Alfred: Wifi-Enabled Automated Mixed Drink Maker

Patrick Barron, CSE, Chris Wong, CSE, Ben Ivaldi, EE, and John Fouad, CSE

***Abstract*—College towns and major cities are often filled with overpopulated bars. Alfred helps alleviate congestion while also expediting the process of buying a drink by allowing customers to order a drink through their mobile device. When a drink is ordered by the customer, that drink choice is sent to a controller, which then rotates a base holding 8 cups. The correct cup is rotated to a certain position where a drink is dispensed with the correct proportions, and then rotated again to the dispensing door. The customer will receive a personal identification code that he/she/they will be able to scan at Alfred, and the dispensing door will vend their drink to the customer. Alfred will pause its execution until the drink is removed by the customer, then continue its execution.**

I. Introduction

HAVE you ever been at a bar and faced the struggle of waiting in an overcrowded line to order a simple mixed-drink? Or have you ever complained upon receiving the wrong drink at a bar, or a drink that is too strong or not strong enough? Alfred will allow users to order simple mixed-drinks from their mobile device at the bar without having to wait for the bartender or having to deal with pouring errors or receiving the wrong drink.

There have been inventions [11, 12] that are somewhat similar to our design but none of those have the fully automated functionality that Alfred contains. Also, similar designs that are used in society today such as the "Coca-Cola's Freestyle are available on lease for $320 per month" which is very expensive compared to our $500 dollar budget to make Alfred [1, 14]. Our design will also save people time in crowded bars and in today's society time is very valuable to people. Our design will allow customers to spend more time with their friends, co-workers, family etc. instead of wasting their time waiting to order a drink which will increase their enjoyment of the bar.

Our requirements were developed based on the size of our design and how Alfred will be able to be placed on a bar and the bartender will only have to insert cups of ice and bottles of the drink choices into the machine periodically. Alfred will be powered by plugging the system into a standard outlet. The requirements are also geared towards the customers as they will have to wait limited time to receive a drink and also can order from their mobile device. Table I shows a list of specifications.

Table 1: List of requirements and specifications

Requirement	**Specification**
Pour a mixed drink	Correct portions and completed in under 2 minutes
Multiple drink options	Bartender can insert choice of alcohol (750mL) and mixers into dispensers. Choice of 4 different drinks
Minimizes spilling	Spills less than 5% of drink
Online ordering	User orders through mobile website
Minimizes transaction costs	Tab system for ordering
Drink served to correct customer	Serving door only opens when id code is presented
Simultaneous pouring of liquids	15.9" rotating base with 8 cups of ice and 6 pumps
Failsafe: detects positioning of base and cups	Sensors to make sure cup is removed before closing the serving door. Sensor used to align base

P. Barron from Bridgewater, MA (e-mail: pbarron@ umass.edu).
C. Wong from Norwood, MA (e-mail: cmwong@ umass.edu).
B. Ivaldi from Bridgewater, MA (e-mail: bivaldi@ umass.edu).
J. Fouad from Framingham, MA (e-mail: jfouad@ umass.edu).

II. DESIGN

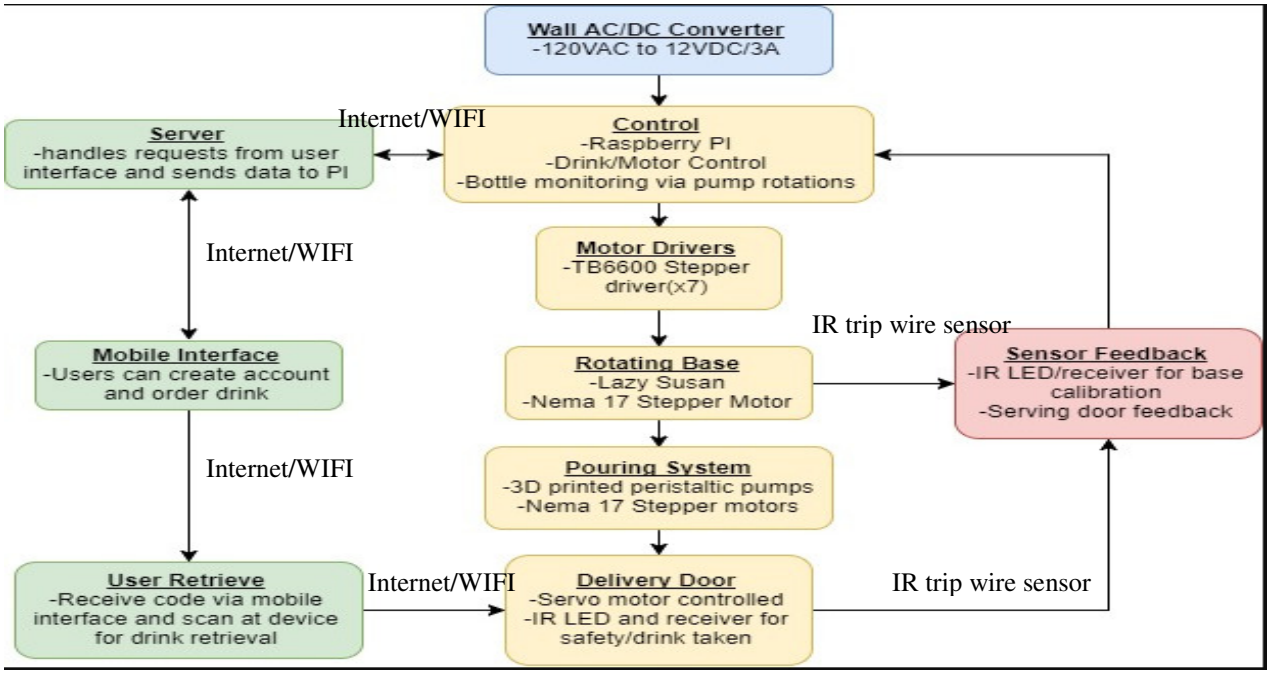

Figure 1. Block Diagram

A. *Overview*

For Alfred to function properly, it will need multiple motors and controllers. The use of an Arduino and a Raspberry Pi were needed to get the Alfred to dispense one beverage with the correct proportions. An "order" button is pressed, which then initiates Alfred to begin creating a drink. The code on the Arduinos will be transferred to the Pi. Three stepper motors were used for the prototype: two for pumping the liquids and one for rotating the base. Four more stepper motors will be added, as there will be six pumps pushing liquid into cups, and still the one motor rotating the base carrying the cups.

A serving door will be implemented that has an IR-LED transmitter-receiver connected to each side – this will let the system know when the cup is removed from the rotating base so that the serving door can shut again and execution of the machine will continue. There will also be an IR-LED transmitter for the rotating base to make sure the base is calibrated in the correct position i.e. under the correct pumps. The user will be able to select a specific drink that they want made, and Alfred will dispense the correct drink out of four possible drinks. User will receive a specific identification number (barcode, QR Code, etc.) that they will be able to scan and receive their drink.

According to figure 1, the power coming from the wall outlet goes into the Raspberry Pi. From there, the Pi controls the motor drivers which in turn control the stepper motors and servo motors responsible for the rotating base, the pouring system, and the delivery door. The Pi also communicates with the server which handles requests from the user (mobile) interface, where users can select the specific drink they would like to order. A code will then be sent to the user which will allow them to retrieve their drink. Once the delivery door opens, and IR-LED/receiver will let the Pi know when the cup has been removed, allowing it to close. Once a drink has been successfully retrieved by the user, Alfred will continue making drinks and the process starts all over again.

Other alternatives we considered were Chinese peristaltic pumps, but their flow rate was too slow [13]. We also looked into mechanical shot fillers that perfectly pre-filled a shot, but we would have to create another mechanical arm that pushes the shot filler up.

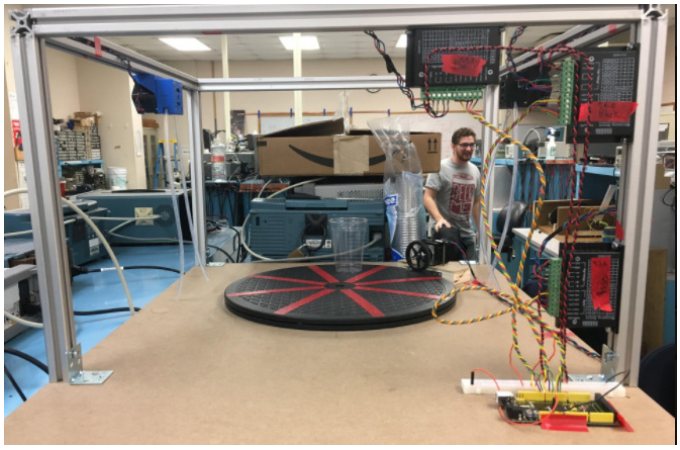

Figure 2: Alfred back view

B. *Rotating Base and Calibrating IR-LED Transmitter*

The rotating base is a Lazy Susan that is screwed into a wooden platform. A stepper motor driven by a motor driver has a rubber wheel attached to the end of it, and when the motor rotates, the wheel rotates [3, 4]. This wheel is placed on the base. As a result, the base rotates synchronously with the wheel

with but with a larger turn radius. Code written on the Arduino is controlling the rotating base to move to certain positions based on where we want a cup to be rotated to. Specific drink nozzles are located in certain areas, so the base rotates a certain amount of steps to a specific position to receive the correct liquid. The calibrating IR-LED transmitter will ensure that we know exactly where the base is at all times [6]. This transmitter will be aligned after each rotation so error does not accumulate from minute degree inconsistencies in the rotation of the base/stepper motor. Once the IR-LED transmitter is lined up, we can continue rotation of the base knowing the exact location of each cup. Knowledge learned in Software Intensive Engineering and Computer Systems Lab 1&2 allowed us to control the Arduino, control I/O, and control the motors. We will implement a test and check method that ensures the correct functions of the IR-LED transmitters and that the rotating base moves to the correct position via the shortest path.

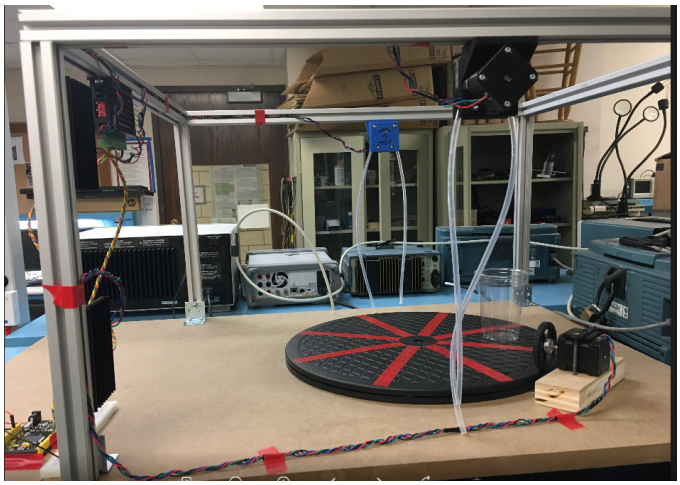

Figure 3: Alfred side view

C. *Serving Door and IR-LED Transmitter*

A serving door will be at the forefront of Alfred which will open up once a customer scans their personal identification code. This door is in the shape of a cube with two faces missing (see Figure 4). If one was looking directly at the front face of the cube, the bottom face and the back face will be the ones missing. The cube will be rotated back 90 degrees with the help of a servo motor, so that the front face is now on top. This "cube" rotates around the final cup position on the base, which will allow the customer to reach and in and grab their drink without being able to access the other drinks in the machine. In the two side faces of the door, there will be another IR-LED transmitter that will let us know whether the completed cup has been removed or not. Once the system communicates that the cup has been removed, the serving door will be shut and Alfred will continue execution. Knowledge obtained from Software Intensive Engineering and Computer Systems Lab 1&2 will allow us to control the serving door. The servo motor and the communication from the LED transmitters will be controlled by the Raspberry Pi. We will have to take measurements to make sure the cube fits over the existing cup but does not hinder the rest of the machine (other cups/equipment).

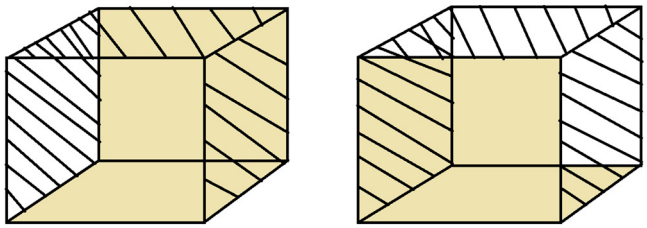

Figure 4: Image on the left portrays the serving door with the bottom and back faces removed (colored in tan). Image on right is after it is rotated 90 degrees, with the front and bottom faces removed.

D. *Pouring System*

This subsystem of Alfred is responsible for the liquid distribution. The main design goals of this system are speed, accuracy, and affordability. In order to meet these goals we used an open source 3D model of a peristaltic pump, designed by silisand[1],[2] that is driven by a Nema 17 stepper motor [3]. By doing so, we were able to utilize the unique properties of these types of pumps, without paying the high market cost for them. These allow for high precision liquid distribution, without having liquid touch any mechanical or electrical components.

Because these motors use a significant amount of power, they each require a motor driver. The drivers chosen for these motors was the TB6600 [4]. These drivers allow for variable micro-stepping if needed, but we found that the normal 200 step per rotation of the motors themselves was sufficient.

In addition to the drivers, the pouring system will also require 3.3V to 5V logic level converters for each driver. This is because the drivers require 5V signals to operate, while the Raspberry Pi I/O pins have a high level of 3.3V. Because of this, the logic level converters will need to be designed, which will be accomplished by making a PCB of transistor amplifiers. The knowledge gained for Electronics 1 and 2 will be sufficient in designing this, and the knowledge gained from Computer Systems Lab 1 will allow for the digital control of the motors.

After printing, building, and wiring our preliminary pumps, we found that they did not have enough holding strength for the amount of liquid required for our design without dripping. Since dripping liquid into cups without command is not desired, and could be a hazard, we needed to solve this. By implementing code to run the pumps backwards after distribution to the designated cup was complete, the tube line is cleared and no liquid is available to drip into the cup.

Overall, the use of these printed pumps in conjunction with the drivers have allowed for quick and accurate distribution of liquid. In comparison to the initial Chinese peristaltic pumps we bought, which had a flow rate of about 2.7 fl oz/min, our printed pumps have a flow rate of about 12.5 fl oz/min, which greatly improves our production time [13].

E. *Mobile Interface and User Retrieve*

The mobile interface for Alfred will consist of a site that is

accessible from a smartphone and enables the user to create an account. Upon creation of the account the user will then be able to place orders from a choice of 4 mixed drinks. From there the users order will be sent through the control system and initiate the drink making/pouring process. When the users drink is complete, they will then receive a notification with a code and can proceed to use their code to retrieve their drink.

The website will be created using HTML, JS, and CSS code for the user interface side of the website. This code will allow users to create an account and store link their account with the sites database so they can proceed to order drinks. For the ordering itself, code will be used to form a selection menu where the user can click which drink they want and then click a separate place order button. For MDR we implemented a simple "do" button to initiate the drink order, but for future implementation we will improve upon that and make our ordering platform through the mobile interface more user friendly.

To go about completing the mobile interface we will use our gained knowledge from Software Intensive Engineering with regards to coding in HTML and also making websites user friendly. Going forward we will continue to learn more JS and CSS to improve the interface and also learn more about working with the front-end and the database to store the users account information. To design and test the mobile interface we will check our code files on a web browser to see exactly how the site looks and also we will test the ordering and account portions by going to the site and placing orders and creating accounts respectively. To analyze these results we will make sure the website looks nice on the mobile device and is capable of creating an account and then placing a drink order.

Upon completion of the drink being poured, the user will receive a personal identification code to then go and retrieve their drink. To use this block we will start by sending the user a personal code that is a number sent to their phone and then the users will be able to use that code to go up to Alfred and get their drink. From there we would like to advance our project to send the user a unique QR code or bar code that they will be able to scan when they go to retrieve their drink. To implement that we will have to build upon our knowledge in image processing from Computer Systems Lab II. To do this we will have to create code to generate a unique code upon the users order that will be sent to the user when their drink is complete. Also we will have to have a scanner that can read in the QR/bar code when the user presents it to retrieve their drink and that code will have to be processed and matched with the code of the drink that is being made by the control system. Once those codes are matched properly, the drink will rotate to the door position and the door will open for the user to get there drink. We will test and analyze this block by running the code and making sure it can recognized the matching codes to pair the order to the drink that's being made.

F. Server and Client Block

The purpose of this block is to allow for communication between each of the major components of Alfred to control the system as a whole. For quick implementation and testing, CPython was initially used to prototype the server and client [7]. However, with the scaling of the system to include multiple processes, the CPython was transcribed to C since CPython's Global Interpreter Lock prevents threads with shared memory from operating in parallel. The server and client use sockets and TCP to communicate via the internet. The server will be hosted remotely with another company and the client will run on a Raspberry Pi 3 Model B [9]. The Raspberry Pi's Broadcom BCM2837 ARM Cortex-A53 is a 4-core, 2-way superscalar processor which should enable it with the ability to run multiple process threads at the same time [8, 10]. This is useful since the complexity of maintaining communication with the server and processing the information to control the system will require a multithreaded solution for the fastest performance.

The server and client block is very software heavy. Therefore, there were many techniques that were used from the more software-based classes. Data Structures and Algorithms was useful for determining how the data should be saved and processed in an efficient way. Introduction to Computation and Cryptography aided in the selection of a mathematical ring using modular arithmetic to provide a mathematical model that the client could interpret. Computer Systems Lab I and II helped with mapping the operating system's address space to the physical addresses that control the general purpose input output pins on the Raspberry Pi's board. Computer Networks and the Internet assisted in understanding and implementing the internet protocols. Software Intensive Engineering helped with controlling multiple thread resources so that there are no errors or deadlock between the process threads. Computer Architecture aided with choosing a processor that would fit our needs for the project.

While the prior knowledge that we had helped with this block, we still needed to learn specific hardware-software interactions for the Raspberry Pi. Also, we need to learn more about developing a solution that could be ported to industry-scale tools. For example, hosting our server on someone else's server.

To design and test this block, the software will have to be run under realistic situations and workloads. The remote server will need to serve webpages and communicate with a reasonable number of clients. Also, the client will need to handle a reasonable number of drink requests. In order to test the performance, we will put the system under realistic to heavy workloads and then measure the system's response time, utilization, processing time, and task completion time. All of these metrics can be analyzed to see if the system is performing better or worse. Lower response times, processing times, and task completion times mean that the system is performing better. Utilization is harder to analyze. While it is preferred that the system is being utilized a lot, which could mean that the system is working to the maximum potential, having too much utilization could result in overworking the system and causing rapid deterioration of the hardware, and it could also mean that there is a bottleneck in the system.

III. Project Management

MDR Goals	
Deliverables	**Value**
Pumps will pour correct amount of liquid into cups within one minute	100
Rotating Base will rotate the cups to their correct positions	100
Server will be implemented and can communicate with clients	100
Client will be able to order one drink and communicate with the server	100
Website will be implemented	100
Serving Door will open when it receives confirmation code	0

The majority of our Midway Design Review goals were reached. Initially, we planned on having multiple different parts working independently. However, it was advised that we focus on integrating all of the essential parts than implementing some of the more auxiliary parts. Therefore, for the prototype, we integrated communication between the browser client, server, Raspberry Pi, microcontroller board, and motors, thus allowing us to send a message from a user to the rest of the system where it is translated into mechanical movements and a mixed drink is produced. So, the client can make an HTTP GET request to the server and receive a webpage in response. Also, the client can make an HTTP POST request to the server to order a drink, and it is processed and sent to the Raspberry Pi, which sends a signal to the microcontroller. The microcontroller then interprets this signal to tell the motors to rotate the base and pump the liquids to successfully pour a mixed drink.

Due to the fact that we focused on integration rather than total functionality of all of the aspects of the design, the serving door, feedback sensors, and personal identifier systems still need to be implemented. In addition, the system needs to be scaled up to support 8 simultaneous drinks at the same time, the server needs to be moved off of the local host to a remote host, and the microcontroller needs to be removed and integrated with the Raspberry Pi.

The team works as a cohesive unit. Each team member has a specialty that they can apply to specific subsystems, and there is enough overlap in knowledge between each team member, which promotes joint solutions to problems that occur in any given subsystem.

John Fouad and Benjamin Ivaldi's specialties regard mechanical, power, sensor, and microcontroller devices. Patrick Barron's specialty lies in frontend development. Christopher Wong's specialty lies in backend development.

This collaboration is aided by the open communication within the team. Often, there are daily discussions between group members with weekly team meetings. In addition, there are weekly meetings with our advisor scheduled to check our progress and ensure that the project is progressing forwards.

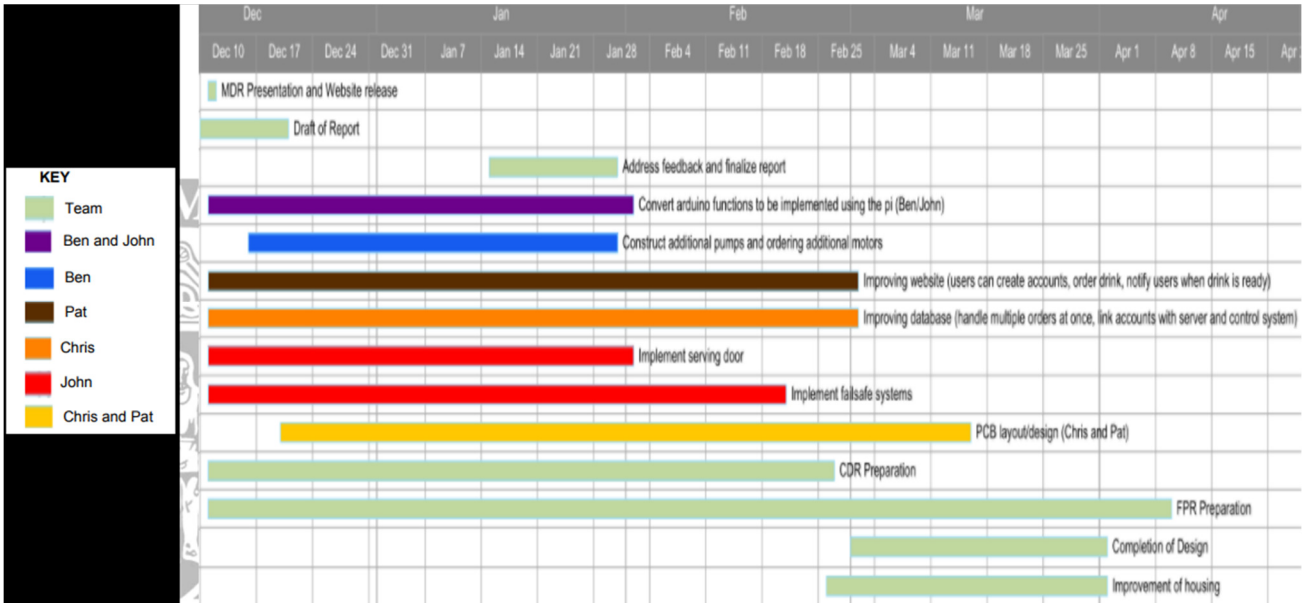

Figure 5: Gantt chart of progress

IV. Conclusion

Currently, the essential systems of Alfred have been implemented, which satisfies our MDR deliverables. We have two complete pump systems integrated, the rotating base, and basic user-device communication via website. As of now, Alfred is able to receive a virtual button click on our website and then make one drink by rotating a cup to the two locations under each pump, and then rotating the cup to the dispense position.

The next phase of Alfred is to scale-up and finish the pump system, implement the serving door, and make the website/server be able to intake a personalized drink order.

The scale-up of the pump system will be fairly straightforward. Four more pumps need to be 3D printed, assembled with bearings, and mounted. Each pump will need a driver which will be connected to the Raspberry Pi for control. In order to communicate from the Raspberry Pi output, 3.3V, to the pump motor drivers, 5V, a logic level converter will be required. This is intended to be our PCB design and will require 7 amplifiers: 6 for the pumps and 1 for the base motor.

Implementing the serving door will require the construction of the door, an infrared LED and sensor, and a servo motor. The Raspberry Pi will need to use a PWM signal to control the orientation of the door and also an analog input from the infrared cup sensor.

The website will need to allow users to create a personal account, order drinks with variable proportions, and send a personalized retrieval code for when their drink is ordered. Implementing these functions will in turn require Alfred to plan the drink mixing in an efficient way, regardless of the number of orders placed.

These goals will be accomplished by staying focused on our assigned tasks and staying on schedule. Group communication is an essential part of achieving our goal, and if we stay on our current motivated pattern, these goals will be accomplished.

References

[1] Pullen, John Patrick. “5 Technologies Changing the Restaurant Industry.” *NBCNews.com*, NBCUniversal News Group, 9 Sept. 2012, www.nbcnews.com/id/48959179/ns/business-small_business/t/technologies-changing-restaurant-industry/#.WjsylFWnHIV.

[2] Thingiverse.com. “Peristaltic Pump Improved for Nema 17 by Silisand.” *By Silisand - Thingiverse*, www.thingiverse.com/thing:1134817.

[3] Clifford, Paul. “Stepper Motor Specifications, NEMA 17 1.8 Degree 200 Steps-per-Revolution Four-Phase Unipolar Permanent-Magnet Stepper-Motor.” Find Controllers for Instrumentation and Automation at the Mosaic Industries Site, Mosaic Industries, Inc., www.mosaic-industries.com/embedded-systems/microcontroller-projects/stepper-motors/specifications.

[4] “TB6600 Stepper Motor Driver SKU: DRI0043.” DFRobot, 28 June 2017, www.dfrobot.com/wiki/index.php/TB6600_Stepper_Motor_Driver_SKU:_DRI0043

[5] “Ks0002 Keyestudio Mega 2560 R3 Development Board.” Ks0002 Keyestudio Mega 2560 R3 Development Board - Keyestudio Wiki, Keyestudio, wiki.keyestudio.com/index.php/Ks0002_keyestudio_Mega_2560_R3_Development_Board.

[6] “IR Receiver Modules for Remote Control Systems.” Vishay, www.vishay.com/docs/82491/tsop382.pdf.

[7] “Python/C API Reference Manual.” Python/C API Reference Manual - Python 2.7.14 Documentation, Python, docs.python.org/2/c-api/index.html.

[8] “ARM® Cortex® -A53 MPCore Processor.” ARM, docs-api-peg.northeurope.cloudapp.azure.com/assets/ddi0500/g/DDI0500G_cortex_a53_trm.pdf.

[9] “Raspberry Pi Schematic.” RaspberryPi.org, www.raspberrypi.org/documentation/hardware/raspberrypi/schematics/Raspberry-Pi-3B-V1.2-Schematics.pdf

[10] “ARM® Cortex® -A53 MPCore Processor.” ARM, infocenter.arm.com/help/topic/com.arm.doc.ddi0500d/DDI0500D_cortex_a53_r0p2_trm.pdf.

[11] Jamiep, and Instructables. “Build a Mobile Bar - BaR2D2.” Instructables.com, Instructables, 8 Nov. 2017, www.instructables.com/id/Build-A-Mobile-Bar-BaR2D2/.

[12] Lomas, Natasha. “Barobot Is A Hackable Cocktail Mixing Robot.” TechCrunch, TechCrunch, 20 May 2014, techcrunch.com/2014/05/20/barobot/.

[13] “Dosing Pump12V DC Peristaltic Liquid Pump Hose Pump Dosing Head for Aquarium Lab Analytical Water (Green).” Amazon.com, www.amazon.com/dp/B01HRPKBAE/ref=sspa_dk_detail_1?psc=1&pd_rd_i=B01HRPKBAE&pd_rd_wg=HgS0y&pd_rd_r=H11V96JM160276XH0KWT&pd_rd_w=q4xRM.

[14] “Coca-Cola FreeStyle.” Coca-Cola, www.coca-colafreestyle.com/.