AUTOMATED BARTENDER
DESIGN DOCUMENT

March 10, 2017

Greg Wajda, Max Dribinsky, Austin Gram
UIUC
ECE 445

Contents

INTRODUCTION

Objective

The first image that comes to mind when imagining a crowded bar is a scene of blood, sweat, and tears. Fighting to attract the attention of the overwhelmed bartender, surrounded by sweating bodies and overwhelmed with a pounding bass. We began our project with a goal to optimize that process, to make ordering a drink akin to purchasing an item on Amazon with two-day shipping, with similar guarantees of quality and safety upon delivery.

Taking matters into our own hands, we brainstormed how we could improve the security of the exchange that takes place at a bar when ordering a beverage. Essentially, the process of ordering a drink can be broken down into the following steps:

1. Customer attracts the attention of the bartender
2. Bartender begins preparing the order while customer continues socializing. Note that this step requires complete trust in both the attention and good intentions of the bartender and those around him/her.
3. Drinks are ready. The customer pays the bartender and leaves with the drinks.

In that process, we noticed that the second step provides much more vulnerability than necessary. Our project seeks to resolve and mitigate those vulnerabilities so that the process is streamlined and secure. As the title of our project implies, on a high level, we are designing a machine which can operate in place of a bartender. The machine will have the ability to pour a mixed beverage in any proportion, combining two separate drinks to the user's liking. However, by protecting the beverage until the moment it is prepared and the customer has authenticated themselves, the user can maintain full confidence that the drink is safe and delicious. Our goal is for this machine to be more efficient, safe, and reliable than a traditional human bartender would be. We seek to accomplish this with a heavy emphasis on security and it is our hope that in doing so, we will further aid the movement and societal shift to prevent sexual assault.

Background

After conducting some research, we saw that the problem of automating the bartender has been solved before. It was already an item on the market that users were purchasing and

loving. However, there was only one thing missing, which turned out to be the most important part: security.

What began as a concept to improve the ordering process at a bar has transformed into an idea which aims to protect campuses and bars across the world. We focus heavily on security to prevent anyone who is not permitted to interact with the drink as it is being prepared.From our observations and experiences on this campus, alcohol is an enormous part of life for a significant percentage of the student population. With the current standards in place at bars on campus, the threat of a predator slipping an unwanted addition into someone's drink it too large to be ignored, and the inconvenience of ordering drinks is significant enough to take away from the enjoyment of bargoers.

High Level Requirements

1. **Security:** Drink should not be accessible to any customer until the drink is fully prepared, and the customer who placed the order has authenticated him or herself.

2. **Efficiency:** Drink should be prepared and awaiting the user's authentication in 20 seconds or less from the moment it is ordered.

3. **User-friendliness:** User interface on mobile application delivers a straightforward and pleasant experience to the user, allowing him or her to order a beverage to their desired proportion.

Our implementation focuses directly on designing a machine which fulfills these three requirements. For the security, the beverage is only accessible to the outside world immediately after the user has authenticated, until the moment the door shuts. If a user authenticates but does not claim their drink within a few seconds, the drink is dumped. This may be an inconvenience to users if they are ready to pick up their order. However, we are confident that there exists a timing solution which will preserve the safety and security of the drink without making the experience frustrating for the user.

As far as efficiency, our second requirement, our program loops endlessly and never stops working. Although in the first version it will only be able to service one request at a time and will not have any sort of request queue, we feel that in a later version, this aspect would be straightforward to implement with threading and multiprocessing. More machines could also be added which could divide up the labor. There would be some sort of routing protocol to

forward the requests to the individual machines from some central server. Currently, however, this machine will be designed to serve one request at a time, which should be approximately as fast as an average human bartender.

Finally, our third requirement requires a flawless design of our mobile application. We will deliver a user-friendly and intuitive experience by offering the user several options. They will have a few choices between suggested beverage proportions, or have the option to customize their beverage, and choose any proportion they'd like. In a later version of the project there would be more options of beverages, and the mobile application would provide more combinations and mixes, as well as more customization if a user wanted to create their own mixture.

Commercial Product Vision

Currently, bars aren't scalable. The more people in your bar, the harder it is to supply the demand, and the harder it is to account for the safety of everyone. In commercial use of our product, we would expand the drink choices available by increasing the size of the conveyor belt and amount of drinks. We would create multiple pick-up areas that are still protected from outside factors to allow for multiple orders to occur in parallel. Customers would be able to purchase drinks using their app and wait to be notified that their drinks are ready. This step allows bars to scale because people will waste less time trying to get drinks and in turn have more time to enjoy with their friends, making them less likely to leave when the bar gets too crowded. It also increases safety because less people are standing together pushing against each other to get closer to the bar, and instead spread out naturally.

In addition, and what we find separates our project from most other automated bartender project, is our focus on safety. The drinks will never be exposed to the outside until the user is verified. In practice, this would be implemented at scale by scanners in every pick-up area mentioned above. In college bars, bartenders frequently line up eight drinks on the bar and pour them. Then they slide each one over individually, exposing them to every single person sitting at the bar. After that, the person who just bought the drinks hands them back to their friends, who are usually reaching over/around a crowd of people surrounding the bar. This method is extremely risky compared to simply going to a pick up point when notified.

We believe that our design can easily be scaled to commercial use. Whoever uses our design

will also be able to scale their business, drastically increasing safety of the bar patrons along the way.

DESIGN

Block Diagram

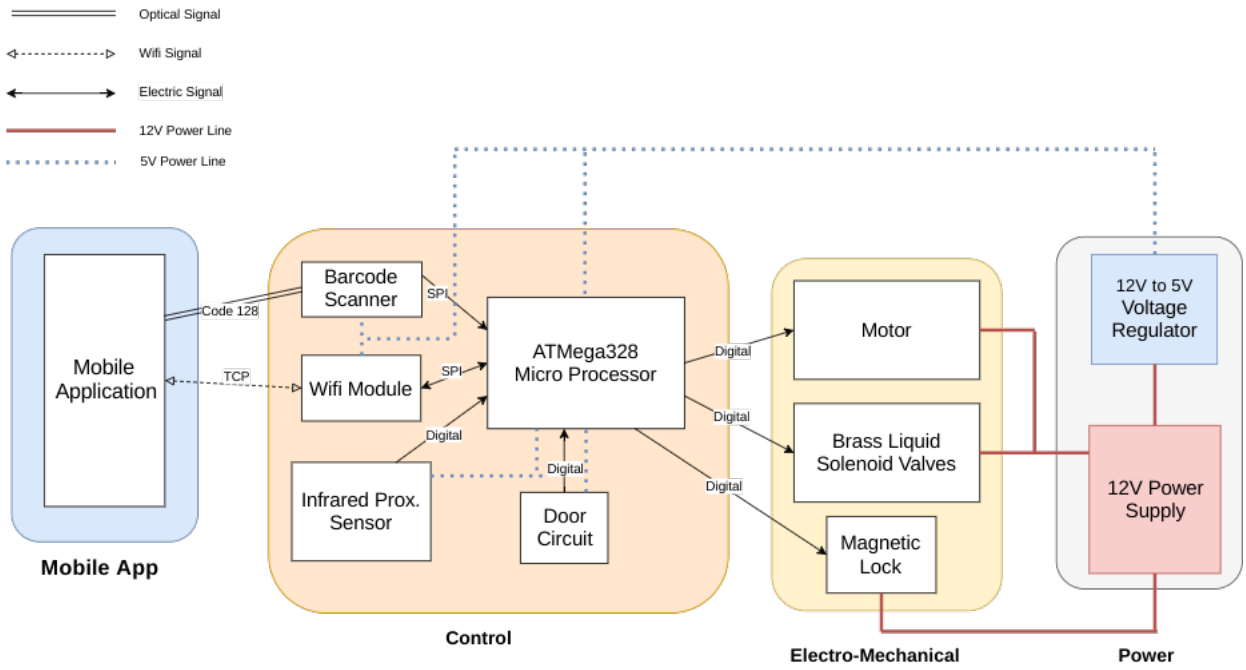

Figure 1: Block Diagram

Modules

Micro-processor: ATmega328	
Input	3 Bytes from WiFi Module, 128 Bytes from scanner, 5V±5% or 0 from sensor and door circuit
Output	Digital signals to various modules, see block diagram above.

The ATmega328 houses the control logic of the entirety of our project, outside of the user interface to obtain drink orders from the users. It has pins that allows for multiple protocols such as SPI, I^2C, USART, and PWM, which we used to formulate our requirements for the rest of our components.

WiFi Module: Adafruit ATWINC1500 WiFi	
Input	3 bytes (drink order) from Android Application via TCP
Output	3 bytes (drink order) to ATmega via SPI

In industry, we would want our users to be able to order a drink from anywhere in the bar. To accomplish this, we decided to go with WiFi for communication between the Android app and the ATmega328. This module uses SPI as its protocol, which meets our requirement because the ATmega can interact over SPI.

Barcode Scanner: Unitech AS10-P AS10	
Input	Barcode data (128 bytes) from Android Application
Output	Barcode data (128 bytes) to ATmega via SPI

Our project verifies the user's identity by utilizing a barcode scanner. By mounting it above the drink dispensing area, the user will be able to scan the barcode they receive from their app and grab their drink after the door is unlocked. As mentioned in the output, the barcode scanner outputs 128 bytes over SPI, which meets our requirement because the ATmega can interface with SPI.

UHPPOTE Magnetic Lock	
Input	12V±5% to lock, 0V to unlock
Output	No Output

This magnetic lock will allow us to control user access of the door without adding mechanical components, saving us time while doing just as good of a job as any other lock. The only logic this module requires is whether the lock needs to be on or off. This can be handled by our ATmega328 by utilizing one of its GPIO pins along with a transistor to control a 12V line to one of the lock's terminals, with the other one attached to ground (see figure 7 on page 15).

Brass Liquid Solenoid Valve - 12V - 1/2 NPS	
Input	+6-12V to open, 0V to close
Output	No Output

We will be using this valve for the dispensing of both of our drinks. It has two terminals which cause the valve to open when a voltage difference of 12V is applied, letting the liquid through. It doesn't have a gasket, so there is no minimum pressure required. This was necessary to meet our requirement because we do not pressurize the liquids in our project. It can still function down to 6V but it is slower to open. This can be handled by our ATmega328 by utilizing one of its GPIO pins along with a power transistor to control a 12V line to the valve, as well as a kickback diode to prevent damaging any parts (see figure 7 on page 15). A power transistor is necessary to handle the high current that this component pulls (3A @ 12V).

Infrared Proximity Sensor: Sharp GP2A200LCS0F	
Input	+5V±5%
Output	5V±5% without sensing, 0V when sensing something

This proximity sensor has a very short range (2mm - 22mm). It is exactly what we need to ensure that the cup is in the correct location for the pouring of beverages. This module has 3 pins: Vcc, Vout, and GND. Vout will be attached to a GPIO pin on the ATmega328. When Vout goes low to 0V, we know that there is a cup in place and to stop the conveyor belt.

Android Application	
Input	User drink input via UI
Output	3 bytes (drink order) to WiFi Module via TCP

Our android application has a simple UI that allows the user to order a combination of two drinks with different ratios. It communicates over the local WiFi network to send data in the form of 3 bytes to the WiFi module. The WiFi module then relays it to the ATmega328.

12V 7A Power Supply	
Input	120VAC @ 60Hz from wall socket
Output	12VDC with 7A max (84W)

Our power supply will need to supply 12VDC at 7 Amps. We will use a voltage regulator to provide a 5VDC line as well. See our calculation in the Supporting Material section to see how

we came up with the power requirements.

Door	
Input	No input
Output	+5V±0.25 when door is closed via completed circuit

The door will have a latch that will allow it to open and close. It will have the magnetic lock attached to it. It will complete a circuit when closed and will be an open circuit when open (Figure 7 top left). This circuit will be attached to a GPIO pin on the ATmega328 to notify when the door is open.

Conveyor Belt Module - Motor and Driver

Stepper Motor	
Input	12V differential controlled by driver.
Output	Movement of conveyor belt

Bipolar Stepper Motor Driver - A3967SLBTR-T	
Input	5V logic, 12V load, PWM from ATmega
Output	Stepping control to motor via motor's 4-wire cable

Our stepper motor and driver allow us to control the conveyor belt using the ATmega. By hooking up our driver to the ATmega's PWM pin, we can vary the speed of the conveyor belt to accelerate and decelerate the cups into the correct position in front of the door without them falling over. The driver circuit will utilize the A3967SLBTR-T chip in an adaptation of Sparkfun's EasyDriver - Stepper Motor Driver circuit (Figure 2) by eliminating unnecessary components such as the step down from 12V to 5V because we already have a voltage regulator, or the handling of 6 & 8 wire stepper motors.

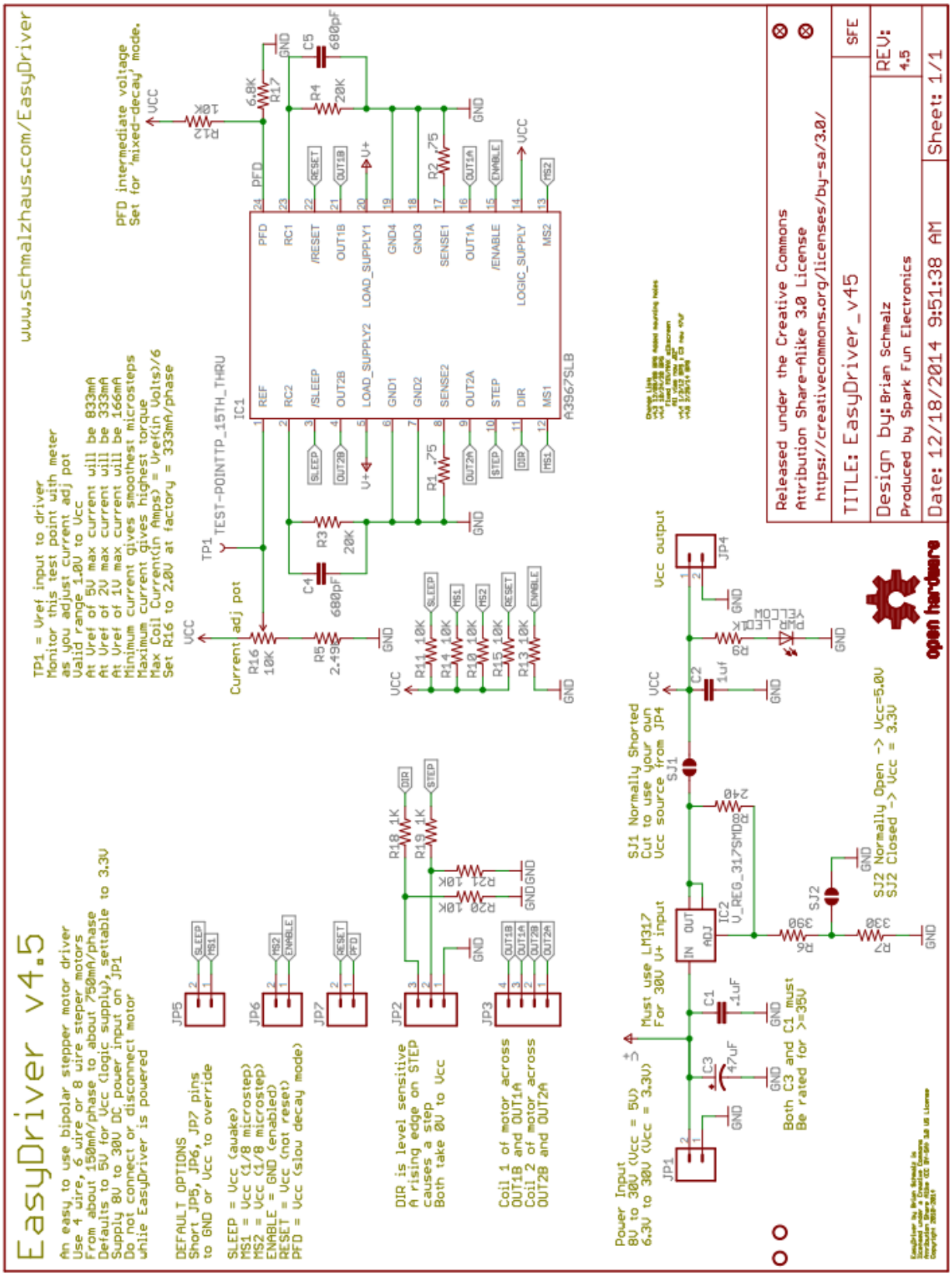

Figure 2: EasyDriver - Stepper Motor Driver schematic from Sparkfun

Physical Design

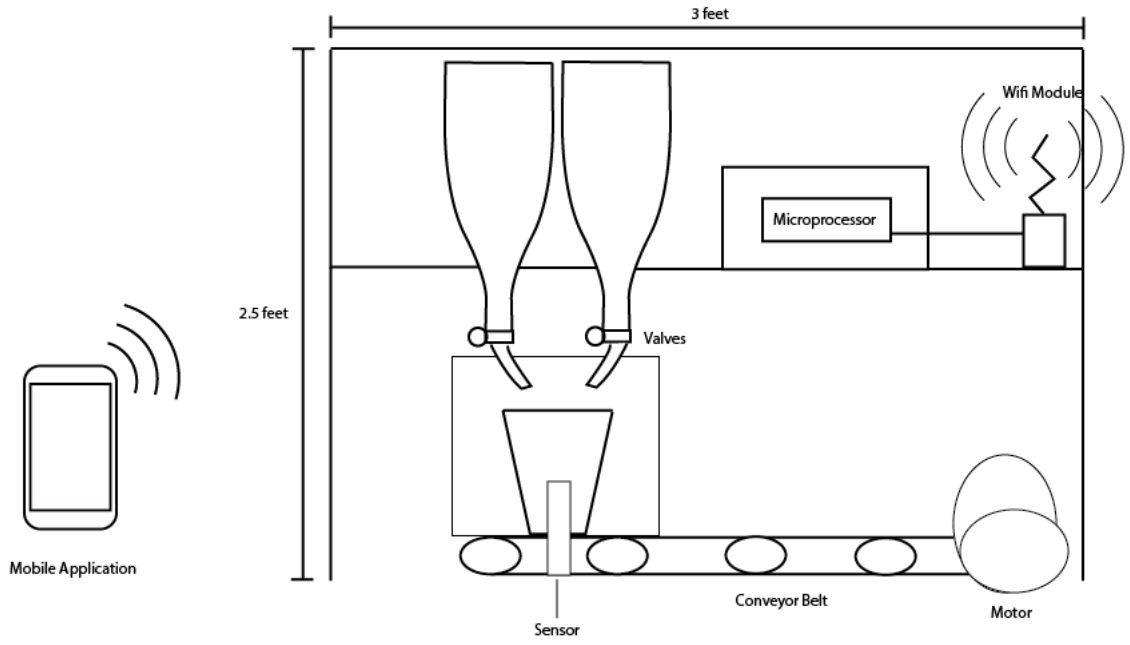

Figure 3: Back View

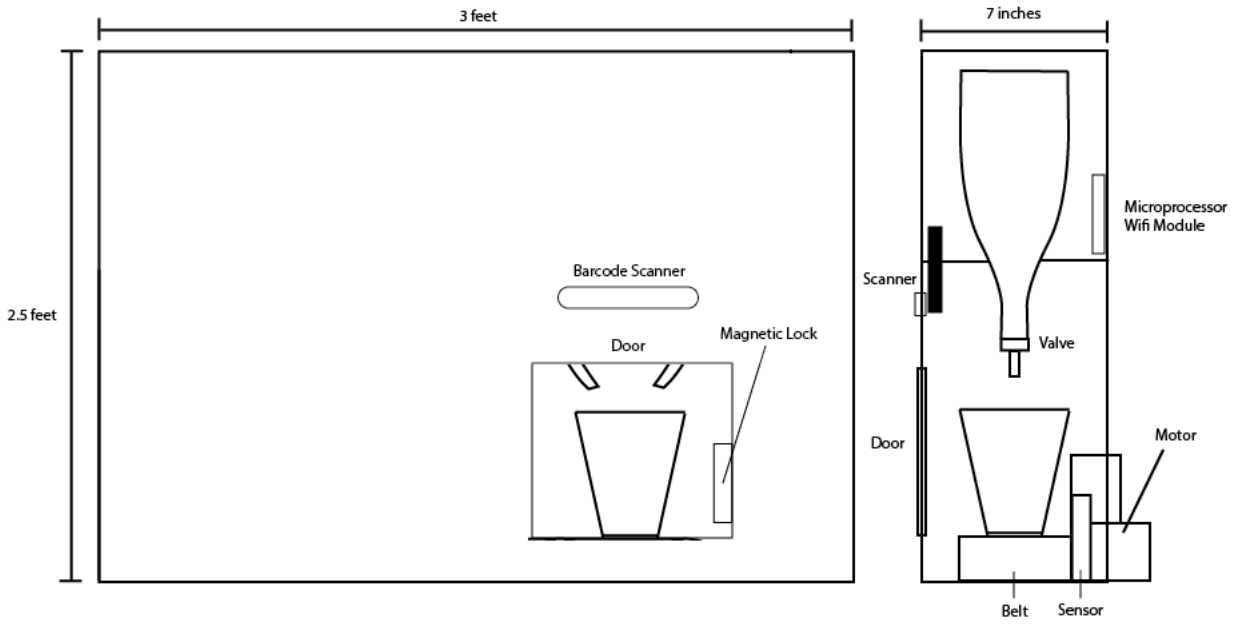

Figure 4: Front and Side View

While somewhat cartoony, these figures get the general idea across of the physical design of our project. This design abstracts the power supply away to make it more clear. See our block

diagram (Figure 1) for how power is handled.

Our project is only concerned with the dispensing of drinks and authentication of the user. In Figure 3, the back view can be seen, where we utilize a conveyor belt driven by a motor to transport the cup. The cup will eventually be detected by the sensor in the pouring area, indicating that it is in the correct spot and ready. The drinks will then pour. The view that the user sees when picking up the drink is on the left side of Figure 4, or the front view. The user can see very little of the inner workings of our project besides the things immediately surrounding the door.

All communication with the user is done via the Mobile Application's interaction with the ATmega microprocessor with the WiFi module as the middle man.

We assume, for the purpose of a 1 semester project, that empty cups will always be loaded on the conveyor belt, and the user will always take the cup. In industry, we would ideally have the cups being put onto this belt after being washed or prepared in some way. If the drink is not claimed by the customer, the drink will be moved off to the left (of Figure 3) to the disposal for unused drinks, which is also abstracted away for us. Our design would be able to scale for industry by increasing the size of the conveyor belt, allowing for more drinks, valves, and drink pouring locations marked by sensors.

Functional Overview

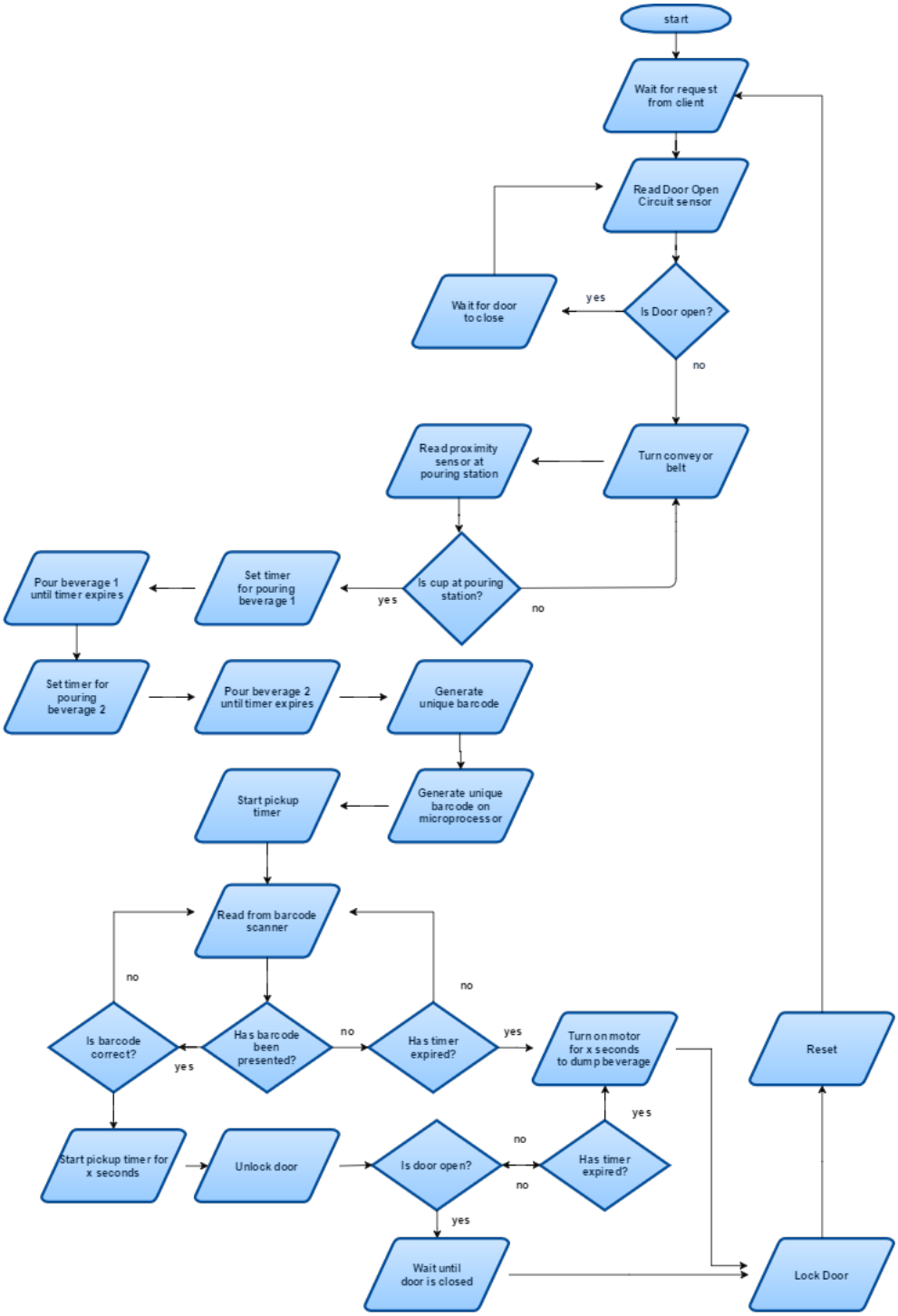

Figure 5: Program flowchart

Our program flowchart above maps our program's execution from listening for a request from the mobile application, through each step of the pouring process, to the moment of delivery.

Before an order is made, there will be a listener program running on the ATmega, which interfaces with the WiFi module using the SPI protocol and listens for a request being made from a mobile device on the same network. Once a connection is established, we are able to communicate with the client directly and securely. The program flow leads through the procedure of pouring a beverage and generating a barcode. It sends this barcode to the user via TCP, and begins scanning for barcodes in the near vicinity. The security requirement is satisfied through our authentication procedure.

While the majority of the security and protection comes from the physical barrier that separates the user from the drink as it is being prepared, the core functionality stems from the protocol interfaces between the micro-controller (ATmega328), and the various modules. This microprocessor is extremely versatile in that it natively offers SPI, I^2C, USART, as well as several analog and digital GPIO pins from which we can read, and to which we can write values. The ATmega328 is the processor used in many Arduino applications, which allows programming and developing with it to be straightforward and well-documented. We intend to set it up primitively as a breakout board with the schematic as shown below.

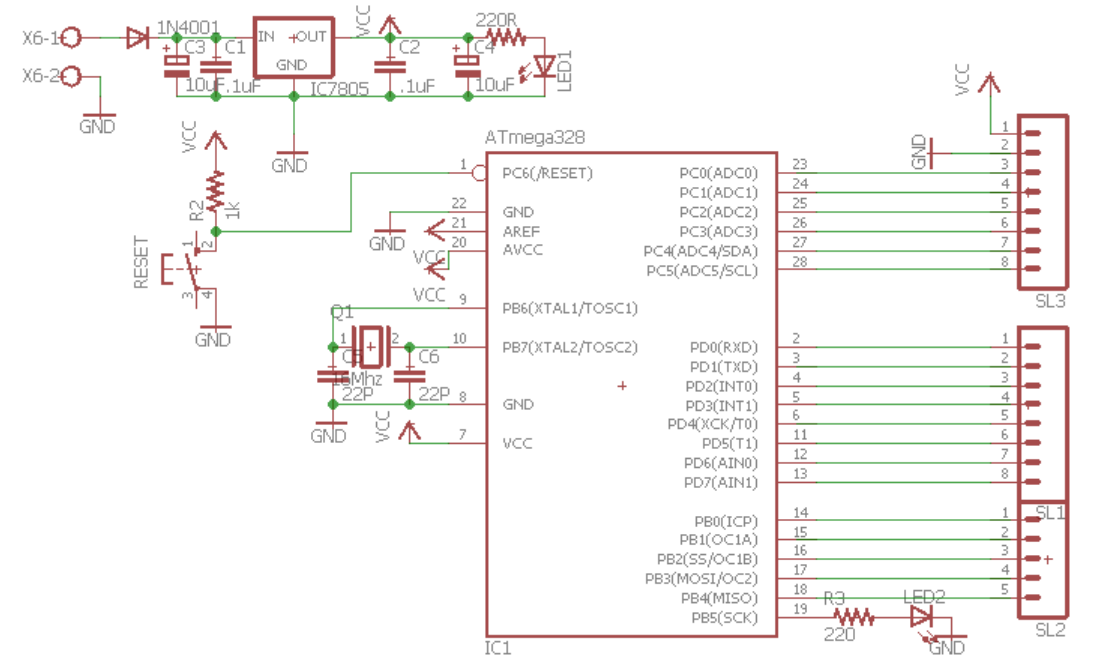

Figure 6: ATmega328 basic schematic [1]

With such a flexible layout, we will be able to add and remove our modules very easily, allowing for extremely modular and streamlined testing and development. All modular components can be represented as black boxes with their corresponding inputs and outputs attached to the correct protocols of the ATmega. A PCB will be made for the sensors, with leads for connections to the other modules.

Beyond the layout, the microprocessor will utilize GPIO pins to control the pouring valves, as well as the electronic devices including the lock and open-door sensor as shown in Figure 7. These signals are easy to manipulate and read. The door circuit will have a pull-down resistor so that when the circuit is open, the GPIO pin reading will see 0V, and when the door closes, the circuit is complete and it will read 5V.

For communication with the scanner and WiFi module, we will be implementing the SPI protocol, transferring serial data to and from the microprocessor. We only have to read from or write to one module at a time, so after spending some time evaluating our options and seeing what was available, we decided SPI was the most efficient choice for our purposes. The USB protocol would have been easier to find scanner devices for, but after looking at some research, it seems that the microcontroller cannot act as a USB master, making it unnecessarily complicated to create a workaround library considering our limited time frame.

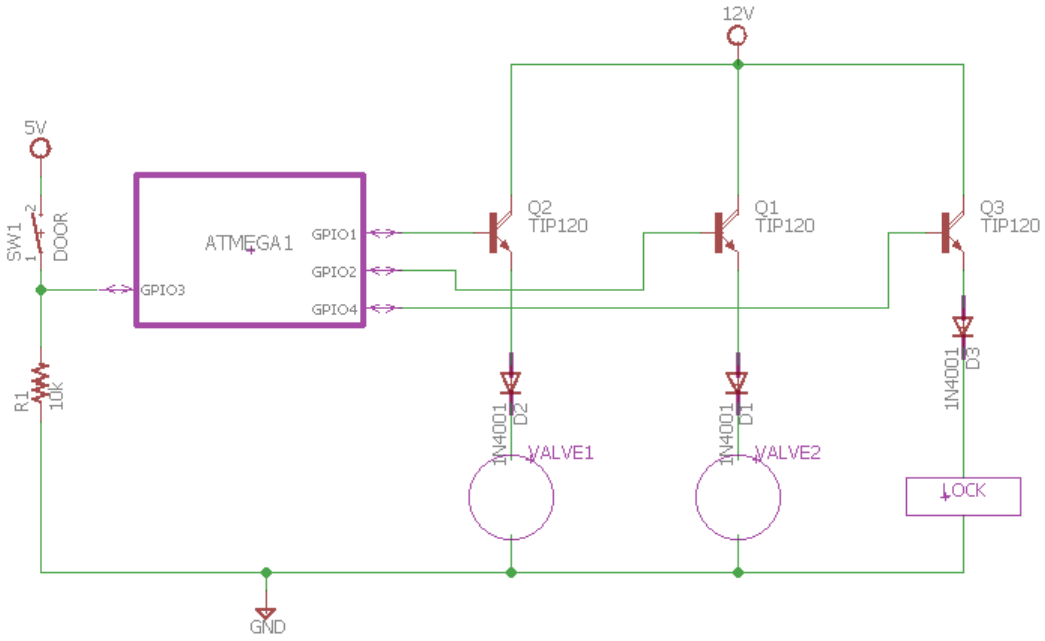

Figure 7: Circuit design for electromechanical controller

Requirements/Verification

Micro-processor: ATmega328	
Requirement	**Verification**
Must be able to run at 5V ±10%	Using voltage generator, we will run simple programs at 4.5V and 5.5V and evaluate the performance.
Must be able to provide a timer interface to the programmer, and time events up to 120 seconds.	Write a function which will light up an LED at a certain sequence of time steps and compare it against a stopwatch.
Must be able to generate 128 random bytes and send them to the mobile device in less than .5 seconds	Display generated barcode on mobile device, and compare with hardcoded "random" bytes on ATmega.
Must be able to compare two 128-byte strings to determine if they are equal in less than .5 seconds	Hardcode several strings for the ATmega to compare, and display the results on an LED (1 if same, 0 if different), and compare timing against a stopwatch.

WiFi Module: Adafruit ATWINC1500 WiFi	
Requirement	**Verification**
Must be able to send 128 bytes of barcode data in under 1 second.	Create socket connection on ATmega to send packets of data. Send 128 bytes of bytes from ATmega to Phone. Will send bytes over connection and then print bytes on screen to confirm.
Must be able to receive 3 bytes of user/drink data in under 1 second.	Create socket connection on app to send packets of data. Send 128 bytes from Phone to ATmega. Will send bytes over connection and then print bytes on screen to confirm.

Barcode Scanner: Unitech AS10-P AS10	
Requirement	**Verification**
Scan code correctly 95% of the time or more.	Using a set barcode, scan hundreds of times. Store correct scan value on ATmega. Light up an LED for matching stored and scan value to obtain data points. Calculate percentage using these data points until we reach confidence in the requirement being met.

12V Power Supply from Kit	
Requirement	**Verification**
Must be able to provide 12V±.25 to our 12V modules.	Use multimeter to test voltage from power supply.
Must be able to be regulated to provide 5V±.25 to our 5V modules.	Use multimeter to test voltage from power supply.
Must be able to provide at least 1A current.	Use multimeter to test current.

UHPPOTE UT0511-130NO Magnetic Lock	
Requirement	**Verification**
Must be able to handle 100lbs of force pulling against the lock.	Fasten both lock components and lock by applying 12V differential. Clamp the mobile component, pull with a spring scale. Our design will not allow enough grip for humans to provide that kind of pulling force.
Must only have 2lbs or less of resistance when unlocked.	Fasten both lock components. Unlock by applying 0V. Pull with spring scale.

Brass Liquid Solenoid Valve - 12V -1/2 NPS	
Requirement	**Verification**
Must be able to dispense liquid using only gravity (no pressure necessary) at a rate of at least 1.2 ounces per second for a total of 10 seconds	Hold container of liquid upside down with valve attached. Apply 12V differential to the terminals and observe results.

Infrared Proximity Sensor: Sharp GP2A200LCS0F	
Requirement	**Verification**
Must emit digital signal when an opaque cup is at a distance of 2cm or less.	Wire up sensor. Hold opaque cup at distance 2cm or closer and measure output.
Must emit digital signal when a glass cup at a distance of 2cm or less.	Wire up sensor. Hold glass cup at distance 2cm or closer and measure output.

Android Application	
Requirement	**Verification**
Allow users to select or create drinks with a custom proportion between two mixtures (any ratio from 100%:0% to 0%:100% with increments of 10%).	Create user interface that allows such options. Test functionality works as intended.
Must be able to receive 128 bytes of barcode data in under 1 second.	Create socket connection on ATmega to send packets of data. Test sending data from ATmega to Phone.
Must be able to send 3 bytes of user/drink data in under 1 second.	Create socket connection on app to send packets of data. Test sending data from Phone to ATmega.
Must be able to execute on Android Phone running Lollipop.	Test app launch on Phone or emulator with Lollipop.
Must be able to display barcode data on screen to be read by barcode scanner.	Test receiving and display of barcode data.

Door Circuit	
Requirement	**Verification**
Must change digital value when door is open or closed within .25 seconds	Set up ATmega to read from circuit pin and open and close the door quickly, displaying readings through LED output

Supporting Material

Conveyor belt motor RPM calculation:

2’ long conveyor belt
1" diameter motor

Distance per rotation of motor = circumference = $\pi D = \pi$ inches
Desired time to travel 2': 10 seconds
Desired velocity = $\frac{dist}{time} = \frac{2'}{10s} = \frac{24"}{10s} = 144\frac{inches}{min}$

Desired RPM = $\frac{144\,inches}{min} * \frac{1\,rotation}{\pi\,inches} = 45.8$ rpm

Pouring speed calculation:

Total time requirement: 20 seconds
Conveyor belt time worst case: 10 seconds
Drink size: 12oz
Desired pouring rate: $\frac{amount}{time} = \frac{12}{10} = 1.2$ ounces per second

Power Requirements

Part	Voltage	Rated Current
Valve	12V	3A
Valve	12V	3A
Stepper Motor	12V	0.33A
Lock	12V	0.10A
Voltage Regulator	5V	1A Max

$P = IV$
Worst Case Power = $I_1V_1 + I_2V_2$, we assume maximum current across regulator.
$= 12V \cdot (3+3+0.33+0.1)A + 5V \cdot 1A = 82.16W$
Power supply is 12V, so we need it to handle $\frac{82.16W}{12V} = 6.847A$

Tolerance Analysis

The ATmega328 chip is able to operate at voltages between 1.8-5.5V[2], with an absolute maximum rating of 6.0V. We will provide this voltage to the ATmega, as well as the other modules on the 5V line (WiFi module, proximity sensor, and barcode scanner) by utilizing the LM7805 voltage regulator device, which can convert 12V to 5V. It has an output voltage tolerance of ±4%[4], which in our case equates to $4\% * 5V = .20V$. Since the ATmega328 chip

is able to function normally anywhere between 1.8-5.5V, and our regulator will provide us anywhere between 4.8-5.2V, we should be well within our limits in operation.

Cost

Labor			
Name	**Hourly Rate**	**Hours Invested**	**Total ($)**
Austin	$30.00	200	6,000
Greg	$30.00	200	6,000
Max	$30.00	200	6,000
Total		600	18,000

Parts					
Description	**Source**	**Part Number**	**Quantity**	**Cost ($)**	**Total ($)**
ATmega328	Sparkfun	ATmega328	1	6.00	6.00
WiFi Module	Adafruit	Adafruit ATWINC1500	1	25.00	25.00
Barcode Scanner	Amazon	Unitech AS10-P	1	30.00	30.00
Lock	Amazon	UHPPOTE UT0511-130NO	1	23.00	23.00
Brass Liquid Solenoid Valve	Adafruit	Adafruit Product ID: 996	1	24.95	24.95
Infrared Proximity Sensor	Digikey	GP2A200LCS0F	1	7.11	7.11
Wood for creating casing (if machine shop can't make)	Menards	3/4" x 8" x 8'	3	23.98	71.94
Motor	Machine Shop	Miscellaneous	1	N/A yet	N/A yet
Circuit miscellany	ECEB?	N/A	Depends	0.00-30.00	0.00-30.00

Table	**Total**
Labor	$18,000
Parts	$188.00-$218.00
Total	~$18,203

SCHEDULE

Week	Task	Duty
Feb. 27th	Finalize purchases	All
Mar. 6th	Program blink LED program onto ATmega328 successfully	Max
	Complete testing of all parts in our possession	Austin
	Work on account creation for Android App	Greg
Mar. 13th	Coordinate with Austin to have sensors/valves programmatically controllable	Max
	Fix bugs. Work with Max to integrate sensors/valves with ATmega.	Austin
	Create socket connection on Android App and Test.	Greg
Spring Break	Coordinate with Greg to get WiFi module reading/writing to ATmega	Max
	(Maybe) Work on physical frame of project, test parts, work on motor driver.	Austin
	Create socket connection on ATmega and Test.	Greg
Mar. 27th	Coordinate with Austin to read from barcode scanner programmically	Max
	Work with Max to integrate barcode scanner. Assist others.	Austin
	Keep testing connect between App and ATmega. Help with Barcode.	Greg
Apr. 3rd	Implementing flowchart & assemble modules. Connect to mobile app.	Max
	Focus on fixing any bugs involving specific parts. Assist others.	Austin
	Test a complete order and make sure it gets validated correctly	Greg
Apr. 10th	Complete implementation of flowchart and assembly of modules	Max
	Begin mounting pieces on frame, assist others as necessary.	Austin
	Test and fix bugs.	Greg
Apr. 17th	Mount all pieces on frame, Work on Mock Demo	All
Apr. 24th	Work on Final Paper and Presentation	All
May. 1th	Complete Final Paper	All

ETHICS AND SAFETY

The ethical aspect of this project is what makes it stand out among the others. Our product is designed to make the world a safer place by removing the opportunity for malicious bartenders or patrons to inject someone's drink with a potentially harmful substance. By automating the drink preparation process and ensuring that only the person who ordered the drink is coming in contact with it, we successfully eliminate any uninvited interactions between the drink and a stranger.

The first rule stated on the IEEE Code of Ethics [3] describes an individual's responsibility to the welfare of the public in all circumstances. In our scenario, we have observed a dangerous process that occurs frequently, and thus it is our responsibility to do everything in our power to rectify it. We can thus improve the safety of bars, which are so often known to be dangerous places.

In the actual implementation and physical design of our automated bartender, we will be extremely diligent to produce no harm. We will accomplish this by observing all the necessary security precautions to ensure that tampering with our device is difficult, if not impossible. These precautions include establishing a secure connection with the client, and preventing our messages from being tampered with or falsified. As for the machine itself, any circuitry will be shielded from the user so there is no electrical danger, and our design will not use enough power to start a fire. Although fluids may be splashed by user error in grabbing the drink, all of our hardware will be protected and insulated.

Finally, we need to take into consideration the legislation surrounding consumption of alcohol in the United States. Currently there are laws in place to limit the consumption to only be allowed for those aged twenty-one years or older. Especially considering the legal age of bar entry for campus town is nineteen, this factor needs to be taken into consideration. One solution which we propose is a secure registration system which requires users to authorize their age to be twenty-one or older before allowing them to make a purchase. It will be unlawful to lie about one's age during the registration process, and offenders will be persecuted to the fullest extent of the law. The process will be very similar to how bars and the Champaign Police coexist on campus, with occasional raids to ensure everyone is complying. Additionally, we recognize that there are FDA requirings for using plastic tubing to serve beverages, and we will be sure to adhere to these. The policies presented are to

protect the safety of our customers and ensure a business' compliance with the law.

Ultimately, our main objective is to design a machine which will allow partygoers to have a good time and ensure their safety, while saving the bar itself money in the long run. We realize that this will not stop all sexual assaults from occurring, as tragically there are countless more situations in which these incidents occur, but we hope to make a dent in what seems, at first glance, to be a daunting problem.

Bibliography

[1] A. Sanjeev, "How to Make Arduino Board: The Easy Way - DIY Hacking", DIY Hacking, 2014. [Online]. Available: https://diyhacking.com/make-arduino-board-and-bootload/. [Accessed: 16- Feb- 2017].

[2] "8-bit AVR Microcontrollers ATmega328/P DATASHEET COMPLETE", Atmel, 2016. [Online]. Available: http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf. [Accessed: 22- Feb- 2017].

[3] "IEEE IEEE Code of Ethics", Ieee.org, 2017. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 22- Feb- 2017].

[4] LM78XX/LM78XXA 3-Terminal 1A Positive Voltage Regulator, 1st ed. Fairchild, 2017, p. 1. [Accessed: 24- Feb- 2017].