Hands Free Drinks Mixer

ECE 445 Design Review

Spring 2019

Team 40: Dave Ha, Eric Mysliwiec, Matthew Gross

TA: Channing Philbrick
Hands Free Drinks Mixer

1 Introduction

1.1 Objective

1.2 Background

1.3 High-Level Requirements

2 Design

2.1 Block Diagram

2.2 Physical Design

2.3 Power Circuitry

2.3.1 120VAC to 12VDC Converter

2.3.2 12VDC to 5VDC transformer

2.4 Motors

2.4.1 Pumps

2.4.2 Disk Stepper Motor

2.5 Sensors

2.5.1 Flow Meters

2.5.2 Photocell and Laser

2.5.3 Weight Sensors

2.6 Logic

2.6.1 ESP8266 Microcontroller

2.6.2 Motor Drivers

2.6.3 Refill Indication Control

2.7 User Interface

2.7.1 RFID

2.7.2 Push Buttons

2.7.3 LCD

2.7.4 Refill LED

2.8 Server

2.9 Supporting Material

2.9.1 Refill Indication Control Circuit and Simulation

2.9.2 Motor Drivers

2.9.3 Stepper Motor Driver

2.9.4 Software Flow Chart

2.9.5 Photocell/Laser System Test

2.9.6 Stepper Motor Calibration Calculations

2.9.7 Throughput Analysis of ESP8266

2.10 Tolerance Analysis

3 Cost and Schedule

3.1 Cost Analysis

3.2 Schedule

4 Safety and Ethics

5 References
1 Introduction

1.1 Objective

Waiting in lines is something that most people cannot avoid on a day to day basis. In fact, people wait in lines so often that Richard Larson, an MIT professor, estimates “some people spend a year or two of their lives waiting in line.” [1] A great way to minimize this waste of time is to decrease the wait time to get a drink at the bar. For bartenders, trying to serve every customer and finding the right tab behind the bars can be challenging. Adding more staff to speed up the process is neither space nor cost effective, especially since some bars have very limited space behind the counter. Smaller lines and faster processing at bars is needed in order to increase the efficiency of sales, as well as reduce the amount of stress for both the bartenders and the customers.

Building an automated drink mixer will ensure that there are fewer tasks for bartenders, so the waiting time could be cut significantly. This machine will assist bartenders by serving customers who want preselected drink specials for the day. Having this system in place will cause a significant increase in the amount of drinks served in a given amount of time, by reducing the amount of tasks for bartenders.

1.2 Background

Places such as Pour Bros have incorporated RFID cards to track customers’ drinks and allowing the customers to pour their drinks themselves. [2] However, not all bars can transition to this style and atmosphere. Furthermore, Pour bros incorporates a large, clunky station as opposed to a personalized pouring experience. This results in almost no lines and requires much less workers. The proposed solution will differ in that it is smaller in size and is personalized, whilst adding the ability to transmit total sales data via Wi-Fi upon shutdown. This will assist bartenders when lines get long instead of functioning as a self serve device. The increased rate in drinks being made will keep customers happy and streamline sales during extreme rush hours.

Additionally, through having an RFID scanner for cards, the system will remove the need for the bartenders to look for the right tab among many cards. To make the drinks, all that the bartender would need to do is scoop some ice with the cup, and put it on a slot in the machine. The only task that the customer needs to do is to place his RFID tag on the sensor and select the drinks through using the buttons; Ultimately, the responsibility falls upon the customer.
1.3 High-Level Requirements

The success of the project will be determined by the following quantified measurements:

- Has to be able to fill a standard 16 oz solo cup with 6 oz (236 ml) of liquids in 2 minutes using at least 2 different liquids.

- Has to be able to scan at least 3 unique RFID cards and keep a log of tabs, each tab consisting of 20 or less drinks, until shutdown of the system.

- Has to be able to send the log of tabs in the form of a small text file that is less than 10 kB in size to an email through the Wi-Fi in less than a minute, with full connection to an internet network.

2 Design

The block diagram in Figure 1 contains five main modules: the power circuitry module, motors module, sensors module, logic module and user interface module. The power circuitry ensures that the standard 120 volt wall outlet gets properly converted to usable voltages – 5 V DC for the microcontroller and sensors and 12 VDC for the custom PCB and the motors module. The motors module controls the core of the system. This includes rotating the disk with the stepper motor and pumping/dispensing the liquids. The sensors module is used for the fine tuning of the system. Things like measuring the flow of liquids, calibrating the position of the cup and warning when an ingredient is low can all be done within the sensors module. The logic module is the brains of the system. This module uses data in two different ways. Firstly, the microcontroller reads various inputs and displays the menu on the LCD for proper drink selection. Secondly, the refill indication control uses driving logic from the weight sensors beneath each liquid tank to identify empty bottles. Finally, the user interface module allows the user to interact with the system. Any request for a drink or information being delivered to the user is done here. Independent from the system, a server or a network-connected computer is necessary to receive tab log file from the machine upon system shutdown sequence.
2.1 Block Diagram

Figure 1. Block Diagram

Data Line (3.3 VDC unless otherwise stated)
Wireless Connection
Power Connection (DC unless otherwise stated)
2.2 Physical Design

The blue parts in Figure 2 will be cut from wood and fastened with screws and glue. The yellow disks will be 3D printed with the black cutout (see Figure 2) used to hold the cup while the red cutout (see Figure-2) is used for calibration with a photocell sensor. The top disk will be attached to the wooden structure with screws. The bottom disk will be attached to the stepper motor which will be sitting in a cutout in the wood (not shown). In the case that the disk struggles to support the weight of the cup, two supports can be easily added.

2.3 Power Circuitry

Distributing power to components is essential to the success of this project. Not enough power will cause the components to stop running, while too much power could potentially destroy components.
2.3.1 120VAC to 12VDC Converter

Most components in the project will be unable to handle the 120V supplied by the wall outlet. This converter will lower the voltage to a much more usable 12V.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Verification</th>
</tr>
</thead>
</table>
| 1) Must supply 12VDC +/- 20% and 2A +/- .25A from a 120VAC source. | 1) To verify:  
a) Use a multimeter to measure the voltage supplied by the converter.  
b) Then use an oscilloscope to verify stability. |

2.3.2 12VDC to 5VDC transformer

There are some components in the project that can only withstand 5V, and some logic circuitry will begin failing outside of the 4.5 to 6.5 Volt range. A switching voltage regulator will be used to step down the voltage instead of a linear regulator in order to avoid needing a heatsink and fan. [3]

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Verification</th>
</tr>
</thead>
</table>
| 1) Must supply 5V +/- 10% with at least 0.5 Amperes[4] | 1) To verify:  
a) Use a multimeter to measure voltage across transformer  
b) Use an oscilloscope to verify stability.  
c) Then use a 10 Ohm resistor and multimeter to verify amperage output. |

2.4 Motors

The motors carry out the mechanical tasks of the machine such as moving liquid from the tanks to the cup. Furthermore, it moves the cup such that it accurately rests below dispensers for the duration of the pour.
2.4.1 Pumps

The pumps will have different dispensing rates depending on the type of liquid being dispensed. Per expectations, there will need to be a larger rate of dispensing for non-alcoholic drinks than alcoholic.

### Requirement

1) **Pumps must provide at least 250 ml of fluid per minute.**

### Verification

1) To verify:
   a) Turn on the pump until water is being dispensed.
   b) Then turn the pump off, wait one minute and measure how much fluid was dispensed.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Pumps must provide at least 250 ml of fluid per minute.</td>
<td>1) To verify: a) Turn on the pump until water is being dispensed. b) Then turn the pump off, wait one minute and measure how much fluid was dispensed.</td>
</tr>
</tbody>
</table>

2.4.2 Disk Stepper Motor

The stepper motor will allow us to position the cup under whatever nozzle is about to be dispensing. It will have a one way communication to the microcontroller and can be re-centered when necessary. [5] The stepper motor will give 28 N of force at 20 mm which translates to 5.6 N of force at the center of the cup (100 mm). [6]

### Requirement

1) **Stepper motor must be able to consistently align a given nozzle within 1 inch from the center of the solo cup. This translates to being within around +/- 11.46 degrees of the target.**

### Verification

1) With the disk connected to the stepper motor and the stepper motor connected to a cutoff switch:
   a) Engage cutoff switch.
   b) Mark how far the disk continues to rotate after cutoff switch has been pressed.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Stepper motor must be able to consistently align a given nozzle within 1 inch from the center of the solo cup. This translates to being within around +/- 11.46 degrees of the target.</td>
<td>1) With the disk connected to the stepper motor and the stepper motor connected to a cutoff switch: a) Engage cutoff switch. b) Mark how far the disk continues to rotate after cutoff switch has been pressed.</td>
</tr>
</tbody>
</table>

2.5 Sensors

This project’s sensors will be purposed to re-calibrate the position of the cup, measure the amount of fluid dispensed, and detect when the weight of an ingredient is below an arbitrary threshold.

2.5.1 Flow Meters

The flow meters will increase accuracy in dispensing of the liquids. Attached to the tubing, it will collect the flow data real-time and send it to the microcontroller to logically calculate when to stop the dispensing. The standard liquid volume of a full red solo cup is 12 fl.oz (355ml) [7] so the dispensed volume must be half of that to account for ice volume.
### Requirement Verification

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Verification</th>
</tr>
</thead>
</table>
| 1) Must be able to calculate a total dispensed volume of 178ml +/- 10 ml. | 1) To verify complete the following 3 times:  
   a) Run a premeasured amount of fluid through the meter.  
   b) Compare the meter value to the actual value. |

### 2.5.2 Photocell and Laser

The photocell and laser will allow us to calibrate the stepper motor positioning of the cup using the microcontroller. This is necessary to do at startup as well as during use in order to avoid error in the stepper motor positioning.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Verification</th>
</tr>
</thead>
</table>
| 1) Photocell must be able to detect the laser going through a 1/8” radius hole upon light contact in under a second to limit the disk from rotating too far past the calibration point. | 1) Write a simple program that does the following:  
   a) Read the photocell data.  
   b) Print this data along with time stamps.  
   c) Shine a laser at the photocell and see how quickly it detects this change. |
2.5.3 Weight Sensors

The weight sensors will be used to detect when an ingredient is getting low. It will then notify the user by lighting up an LED. In order to know the proper weight at which the system notifies the user, custom bottles of uniform weight (plastic 2L bottle) will be used, which weigh 45 g empty and 2042 g full. [8] [9]

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Verification</th>
</tr>
</thead>
</table>
| 1) Must be able to differentiate 270 g from 270 g + 75 g. | 1) To verify:  
   a) Measure 270 g on a precise scale and compare value to the sensor’s value.  
   b) Then measure another 75 g and add it to the sensor. |

2.6 Logic

The logic unit serves as a central processing unit for all of the data collected through the sensor, giving orders to specific modules or devices as needed. It will get the RFID input and store it in a log of tabs, as well as sending the data to the server (or an email). It will control the selection of drinks through push buttons, as well as displaying different drinks to the LED. Furthermore, it will make sure that the cups rotate the right amount of steps to correctly position beneath the required nozzle for the drink recipe.

2.6.1 ESP8266 Microcontroller

The microcontroller, powered by 3.3VDC, will run the written program to interface all data and make decisions accordingly. The program can be divided into seven parts, as seen in the software logic diagram (Figure 8) in Section 2.9.3:

1. Establishing connections with all sensors and modules, and initializing all signals and data structures
2. Wait for customer interaction through an RFID sensor, and show selection menu on the LCD screen when the user scans his/her id/sensor
3. Interact with user choices through button signals: navigating through the menu, and selecting/cancelling order
4. Updating the data structure with proper key-pair value of user ID and number of drinks bought
5. Sending rotation signals to the step motor, and liquid dispensing signal to the peristaltic pump to make the mix according to the recipe stored in the program.
6. Reset and align the disk to its zero position, indicating that the drink is done. Return to step 2
7. Upon holding the cancel button for 3 seconds, initiate the shutdown sequence; send the log file to an email(server), and power off the motor. Indicate on LCD that the machine is good to be unplugged.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Must be able to send and receive data to/from RFID sensor through SPI connection.</td>
<td>1) After initializing the RFID module, read the value through mfrc522.PICC_ReadCardSerial() and print values inside mfrc522.uid.uidByte buffer onto terminal. Exact steps in section 2.7.1.</td>
</tr>
<tr>
<td>2) Must be able to send pulses of 50 ± 5 ms (to account for programming constructs), of 3.3V onto the stepper motor driver through digital I/O pin.</td>
<td>2) Connect the motor driver circuit to pin 3, and run the following sketch code: a) inside setup(), call pinmode(3, OUTPUT); b) inside loop(), call digitalWrite(3,HIGH); c) call digitalWrite(3,LOW); d) delay(50); to let the motor catch the pulse e) visually confirm that motor rotated half a step.</td>
</tr>
<tr>
<td>3) Must be able to send 3.3V signal to any selected flow meter &amp; peristaltic pump driver, and read the output pulse from the flow meter through the digital I/O pin.</td>
<td>3) Connect the flow meter driver input to pin 3, flowmeter output to pin 2 to be read, and peristaltic pump driver input to pin 4 of ESP8266. Pump pipe must be fully immersed in any liquid. Then run the following sketch code: a) inside setup(), call pinmode(3, OUTPUT); b) call pinmode(2, INPUT); to control the flowmeter. c) call pinmode(4, OUTPUT); to control the pump. d) inside loop(), call digitalWrite(3,HIGH); and digitalWrite(4,HIGH); to power the pump and flowmeter e) serial.print(digitalRead(2)); to see the flowmeter value. f) visually confirm that motor, and flowmeter are operational.</td>
</tr>
</tbody>
</table>
4) **Must be able to set the pinhole laser high through digital I/O, thereby sending 3.3V, and read the photocell value between zero and non-zero values between on/off**

4) **Connect the laser to pin 3 and photocell to pin 4, and run the following:**
   a) inside setup(), call pinmode(3,OUTPUT);
   b) pinmode(4,INPUT);
   c) inside loop(), call serial.print(digitalRead(4)) and verify the read value is 0.
   d) call digitalWrite(3,High); and verify the laser shooting.
   e) serial.print(digitalRead(4)) and verify that the read value is non-zero.
   f) call digitalWrite(3,Low);
   g) visually confirm that the laser turned off.

5) **Must be able to set the LED through digital I/O pin to indicate successful WiFi connection.**

5) Connect LED to pin 3, and run the following sketch:
   a) `#include “ESP8266WiFi.h”` and set const char* ssid and const char * password to right values
   b) in setup(), call Wifi.begin(ssid,password);
   c) if(WiFi.status() == WL_CONNECTED)
   d) digitalWrite(3,High);
   e) Visually verify that the LED was lit.

6) **Must be able to send a file smaller than 1MB in size in 91 ms, on a 802.11 a/b/g/n network (802.11b network has speeds up to 11 Mbps, whereas others go up to 600 Mbps. Across different networks of a/b/g/n, the speed should average higher). Due to speed of additional programming constructs, the upper bound for acceptable speed will be 1s.**

6) Call the following sketch code:
   a) get time before sending email through time = millis()
   b) Send email through gsender->Send(to, message);
   c) get time afterwards, and subtract time to see if it meets requirements by running Serial.print(millis() - time).
   Thoroughly discussed in section 2.8
2.6.2 Motor Drivers

In order to prevent damaging the microcontroller, a separate system of transistors is needed to control the pumps and stepper motors. To use the least amount of current through the microcontroller, the control pin will drive a MOSFET which allows a 5VDC voltage to turn on a BJT. The BJT will then take the 12VDC from the power supply to drive the motors. The motor circuitry will also be protected using diodes to handle any sudden spikes in voltage from a blockage or mechanical failure.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Verification</th>
</tr>
</thead>
</table>
| 1) Must provide 12VDC +/- 1V to all motors. | 1) To verify:  
   a) Connect voltmeter in parallel with motor to measure the voltage across it  
   b) Apply power to the 12V supply for the motor driver  
   c) Apply a 3.3V Logic signal to the input of the logical control for the motor driver  
   d) Report the voltage from the voltmeter |
| 2) Must provide at least 0.8 +/- 0.05 A to drive the high flow pump motors. [10] | 2) To verify:  
   a) Connect ammeter in series with motor to measure the current through it  
   b) Apply power to the 12V supply for the motor driver  
   c) Apply a 3.3V Logic signal to the input of the logical control for the motor driver  
   d) Report the current from the ammeter |

2.6.3 Refill Indication Control

This will be a custom built PCB, utilizing a voltage divider, an op-amp and an n-channel mosfet. This drives an LED when the weight drops to near 10% of the maximum fluid in the tanks. [7][11][9]

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Verification</th>
</tr>
</thead>
</table>
| 1) Must drive an LED at 5V +/- 2V with 50 mA +/- 20 mA. | 1) Place weight matching 10% of the maximum weight of the tank on a powered weight sensor  
   2) Measure the resistance across the weight sensor |
3) Match the resistance of the weight sensor with one potentiometer in a voltage divider made up of two potentiometers.
4) Adjust the other potentiometer until 50mA is found to be passing through the LED.

2.7 User Interface

The user interface will provide a means for the user to interact with the system. Through the RFID sensor, the user will be able to let the system know who is buying the drink, and increment the value on the tab. Through the buttons, the user will be able to select the drinks, and cancel the order during the selection stage. The LED will display relevant information as the user progresses through the order. Furthermore, the refill LED will light up when the liquid tank runs low.

2.7.1 RFID

The RFID sensor will need to correctly read the unique value stored in a card or a RFID tag. Furthermore, it will need to communicate with the ESP8266 microcontroller and send the data over to be processed.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Must be able to read at least 3 different RFID tags through SPI, each within 0~3 seconds (clock speed on ESP8266 is 80 MHz, and on SPI, half. With slowest rates of 8bit/reads, a 32-byte string should take 32 * 40MHz = 0.8ms. Adding time for other programming constructs such as loops, should be less than 1s)[12].</td>
<td>1) Run the following: a) time = millis() b) read in the RFID card through mfrc522.PICC_ReadCardSerial() c) check time difference by serial.print(millis() - time); d) Do it three times for different cards, and see if the content in the buffer differs. Repeat, and see if the content matches that of earlier iteration.</td>
</tr>
</tbody>
</table>

2.7.2 Push Buttons

Push buttons, powered by the 5V DC transformer, will need to correctly send the push signals to the ESP8266 microcontroller upon being pressed and released. There will be four buttons: left, right, select and exit/shutdown. Button logic is explained in the Microcontroller submodule section 2.7.1.
2.7.3 LCD

The LCD, powered by 3.3 VDC, will display the value formed by the Logic module through getting the input from the microcontroller. Will display drink names during selection, and the ID upon scanning the RFID.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Must display the value set by the logic module (ESP8266) through I2C;</td>
<td>1) run the following:</td>
</tr>
<tr>
<td>since 2 rows of 16 characters are transferred at most, 32 bytes of</td>
<td>a) #include &lt;LiquidCrystal.h&gt;</td>
</tr>
<tr>
<td>data are transferred; these need to be transferred within 50ms(since</td>
<td>b) set button constants for different</td>
</tr>
<tr>
<td>clock speed of ESP8266 is 80MHz, and 32*80MHz = 0.4ms, but accounting</td>
<td>digital i/o inputs, as following:</td>
</tr>
<tr>
<td>for other programming constructs, upper bounded to 50ms) .</td>
<td>RS=D2, EN=D3, d4=D5, d5=D6,</td>
</tr>
<tr>
<td></td>
<td>d6=D7, d7=D8;</td>
</tr>
<tr>
<td></td>
<td>c) in setup(), set row, col by</td>
</tr>
<tr>
<td></td>
<td>lcd.begin(16,2)</td>
</tr>
<tr>
<td></td>
<td>d) time = millis();</td>
</tr>
<tr>
<td></td>
<td>e) print message by</td>
</tr>
<tr>
<td></td>
<td>lcd.print(time-millis());</td>
</tr>
<tr>
<td></td>
<td>f) visually confirm message on</td>
</tr>
<tr>
<td></td>
<td>lcd, and see time spent</td>
</tr>
</tbody>
</table>

2.7.4 Refill LED

The refill LED will light for the operator when a tank is below an arbitrary threshold and needs to be refilled.

2.8 Server

The server will receive the log of bar tabs upon the shutdown sequence from the microcontroller. Through the Wi-Fi module, the microcontroller will send a text file of entries, which consist of key-value pair from user ID to number of drinks purchased. Since having a server is outside the scope of this class and project, data will simply be received through an email transmission (SMTP).

2.9 Supporting Material

2.9.1 Refill Indication Control Circuit and Simulation

Figure 3 shows a mock design of the refill indication control can be achieved using a voltage divider, operational amplifier, n-channel mosfet and an LED. As seen in the simulation results (Figure 4), the LED would have been theoretically illuminated for values above 14 Ohm.
Figure 3. Refill Indication Control Schematic

Figure 4. Refill Indication Simulation
2.9.2 Motor Drivers

Figure 5 contains the simulation of a brushed motor driver. The motor is simulated with a resistor and inductor since LTSPICE does not have a motor component to test. Motors in real life will not be linear and will have built up inertia which will serve as a source when the Bipolar Junction Transistor moves from saturation to cut-off. This excess energy would destroy most BJT’s and thus the diode next to the simulated motor provides a path back so the current winds down with the motor’s natural friction instead of destroying other components. Figure 6 and Figure 7 show the results of this simulation as V2 is swept from 0V to 5VDC. The motor is rated for 12VDC and 0.8A which the system operates within as shown by this simulation.

Figure 5. Motor Driver Circuit Schematic
2.9.3 Stepper Motor Driver

Figure 8. Stepper Motor Logic Control
Figure 9. Stepper Motor Power Circuitry
2.9.4 Software Flow Chart

![Software Flow Chart Diagram]

2.9.5 Photocell/Laser System Test

A simple arduino program was written to print the data received at the analog to digital pin. A photocell from the 3.3V power supply was connected to the A to D pin and a resistor was connected from this pin to ground. Finally, a laser was shined onto the photocell, giving the output shown in Figure 9. The zeros represent no laser and the 1023 represents the laser being shined. This test not only proved this system would work but also showed that the system is not sensitive to nearby lighting and a pinhole is not needed for the photocell.
<table>
<thead>
<tr>
<th>Loop #</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>0</td>
<td>0</td>
<td>1023</td>
<td>1023</td>
<td>1023</td>
<td>1023</td>
<td>1023</td>
<td>1023</td>
<td>1023</td>
<td>1023</td>
<td>1023</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Photocell/Laser System Output
2.9.6 Stepper Motor Calibration Calculations

Figure 10 demonstrates how values of tolerance was calculated for the cup positioning and calibration.

![Diagram showing cup positioning calculation](image)

\[ Tolerance = 2 \times \arcsin(r/2/R) \quad (1) \]

\[ \theta = \sin^{-1}(\frac{r^2}{R}) \] where \( \Theta \) is the degrees of the circle of radius R needed to catch a circle of radius r.

The design most sensitive to tolerances in this project is the calibration and rotation of the bottom disk. If the stepper motor cannot align the cup under each nozzle the fluids will be spilled everywhere. Also if it is unable to calibrate accurately and consistently, the stepper motor will struggle to align the cup with a nozzle over time.

The numbers were chosen carefully by using measurements and trigonometry calculations. Various brands of red solo cups were measured, yielding that all brands have a 3.75 inch diameter or larger at the opening of the cup. Given this length, it was determined that, for safe functioning, the stepper motor must center the cup under each nozzle within one inch from the center of the cup. This will ensure that no fluids are being dispensed outside of the cup. Since the stepper motor has one inch of error allowed, there is, at the least, only .875 inch left until fluids miss the cup. This is why .5 inch was chosen as the max calibration error. Given the worst case scenario, the stepper motor misaligns the cup by 1 inch and the calibration is off by .5 inch. There still remains .375 inch of space until fluid will be dispensed outside of the cup.
Example: the cup has a diameter of 3.75 inches, which gives a radius of 1.875 inches. The cup will sit 3 inches out from the center of the turning disk; thus \( \theta = \sin^{-1}\left(\frac{1.875/2}{1.875}\right) = 18.21^\circ \). The dispenser should be 1” from the center of the cup so the radius shrinks from 1.875 to 1 = r, which gives 9.594°. The recalibration distance is arbitrarily set at 0.5 inches which gives a +/− 4.78° tolerance. These are both well within the 1.8° step size of the chosen stepper motor so tolerance will not be exceeded.

2.9.7 Throughput Analysis of ESP8266

The average data transfer rate from an ESP8266 to a local machine, with underutilized download speed for the machine is shown in Figure 11 representing time series of data.

![Time-Series Data Transfer Throughput from ESP8266, with Underutilized Download Capacity](image)

*Figure 12. Photocell/Laser System Output[13]*

There are some drops and peaks, and the speed is not stable throughout; however, the mean speed comes out to 341 KBps, which definitely satisfies the project needs to transfer a small text file.
2.10 Tolerance Analysis

By putting the stepper motor’s idealized resistance up to double its rated resistance and by increasing the inductance of the motor by 0.6 mH then the stepper motor is put into its worst case scenario where it will create the largest amount of charge time possible. This is worst case scenario is enhanced further by putting the power sources at their minimum values within working tolerance. For the 12V power source the maximum tolerance in this project is 1 Volt so the 12V source became 11 Volts, and the 5V source has a 0.5V tolerance so that was put at 4.5 Volts. Even with all of these changes, the stepper motor still has a charge and discharge time of about 1.5 milliseconds which means the stepper can perform a full step in just over 3 milliseconds. Rounding up to 5 milliseconds per full step then the stepper motor can make 200 steps in a single second which allows for this particular subsystem to have a negligible effect on the delivery time of the drink. This serves to fulfill the high level requirement of serving a drink within two minutes by only adding a maximum of 3 seconds to the delivery time which is maximized when using a mixture of all five possible drinks.

3 Cost and Schedule

3.1 Cost Analysis

When the project is complete, the overall cost of development will include both the time each project member put into working on the prototype as well as the price of all the parts used for the prototype. To
calculate the labor cost, it is assumed that each of the three members will be making around 40 dollars an hour, working 10 hours a week and will do this for 16 weeks. The total labor cost is calculated as shown in Equation 1.

\[
Total \ Labor \ Cost = 3 \ (members) \times \frac{40 \ dollars}{hr} \times \frac{10 \ hr}{wk} \times 16 \ (weeks) \times 2.5 = 48,000 \ (2)
\]

The parts list and costs for both the prototype and the potential bulk production price are shown in Figure 12.

<table>
<thead>
<tr>
<th>Part</th>
<th>Cost (prototype)</th>
<th>Cost (bulk ~100)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFID reader (Amazon)</td>
<td>$6.99</td>
<td>$5.59</td>
</tr>
<tr>
<td>Push Button x4 (DigiKey)</td>
<td>$.96</td>
<td>$0.80</td>
</tr>
<tr>
<td>Mini LCD (DigiKey)</td>
<td>$10.25</td>
<td>$8.10</td>
</tr>
<tr>
<td>LED x5</td>
<td>$0.75</td>
<td>$0.60</td>
</tr>
<tr>
<td>12V DC to 5V DC transformer (DigiKey)</td>
<td>$2.88</td>
<td>$2.56</td>
</tr>
<tr>
<td>120V AC to 12V DC Converter (Amazon)</td>
<td>$15.84</td>
<td>$12.68</td>
</tr>
<tr>
<td>High Flow Rate pumps x5 (Amazon)</td>
<td>$129.00</td>
<td>$60.75</td>
</tr>
<tr>
<td>Stepper motor (Amazon)</td>
<td>$13.99</td>
<td>$11.19</td>
</tr>
<tr>
<td>NodeMCU (Amazon)</td>
<td>$8.39</td>
<td>$2.00</td>
</tr>
<tr>
<td>MCP23017 Expander chip (DigiKey)</td>
<td>$1.24</td>
<td>$0.94</td>
</tr>
<tr>
<td>Shields (PCBWay)</td>
<td>$8.00</td>
<td>$0.60</td>
</tr>
<tr>
<td>Refill Control Circuit (PCBWay)</td>
<td>$4.00</td>
<td>$0.10</td>
</tr>
<tr>
<td>Flow meters x5 (Amazon)</td>
<td>$42.25</td>
<td>$33.80</td>
</tr>
<tr>
<td>Photocell (DigiKey)</td>
<td>$0.95</td>
<td>$0.90</td>
</tr>
<tr>
<td>Weight sensor x5 (DigiKey)</td>
<td>$60.72</td>
<td>$36.70</td>
</tr>
<tr>
<td>Laser diode (DigiKey)</td>
<td>$5.95</td>
<td>$4.76</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>$312.16</strong></td>
<td><strong>$182.07</strong></td>
</tr>
</tbody>
</table>

*Table 2. Cost Analysis Table*
With a labor cost of $48,000 and a prototype cost of $312.16, the total cost of the project will be about $48,312.

### 3.2 Schedule

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3/4</td>
<td>1. Finalize physical design with Machine Shop 2. Build Prototypes 3. Design, get approved and order PCBs</td>
<td>1. Test Prototype Circuits 2. Adjust schematics as necessary</td>
<td>1. prototype all logic on the microcontroller: a. connect to Wi-Fi network with proper credentials b. connect MCP23107 expansion chip c. connect and verify RFID, LCD, Buttons d. connect MUX and verify signals sent to stepper motor, flowmeters, peristaltic pumps, pinhole laser and Wi-Fi indicator LED e. connect and set thresholds for pinhole photocell analog-to-digital reading</td>
</tr>
<tr>
<td>3/11</td>
<td>1. Test and prototype all circuits</td>
<td>1. Test Motor Driver 2. Test Power Circuit</td>
<td>1. Test Wi-Fi file transmission through email</td>
</tr>
<tr>
<td>3/18</td>
<td>SPRING BREAK</td>
<td>SPRING BREAK Pump Motor Characterization</td>
<td>SPRING BREAK</td>
</tr>
</tbody>
</table>
1. Integrate Sensors with breakout logic
1. Integrate Power Circuit with motor driver, breakout logic
1. Integrate physical container and disc system with microcontroller
2. Integrate the PCB together with all the parts
3. Test make a drink

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Integrate Sensors with breakout logic</td>
<td>BUFFER TIME</td>
<td>BUFFER TIME</td>
<td>BUFFER TIME</td>
<td>Mock Demo’s</td>
</tr>
<tr>
<td>1. Integrate Power Circuit with motor driver, breakout logic</td>
<td>BUFFER TIME</td>
<td>BUFFER TIME</td>
<td>Mock Demo’s</td>
<td>Real Demo and Mock Presentation Prepare final report</td>
</tr>
<tr>
<td>1. Integrate physical container and disc system with microcontroller</td>
<td>BUFFER TIME</td>
<td>Buffer Time</td>
<td>Mock Demo’s</td>
<td>Real Demo and Mock Presentation Prepare final report</td>
</tr>
</tbody>
</table>

Table 3. Proposed Schedule

4 Safety and Ethics

There are two main issues that the project could potentially involve. The first is the potential to over-serve customers by having an automated drink system put into place. This is a conflict with IEEE Code of Ethics 7.8.1 which states that engineers must hold in high value the “health, and welfare of the public.”[14] To solve this, a drink limiter logic will be implemented, where each individual’s account will hold the number of drinks bought. After a predetermined limit is reached, the customer will need approval in order to place his/her order. The second issue is the potential to serve customers who are underage. Allowing this with the project, especially on a college campus, would be a clear conflict with the pledge to “to strive to comply with ethical design”[14] In order to prevent this, bars would have a system to register an RFID only to customers who provide proper identification. This project does not deal with batteries or any other potential safety issues, and uses no voltages above a normal household outlet. Given that the outlet is following IEEE standards, it should have its own circuit protection. In addition to this, and to protect the people and technology in use within and around this product, power source was chosen carefully -- that contains short-circuit protection, over-voltage shutdown, and an internal fuse to ensure the protection of the community’s health. Another possible safety issue is a faulty stepper motor. If by some Act of God or through some mechanical problem the stepper motor malfunctioned, it could spin out of control and send parts of this project and drinks flying. To solve this problem, an exit sequence is implemented to manually force a shutdown on motors sub-system upon hitting exit button over 3 seconds in the main menu.
5 References


