Brewin' Brews

Eduardo Ramirez, Fabrizio Castaneda, Jason Salgado, and Ryan Kosasih
Team 7

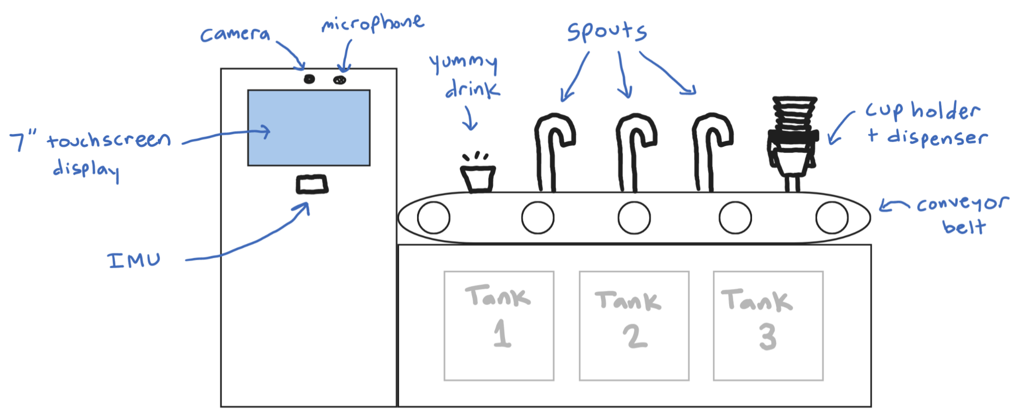
I. INTRODUCTION

The bar service industry has not changed the way they operate for a long time. Since it's inception, bars have revolved around the role of the bartender. The problem is that bartenders are one major source of lost revenue. They are responsible for spilling many drinks, slow to service customers, and can't for extremely long/late hours. To keep the party going, our solution is automation. Just power up and fill the Brewin' Brewer with your favorite ingredients (base liquor, syrup, enhancements) and it will automatically take orders, make drinks, keep track of customer tabs, make sure customers can legally drink, conduct sobriety tests, and facilitate payments. This way, bars can provide quick and consistent drinks for longer hours. Bar owners will also have access to an admin page that they can track the Brewin' Brewer from.

II. DESIGN

Product Concept

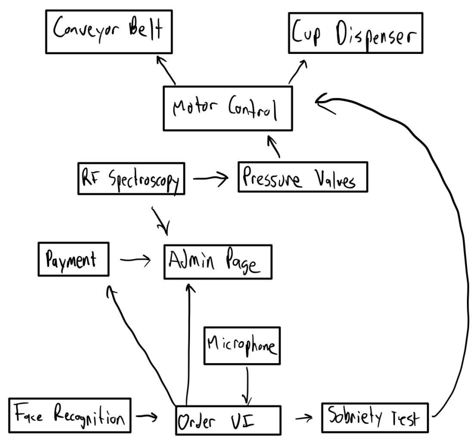
1. The device will have a touchscreen display where users will be able to place orders. A camera above it will be used for facial recognition, ID scanning, and Venmo QR Code scanning. Alternatively, users can order or initiate payment using voice commands by speaking towards the microphone near the screen. An IMU will be underneath the screen to facilitate a tap-based "Simon Says" sobriety test. Once the order is in, the cup holder will dispense one cup onto the conveyor belt and move the cup along the spouts and dispense the amount needed of each liquid to make the drink. At the end of the belt, the customer will be able to grab their finished drink and it is added to their tab.



System Dependencies

1. Each system will receive and/or produce a signal that will go into another system. The robot's system feedback starts at the face recognition system, as the user inputs their face and ID into a camera that will signal the UI system to store the user's data and tab to the admin page. A microphone can then also be incorporated to make orders and/or pay the tab once the face recognition step is completed. After an order

is placed, the IMU once again takes an input for a sobriety test, on which a successful test provides a signal to the motor control system of the conveyor belt and cup dispenser. At the same time, that signal would also propagate to the drink dispensing system to use the mass measurement tools and pressure system, dispensing at each drink checkpoint until the order is complete. To track all of these steps, a central signal and state microcontroller will be in charge of tracking the state of the robot. A general “idle” state will be the default state of the system where it will do nothing until someone uses the face recognition camera system. Each step will have certain time constraints, completion conditions, and failure conditions. At each failure or time expiration, the system will go back into idle; otherwise it will continue through the cycle as aforementioned.



III. RESEARCH

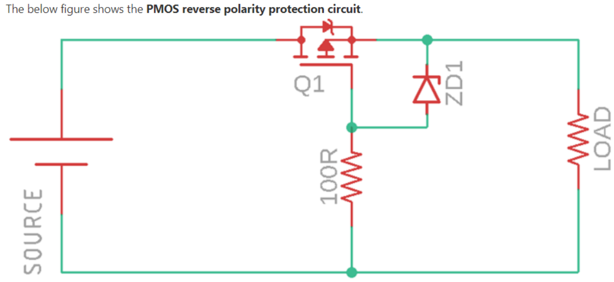
A. Required Hardware
1. Raspberry Pi Model B
2. Raspberry Pi Zero WH
3. Camera Module
4. BerryIMU
5. Multiple DC Motors
6. Pressure Valves
7. 120AC/DC Rectifier
8. Pelican Case + JBox Panel + Terminal Blocks
9. Tiny Vector Network Analyzer (VNA) + USB Realtek Software defined radio (SDR)

B. Required Software
1. OpenCV Library (Video)
2. Scikit-Learn Library (ML)
3. Google Cloud Speech API (Speech Recognition)
4. Venmo API (Payment Processing)
5. HTML/CSS/JS (Frontend)
6. Python (Backend)
7. MySQL (Database)

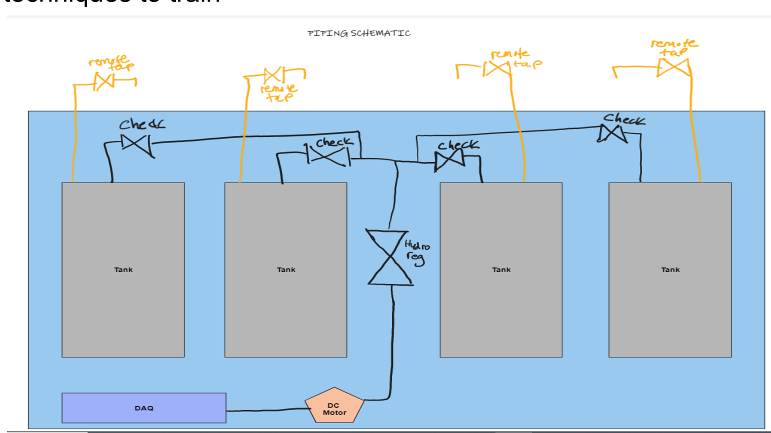
C. Subsystems
1. The Drink
   a) Power Supply + Intrinsic Safety
      (1) The product will run off of a 12V bus obtained by using a phone charging rectifier. The 12V bus will have intrinsic safety through the use of a PMOS reverse polarity protection circuit.

(2) In order to prevent floating voltages and ESD damage from the use of single point ground tied to earth ground. Local grounding done on chassis ground.
(3) An optional Uninterruptible Power Supply (UPS) will be available and auctioneered using a quint diode-or to take over should 120VAC facility power fail
(4) Normally open and normally closed valves along the piping will be defined for user manual vent should the system malfunction

The below figure shows the **PMOS reverse polarity protection circuit**.



b) Piping and Instrumentation (P&ID)
(1) Each tank will be pressurized by a common 12V pump, with a hydrostatic pressure regulator on each line, a remote valve will be at each spigot to administer drinks
(2) Manual venting valves will be installed before the remote valves, and each tank will have a drain accessible from the outside
(3) There will be a redundant fill level meter in the form of an RF mass spectrometer. The pressure differential measurement will train the mass spectrometer. We can use ML regression and classification techniques to train



c) Valves
(1) Thermocouples/RTD, Differential pressure transducers, limits, relays, will be installed.
(2) The biggest limitation is how many Analog and Digital Channels we have.
(3) Signals are inherently analog so we must condition them in order to make full use of each channel. Digitize the ones which make sense (Limits+ relay). The relay is a hall probe.

(4) Don't put high current draw near your instrumentation signals. Interference bad. Use TSP, ground drain, foil, and single point grounding. M22759/16 is preferred.

(5) This system segment will still be a lumped element (60Hz) so impedance matching is of minimal concern.

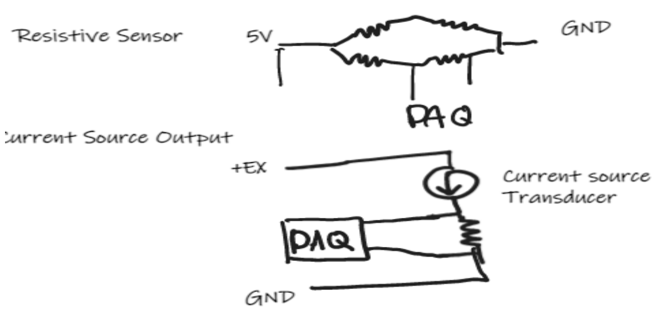
d) Sensing and Control

(1) On the front end for all the instrumentation we will use very basic signal conditioning. All current outputs will be 4-20mA and all voltage will be 1-5V , or 0-5V depending on the type of sensor we obtain. There are tradeoffs for both types, however since the lines are short both are viable.

(2) In the end all readings become voltage inputs into the DAQ. We can divide across resistors, probe hall sensors, etc. We think of the daq input as a very large resistance similar to an op amp or 1G ohm resistor

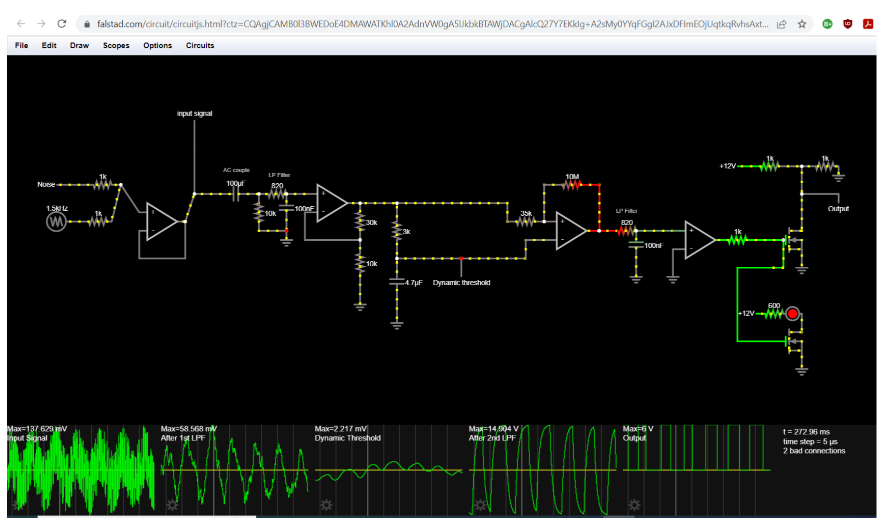
(3) The basic sine wave to digital converter below will be used for all the analog signals we can digitize and remain meaningful

BASIC SENSING SCHEMATIC



Single Point GND, to Chassis GND to Earth GND

Voltage Divide or otherwise mainpulate so DAQ sees
0-5V for resistive sensors
or
1-5V/4-20mA for Current output



e) Spectroscopy
    (1) Use Eigen modes measured by S11 to classify fill levels
    (2) Use basic circuit diagram below.
    (3) Use SDR to measure output of antenna when connected to VNA. Making a mixer and RF transmitter would be impractical since fill levels eigenmodes would change based on fill level
    (4) Radar no good since water absorbs RF well. Ultrasonic an okay alternative but no way to shield that so it stays functioning in wet environment

2. The Movement
    a) Signal Control Microcontroller (Raspberry Pi)
        (1) Each state of the robot will be tracked by the microcontroller, as each system will be connected to it to give it the signal. It will be constantly running upon starting the system to constantly check the system's state. The main use of the signal control system is to error check and control the final stage of the order where the cup moves to each dispenser valve.
    b) 3D Printed Cup Dispenser
        (1) The cup dispenser will store a stack of cups on top of the conveyor belt. A DC motor will move the bottom of the dispenser to release a single cup onto the belt, and will move back for the next order. It will grab cups individually by separating them with an arm that also moves up and down to grab and release each cup.
    c) 3D Printed Conveyor Belt
        (1) The belt consists of two rollers controlled by two DC motors, a large chassis, and a water-resistant material for the belt. The rollers will move the belt in one way to keep the cup moving through each dispensing valve for the drink components. This belt will be attached to the right of the screen and IMU system and on top of the encasing for the drink containers for best cabling and piping efficiency.
3. The Legal Check
    a) Face Recognition
        The program relies on Raspberry Pi camera to detect the user's face and ID.
        (1) All users (both first time and returning users) are required to approach the camera for verification and validation.
        (2) New users will have their faces not recognized (as their data has not been stored), thus they are required to provide an ID to store their face (from the ID), name and date of birth. Then the collected data will be matched with the face of the person standing in front of the camera. If the faces match and the user is legal or 21+, the users can obtain drinks.
        (3) Returning users only need to stand in front of the camera as their data has been stored. If their faces are recognized, then they are able to get drinks.
        (4) Data will be loaded and stored to the cloud storage.
    b) Sobriety Test
        The program relies on IMU to detect the taps done by users. The test is only

prompted when the user is getting their third or more drinks of the day. The instructions are in the form of tap patterns.

(1) The test is similar to the “Simon says” game - if the instruction given is started with “Simon says”, then the user is expected to follow the instruction given, whereas if the instruction does not start with “Simon says”, then the user must not follow the instruction.

(2) In order to pass the test, users must be able to follow three instructions given, with at least one of the instructions having “Simon says” in the beginning.

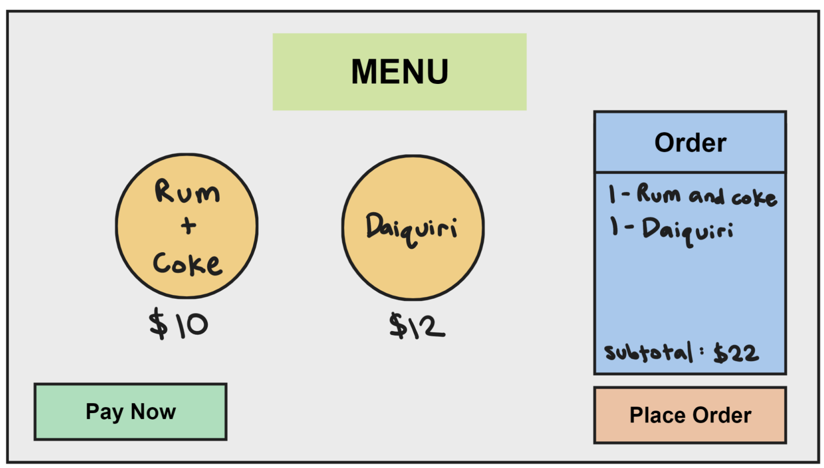
(3) The tap patterns will be made (generated) as the test is being programmed. Then out of the instructions generated, three of them are used to test each user, in addition to adding at least “Simon says” at the beginning of one of the three instructions.

(4) Users must do all three instructions stated in order to get their third or more drink.

4. The Menu

a) Ordering System

(1) After successful identification, the user gains access to the menu. Here, they can choose from the available drinks. The “Place Order” button adds their order to their tab and the device starts making the drink(s). The “Pay Now” button prompts the payment process. The user may also use voice commands like “Hey Brewin, add Rum and Coke to my order”.



5. The Money

a) QR Code Scanning

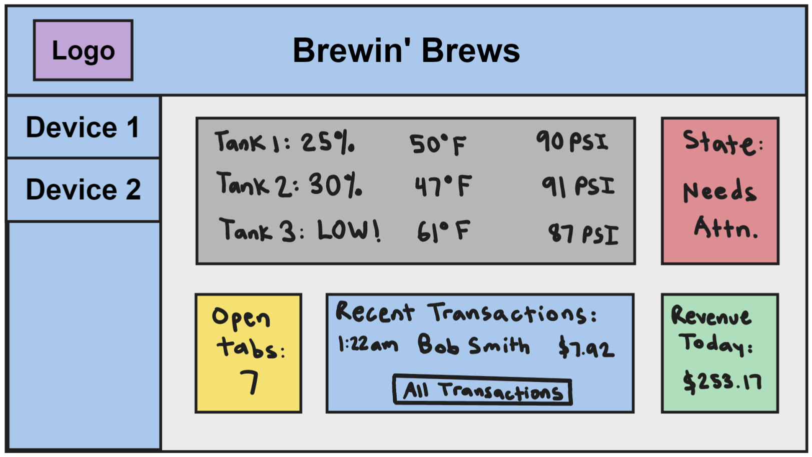
(1) Once the payment process is prompted, the touchscreen display will stream camera video and instructions for the customer to show their Venmo QR code. The camera will be active until the program can legibly read the QR code and extract the user ID.

b) Venmo API

(1) Once we have the user ID, we use the Venmo API to send a request for the amount of money on their tab. The system will periodically check if the request was completed, and if so, will show that the transaction went through and clear their tab.

6. The Admin Page

a) UI Web Page
   (1) Displays the device(s) data tied to an administrator account



b) Database
   (1) Device data (e.g. tank levels, liquid temperatures, pressure, open tabs) is sent to a database over the internet. The web page will use this database to display up-to-date data.

7. The Clean Up
   a) UI for Request
      (1) Will be present as a button on the admin page
      (2) Touch screen will prompt user to flush selected canisters
      (3) Post flush prompt for soap and water
      (4) Post Soap and water, prompt for water for final flush
   b) Controls
      (1) Manual drain required of selected canisters
      (2) Canisters fill w/ Soap+Water, then flush
      (3) Canister fill w/ water the flush
      *Flush is a full pressurization of piping to force liquid through

IV. EXPERIMENTAL TESTING
   A. Preliminary Testing Phase
      1. Test individual components for functionality out of the system (IMU, gesture, camera face recognition, voice recognition, valve dispensing, conveyor belt, cup dropper actuation)
      2. Mostly individual work, some collaboration as needed
   B. B. Grouped Testing Phase
      1. Connect small groups of electronics and code to ensure the immediate subsystems work together (e.g. conveyor belt movement and valve dispensing)
      2. Add code/hardware necessary to get subsystems connected
   C. Full Testing Phase
      1. Combine 2 groups at a time (e.g. valve/conveyor + face recognition/payment processing)
      2. Amalgamate groups until full product functionality achieved
      3. Multiple test cases for code to ensure exception catching for user error, full stress tests for hardware to ensure user safety

V. WORK DISTRIBUTION
   A. Scrum Master/ Team Lead: Fabrizio

1. Responsible for touchscreen UI, Admin page, voice commands, and payment processing

B. Hardware Lead: Jason
   1. Responsible for Electrical schematic and Piping+Instrumentation. Pressurization systems and junction box design and wiring. Hardware counterpart to Eduardo.

C. Software Lead: Ryan
   1. Responsible for facial recognition: face detection, face matching, store and load saved data of users.
   2. Responsible for sobriety test: generate tap patterns for the test and ensure IMU works for tapping, plus the test works as intended.

D. System Integration Lead: Eduardo
   1. Responsible for conveyor belt and cup dispensing motor systems, and signal control system integration with all system components. Assistance in other hardware aspects of the robot as needed

VI. TIMELINE