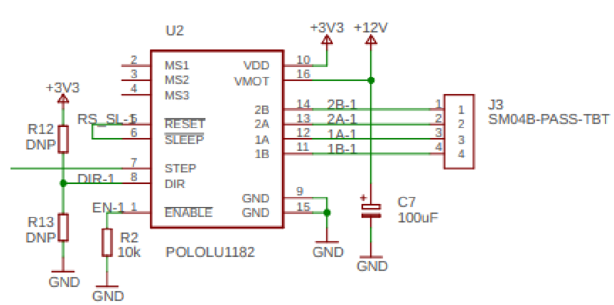

Figure 4 – Motor Wiring Diagram

B. RFID Module

There are six main frequency range that the RFID system transmits data over. Our system will implement an RFID system that operates on the 120-150 kHz frequency range because this is a short-range frequency band that is unregulated and usually used in smaller applications such as authentication card scanning. This is unlike the 13.56 MHz band which falls under ISM band regulations. ISM bands are usually reserved for industrial, scientific and medical purposes.

C. PID Controller

The design for the PID controller is shown in Figure 5. The P, I and D terms are configured to be 1, 1.034e-3, and 967.1 accordingly. The schematic consists of a difference amplifier which takes the difference of a reference voltage and the temperature sensor's output. This is followed by proportional, integral, and differential op-amps that represent the P,I, D terms. A summing amplifier adds all three terms.

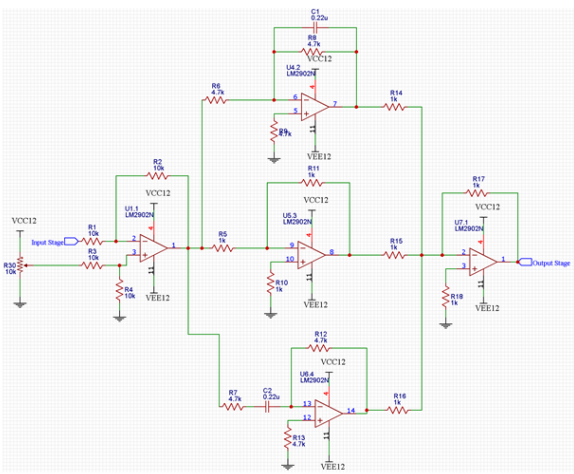

Figure 5- PID controller Schematic

Input and output stages were added to control the temperature of the boiler. The input stage is used to measure the temperature and the output stage is responsible for converting the analog PID output into a duty cycle. The first part of the input stage shown in Figure 6 consists of an AD595 instrumentation amplifier. It produces a 10mV/°C output from a K-type thermocouple signal. In this case, the K-type controller used is the Type-K Glass Braid Insulated Stainless Steel Tip because it can withstand higher temperatures than thermocouples covered with vinyl and because it is breadboard-compatible. This output is amplified by a non-inverting before reaching the difference amplifier.

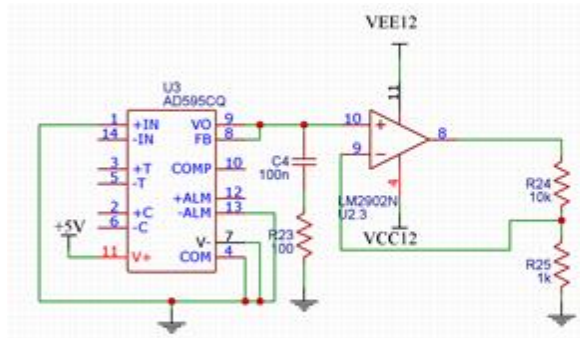

Figure 6- Input Stage: AD595 and non-inverting op-amp

This PID controller cannot control the 1300 watt boiler due to the limitations of the components used. Instead, the controller is used to control a 40 A, 250 V solid state relay. The continuous PID output can be turned into a duty cycle if it is inputted into a comparator along with a triangle wave. This is explained by Figure 7.

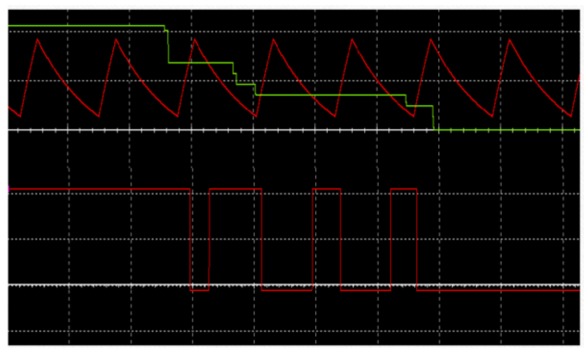

Figure 7- Conversion of Continuous PID output into duty cycle

Finally, the schematic used to generate a triangle wave is seen in Figure 8.

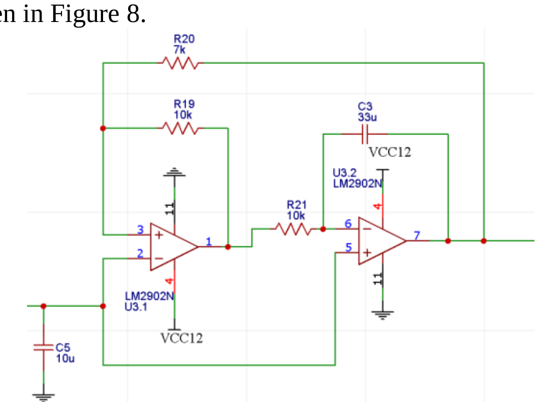

Figure 8- Output stage: Triangle waveform generator

D. Boiler

The boiler used is a 120 V, 1300-watt thermoblock. A thermoblock is the ideal boiler for this application because it is more compact than a single boiler. It also heats water much faster since it only needs to heat the small amounts coming in. It is also ensuring that all the water being sent by the pump ends up in the cup of coffee. The main disadvantage it presents is the fact that it complicates the control of the heating. In a single boiler, the water can simply be heated until the desired temperature is reached. A thermoblock needs to regulate the temperature as the water is flowing in from the pump.

E. Power Module

Choosing a power supply which does not provide the correct voltage and amperage to your components can cause them to behave erratically and has the potential to be dangerous. Therefore, choosing the ideal power supply to power the different subsystems in our coffee machine was a very important task. The components in our system are rated at either 12V or 5V. Therefore, we decided to choose a 12V with enough amperage to power the peripherals. To power the components which require 5 volts we will manually step down the voltage using a DC-to-DC voltage converter. This will let us power any component or peripherals in our system with the correct voltage. The table below summarizes the specifications for the power supply we have chosen for our coffee machine.

Table 1 – Power Supply Specifications

Output Power	360W
Output Voltage	12 V
Output Current	30 A
Input Voltage Range	90 - 264 VAC
Frequency Range	47-63 Hz
Operating Temperature	-10 to 60°C
Size	7.8 x 4 x 2”

For our 12V to 5V converter we decided to go with the “BINZET DC Converter Step Down Regulator”. It is cheap and easy to implement into our design making it an ideal choice. The converter outputs 10A at 5V, and will be used to supply power to the two PCB’s in our system.

We have also included a 20A circuit breaker in our design. In the event of an overcurrent failure, such as a power surge, the breaker will trip and will protect the machine and its components from damage.

F. Enclosure

The enclosure for our coffee machine will be made out of plywood. Its dimensions will be 2.5”x2”x1”. Plywood is inexpensive and easy to work with, making it ideal for our system. The main components of our machine will be mounted inside on the plywood. This includes the boiler, pumps, coffee dispensers, and coffee filters. In addition, to account for the high temperatures the boiler will reach, it will be mounted on two thin cement sheets. This will keep the plywood safe from burning even with the boiler mounted in it. The components have been strategically placed in order to efficiently perform their duties. The PCB’s and power supply will be stored inside electrical boxes.

G. Pumps

In order to deliver a precise amount of water to brew the coffee, we decided to implement peristaltic pumps in our design. These pumps will be controlled by the microcontroller PCB and powered by the power supply. The PCB will send a signal to the pump and it will pull water from the reservoir and dispense it in the filter. Due to concerns regarding how fast the pump is able to dispense water, we have chosen to implement a powerful diaphragm pump with a large flow rate. The flow rate can be controlled by adjusting the voltage sent to the pump, therefore even though the pump is rated at 12V, we will power it using 5V. This will be a good compromise between speed and reliability. As speed is not an important factor, the slower peristaltic pump will be used to dispense creamer in the customer’s coffee if they so choose. This peristaltic pump will also be controlled by the microcontroller PCB and powered by the power supply. We have used food grade silicone tubing in both pumps to ensure neither the water nor the coffee gets contaminated.

V. SOFTWARE DETAIL

A. *Mobile Application*

During development, web pack is watching all of our JavaScript and SCSS and compiling them into a native mobile application that can be installed on either Android or iOS.

The frontend of our application is based around the React Redux Flow model. In this model, React Native is in charge of rendering views to the user and Redux is in charge of handling the data for the mobile application. Figure 9 below shows the React Redux Flow model.

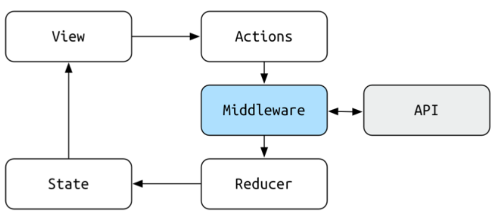

Figure 9 – React Redux Flow Model

When the application first loads, React Native first renders the App component to the main view encapsulated by a Redux Provider module. This provider module contains the current global state of the app which is handled by Redux. With the provider in place, the state can be accessed by any component via the mapStateToProps function.

Figure 10 shows the process of creating their custom coffee preset. First, they would start off at the home page for the application. Here the user has control over their coffee presets. They can choose which coffee preset will be the default preset to be used by the coffee machine when they scan one of their cups. They can also choose to delete a coffee preset by clicking on the X icon on the top right corner of the panel. To add a new coffee preset, the user can tap on the “add” panel which will always be the last panel with the plus icon. The user will be prompted to select their coffee flavor. This will be based on the grind flavors that are available in the Mokka coffee machines near the user’s location. Once the user has selected their desired flavor, they will then choose how much cream/milk they want in their coffee. Then the last step would be for the user to pick the size of their drink.

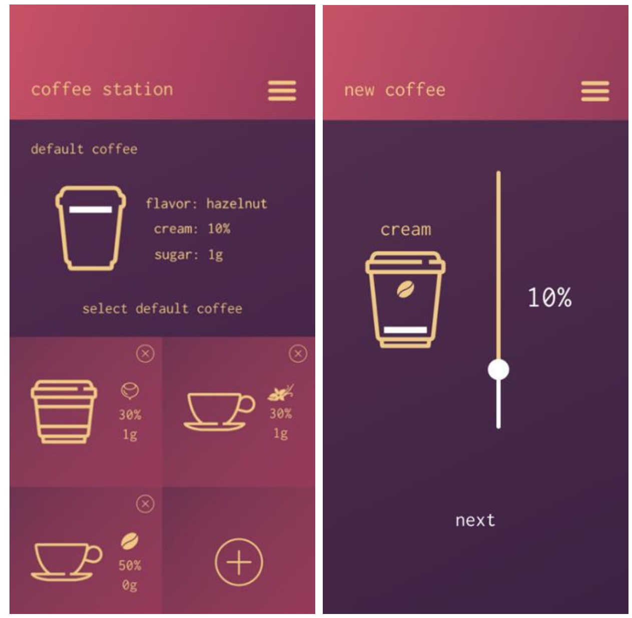

Figure 10 – Coffee Station Page

B. *RESTful API*

Our system architecture matches closely with the Client-Server model. The server acts as the central database, where multiple users are allowed to login and access the system, as well as take requests from clients to send and receive data. The server handles the storage of all coffee settings as well as information on all the users. Users are able to access all user functions (coffee settings, payment information, etc.) and to update or modify the data. Because of the needs of our system, and our current description, the Client-Server model represents our system the best.

The backend will be built using the Node.js. We designed an Restful API that will be accessible by our coffee machines via online requests. To handle the creation of the API, we used the Node.js library Express. Express allows us to define GET, POST, PATCH, and DELETE routes for our API that will allow the user to modify the data in the database from the mobile application. Creating an API will also allow other developers to interface with our data. The following tables detail the different routes that our API provides.

Table 2 – Express Library

Request Type	Route	Description
GET	/users	Returns a list of all users and all of their coffee information (useful for developers who want to take advantage of our API to gather statistical or analytical data).
POST	/users	Creates a user with the given data. The user data provided must adhere to the model detailed in the data models section. This route will return the new user as a response or an error if the data provided is invalid.
PATCH	/users/:id	Modifies the user with the given ID. This route will be used by the mobile application to modify user payment information, coffee preferences and other information such as email or phone number. This route will return the modified user as a response or an error if the data provided is invalid.
DELETE	/users/:id	Deletes a user with the given ID. This route will return the deleted user as a response or an error if the user is not found.

C. Server Environment

We are hosting our backend server remotely on a VPS (Virtual Private Server) on DigitalOcean. The backend server is running Ubuntu 16.04 as its main operating system. By having a public IP address for our server, we are able to make requests to it from anywhere in the world.

D. Database

The database handles the storage of all models being used by our system. The models include the User Model and the Coffee Machine Model. The database is hosted on a server that can be used to transmit and receive messages to the network.

To store the data, we chose to use MongoDB. MongoDB is a NoSQL database. MongoDB utilizes a variety of data models including document, graph, key-value, and columnar and also uses documents with JSON formatted data to store information.

The mobile application sends information to the backend Node.js server and the server utilizes a library called Mongoose to interface with the MongoDB database. Mongoose provides us with several functions that allows us to pull data from the database and send it to the user when needed.

The following table details the required fields in the database for the user model. The field column contains the key that MongoDB will use to reference the data, the data type field contains the data type of the given field, the required column states whether or not the field is required for each instance of the user or machine. Table 3 shows the database model diagram.

Table 3 – Database Model

Field	Data Type	Required
_id	ObjectId	Y
name	String	Y
email	String	Y
password	String	Y
payment_info	Array[Object]	
payment_info.name	String	Y
payment_info.address	String	Y
payment_info.cardNumber	String	Y
payment_info.expiration	Date	Y
payment_info.cvc	String	Y
cups	Array[Object]	
cup.uid	String	Y
cup.size	Number	Y
coffee_prefs	Array[Object]	
coffee_pref.type	Number	Y
coffee_pref.cream	Number	Y
coffee_pref.sugar	Number	Y
tokens	Array[String]	Y

VI. BOARD DESIGNS

The following section will give a brief overview of the printed circuit board designs used to control the system.

A. Microcontroller PCB

Figure 11 below is the system's microcontroller PCB. It is designed to handle 3 stepper motors for dispensing different types of coffee grinds and 3 pumps for moving fluid throughout the machine. We power the PCB with a single 12V line which is inputted at the right. The voltage is then stepped down to 5V and 3.3V via a buck regulator. The maximum trace width is 0.8mm and the minimum trace width is 0.3mm.

The brains of the system is the ATSAMW25H18-MR510PB. This is the micro processing chip that allows us to drive GPIO pins high or low and connect to the internet to make requests to the backend server. We have also included test points connected to all the pins of the microprocessor in case any modifications would want to be made to the board.

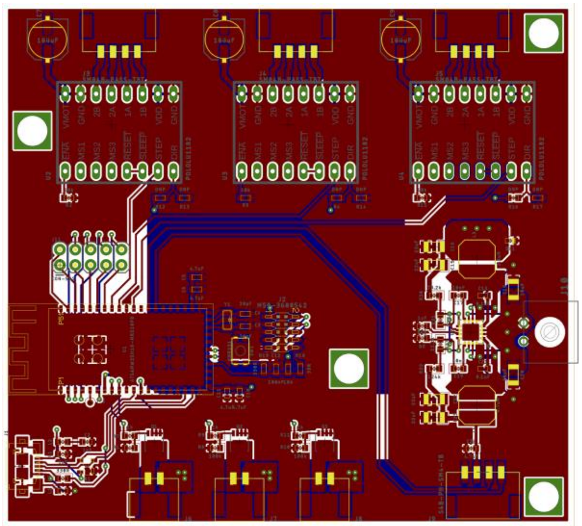

Figure 11 – Microcontroller PCB

B. PID Controller

The PID controller PCB uses through-hole components to facilitate testing. It uses an 8-pin terminal block for all the different supply voltages, ground, and the thermoblock wires. Three quad op-amps make the center of the board for easier access to all the components. This op-amp is the LM2902 and it was chosen due its low price of $0.45 and wide power supply range. It also met our preference of being a quad op-amp and had a maximum operating temperature of 125°C. Another reason it was chosen is that it is compatible with both, single and dual supply voltage ranges. Another op-amp considered was the NE5532 but it did not have the same benefits of the quad LM2902. Most low noise op-amps considered had a much higher price and had a very limited supply voltage range. The AD595 amplifier is placed on top of the board to allow for connection directly to the thermocouple. The only potentiometer on the board controls the setpoint of the temperature through the use of a voltage divider.

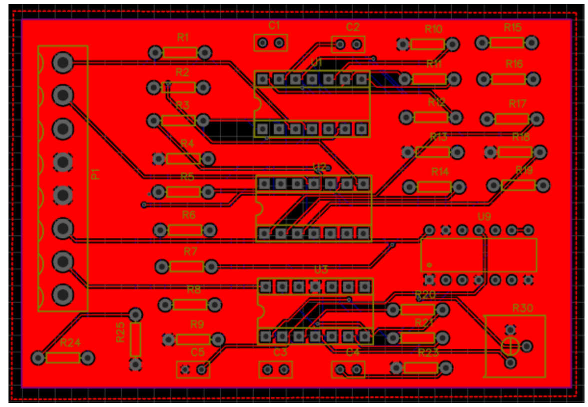

Figure 12 – PID PCB

VII. Conclusion

This report goes over the design required for the development of the Mokka coffee machine. Everything is explained thoroughly starting from a system overview all the way down to the schematic and software explanations. All of the design details that went into this project were researched extensively to ensure that all aspects of the machine function as expected. Although everything does work, there is still a lot of room for improvement in many areas of the machine. The Mokka coffee machine would be a great product ideal for small companies and coffee shops. Simply by using an app, users could be able to select their preferred coffee and pay instantly without having to take out their wallets. Alternatives to the Mokka coffee machine are premium high-quality coffee machines. However, the Mokka coffee machine would offer an automated system for selling coffee at a reduced cost. This product would create a new kind of product for commercial coffee machines.

References

[1] Cetinkunt, S. (2007). Mechatronics. 2nd ed. Hoboken, N.J.: John Wiley & Sons, pp.770-774.

[2] 12V 0.4A 36oz-in Unipolar Stepper Motor. (2018). Retrieved from https://www.robotshop.com/en/rbsoy03-soyo-unipolar-stepper-motor.html