BarBoi Documentation

Corey Du Val and Drew Ehrlich

Release 1.1, July 2020

1 Project Vision

The purpose of the BarBoi project was for Corey and Drew to apply the skills they have learned both in class and out of class to design, build, and test a fully operational machine. This documentation is written with the goal of walking others though our design to allow other groups (amateurs) to study or replicate our project.

2 Non-technical Overview

The BarBoi (also known as the AutoBartender) is a machine to pour mixed drinks without the need to have any human interaction. To do this, nine bottles are hung from the ceiling of the BarBoi, each with a computer controlled valve. When the user visits a website and makes a selection, the computer opens the correct valves for a set amount of time to use gravity to pour a drink. We considered the BarBoi a success when it satisfied the following three criteria:

1. **Correctly Pours Drinks:** Obviously, the machine needs to work in the way we intended, and cannot pour the wrong drinks or pour the wrong ratios in those drinks.
2. **Easy to use:** We wanted to avoid relying on constant babysitting of the BarBoi while it is in use. Ideally, the only manual work needed from the user would be to plug in the machine, fill the bottles that hold the liquids, and clean up after it is done being used.
3. **Sturdy:** Because the nature of the machine is that it will often be around drunk users or operators, we need to make sure that the BarBoi is resilient enough to handle both hardware and software errors easily.

While not particularly complex, a BarBoi will take a significant amount of time and money to complete. Version 1 was worked on over 6 months and costed about 500 dollars in materials alone. Technical knowledge is NOT required for this project, but it certainly can't hurt!

3 Hardware Design

The hardware aspect of the BarBoi can be split into sections as follows: the housing (big wooden apparatus), the tubing (starts at the bottle and ends right about the drink), and the circuitry. Each of these sections needs to be complete AND integrated with one another before the system will work.

3.1 Materials

A list of all the major materials needed for one nine-bottle BarBoi is shown in Table [1]. Future revisions may feature more than nine bottles, as there is theoretically space for 25 (although 25 solenoid valves and 50 sets of barb connectors is quite expensive).

It is important to note that this project will cost much more than just the raw materials if you do not already have the tools to handle them. Woodworking materials are essential to this project, and while the

Amount	Material	Vendor
9	12v Solenoid Valves	Amazon[3]
1	Raspberry Pi 3b+	Amazon[6]
9	N-Channel MOSFET	Amazon[5]
1	12v Power Supply	Amazon[2]
4	2'x3' Wood Planks, 6ft	Home Depot
1	$\frac{3}{4}$inch Plywood Sheet, 2'x 6'	Home Depot
4	$\frac{3}{4}$inch Plywood Sheet, 2'x 1.5'	Home Depot
2	$\frac{3}{4}$inch Plywood Sheet, 2'x 2'	Home Depot
9	Bottle Cap to $\frac{1}{4}$inch Hose Adapter	Amazon[4]
18	$\frac{1}{4}$inch Hose Barb to $\frac{1}{4}$inch Male Pipe fitting	Amazon[1]
2	$\frac{1}{4}$inch Tubing, 50 ft	ACE Hardware
9	$\frac{1}{4}$inch Tee Joint	ACE Hardware
9	Small Screw-Hooks	ACE Hardware
9	Soda Bottle	Safeway lmao
1	L-bracket, 20ct	Amazon

Table 1: This table lists all of the major materials we used to complete this project. Please note that tools used as well as consumables like nails, solder, and wiring are not shown here. Vendors with superscripts provide links to items that are available easily online.

circuitry could be done without a soldering iron, our personal experience says that the final product should be soldered to avoid complications in debugging. This pushes up the price of a single BarBoi significantly, but the tools can be used many times. Consumables like electrical tape, wire, nails, and solder are not listed for simplicity's sake, but were used in some quantity in the final build. These will likely be bought in a quantity that is far greater than needed for one BarBoi.

3.2 Housing

The housing itself is 6 feet tall, and 2 feet square. It features a sturdy ceiling and a second shelf at around table height. I would recommend that the housing is built before any other pieces are attempted, as having a permanent place to store finished sections was handy. Figure [ ] shows a render of the housing without anything in it.

When it comes to the actual construction of the housing, exact rules do not need to be followed. The only important part is that the final housing is strong enough to suspend all nine of the bottles in the air (full of liquid) and not be unsteady. I attached the horizontal plywood to the 2'x3' legs using l-brackets and strengthened the system with the plywood crossbars that can be seen on the top and bottom of the housing. Most of the wood is not glued, with only nails or screws to join individual sections. I then finished by sanding and applying a coat of water-resistant paint, though these steps could probably be skipped if the wood is nice enough by itself.

3.3 Tubing Connections

Each of the 9 bottles included requires the hosing, adapters, and solenoid valve to be in a specific arrangement, shown in Figure [ ]. The tee joint is required to let air enter the area below the valve when the valve is closed, as otherwise the difference in pressure will not allow the liquid to finish flowing. To solve this problem above the tee joint, each of the holding container bottles has a hole drilled in at the top (bottom if being held normally) for airflow.

3.4 Schematic

The circuit pictured in Figure [1] shows how each 12 Volt solenoid gets powered by a 3.3V output from the Pi using an external 12V source. Each bottle needs its own control MOSFET, so the individual circuit needs to be soldered 9 times, in parallel with respect to the 12V power source.

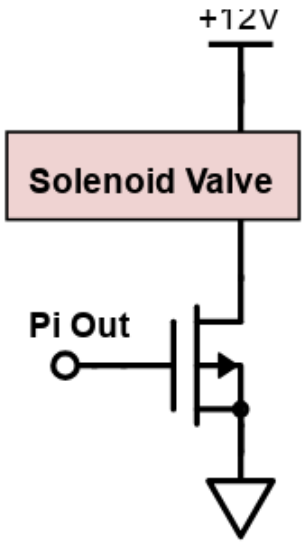

Figure 1: This is the individual driver circuit for each solenoid. While each Pi out needs to be discrete, the +12V and GND may all be tied to put all of the solenoids in parallel. Figure made in SchemeIt[7].

One issue we needed to fix was that the solenoids actually pull so much current when they are turned on that the cheap 12V wall converter could not open more than three at a time (remember, they are wired in parallel). This prompted us to add in a segment of code that separates each recipe into sections where no section has more than three solenoids open at a time. This code could be avoided altogether by using a more potent power source or by using two or three of the cheap ones in parallel (we didn't test this).

4 Software Design

4.1 Introduction

The software design of this project consists of a few main components: the front-facing web server that users can use to order drinks, the script the server calls that opens and closes the physical valves to pour the drinks and the 'recipe grammar' that was created to pass information between the server and the script in a consistent and easy to parse manner.

4.2 Web Server and Concurrency

The software design of the device consists of two parts, the server and the client. The server does all the hardware triggering and listens for requests from the client, which sends GET requests to the server whenever and order is placed. As of this version, the server and client are officially distinct so the server can act as an API endpoint so other devices can send signals to it very easily. The site has a few pages, a main page for normal operation, a 'mystery shot' page for an experimental mode of use, and an 'admin' page for more subtle control over what is being poured (opening single valves instead of a set).

Obviously, matters of concurrency are important here since if two people order drinks simultaneously, but our system does not have a native way to detect that and stop the two drinks from being mixed together.

However, our use case is so small in terms of number of users that this should not present itself to be an issue as long as the users are ordering drinks while looking at the device itself.

4.3 Recipe Grammar

The 'recipe grammar' is used by the server and handler to communicate messages of what drinks to pour, taking the shape of well-formatted .csv files. The format of which is the following:

p0 p1 p2 ... pN
e1 e2 e3 ... eN

where pX is in the set of pins that are set up to pour drinks, and eX is in a OUNCE, RANDOM where a is some float greater than 0.00 with two maximum decimal points, and RANDOM is some random a less than 5.

4.4 Recipe Handler

The recipe handler takes in a drink name (i.e. 'cosmo') and first checks to see if a recipe for that name exists in the directory of recipes stored in the project files. Then, it loads the .csv for that corresponding recipe and parses the data and sends power to each pin pX for e1 * 3 amount of time (in seconds) since our tests indicated it took 3 seconds to pour one ounce of fluid. It does this in a multi threaded manner to help improve overall pour times since pouring 4 fluids at once is much faster than doing them one at a time.

5 Acknowledgements

This project would have stayed only a dream of mine (Corey's) without the help and support of my friends and family. I would like to first thank Drew Ehrlich, who put in countless hours arguing with me (for good reason, I have many stupid ideas) and doing a vast majority of the code himself. Without his input this project would never have taken off the ground. I also enjoyed working physical labor with my friends Nick Adair, Steaven Ballesteros, Justin Hinman, and Danny Li, all of whom helped with the construction of the housing. Finally, a huge shout-out to the residents of 149 Palo Verde Terrace, who allowed me to build and house this project over the course of many months in our living room. I cannot understate how important everybody around me was to being able to bring the BarBoi to life.

References

[1] Amazon. *1/4in Barb Tips*. URL: https://www.amazon.com/Anderson-Metals-57001-0404-Fitting-Adapter/dp/B002SAO7XQ/ref=pd_rhf_ee_s_rp_c_0_1/132-4047080-4074111?_encoding=UTF8&pd_rd_i=B002SAO7XQ&pd_rd_r=0f86d8d0-4129-4fbf-86f2-7db6f04eac06&pd_rd_w=vlRhH&pd_rd_wg=4vTU9&pf_rd_p=779be473-677c-4bc8-b158-4c7b19dd7a44&pf_rd_r=QAKM60YA3TB4ZZVT5V61&psc=1&refRID=QAKM60YA3TB4ZZVT5V61 (visited on 02/19/2020).

[2] Amazon. *12v DC Power Supply*. URL: https://www.amazon.com/inShareplus-Mounted-Switching-Connector-Adapter/dp/B01GD4ZQRS/ref=pd_rhf_ee_s_rp_c_0_1/132-4047080-4074111?_encoding=UTF8&pd_rd_i=B01GD4ZQRS&pd_rd_r=94d39e70-b91d-4708-8312-e943780e3bb5&pd_rd_w=SqvTc&pd_rd_wg=0lnEE&pf_rd_p=779be473-677c-4bc8-b158-4c7b19dd7a44&pf_rd_r=CFVGDBDJXX7FMRSSXWKM&psc=1&refRID=CFVGDBDJXX7FMRSSXWKM (visited on 02/19/2020).

[3] Amazon. *12v Solenoid*. URL: https://www.amazon.com/Electric-Solenoid-Normally-Solid-U-S/dp/B00APDNPXG/ref=sr_1_8?keywords=12v+solenoid&qid=1582157552&sr=8-8 (visited on 02/19/2020).

[4] Amazon. *Bottle Cap Adapters*. URL: https://www.amazon.com/gp/product/B00RKM3JZ4/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1 (visited on 02/20/2020).

[5] Amazon. *N-Channel MOSFET, 10 Count*. URL: https://www.amazon.com/gp/product/B07CTF1JVD/ref=ppx_yo_dt_b_asin_title_o04_s00?ie=UTF8&psc=1 (visited on 02/19/2020).

[6] Amazon. *Raspberry Pi 3+ Starter Kit*. URL: https://www.amazon.com/CanaKit-Raspberry-Premium-Clear-Supply/dp/B07BC7BMHY/ref=sr_1_3?keywords=raspberrypi&qid=1582157245&sr=8-3 (visited on 02/19/2020).

[7] Digi-Key. *Scheme-It: Free Online Schematic Drawing Tool*. 2014. URL: http://www.digikey.com/schemeit (visited on 05/20/2019).