**MmmBar**

-DMP project-

Name: Deac Melinda-Anca

Group: 30431/1

**Contents**

Introduction………………………………………………………………………pg. 3

Bibliography……………………………………………………………………...pg. 3

Implementation…………………………………………………………………..pg. 4

Testing……………………………………………………………………………pg. 6

Conclusions………………………………………………………………………pg. 7

**Introduction**

I designed the project while talking with former highschool colleagues about our experiences from the past. I started to miss them and all the fun we used to have and decided to build something that we can have fun with when we will meet again. This is how the “Mmmbar” got its first spark.

Initially, it was designed to mix drinks (not childproof drinks), but on second thoughts, I will present it as a coffee-based drinks mixer.

I know that there are many such machines, but I wanted to have one of my own. Also, all the implementations I have seen online seemed incredibly fun to build and I wanted to design one of my own.

**Bibliography**

I did find some implementations similar to what I want to do on multiple sites. I will compare the three that stayed with me for longer.

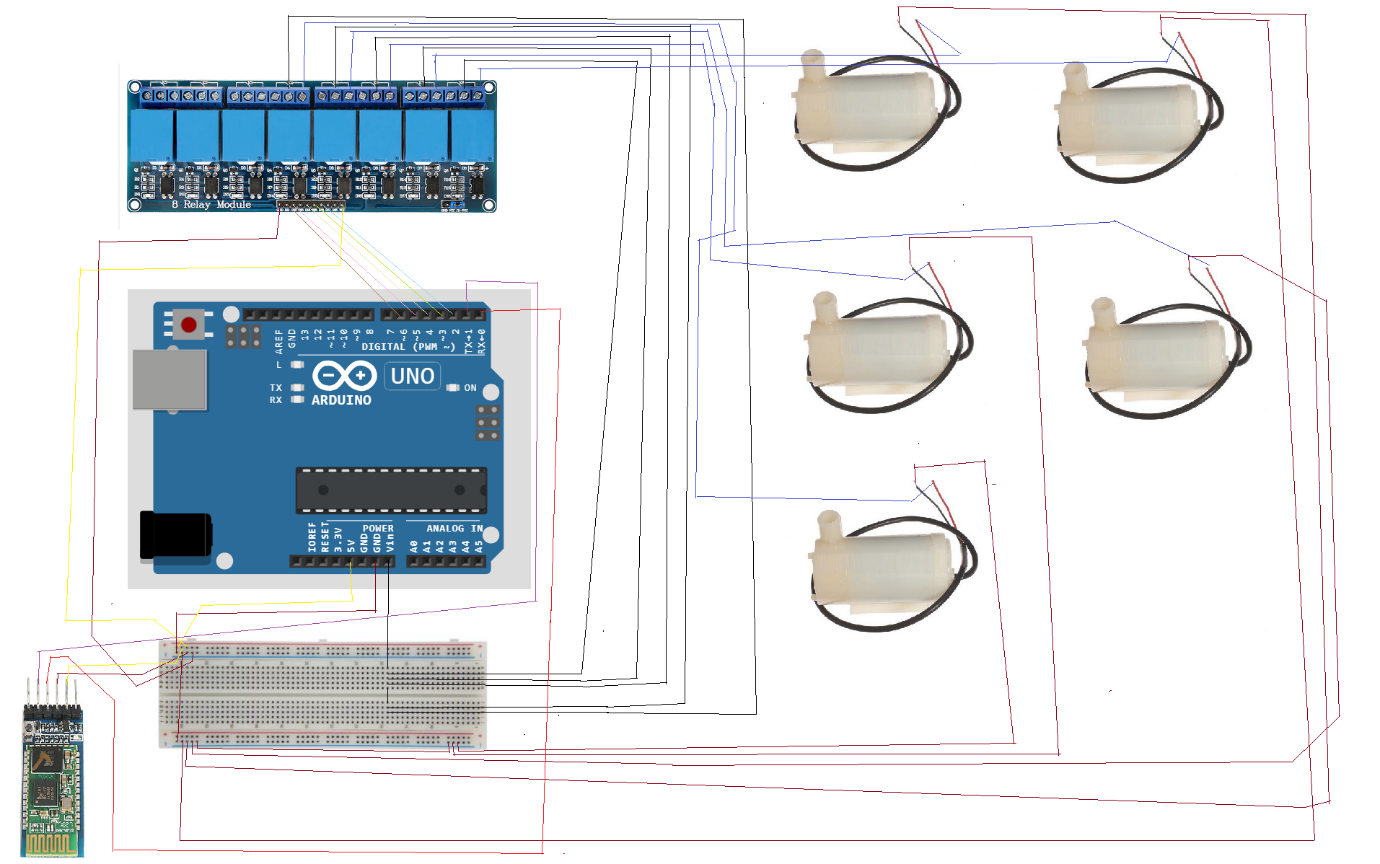
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name/Link | Working method | Aestethics | Price | Transfer Methods |
| Cheap Barboot / [Arduino cocktail maker | Cheap Barbot - YouTube](https://www.youtube.com/watch?v=UyE59p9K1fA) | Uses Arduino UNO, relays, pumps (not submersible), and a phone app written in MIT App Inventor. | Not incredibly nice to look at, but easy to use. | Convenient, not expensive | Probably easy to carry around. |
| The Inebriator / [The Inebriator - Arduino Cocktail Machine - Dispensing Signature Cocktail - YouTube](https://www.youtube.com/watch?v=hJIkJ9x0-JQ) | Also uses Arduino uno. Hard to find more details, but has an interesting way of pouring the liquids. | Impressive, eye catching | Probably more time consuming to build than the one above and also more expensive. | Difficult to even get it through the door. |
| Home Made Automated Cocktail Bar / [Homemade Automated Cocktail Bar using Arduino - YouTube](https://www.youtube.com/watch?v=u4aNY_1bjME) | Most of it is 3D printed, uses servomotors. Has a similar way of pouring as the inebriator, which is pretty impressive. The pumps are similar to those used in the cheap barboot. Has Arduino uno and relays. | Incredibly fine to look at. Eye catching and entertaining. | Time consuming to build, probably equally expensive as the inebriator. | More compact than the inebriator, but still difficult to carry around. |
| MmmBar | Like most of the previously described implementations, it uses Arduino Uno, and relays, but submersible pumps (since they were cheaper) | Not impressive. Rather cute. | Cheap, 30-40 euros. | Easy to carry around, taking up little space. |

**Implementation**

To implement the MmmBar, I tried firstly, to make use of the components I already had at home. So, I used an Arduino Uno and some wine hoses, some party cups and some zip ties. Then, I ordered 8 relays and the submersible pumps. I tried to stay as non-expansive as possible.

In order to get the liquids out of the containers, I connected a set of 8 relays to the Arduino uno’s VIN pin and to the gnd. The signals were sent from the 2-6 pins from the board. Pretty easy to get around with. The VIN workaround (for not using a separate power source) is possible because I used 3-6V water submersible pumps, which don’t need more power from the relays than what the board can send. The initial idea had more powerful relays in plan, but they were way out of the budget.

A demo with the schema will be attached below, but I will explain it better under it.



I connected the HC05 bluetooth device to the Rx, Tx pins (pin 0, respectively pin 1) and to the GND and 5V pins.

Then, I powered the relay board with the 5V and GND pins of the Arduino and I connected the input pins of the relays to pin 2, 3, 4, 5, 6 and 7.

Each relay gets its voltage from the VIN pin of the Arduino and it feeds the Vcc wire of the pumps.

The other wire of the pumps is connected to the GND of the Arduino board.

There is nothing smarter or more complicated than this. The idea was simple, but the physical implementation was a real challenge.

**Wall 1. : Weak pumps!**

As a starting point, I tried to fix the pumps on a wooden plate and put a hoses in the cups (instead of the pumps themselves), but no matter how much power I would feed the Arduino (which meant more power for the relays thanks to the VIN pin), the pumps just wouldn’t make it. Maybe it was because I didn’t seal it properly or because I sent the power wrongly to the relays. Anyhow, the pumps wouldn’t work and I desperately needed to regroup and rethink my solution.

Two hours later, I was placing the pumps carefully inside the cups and closing the lid on them. This is how they looked in the final design too.

**Wall 2. : No place left for the composition cup**

After setting the whole project on the board, the cup where the composition was to go in, had no space left for placing. So I nailed in a new board, to extend the initial one.

**Wall 3. : Cups would not stay in place**

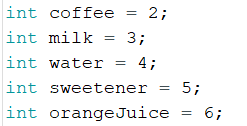
The cups holding the pumps would get pushed by the hoses and lose their spot, so I had to fix them with some zip ties of eachother and of the wooden board.

**Wall 4. : Powering the pumps**

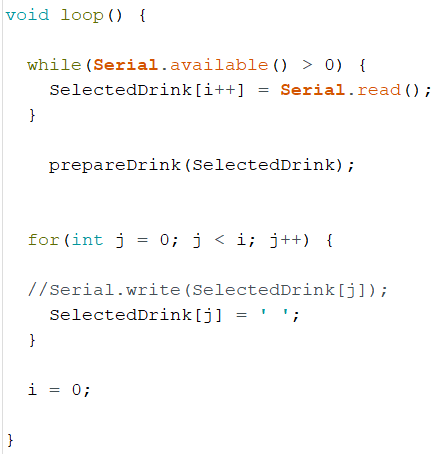
This would not have been a problem if the initial design would’ve hold, but since I had to place the pumps in the cups, I also had to come up with an idea to get the power to the pumps’ wires. So I asked a friend to teach me how to take off the isolation of the wires and how to bond two such wires, then fix them with tape. A real interesting part since I failed my previous (in 2017) tape applying course. So, the aesthetics might have had a little bit to suffer, but after this final modification, the project was finally ready to test!

**Algorithms**

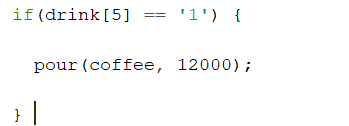
I store the componence of each cup as the name of the pin at each it is placed:



At each loop, if I received a new signal from the Bluetooth device, I read it and prepare the drink accordingly.

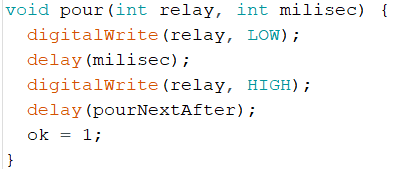


An example of drink preparation:



This code prepares 300 ml of coffee.

When pouring, I turn the corresponding relay on for the requested amount of milliseconds, then turn it off.



**Testing**

Like I said throughout the Implementation chapter, basically the testing was pretty much happening in the same time. After each addition, I had to test the functionality. After each modification, I had to test the functionality.

So a main testing issue that I had was when I realized that the pumps were not strong enough to pull the liquid through a hose. I obviously had to test the pumps to find out that they won’t work like this.

After all, when the design was ready to roll, I had to find the perfect seconds combination to prepare the drinks written in the mobile application. This could only be done by testing and testing and testing and testing again with newer configurations. I managed to compute the fact that 2000 milliseconds meant 50 milliliters of liquid (give or take). After this, things were pretty simple, I just adjusted the timing in the code and tested to see if each pump was giving the expected response, which they did.

**Conclusion**

In conclusion, I would like to say that I never feel like working when I do things like this. From the beginning, I imagined this project in my home, I thought about how I will use it, I designed it specifically for my own use and I had the most of fun making it. Even now, that it is completed, I still think about all the things I want to improve and add.

As a portable, beginner-like option, I believe the project is pretty nice, but when talking about aesthetics, I want to make huge improvements. I also would love to improve the security of the electronic components, because, as everyone said when they heard about the project, “You put the electronics so close to the water?!”. Even a plastic shield would do the trick. But for the amount of time I had and for the experience I own, the project reflects the contents of my brain with a certainty of about 97%.

|  |  |
| --- | --- |
|  |  |
|  |  |