Project Documentation: Multimodal Internet Drink Mixer

CS 3671 - Prototyping Intelligent Devices | Section A | Spring 2024
Ethan Haarer | GTID: 903586678

Hardware Used:

- 5V Relay Board
- 3 Peristaltic Pumps
- 3 IR sender/receivers
- Arduino R4 Wifi
- Arduino Mega
- Adafruit Sharp Memory Display
- RC522 RFID card scanner
- Ultrasonic Proximity Sensor
- Surge Protector
- Various wires and LEDs
- 2 tactile switches
- 5V Power Supply
- 5m of 5mm silicone tubing
- 12 piping connectors

Libraries Used:

- NewPing
- SPI
- MFRC522
- Wifis3
- Adafruit_GFX
- Adafruit_Sharpmem
- qrcode

Custom Parts:

- 3D Printed Nozzle to consolidate 3 output silicone tubes (see below)
- 5 Custom circular Wood disks to aid in the framing of the main body (see below)

Skills Developed:

- How to process signals form IR sensors to have meaningful input to the Aduino
- How to Use the QrCode Library to work alongside the Adafruit GFX library to make a custom scalable QR code display function
- How to use relays to protect the Arduino and the rest of the circuit from high voltage and potential power spikes

- How to combine/modify IR sensors and water pumps to create a rotary encoder to exactly count rotations
- How to get two separate Arduinos to communicate solely over their Serial1 pins
- How to use an ultrasonic proximity sensor to detect specific ranges

Iterative Design Process:

1. First I needed to ensure that I could control the base dispensing, so I began with having the arduino simply turn on and off the peristaltic pumps with the relay board.
2. Next, I needed to make sure that I could count the exact number of rotations that each pump completes as I could not rely on time. I took apart the pump, covered half of the internals in electrical tape, then drilled a hole on the top to attach IR sender/receivers to check if the current color seen is black or white. After some testing, I implemented some basic signal processing to turn the signal from the IR pump into a square wave, then every time there is a falling edge I count that as one rotation for the pump.
3. For the pump I later added a 5V supply board to power the relay board separate from the arduino to further protect the arduino from power issues and potential power surges. I make sure here to connect the grounds of the the arduino to ensure a common ground across all devices.
4. Next, I wanted to make sure that if there was a drink pouring, there would be a cup underneath so the liquid wouldn't spill anywhere. Here, I used the NewPing library with an ultrasonic distance sensor to have the machine only dispense when there is a cup. Note for future parts of the project I came back to this section to ensure that the dispensing can pause and resume on the same drink if the cup is taken away, and cards can only scan when a cup is detected.
5. Next I needed to add the ability for inputting a drink via RFID cards. My initial plan was to have a separate machine that allowed for the writing of data to different RFID cards, with the main body of the machine decoding this data to produce the drink. However, the cards available in class did not have the functionality to be written to, and the 4x4 input pad I attempted to use for it was inconsistent. Instead I pivoted to having each UID of the cards to be used be hardcoded into the code, then when it sees a card it knows what recipe to output. Each time that card is seen, it is stored so the machine knows not to make that drink twice, unless the master control card is scanned, which resets the scans of all cards.
6. I then wanted an alternative way to input drinks to the machine, so using the Arduino R4 I connected to GT wifi and hosted a web server that allows users to specify how much of each drink they want, then pour it remotely by sending a HTTP request to the server. I also had the proximity sensor blink a blue LED that

denoted if a cup was detected or not to signify to the user that the machine was waiting for a cup before starting to pour.

7. Though the web server was up and running, the user had no way to access the link to the website, as the link changes every time it is set up over wifi (and GT doesn't like static IP addresses) so I first attempted to add an adafruit sharp memory display to the R4. Although I had a qrCode displaying on this display using the mega board, the R4 did not want to output to the display, an issue I still don't know why kept happening. Another problem was that I had run out of usable pins on the Arduino R4.
8. To resolve the issue in 7, I came up with a slightly unconventional solution: I was going to send the IP address across serial to another Arduino, and Arduino Mega, and have that display the qr code. I initially had an issue as I was attempting to send the IP as a string over serial, and no data was being seen and the qrcode held empty text. However, I was eventually able to implement a handshake tactic where the Mega is waiting until it hears serial data from the R4. When the R4 sends the data, it halts its code until it hears an acknowledgement from the R4 that the String has been received and processed, then it continues, ready for inputs from the user.
9. Finally I added a removable shell to the device, 3 LEDs to help communicate feedback from the machine to the user, and reset and stop button on the side of the machine to immediately stop dispensing and clear the current order and reset the R4 respectively.

Photos of project:

Figure 1: Final Project

Figure 2: Project without outer Shell

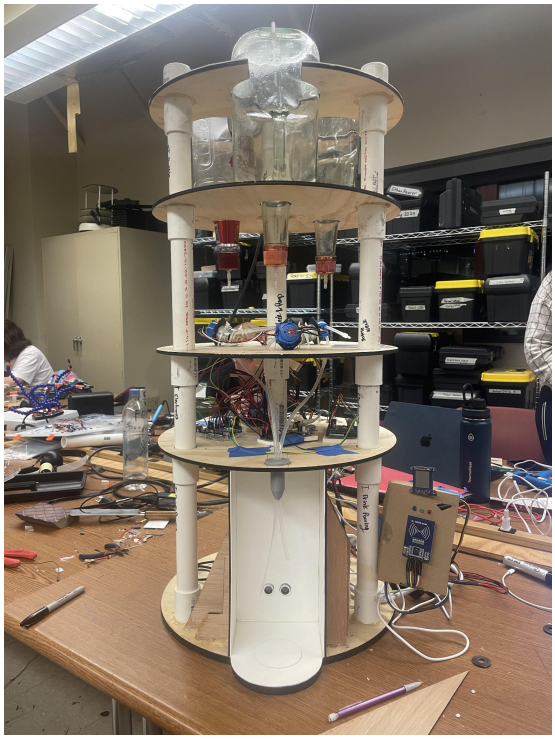


Figure 3: Screenshot of Website

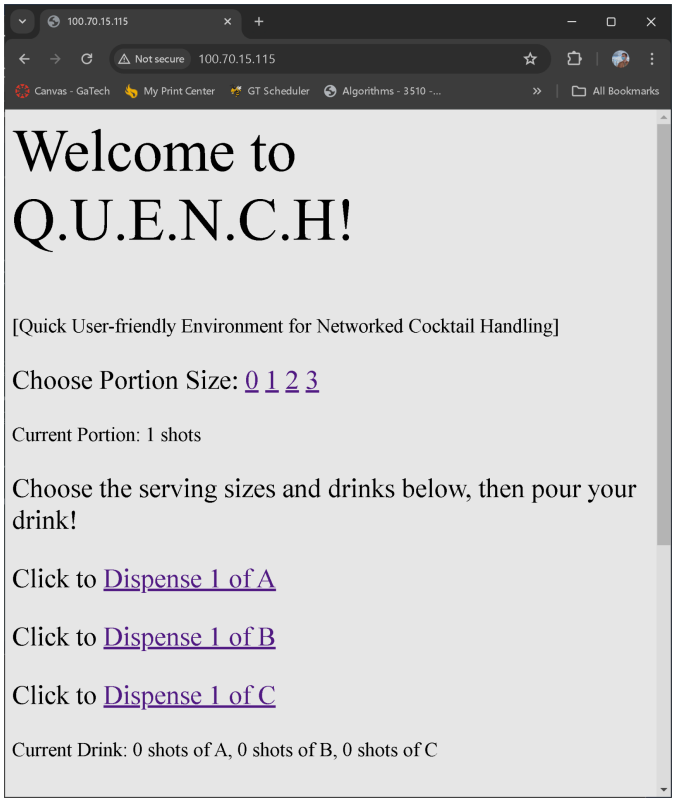


Figure 4: Peristaltic Pump and IR sensor arrangement

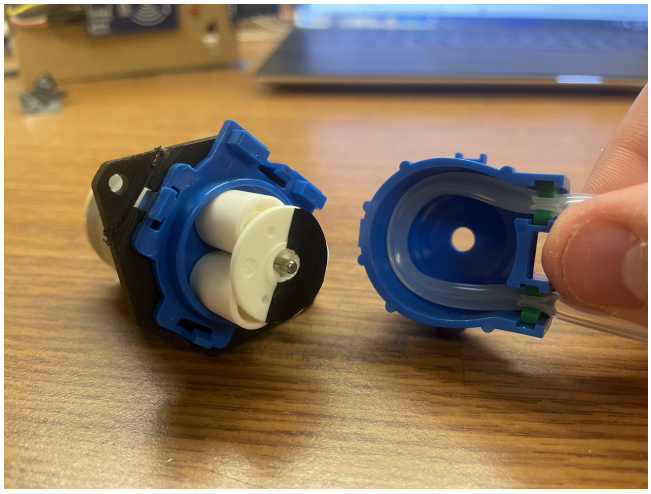


Figure 5: Laser Cut Disks used to form the main body

Figure 6: CAD of the Dispensing nozzle

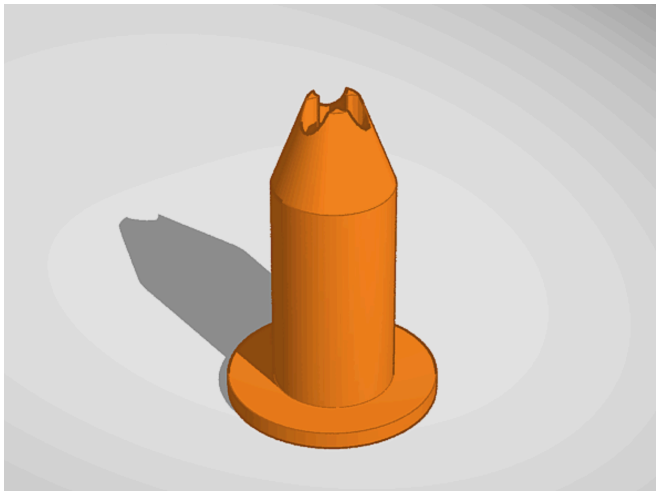


Figure 7: Initial Ideation Design CAD, where the top spun the bottles about an axis

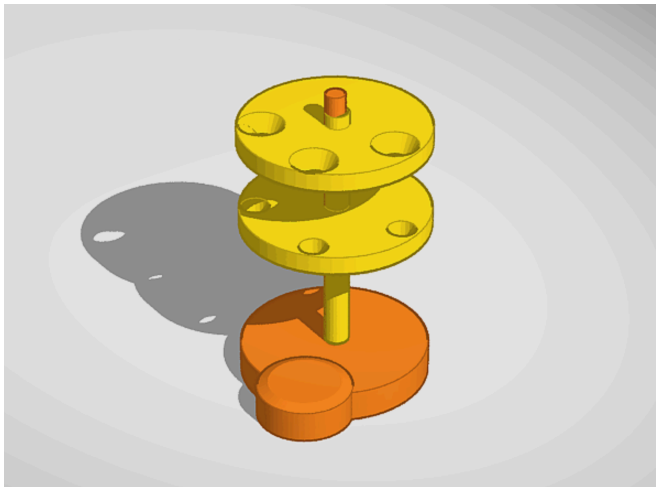
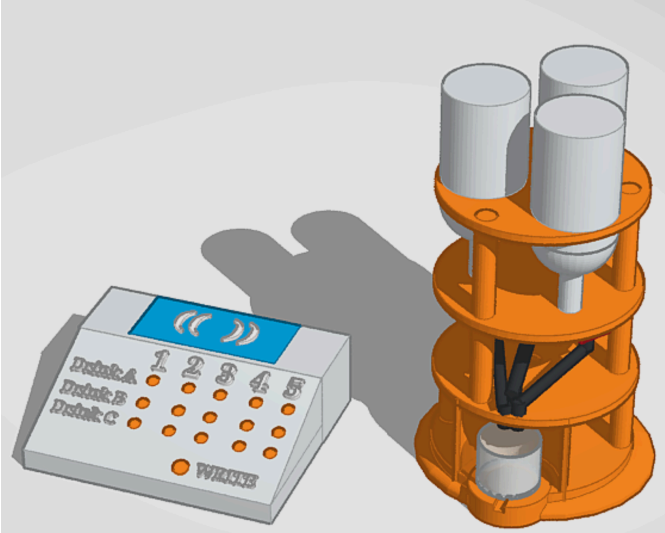


Figure 8: Final Ideation CAD design



Tutorials and References Used:

- https://arduinogetstarted.com/tutorials/arduino-uno-r4-wifi-controls-led-via-web
    - Tutorial on how to set up an Arduino r4 as a web server, I further added the ability for the web page to change based on the drink inputs of the user
- https://www.electronicshub.org/write-data-to-rfid-card-using-rc522-rfid/
    - Tutorial on how to read/write to RFID cards using Arduino, I further added the implementation to track which cards have been scanned or not
- https://www.arduino.cc/reference/en/libraries/qrcode/
    - Reference to an arduino library that converts URLs to a QRcode, used as a library reference. I further implemented the ability from the basic example to output to the serial monitor from a bitmap to instead loop through a bitmap to display the qr code pixel-by-pixel.
- https://www.youtube.com/watch?v=_QETdyeMyZw&t=218s&ab_channel=50Hz
    - This was used as a base for the serial communication, but I later added the handshake implementation to work with longer serial inputs.
- https://www.youtube.com/watch?v=0BNcI8jMcXE&ab_channel=Tinker%26Build
    - I used this as an example of how to power the pumps with a relay board, but I further change it as it usually uses a lot of amps, so instead I have the power come from the 5 Volts in instead of the Arduino itself.
- https://www.youtube.com/watch?v=K8d8BDtR9Jc&t=605s&ab_channel=VincentStevenson
    - This, alongside the basic example code from the NewPing library, helped me set up my proximity sensor, I used this to further add checks to the code to ensure a cup before pouring and pausing the dispensing if the cup is taken away.

Pin Out Schematics:

The following Pins are used for the Arduino R4: (note, GND is connected to 5V power supply ground)

- RFID-RC522
    - Pin 13 to SCK
    - Pin 12 to MISO
    - Pin 11 to MOSI
    - Pin 10 to SDA
    - Pin 9 to RST
    - GND to ground, 3.3V to Arduino 3.3 Volts out Pin
- HC-SR04 Proximity Sensor
    - Pin 6 to Trig
    - Pin5 to Echo
    - Gnd to ground, VCC to power supply
- IR Sensors (x3)
    - For each pump, IR1 goes to A0, IR2 to A1, and IR3 to A2
    - Vin to 5V power, GND to ground
- Relay Board
    - Pin 4 to IN1
    - Pin 3 to IN2
    - PIN2 to IN1
    - GND to Ground, 5v to Arduino R4 5V out
    - JD-VCC to 5v Power Supply
    - RELAY: Set the pumps up in a parallel circuit with the power supply plug of choice voltage, and have the relay break this circuit for a relay on each branch to ensure all pumps can run independently of each other
- LEDs
    - Pin 7 to Red LED
    - Pin 8 to Green LED
    - Pin A5 to Blue LED
- Serial Communication
    - Pin 0 RX to Mega TX
    - Pin 1 TX to Mega RX

The following Pins are used for the Arduino Mega:

- Serial Communication
    - Pin 0 RX to Mega TX
    - Pin 1 TX to Mega RX
    - Ensure Mega GND is connected to R4 GND
- Sharp Memory Display
    - Pin 13 to SCLK

- Pin 11 to MOSI
- Pin 10 to CS
- Vin to 5V
- GND to ground