CM10194 – Arduino Coursework 2

amg208 – ar2499 – al2303

Introduction

Two or more Arduinos, three or more inputs, and two or more outputs – endless possibilities. When given so much freedom, it was important to consider a wide variety of projects. We explored the idea of a GameBoy-style snake console and a robotic condiment delivery system but felt an instant connection with a cocktail machine. Being students, we felt that we would be passionate enough about our idea to stay focused.

Project Outline

Whilst a cocktail machine might appear to be a relatively straightforward concept, pouring different amounts of various liquids, in combination, to produce a particular drink, we wanted to challenge ourselves.

We decided to have a Teams meeting to discuss the project and produced the following whiteboard:

*Please see **figure.1** at the end of the PDF.*

From this whiteboard, the following extension implementations can be found:

- A speaker system, with volume control, that plays various melodies, depending on the system's state.
- An LCD to show the user the various cocktail options, alongside a contrast adjuster.
- A coin system that would allow users to 'pay' for their drink.
- An option selection system

Splitting the workload:

To work efficiently as a team, we decided to split various tasks out to different individuals.

Archie:

- The coin dispenser
- Piezo, speaker system

Aidan:

- Take drink orders, and trigger relays, connected to milk dispensers to dispense the correct amounts.

Andrew:

- Take users' drink order and send them to the cocktail producing Arduino.
- Create a user interface that dynamically adjusts according to various user selections.
- Adjust LCD contrast using a potentiometer.

Technology review:

Liquid dispensing:
Typically, in restaurants, there are dispensers that pour out only one drink, but they do not mix multiple together. This can automate the process of making cocktails. On another note, in a pharmaceutical setting, pharmacists have to measure out volumes of liquid themselves. This could save them time and allow for automated drug manufacturing. This could lead to more accurate measurements in a smaller amount of time, were the correct equipment used.
Another use of our solution could be paint mixing in a hardware shop, by just passing a specific ratio of each primary colour (in addition to black and white for brighter and darker tones). Our project could also be paired with an RGB photoresistor so we could match the colour of whatever it senses, to mix the right colour.

Multi Arduino:
The reason we used two Arduinos for this project is because in a real world application, the menu interface might be in a different place to the pouring circuit and they could communicate wirelessly, like in a fast-food restaurant (order and order making).
Moreover, due to the pandemic, if our system were to be used in a restaurant, should people have to wait in the same queue when placing an order and waiting, large queues would be created, leading to a lack of social distancing.
Additionally, we needed another Arduino for extra pins, since the LCD is pin-greedy and we would therefore not be able to have plugged everything in.

Development process:

Development:

Our project was completed iteratively: The LCD menu was designed by Andrew, using TinkerCad before physically wiring it up on its own independent Arduino. Archie designed the coin dispenser and piezo which plays throughout the menus, yet again on its own independent Arduino. Aiden wired up the solenoids and made the milk dispensers turn on, on his own board.
We then connected the coin dispenser onto the LCD Arduino and used that board as a master Arduino and the relay and piezo Arduino as a slave Arduino. We implemented the functionality of sending the drink ratios as bytes into the slave Arduino. We avoided using delays in both the pouring and the melody by using time since we last declared a variable to be millis() and using while loops. We did not want to use delays as this can make our program unresponsive and could lead to us not being able to run multiple dispensers at once or even the piezo playing at the same time as the dispensers.
We felt that our project would look much more professional if it was enclosed in a box, so we laser cut one out of plywood and superglued it together in the Lovelace lab. In the same lab, we also had to solder pins to wires of the dispensers, as they would not fit into the breadboard. We also soldered pins to the legs of the buttons so we could plug male to female wire adapters onto them, so they wouldn’t need to be plugged into the breadboard.

Research:
Initially, we needed to research how to connect the Arduinos together and send data to each other. We found out that we can use the Wire by sending data as individual bytes and then reading these bytes on the slave Arduino.
We needed to research a way to use a second circuit to supply a different voltage to the dispensers. This was an issue that needed to be fixed since we did not want to either damage our boards and we also

wanted to use variable voltages to power our dispenser. We ended up using relays as they acted as both a switch and a separator of circuits.
We also needed to research various types of dispensing the liquid to the user. We researched multiple gravity-based and motor/pump based systems and weighed each one's advantages and disadvantages until we concluded that milk dispensers were best in terms of cost and complexity.
Another piece of research we did was how to control the volume of the piezo using the components we had. Initially, we wanted to control it digitally, using analogWrite, however, this affected the pitch of our piezo as well as the volume. We ended up using a 10k potentiometer, wired up in series with the piezo. This directly controls the voltage that the piezo gets.
Some skills we also learned were how to solder since we needed to solder multiple pins to our wires, laser cutting for our box and 3D printing for our plastic cup holder.

Problems:

When we decided we wanted to create a drink mixer, our first issue was how we would actually pour the drinks. Initially, we were going to have a gravity-based system using valve-solenoids, which would require a 9V supply. Another issue this caused is that when you flip a bottle upside down, it creates a vacuum at the top of the bottle. To overcome this issue, we would need to have a special cap, with a hole drilled for the rubber tubing and a smaller hole drilled for a straw to let air in at the top of the bottle. Furthermore, this solution requires more complex formulas to determine the amount of liquid poured, depending on how much liquid there is in the bottle.
A solution to overcome this problem would be to use a sheet of copper metal sitting in the liquid. When metal is in water, as opposed to air, its resistance is lower. Based on the resistance, we could calculate the volume of the liquid in the bottle, letting us know when to close the valves. However, the main issue with this would be that we could get copper oxide in the drink, which could poison our users. This solution was a dealbreaker for us since we value user safety rather than accuracy. If our solution was used for paint/colour mixing, then this would not be an issue, however, in a drink or pharmaceutical setting, this is simply not viable.
During development, we had issues with our button debounce time, since the variable we used to measure the time for the last press was an int, rather than unsigned long, meaning that this variable would reach a limit and then subtracting this value from millis() would return a number which would only increase, meaning our debouncing would only work for a short amount of time.
Our LCD contained a variety of bugs when we included a progress bar since the equation used to determine how long the progress bar will take was using integers in a fraction which would typically return a decimal. This would lead to undefined behaviour since we would set the cursor of the LCD to a value that did not exist and would lead to having random characters displayed on our display. This was fixed by casting variables to doubles.
We created a box by laser cutting plywood, made to fit our bottles and interface, including LCDs, buttons, piezo and potentiometers. Unfortunately, our measurements were too small, so we needed to either sand down some parts or get another piece of plywood laser cut.
While supergluing our project, we got superglue in the “down” button, making it stick for a few seconds when clicking it. This made it very unpleasant to use our screen, so we decided to swap the functionality of the up button with the down and change the button when we get to the Lovelace lab.

In addition to these issues, we also faced team issues. One of our team members, (Andrew Grosuleac/ amg208), caught Covid-19, so we had to communicate over Teams, slowing down prototyping. Furthermore, we were limited to three people for a four-person project due to the number of students on the course. This greatly increased each of our workloads and meant we had to work extremely efficiently within the time frame.

Final prototype implementation:

LCD: We needed to make our program easy and fun to use in order to stay in theme with our project. We had to make do with a very basic 16x2 display, so we tried our hardest to make it user-friendly. We believe we achieved this by having a simple and easy-to-follow user interface and a loading bar when we pour the drinks to show people how much of their drink has poured so far.
When selecting a drink, the ratios (either pre-programmed by the premade drinks or user-defined for custom drinks) get sent to the slave Arduino, using the Wire library. This will determine how long each liquid should pour for.

Buttons: The buttons in our final implementation allowed the users to select a menu/drink and navigate the menu.

Coin and note acceptor:

Vending machines have always used a coin and note acceptor to take payment. Having a system that recognised various types of coins and notes felt beyond the scope of the time frame. Instead, we decided to have the user pay for coin credits and have the coin credits be recognised.

The cocktail Machine has a vending machine-style hole in which to insert the coin. The coin blocks the light coming to the photoresistor via the LED. This difference in normal light intensity coming to the photoresistor triggers the credit to increment.

LED: The LED is used to shine a light on the photoresistor to keep track of whether the user inputs a coin.
Photoresistor: The photoresistor is used in conjunction with the LED. When the user inputs a coin, the light reading of the photoresistor drops, indicating that a coin was passed through, since the coin passes between the LED and the photoresistor, blocking the light of the LED.

Piezo: Waiting for your drink can be boring. We decided to implement a piezo speaker system to alleviate boredom during drink waiting times. A potentiometer is connected in series to the piezo for volume control. The project has a different melody play based on the machine's different states: 'Jingle Bells' for the 'insert credit' and selection screen, 'Escape (The Piña Colada Song) for drink making and a custom melody indicating that a drink is finished pouring.

Milk dispenser: Three dispensers (pumps) are used to draw liquid from the bottles and dispense it to the user's glass. They take 3V so they are connected to a secondary circuit containing two AA batteries. This secondary circuit is switched on by a relay for each individual dispenser. The amount of time varies depending on the ratios for each drink.

Relay: The relay solves the issue of separating the 3V circuit needed to run the dispensers from our Arduino board, while at the same time acting as a switch. The positive wire of the dispenser is connected to Normally Open on the relay, whereas the COM of the relay is connected to the positive of the 3V circuit. While our Arduinos do have a 3.3V output, we used a relay as we might want to expand our projects to use more powerful pumps which could require a higher voltage, which cannot be supplied by the Arduino.

Potentiometer: We used two photoresistors in our project. One is used to control the contrast of the LCD, to make it easier to see in different light conditions. The second potentiometer is used to vary the current supplied to the piezo, thus changing its volume.

Intra-Arduino Communication: In addition to operating and interacting with simple devices such as buttons and LEDs, the Arduinos need to communicate data between themselves. Specifically, the drink ratios recorded by the master Arduino must deliver said ratio to the slave Arduino, which control the drink dispensers. We considered connecting to pins on separate Arduinos, one as an input and the other as an output, and having the receiver count how many times the pin is raised high within a given amount of time to transfer integers. After further research, we realised that Arduino software has far more advanced built-in methods of communicating with each other. I2C is a simple form of serial communication over a single wire. It requires to have two connected analog pins as well as connected ground pins. While connected, the Arduinos can read and write data bit by bit directly to the wire.

Conclusion

This project used a wide variety of lab learned skills; LEDs, potentiometers, buttons and piezos. It also uses an LCD, relays and milk dispensing motor which required additional group research. This taught us many skills

Potential Additional Uses:

Whilst this project idea seems rather comical and entertaining by nature, it actually has a variety of applications. For instance, it could be used within an industrial setting, mixing black and white with different primary paint colours to produce the desired paint shade. It could also be used within the medical field, within pharmacy, mixing various ingredients to produce a particular medicine.

Expansions:

To expand the project, we could add more dispensers to allow for more drink combinations. We could also add a sensor for detecting when the level of each of the liquids. We could also add an auto contrasting display, which changes based on the environment.

figure.1

Use a minimum of 2 Arduinos that must communicate in a meaningful way:

1. Coin intake, Music Playing & Drink Choice
2. Drink Dispensing

Include a minimum of 3 unique inputs (buttons, potentiometers, sensors etc)

- Buttons
- Photoresistor
- Potentiometer

Include a minimum of 2 outputs (serial monitor, LCD, LEDs, servo etc)

- LCD
- RELAY / Milk Dispenser
- LED
- PIEZO

Cocktail Machine - Functionality

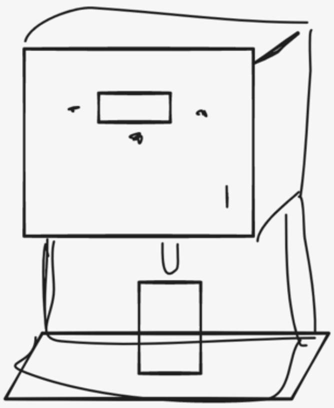
- Take user's cocktail preference via buttons (type of cocktail, alongside strength)

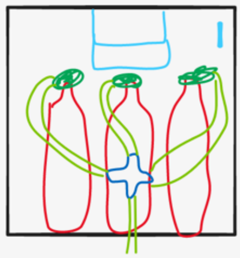
- Other menu for personal ratio preference.

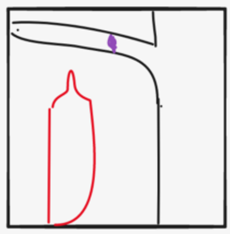
- Check whether coin has been put into machine and react accordingly

- Should play and adjust volume of music via piezo and potentiometer

- Calculate time periods for how long the relays should individually run for.







Coin dispenser

What do we need:

- Milk Dispenser x3 (so 2 packs)
- Piezo
- Relays (Same number as milk dispensers)
- LED x1
- Photoresistor
- Buttons x3
- Potentiometer
- Rubber Bands x3 (Same number as milk dispensers)
- Battery adapter x3
- 9V Battery x3
- Arduino Cable Wire
- Vinyl Tubes

What we need to buy:

- Milk Dispenser x3 (so 2 packs)
- Roll of Arduino wire
- Rubber Bands
- 9V Batteries x3 (Talk to Alvero)
- Battery Adapters x3 (Maybe / Talk to Alvero)
- 4 Piece PVC Connector
- 4 Metal clips
- Plastic back panel

Distributing Workload:

Archie:

- Music and Coin dispenser

Aidan:

- Take LCD Input to send to ratios

Andrew:

- LCD Input

Aidan sends Archie and Andrew completion of drink

Andrew sends Aidan drink ratios

Andrew sends Archie before he's sent to Aidan saying about to start drink

Archie sends to Andrew when coin has been inputed

Schematic and photos: