Cheap and Versatile Breathalyzer Box

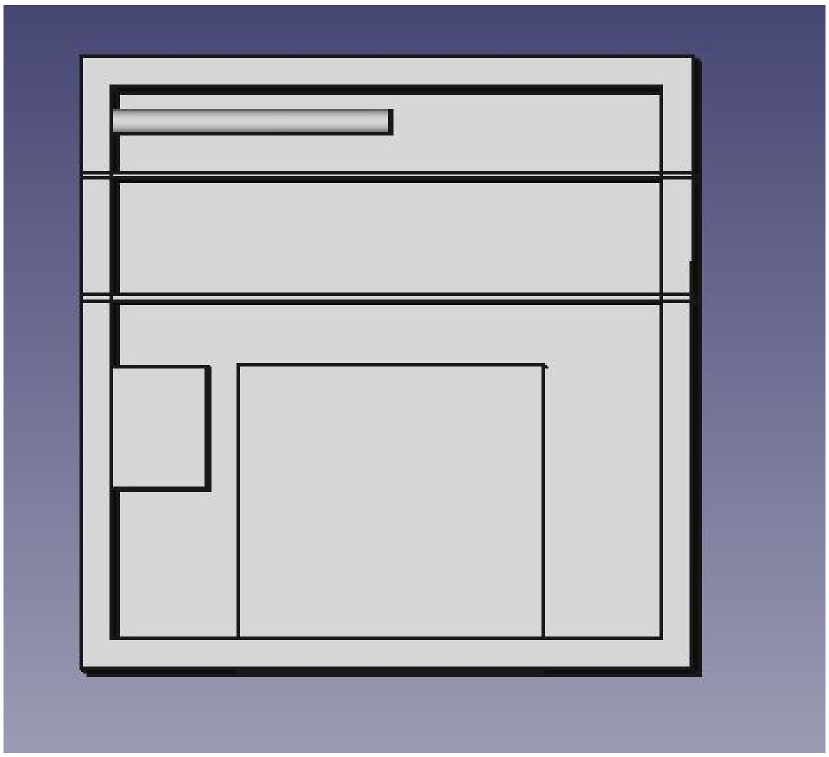

Fig 1. Breathalyzer Box

ECE 445 Design Document

Cameron Palte, Kush Patel, and Stanley Zhao
Team 1
TA: Shaoyu Meng
Professor: Wei He

Fall 2019 - 10/03/2019

Table of Contents

1. Introduction……………………………………………………………………………..2
    1.1 Objective………………………………………………………………………2
    1.2 Background……………………………………………………………………...2
    1.3 Visual Aid..…………………………………………………………………....3
    1.4 High-level Requirements List…………………………………………………3
2. Design……………………………………………………………………………………..4
    2.1 Block Diagram………………………………………………………………...4
    2.2 Physical Design………………………………………………………………..5
    2.3 Subsystems……………………………………………………………………6
        2.3.1 Microcontroller……………………………………………………...6
        2.3.2 On / Off Switch……………………………………………………...9
        2.3.3 Lock Switch…………………………………………………………9
        2.3.4 Passcode Switch……………………………………………………9
        2.3.5 12V Battery and Wall Charger…………………………...……………10
        2.3.6 12V Voltage Regulator……………………………………...…………..11
        2.3.7 5V Voltage Regulator………………………………………………...12
        2.3.8 5V DC / DC Converter………………………………………………13
        2.3.9 Solenoid………………………………………………………………14
        2.3.10 Relay………………………………………………………………..14
        2.3.11 Alcohol Gas Sensor……………………………………………………15
        2.3.12 LED Display…………………………………………………….....16
        2.3.13 RFID Case……………………………………………………….....18
    2.4 Overall Circuit Schematic and PCB……………………………………………19
    2.5 Tolerance Analysis………………………………………………………….....21
    2.6 Future Work...…………………………………………………………………23
        2.6.1 Overview………………………………………………………….....23
        2.6.2 Detailed Breakdown…………………………………………………..24
        2.6.3 Cost and Schedule……………………………………………………25
        2.6.4 Ethics…………………………………………………………………26
3. Cost and Schedule………………………………………………………………………..26
    3.1 Cost Analysis…………………………………………………………………...26
    3.2 Schedule………………………………………………………………………...28
4. Discussion of Ethics and Safety…………………………………………………………29
5. Citations…………………………………………………………………………………30

1. Introduction

1.1 Objective

On average, two in three people will be involved in a drunk driving crash in their lifetime. [1] In 2017, 10,874 people died on American roads in crashes related to someone's blood alcohol content being above the legal limit. Those deaths made up 29% of all deaths involving roads in the United States. [2]. Despite large efforts to stop this, drunk driving continues to be one of the largest preventable causes of death, especially among young adults. Preventing someone from driving under the influence is difficult. Being intoxicated significantly lowers one's inhibitions [3], and as a result, increases the chance of someone making a poor decision.

Current systems to prevent drunk driving are inconvenient and expensive. Our proposed solution would help remedy this problem by making it more convenient to prevent drunk driving, while lowering costs. Current ignition interlocks require a 1+ hour appointment with a certified technician to install and must be used by every single driver of the vehicle. [4] They are also $900, presenting a strong financial barrier in a time when 40% of Americans can't afford a $400 expense. [5] Our proposed key-based solution would be cheap and portable. It's attachment to car keys means it wouldn't require any installation and wouldn't inconvenience other users of the car. Bars or other institutions could mandate it to serve customers. Alternatively, users trying to find a low cost way to stay responsible, such as those with comments like,

> "I was court ordered to have an interlock in my car for 24 months and after I was done I purchased this volunteer interlock devices so I would never have to go through this experience again" [6]

would have a system that doesn't inconvenience other users of their car, and is low cost, while taking advantage of the fact that these people, while sober would not drive intoxicated.

Our proposed solution is a key-based device that allows the user to place their car keys inside and lock, at the start of a night going out. The device will contain a breathalyzer, consisting of an alcohol gas sensor. For the user to retrieve their keys, they need to blow into the breathalyzer, which will calculate their blood alcohol content and send it to an LED display. They will also need to enter in a passcode, through a series of switches, on the side of the device. If their blood alcohol content is below the legal limit, and the passcode is correct, then the user will be able to retrieve their keys from the device.

1.2 Background

The first ignition interlock systems were developed in 1969. As of 2012, all 50 states allow using ignition interlock sensors as a sentencing alternative for drunken drivers. The interlock device underwent a significant improvement in the 1990s to more efficient sensors, however, since then minimal progress has happened on a technological front [7]. In a consolidated industry, with relatively few manufacturers, there is minimal incentive to continue to develop more effective solutions to target subsections of the market.

For our target audience, being able to avoid making a decision while intoxicated, that they would never make while sober is everything. Having a discrete way to do this that doesn't affect other users of their car at a low cost is even better. Additionally, research is being done into voluntary breathalyzers, especially in relation to cars. A recent pilot study with commercial drivers in Sweden showed that investigators believed voluntary breathalyzers to have an overall positive effect on safe driving. However, 61% of the company's with the commercial drivers said their main reason for not having installed them was cost. [8] Our easy to use device should help fill provide a product with new levels of convenience and low cost in the voluntary safe-driving market.

1.3 Visual Aid

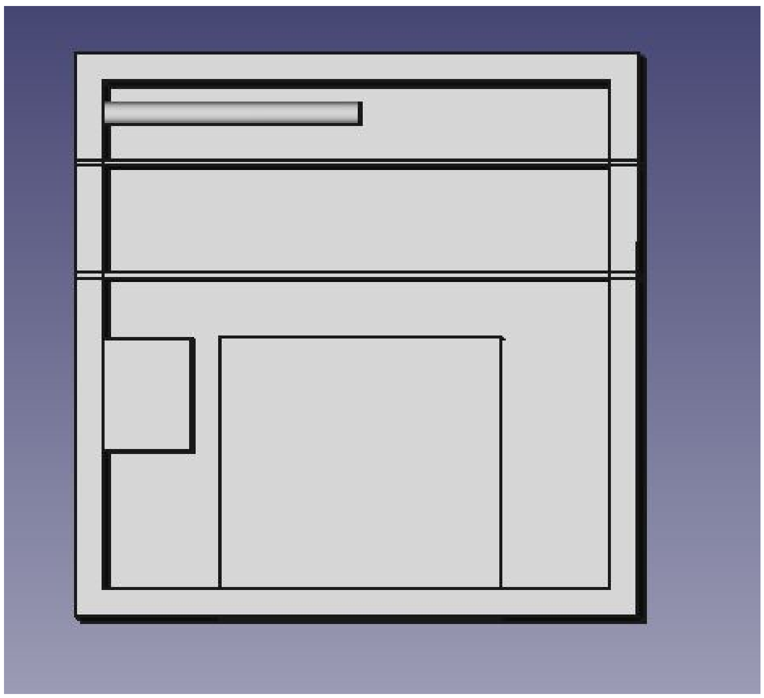

Fig 2. Visual Aid (top down view)

Above is a pictorial representation of our project. The box stands independently from a user perspective, although once in a while charging there would be a separate charging cord entering the device. The battery will be mounted on the bottom of the box, below what's visible, where the wall has been removed to show the solenoid. The user, to use the box, will open it up along the right hand side after blowing into the breathalyzer. They will then place their keys into the box and close it. They will subsequently only be able to open the box to retrieve their keys by again blowing into the breathalyzer (with the correct passcode and a blood alcohol content below the threshold). We have not finalized our decision of where everything component will be yet.

1.4 High-level Requirements List

- Breathalyzer box can accurately measure user's blood alcohol content to a precision of 0.02% and display that information on a 7 segment LCD display.
- An inaccurate passcode will prevent the user from unlocking the breathalyzer box regardless of their blood alcohol content.
- The breathalyzer box can prevent a push-to-start key from being used, and be manufacturable from less than $100 of parts.

2. Design

2.1 Block Diagram

Our project consists of 6 major blocks, the power supply, lock mechanism, user interface, microcontroller, breathalyzer, and LED display. The power supply will contain our battery, which can be charged from the wall, along with the required regulators to supply 5V and 12V to our various components. The user interface allows the user to turn the microcontroller on and off, lock the box initially, and enter their passcode. The lock mechanism consists of a solenoid and a relay to decide when to lock and unlock the box. The breathalyzer consists of an alcohol gas sensor to measure and calculate the blood alcohol content. The LED display consists of a 7 segment display to show the user's blood alcohol content. Lastly, the microcontroller consists of an Arduino Pro Mini 328 5V along with the PCB to handle the logic and interaction between the other components.

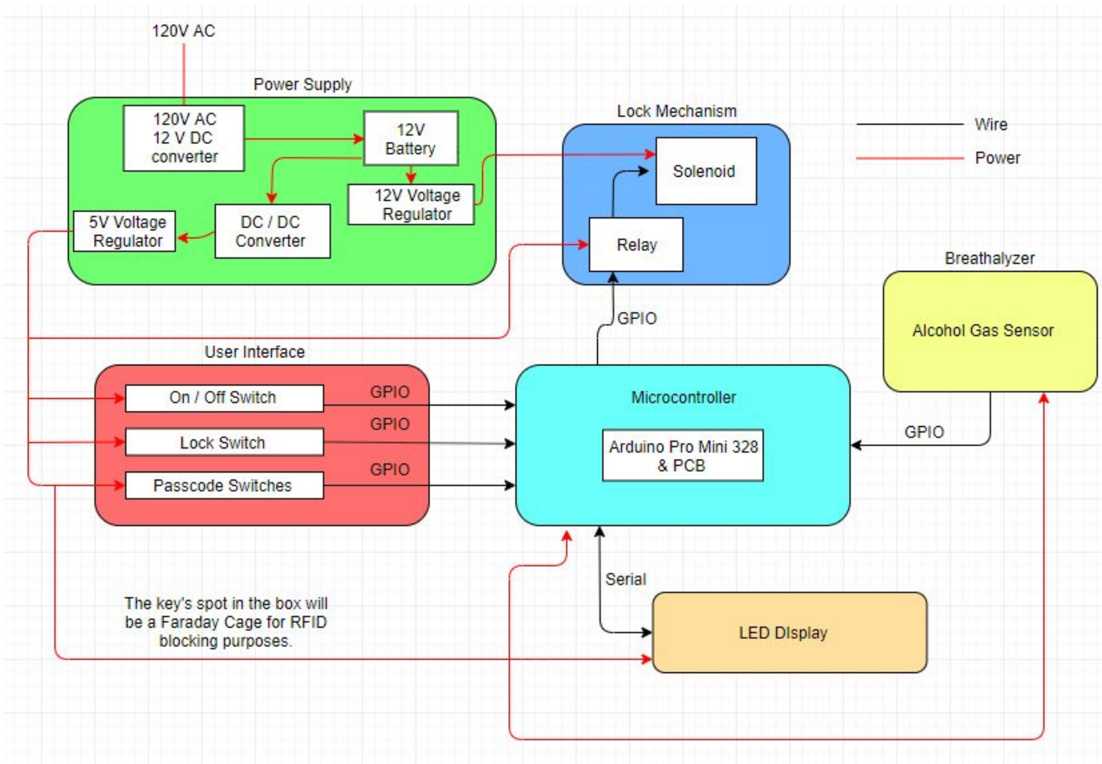

Fig 3. Project Block Diagram

2.2 Physical Design

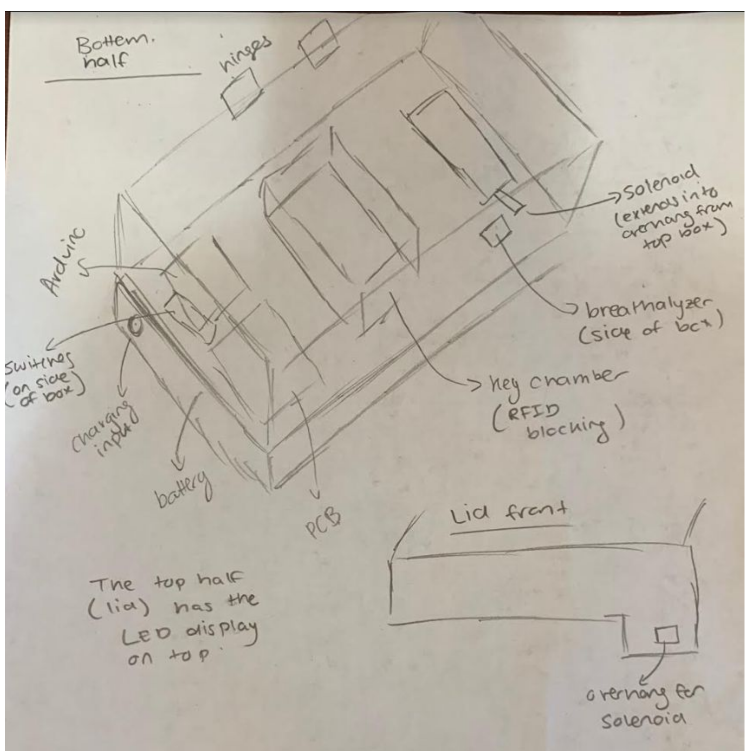

Fig 4. Hand Drawn Container View

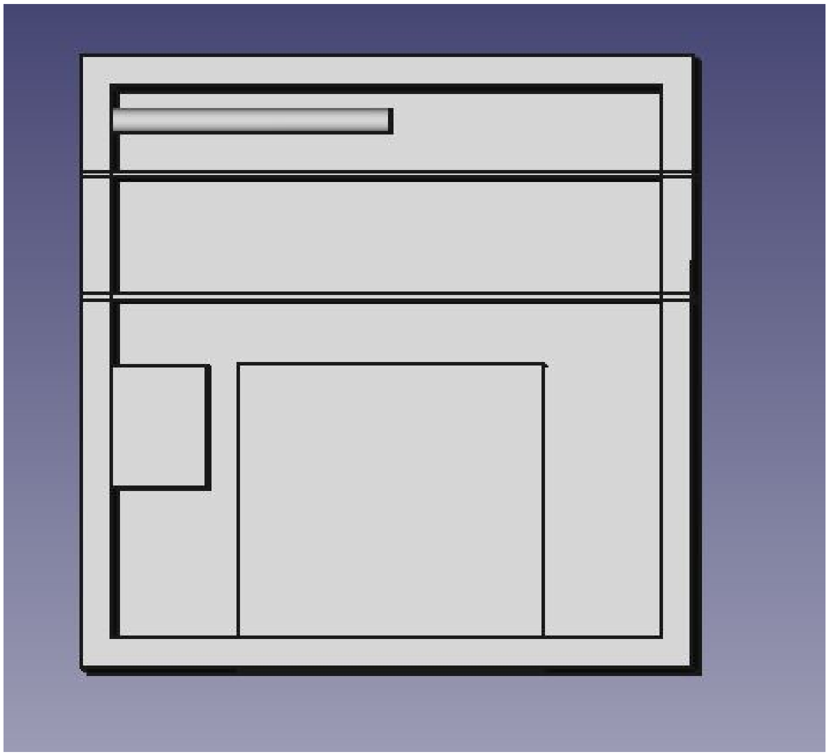

Fig 5. Top-Down Box View, Hinges on Right Side, Battery Below

Our box has a few key components in it which requires 6 inches long by 4 inches wide by 4 inches tall. The box will be opened up by two hinges which are will be on the right hand side of the box diagram (Figure 5). The lid will be on top of this top down view and solely contain the LED display.

These dimensions are up for change depending on future size requirements. More specifically the box must be able to hold the PCB along with all of its circuitry, arduino, the breathalyzer sensor, a compartment to hold the desired items, as well as a battery to power the electrical components. The battery specifically will be attached to the bottom side of the box. Along with all of these parts, the box must also have a locking mechanism. This is supported by a solenoid which will fit into a groove to lock the box as a default state. If the solenoid has retracted from the hole, the cover of the box will be able to rotate about the set of hinges.

2.3 Subsystems

2.3.1 Microcontroller

The microcontroller will be the brain of the unit and handle the logic for communicating between the different components. It will consist of our PCB along with an Arduino Pro Mini 328. Our PCB will consist of our passcode logic, power system, and relay for the solenoid. We chose the

Arduino Pro Mini 328 both for its low cost ($9.95) and its balance between ease of use and small size that makes it simple to fit in the box. Also, by not putting the chip on our PCB, by using the Arduino Pro Mini 328, we're avoiding resting our entire project no the success of our PCB. Inputs come from the on / off switch, lock switch, and passcode PCB logic, and alcohol gas sensor, while outputs go to the LED display and relay.

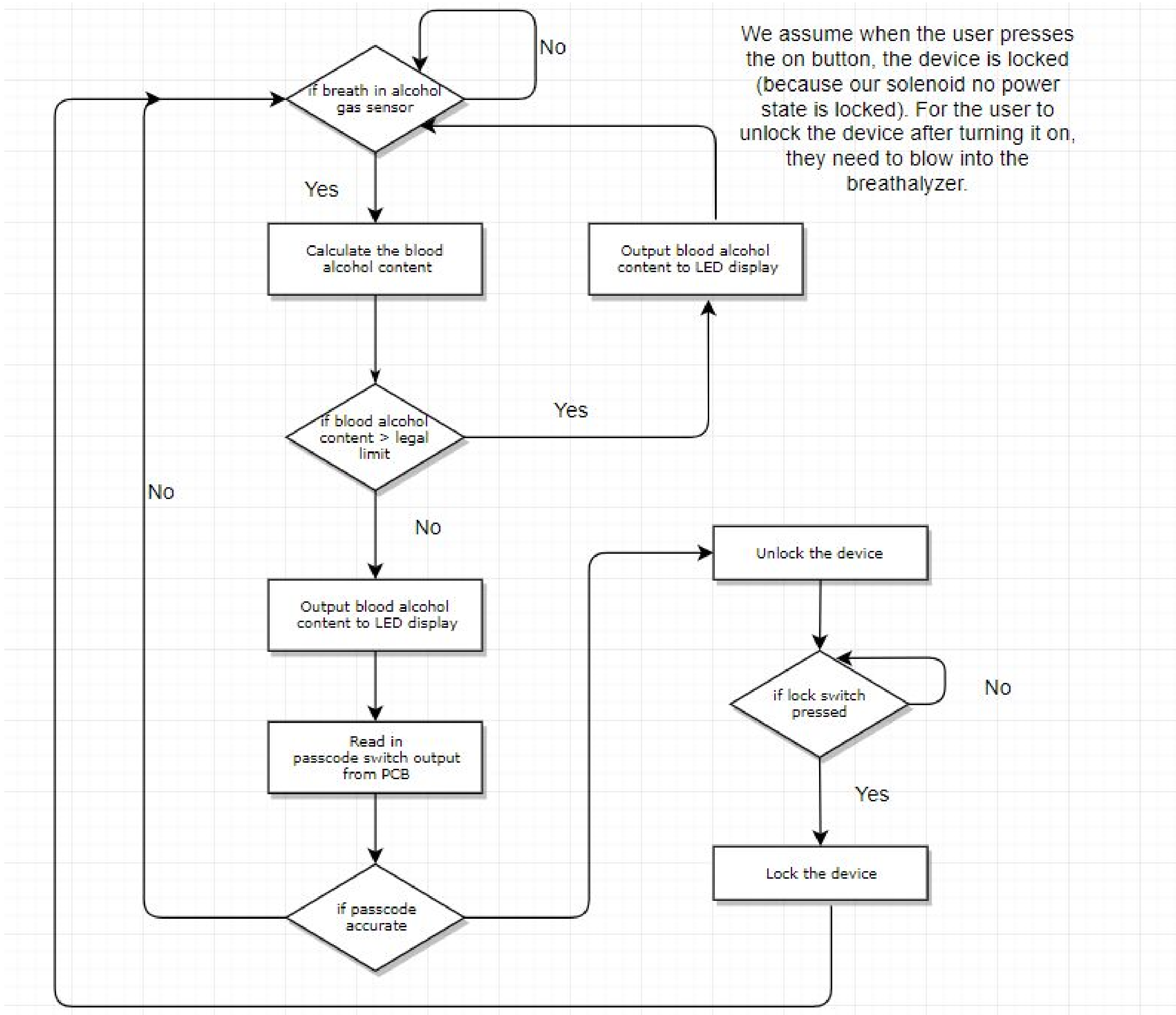

Fig 6. Software Flow Chart

Requirement	Verification
Based on the inputs into the microcontroller, the correct output is sent to the relay.	Manually set inputs and use a voltmeter to verify the correct value is sent to the relay.
Microcontroller runs code to read and analyze blood alcohol content automatically upon being turned on.	Supply with power and confirm that the correct code executes, by having the code turn on and off LEDs on the board as it reaches certain stages.
Microcontroller outputs blood alcohol content (%) to the LEDs.	Write code to input simulated resistance values and confirm that the correct blood alcohol content is calculated and sent to LEDs over serial connection.
Microcontroller can adequately receive input from the switches.	Use a multimeter to verify that the microcontroller receives the correct input from switches and PCB.

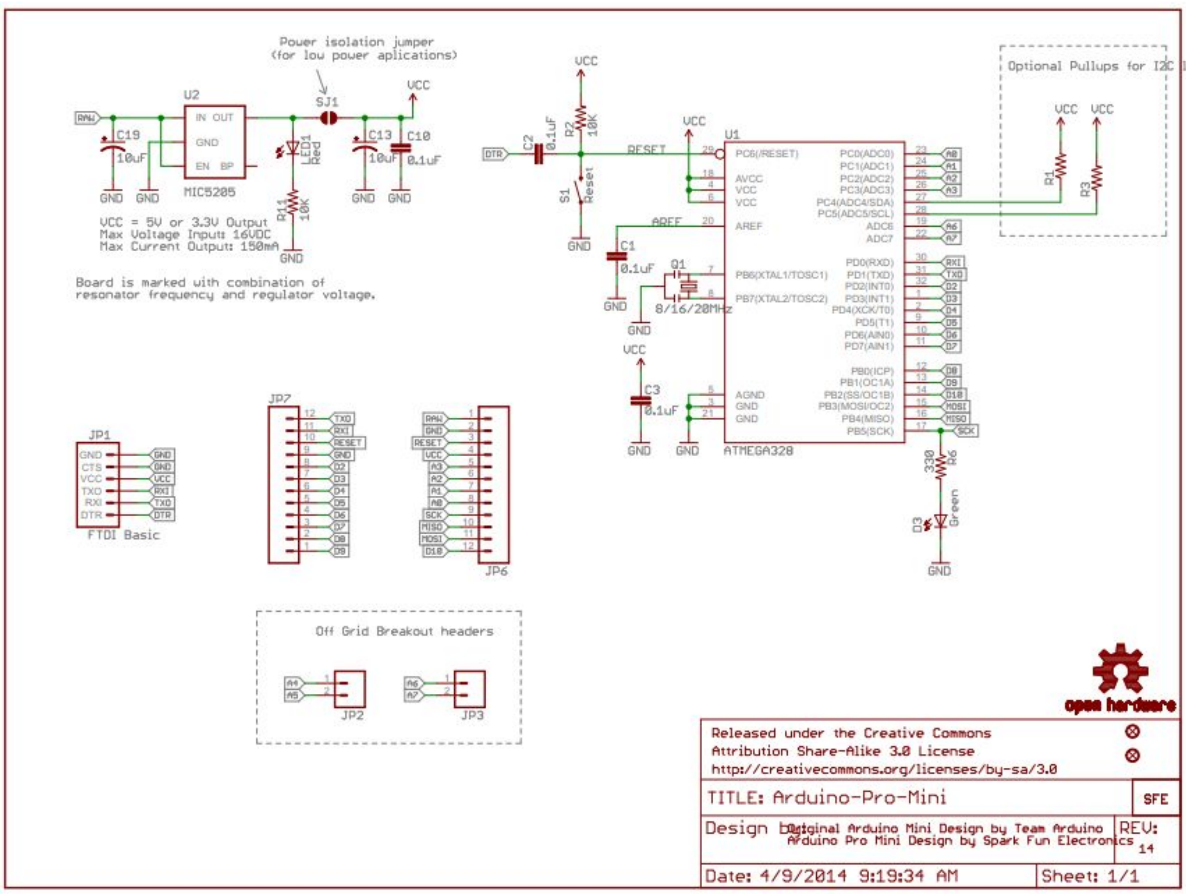

Fig 7. Arduino Pro Mini 328 Pin Assignment Circuit Schematic [9]
We will not use every pin on the schematic.

2.3.2 On / Off Switch

The on and off switch is a simple switch on the side of the container. The purpose of this switch is to enable the user to maximize power savings. Since our solenoid defaults to locked (extended) when unpowered, the container should remain locked when this switch is off. The user can then flip it to on which will turn on our microcontroller and associated circuitry and allow the user to use the breathalyzer. We chose an on / off switch with a solenoid that's locked by default to minimize power consumption of the device.

Requirement	Verification
Container can not be opened when off (and the circuit is unpowered).	Turn switch to off position and make sure that the container cannot be opened.
Once turned on, user can unlock box assuming blood alcohol content is at the correct level.	Once the device is built, test turning on and confirming that the device can be unlocked (this is a full integration test but depends on pressing the on button resultings in the microcontroller booting up, etc.)

2.3.3 Lock Switch

The lock switch is a simple switch on the side of the container. Pressing this switch will trigger the container to lock, giving the user a way to lock the container once the keys have been placed inside. We chose to have a lock switch because the user needs a way to lock the container once they have placed their keys inside.

Requirement	Verification
Pressing switch causes container to lock.	Take an unlocked container version and press the lock switch. Confirm that the container is subsequently locked.
Flipping the switch does not change the state of a locked container.	Try moving switch when container is locked to confirm it does not unlock.

2.3.4 Passcode Switch

The passcode switch is a series of switches positioned on the side of the container. The user enters their passcode into these switches, which pass through a series of logic gates on our PCB, to check the password vs. a hardcoded password. A digital 1 is then sent to the microcontroller if the passcode is correct, otherwise a 0 is sent. The decision was made to hard code the passcode to both speed up the verification of the password, and help keep it and its logic modular, while also preventing someone unauthorized who wants to access the device from simply being able to reset the passcode.

Requirement	Verification
The user can input their passcode on the switches, from the side of the box, while the box is closed.	Close the box and use a voltmeter to confirm that the switches can be set to low or high.
The passcode switch logic gates output a 1 if the passcode matches the hardcoded passcode, otherwise a 0 is outputted.	Enter the correct passcode and then measure the output recorded by the Arduino microcontroller from the output of the logic gates with a multimeter.

2.3.5 12V Battery and Wall Charger

To charge our 12V DC rechargeable battery, we need a 120V to 12V AC to DC converter. We plan to use an off the shelf power supply and converter for this purpose for two reasons. This simple solution would allow us to focus on the more important aspects of our project, while simultaneously allowing users of our device to simply purchase an off-the-shelf power supply to power their device.

While one of our use cases involves commercial institutions such as bars and restaurants being able to use our device, another essential use case is users carrying it around as a portable device. As a result, we need our device to be battery powered. We decided to go with a 12V battery because of the simplicity in using that power for our solenoid which uses a respectable amount of power (12V at 0.6 amps). If we had to step up the power from a 5V battery, we'd have to draw 1.5 amps from the Solenoid alone on top of the power for the microcontroller and other components, which we felt was less optimal.

In terms of power requirements, we want our device to be able to handle a maximum of 10 unlocks across a given night, each lasting 15 minutes. That means our battery needs to power our device for 150 minutes.

In terms of power consumption across 150 minutes, all of the components besides the solenoid will be powered from the Arduino Pro Mini. Therefore our two sources of power consumption are Arduino Pro Mini 328 and the Solenoid. It's worth pointing out that the Arduino Pro Mini consumes very minimal power (5 mA at a 3.3 V input [10]) when its external pins aren't being used. Given that the components we're powering off of the external pins use minimal power, the Arduino Pro Mini 328 should never consume the full 150 mA that it is capable of consuming.

Arduino Pro Mini 328 = 5V at 150 mA [11]
MQ-3 Alcohol Gas Sensor = 5.0V at 160 mA [12]
Solenoid = 12V at 600 mA [13]

Total Hourly Consumption: 8.75 watts
Minimum Battery Capacity: 21.875 watt-hours

Minimum Battery Capacity (mAh): 1823 mAh at 12V
Minimum Battery Output Current at 12V: 802mA (includes a 10% error margin)

Requirement	Verification
The off the shelf charger is capable of fully charging the 12V battery overnight.	Plug the 120V off the shelf charger into the 12V battery and confirm it is fully charged after 8 hours.
Battery can output 12V +/- 0.5V when fully charged.	Fully charge battery and confirm the required voltage is outputted with a voltmeter.
Battery capacity can handle keeping case powered and unlocked for 150 minutes.	Unlock case through the microcontroller and confirm that it remains unlocked and powered for 150 minutes.
Battery can output enough current to support our entire circuit.	Our calculations above show that the battery we're planning to purchase should be more than capable of handling this. Verify empirically by drawing load from battery with a resistor and using an ammeter.

2.3.6 12V Voltage Regulator

The purpose of the 12V voltage regulator is to confirm that our circuit, and most importantly, the solenoid has consistent access to the 12V power that it needs to function. For this, we plan to use a simple L7812 voltage regulator to output the 12V +/- 0.5V for our solenoid. We chose this regulator because of its efficiency and low cost. Our voltage regulator also needs to be capable of handling 600 mA of current to support the 12V Solenoid that it's driving. Our selected L7812 regulator can handle 1.5 Amps, and based on the data sheet outputs 12V +/- 0.5V (attach data sheet).

Requirement	Verification
12V voltage regulator can output 12V +/- 0.5 as long as the battery is at >10% capacity.	Test output voltage with a voltage meter at various battery levels to confirm this requirement is met.
12V voltage regulator is capable of supplying adequate current.	Attach 12V voltage regulator to the solenoid and confirm it is capable of supplying 600 mA.

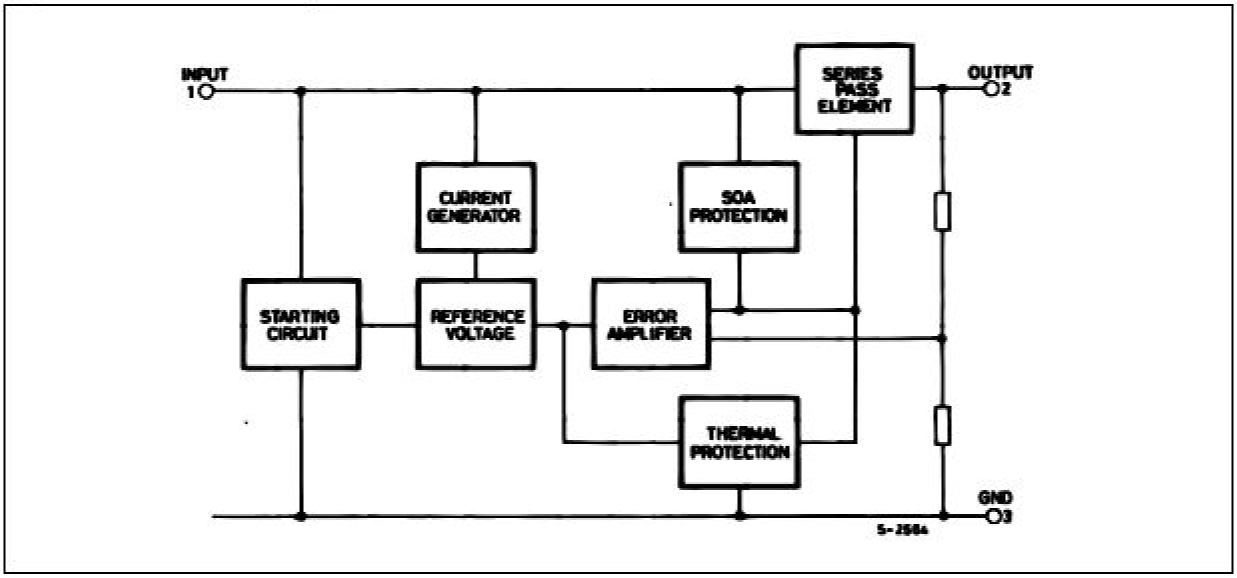

Fig 8. 78xx Internal Schematic [14]

2.3.7 5V Voltage Regulator

The purpose of the 5V voltage regulator is to confirm that the entirety of our circuit has access to the well regulated 5V power that it has to function. For this, we plan to use a simple L7805 voltage regulator to output the 5V +/- 0.2V, primarily for our microcontroller. We chose this regulator for the same reason as the 12V regulator, it gets the job done at a low cost. It's worth noting that our microcontroller has a raw input pin, with a built in voltage regulator, that can handle up to 12V raw. However, since the output from our 12V voltage regulator can potentially slightly exceed this 12V limit, we've chosen to use the output of our 5V regulator, that we have for our alcohol gas sensor, for the microcontroller too.

For current requirements, our 5V voltage regulator needs to support both the alcohol gas sensor and the Arduino. The maximum current between these two components 310 mA. Since our 5V voltage regulator can output 1.5A maximum [14], this should be more than feasible.

One potential issue that we might have, which is discussed additionally in our tolerance analysis section, is that the alcohol gas sensor species a 5V +/- 0.1V input, which is tighter than the range of our 5V +/- 0.2V. Initial experimentation shows that the alcohol gas sensor should work fine with our planned voltage regulator, however, we've also identified a buck-boost converter [15] that we could integrate into our project that would provide us 5V +/- 0.05V as a backup. We are trying to avoid using it because of its additional cost vs. the 5V voltage regulator.

Requirement	Verification
5V voltage regulator can output 5V +/- 0.2V as long as the battery is at >10% capacity.	Test output voltage with a voltage meter at various battery levels to confirm this is true.
5V voltage regulator is capable of supplying adequate current.	Attach 5V voltage regulator to the microcontroller and alcohol gas sensor and confirm it is capable of supplying 310 mA.

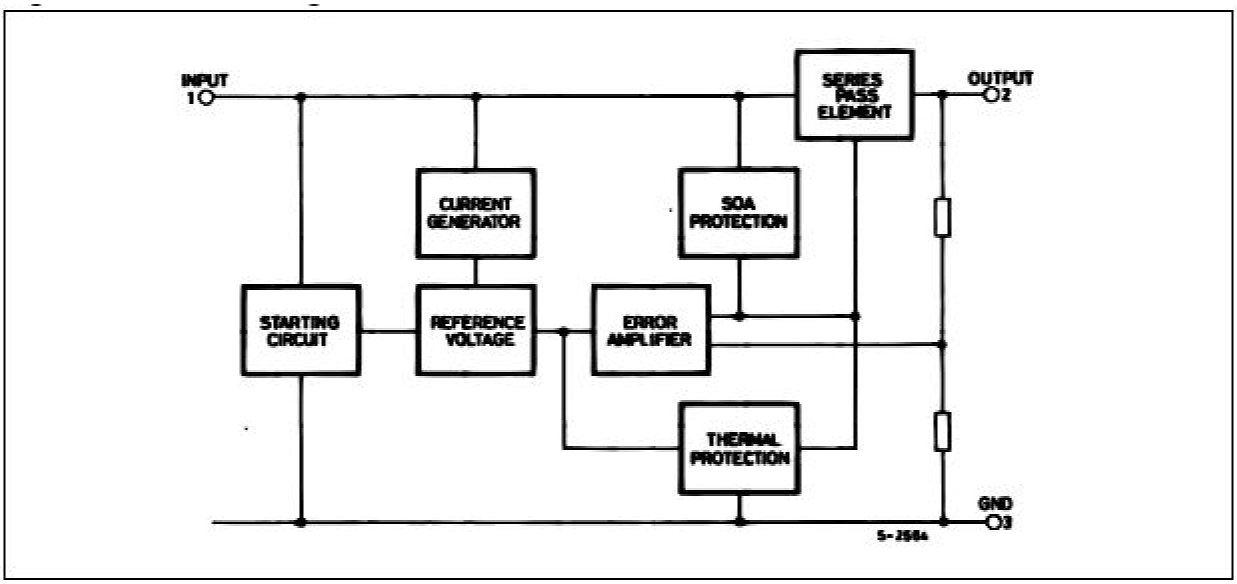

Fig 9. 78xx Internal Schematic (same as the 12V regulator) [14]

2.3.8 5V DC / DC Converter

The purpose of the DC/DC converter is to convert the 12V output from our battery to the 5V that our microcontroller, alcohol gas sensor, switches and LEDs use. For this we plan to use a DC / DC convert (R-78E), that's high enough efficiency that we don't need to worry about using a heat sink on the converter. We chose this regulator over a buck / boost converter board because, while it is slightly less efficient (91% instead of 95%). Since this efficiency difference amounts to 0.155 watts out of 21.875 watts (0.7%), we feel that is a worthwhile trade off for the smaller footprint and lower cost of the DC/DC converter. [16]

The current requirements for the 5V DC/DC converter are:
150 mA - Arduino Pro Mini 328 5V

160 mA - MQ-3 Alcohol Gas Sensor
= 310 mA / 0.91 (efficiency) * 1.10 error margin = 375 mA

Since 375 mA is well within the DC/DC converter output limit of 1000 mA, we have ample room in terms of the R-78E being able to supply sufficient current.

Requirement	Verification
Outputs voltage at 5V +/- 0.25V (based on datasheet) assuming battery at >10% capacity.	Connect to 12V input from 12V regulator and measure output with voltmeter.
Capable of outputting current at 375 mA without overheating.	Construct test circuit using 5V output designed to draw 375 mA and confirm the R78-E can provide this.

2.3.9 Solenoid

The purpose of the solenoid is to keep the container locked unless told otherwise. The solenoid acts as the locking system by having the latch stick out when there is no power, and pulling in the latch when there is power. We specifically picked a solenoid with this characteristic (ROB-15324) to save energy, and prevent the device from needing to be powered on for it to remain locked.

Lastly, we will need to confirm that the solenoid can be powered within the working range of our 12V voltage regulator (12V +/- 0.5V). The datasheet for our chosen solenoid [13] simply says 12V input without giving a range, so one of our first tests will be to verify the +/- 0.5V range.

Requirement	Verification
Solenoid must retract to allow the box to open when powered with 12V +/- 0.5V.	Power the solenoid with 12V and confirm that the box can be subsequently locked.
Solenoid must be able to lock the device when powered off.	Once the solenoid is in the device, with no power being supplied, confirm the device is locked and the lid is unable to open.
Solenoid must be able to maintain device locked state despite movement / being carried around.	Solenoid in device will be carried around in a backpack on (3) 1 mile runs to confirm that it can hold box locked when powered off.

2.3.10 Relay

The purpose of the Relay is to allow the output from our microcontroller to control the power to the 12V solenoid and therefore unlock it as required. We are using a relay for this because the power and voltage output from the microcontroller by itself isn't sufficient to control the Solenoid we are using which requires 12V and 0.6A.

Requirement	Verification
Upon receiving a signal from the microcontroller, relay can activate solenoid.	Manually power the relay and confirm that the solenoid retracts.
Relay prevents solenoid from activating, even with device on, if no signal from microcontroller.	Manually power device without sending output from microcontroller and confirm that the solenoid remains unactivated.

2.3.11 Alcohol Gas Sensor

The alcohol gas sensor is the fundamental block on which our project rests. The purpose of the alcohol gas sensor is to react to the presence of alcohol in someone's breath and subsequently send a value that we can reliably use to determine the blood alcohol content of that person. Using that value will then determine how the rest of the device functions. We didn't have much of a choice in choosing this part, we need some way to detect alcohol for the breathalyzer and this was the only component we could find. We analyze this component, including initial experimental results in our tolerance analysis section.

Testing the sensor involves measuring for a change in the resistance value of the sensor, which should occur based on exposure to alcohol. As we show in our tolerance analysis section, the sensor should be able to distinguish between blood alcohol content to a specificity of 0.0125%. This should be visible by exposing the sensor to two solutions with an alcohol gas concentration difference of equal to or more than 0.0125% and seeing a difference in the resistance value across the sensor. We plan to experimentally this sensor by either using an online blood alcohol content calculator, drinking to a particular blood alcohol content, and reading the output value from the sensor, or by exposing the sensor to an evaporated concentration of ethanol and then measuring the change in resistance value across the sensor from that.

Requirement	Verification
Alcohol gas sensor and circuitry can distinguish between someone blowing into the breathalyzer and someone not blowing into the breathalyzer to prevent it from opening when no one is blowing into the breathalyzer.	Read resistance from the alcohol gas sensor and circuitry and confirm a difference in the alcohol gas sensor resistance value between when someone blows into the sensor and when no one is blowing.
Alcohol gas sensor provides sufficient precision to detect between below and above 0.08% blood alcohol content with resolution of 0.02%.	Create alcohol solutions with a concentration of 0.07% and 0.09% and confirm that the alcohol gas sensor resistance value changes detectably between these two values.
Alcohol gas sensor can adequately detect blood alcohol content up to 0.20% in increments of 0.02% so that we can have a value for the LED.	Create alcohol solutions with concentrations varying by 0.02% increments and then test the alcohol gas sensor at each increment to read the resulting resistance values and confirm a difference.

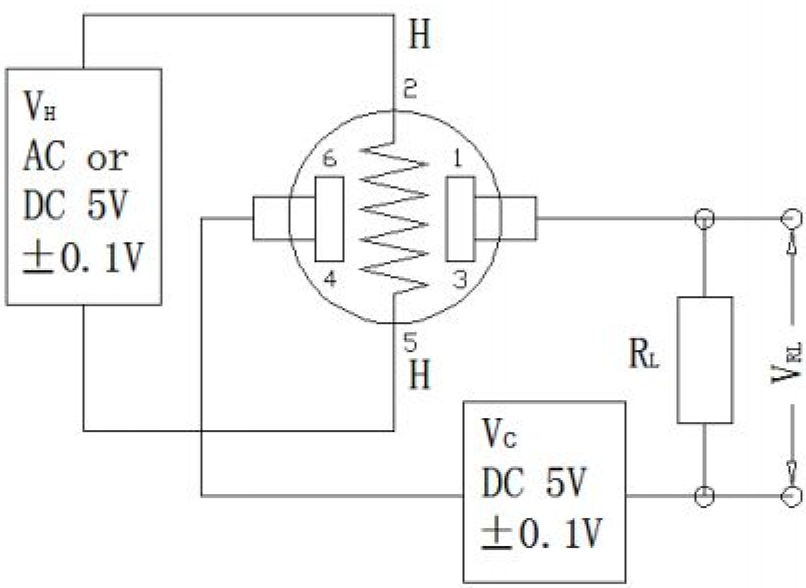

Fig 10. Basic testing circuit for MQ-3 Test Circuit [12]

2.3.12 LED Display

The purpose of the LED Display is to output to the user their current blood alcohol content so that they can understand how close they are to no longer being able to unlock their device, or how far they are above the threshold where they can unlock the device. We chose a simple 7 segment display for this we are only displaying numbers, so this is all that we will need. For simplicity purposes and due to the limitations of our microcontroller, we will have the LED display only communicate with the microcontroller using serial communication.

According to the datasheet [11], the Arduino is capable of outputting 5V per pin at 20 mA, so the Arduino should have sufficient power to power the LEDs.

Requirement	Verification
The output voltage from the microcontroller is sufficient to power the LED display.	Use a multimeter to confirm the microcontroller can supply the LED with a minimum of 3.3V at 3.8mA.
The LED Display is able to communicate with the microcontroller through serial communication.	Use oscilloscope to confirm that we can send a serial output from the Arduino and that the LED Display can receive it.
The LED Display can adequately display the blood alcohol content outputted by the microcontroller.	Send various numbers from the microcontroller (00.00, 00.01, etc.) over serial protocol to confirm the LED displays these.

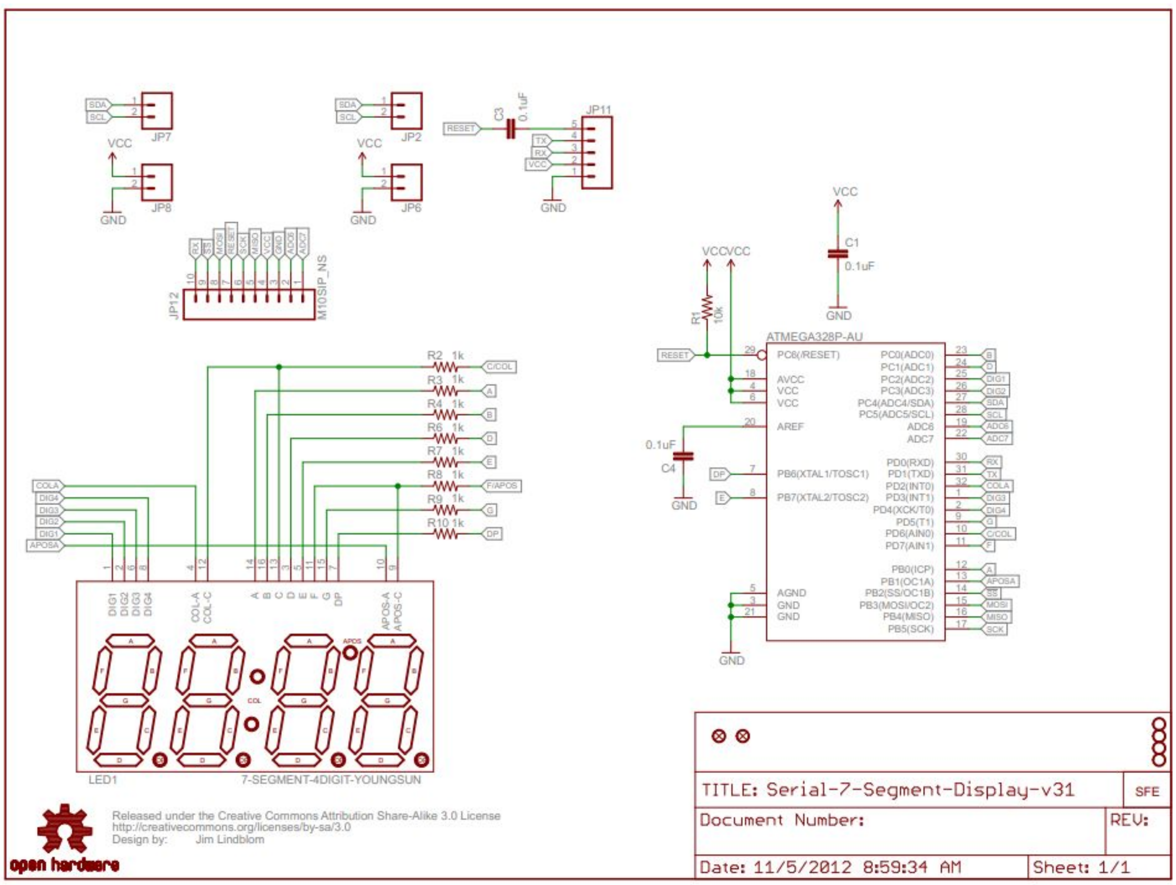

Fig 11. 7 Segment LED Display Schematic Pin Schematic [17]

2.3.13 RFID Case

The core use case of our project is that it prevents someone from starting their car without unlocking the box. As a result, the breathalyzer box will need to be RFID proof in order to stop someone from having a push-to-start car and not needing to unlock the container. We plan to use a wooden box with the key camber being surrounded by metal when the box is closed. The metal will be something like copper or aluminum foil that's low cost and RFID blocking.

Our physical design shows more about what the case looks like.

Requirement	Verification
When the key is inside the case, and the case is closed, a push-to-start car must not be able to start regardless of where the closed case is positioned.	Put push-to-start key in box, close box, and make sure the car cannot start. Test by positioning box in multiple different places around the car.
When the case is closed, the user can still blow into the breathalyzer from the port on the side.	Close the case and verify that the breathalyzer can still pick up values from the user.

2.4 Overall Circuit Schematic and PCB

The purpose of our PCB will be to hold our power supply logic, lock mechanism relay connection, and passcode switch logic. Our microcontroller will be kept separate as part of the Arduino Pro Mini 328 5V circuit board and contain connections to our PCB to receive the data.

Our logic behind keeping the physical switches, alcohol gas sensor, and the LED display off of the board is because they will be in different locations on the case (the lid and the side). Our logic behind keeping the Arduino Pro Mini 328 5V off of the PCB is that it increases the size of the PCB significantly, restricting how we place our PCB in the container, without providing significant benefits.

The reason we chose to hardcode the passcode rather than having the option to "set" the passcode is because we didn't want someone to get their hands on the box and be able to reset the passcode. If anyone could simply reset the passcode that would make the passcode a useless component of the box. It also simplifies our software, by allowing us to have a single line from the output of the passcode on the PCB to the arduino that communicates whether or not the passcode is correct. Lastly, it speeds up our analysis, we will constantly have an input into the Arduino that tells us at any moment whether the passcode on the side of the box is correct. By using 8 binary switches, there are 256 possible passcodes, meaning in our use case it is difficult for someone to guess the passcode.

Additionally, by making our PCB more complicated, our group could run into a situation where the entire success of the project depends on the success of printing the PCB and ordering it, an area in which our group has no experience. By keeping our PCB modular, should something go wrong with it, we can have an easier time finding a workaround.

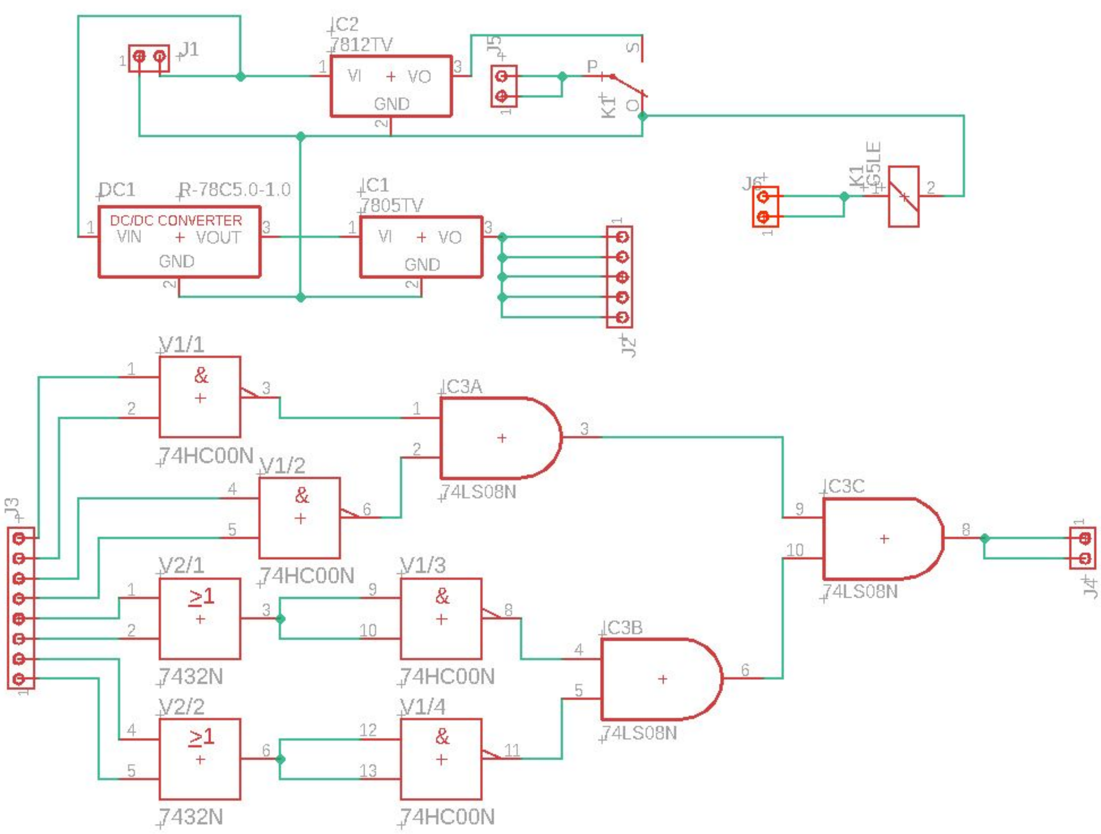

Fig 12. PCB Schematic - Eagle CAD

As is visible in our schematic above, the top half consists of our power regulator circuit and the connection to our relay. The PCB has incoming connectors for ground and the 12V output of our battery. This goes to the 12V regulator and 5V DC/DC converter, which goes to the 5V regulator. The output of the 5V regulator goes to board connectors, so it can be withdrawn, and sent to the various components of our circuit. The output of the 12V regulator is one of the inputs into our relay switch circuit. The coil of the relay is powered by an off board connector (the output from the Arduino Pro Mini 328). When the relay is powered, it flips to the 12V side. The P part of the relay switch, goes to a connector, which goes to the solenoid.

On the bottom is our passcode switch. Here, we have 8 incoming off board connectors, the output from the 8 passcode switches. For the purposes of this schematic, the passcode is [11110000] from top to bottom. The switch output is sent through the circuit logic and then the output goes to a connector, which will go to our Arduino. As a result, if this output is 1, the Arduino can know that the user has entered the correct output.

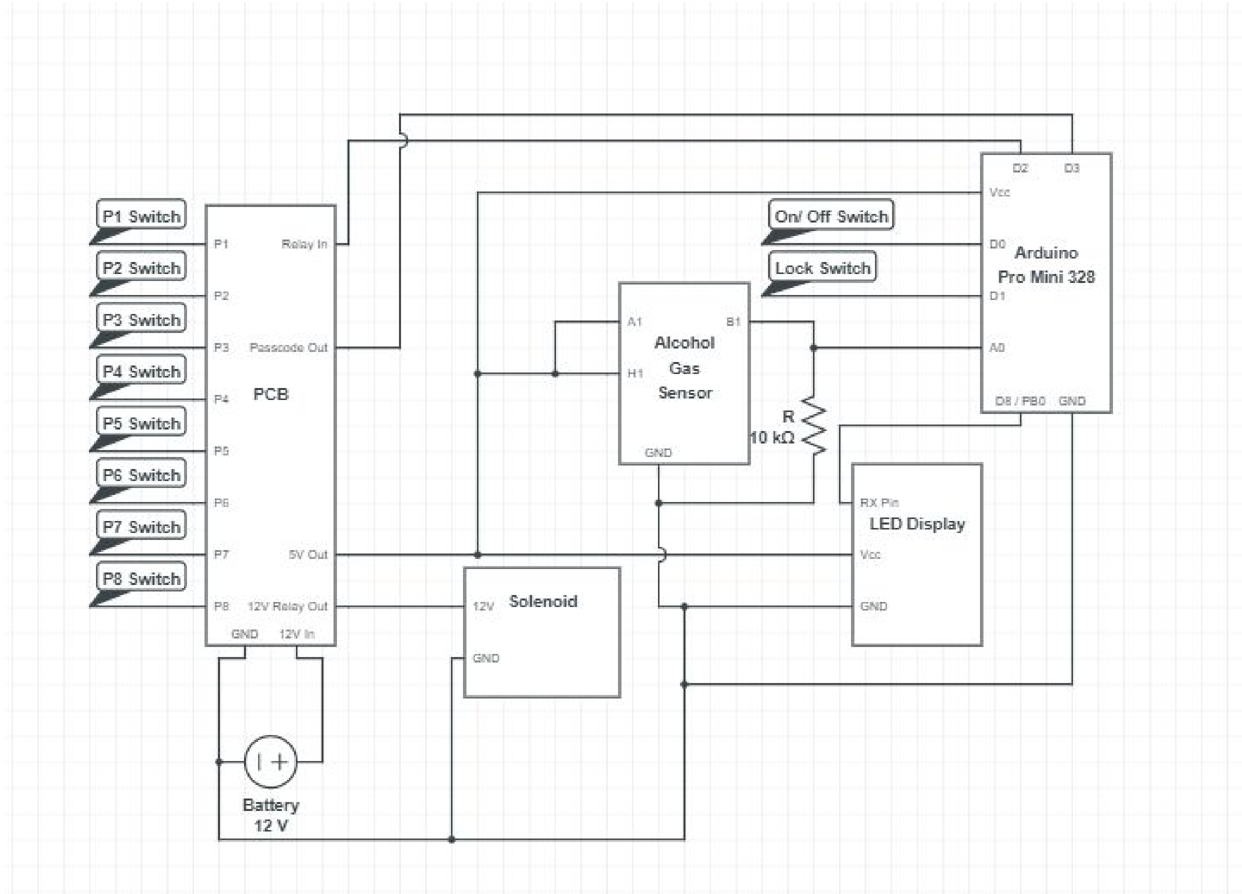

Fig 13. Overall Circuit Schematic

The above image shows our overall circuit schematic after having turned our PCB schematic above into a single block. The named switches above will be simple SPST switches on the side of our box. The passcode switches go into the PCB along with the power source and the relay input. The passcode out is the line that verifies the passcode and goes to the Arduino Pro Mini D3 pin. The On / Off switches and Lock Switch also go into digital pins for the Arduino Pro Mini.

The D8 / PB0 pin on the Arduino Pro Mini acts as the serial line and connects to the RX pin of the LED. The D2 pin on the Arduino Pro Mini acts as the relay input for the PCB which then has a 12V relay output to control the solenoid. Lastly, the A0 pin for the Arduino Pro Mini measures the voltage between the B1 pin output of the Alcohol Gas Sensor and the 10 kiloohm resistor.

This is because the 10 kiloohm resistor acts as a voltage divider with the changing resistance of the alcohol gas sensor.

2.5 Tolerance Analysis

The most fundamental part of our breathalyzer is the alcohol sensor. Fundamentally, our project is a tool to prevent access to the user's keys based on their blood alcohol content. As a result, for our tolerance analysis, we chose to investigate the alcohol gas sensor that we're using (the MQ-3 sensor). Based on the datasheet for this sensor [12], and the requirements of the project, we have come up with 3 different requirements for the sensor.

- The sensor can differentiate between a blood alcohol content in the range of 0.00% to 0.20%, with a maximum error margin of 0.02%.
- The sensor has a noticeable resistance change based on various blood alcohol concentrations.
- The sensor heater can receive power at 5V +/- 0.1V.

For the first requirement, according to the data sheet of the MQ-3 sensor [12] the sensor is capable of detecting alcohol with a concentration scope of 0.05 mg / L to 10 mg / L. In commercial breathalyzers, the generally accepted ratio between alcohol concentration in the breath and alcohol concentration in the blood is 2100 [18]. As a result, taking the concentration scope, that means this sensor is capable of detecting a blood alcohol concentration from 0.01% to 2.1%. This fits well within our requirements to measure above and below 0.08%.

From measuring to a resolution of 0.02%, the data sheet provides the slope of the concentration. As a result, based on that slope, we could calculate the concentration to a desired resolution. We will calibrate the sensor by experimentally determining the Arduino analog input that corresponds to a 0.08% concentration. We will then use the slope provided (0.6) to determine the concentration based on the values that we receive.

Based on preliminary testing, we acquired some information on the specification and accuracy of the sensor. The two goals of this testing to see if the sensor, with the Arduino analog input, could determine blood alcohol content to a resolution of 0.02%. The second goal was to see if we can determine a noticable difference between the sensor reading of someone who is sober against someone who is past the legal limit to be driving.

Since the concentration of alcohol changes the internal resistance of the sensor, we put a 10 kiloohm resistor across the output and measured the voltage drop across the resistor. Since the resistor, with the output, acts as a voltage divider, we could use this, fed into an analog input for our Arduino, to determine the blood alcohol concentration.

Our experiment data from this testing is below:

Capacity	Starting Value	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Avg Diff
Sober	4.81 V	4.74 V	4.72 V	4.75 V	4.74 V	4.72 V	.07 V
4.4 Standard Equivalent (0.10% blood alcohol content)	4.80 V	4.80 V	4.79 V	4.75 V	4.76 V	4.75 V	.03 V

Fig 14. Voltage vs. Drinks

For reference, the resolution on an Arduino analog input is 0.0049V [19], however, our measurements above were conducted with a voltmeter. As can be seen, there is a voltage difference of approximately 0.04V between 0.0% and 0.10% blood alcohol content. That means that given the 8 different voltage measurements possible, at the Arduino's resolution of 0.0049V, we can detect the blood alcohol content to a resolution of 0.0125%, well within our tolerance requirements of 0.02%.

Secondly, based on our tests, we were able to see a substantial difference between the stable value of the sensor in the room versus the breath of someone who was intoxicated. This proved our basic hypothesis, but further experimental testing needs to be done to find the boundary between legally under the limit and past the limit for calibration purposes. We also plan to do additional experimentation, with the Arduino to determine the reliable resolution of the alcohol gas sensor in terms of our project.

One final issue that could potentially arise is from the generally accepted ratio between the ratio of blood alcohol content in the breath and the blood. The currently accepted value from this ratio is 2100 [18], and it is the same ratio used in commercial blood alcohol content sensors. However, this actual ratio can vary in people from 1300 - 3100 [20]. There is currently no accepted workaround for this issue in the industry, and the 2100 value is used as a given. Given that that's the industry standard, that is the value that we will go with. However, we plan to do additional research around potential issues with this problem.

The current sensor we are using for gas detection requires two sources of power per sensor. The first source of power is directed into the heating source to help heat the sensor and provide more accurate results. The other power line is directed into the circuit as shown in the design subsystem section of this report. Both of these sources require 5.0 V of DC power but are able to withstand a tolerance of up to +/- 0.1 V [12]. Our current 5V regulator converts 12V to 5V regulated voltage source that has a tolerance up to +/- 0.2V [14]. We have done some preliminary testing and the current voltage regulator works with our alcohol gas circuit, so we believe that our 0.2V variance in output will be fine.

On the off change the tolerance of our regulator turns out to be incompatible, we have found a Buck Boost Converter (TPS63070) that can provide 5V with a tolerance up to +/- 0.05V [15].

There are a few final points of consideration in regards to our chosen sensor. One important point of consideration is that the data sheet for the alcohol gas sensor, the MQ-3, mentions that the sensor readings become most accurate when the sensor is allowed to heat up for 24 hours [12]. Based on our experimentation above, we believe we can get accurate values by standardizing how long the heater is on before using it. For example, we can calculate our cut-off values using the assumption that the user leaves on the heater for 1 minute before blowing into the breathalyzer, and include that as part of the calculations.

Alternatively, we can set up the circuit such that the heater is constantly powered regardless of whether or not the breathalyzer is on. The battery that we have selected to use is a 72 watt-hour battery, and our other calculations in requirements and verification showed that the minimum battery capacity we need is 21.875 watt-hours. Powering the heating element for 24 hours would use 19.2 watt-hours (24 hours * 160 mA * 5V) of power, meaning our selected battery has ample capacity to change it and allow us to power our heater ahead of time.

The last thing we need to consider is the method of blowing air because a variance in breath temperature can cause issues with reading. We need to test the impact of breath temperatures on the blood alcohol content data collection. However, so far, we have gathered data by placing a straw over the sensor and blowing into it for a 3s duration. Since straws are often easily accessible when people go out drinking, we plan to use further experimentation and testing, but are considering designing the breathalyzer to be used with straws.

2.6 Future Work

2.6.1 Future Work Overview

As a team, we feel that our project has significant potential to be a useful device in the hands of consumers someday, and that one of the best ways to realize that potential is through coming up with additional features we can add. As a result, we have decided to add a goal for future work, that we will work to add to our project, time permitting.

Our future work for our project is to provide users with a way to calculate whether or not they can consume another drink based on the time they wish to travel home. The user will enter their weight, height, age, and gender. Those will be stored in the Arduino Pro Mini's memory. The user will then be able to enter a time they want to leave, in terms of hours into the future.

Based on that information, and the stored information about them, the display will show them the number of additional standard equivalent drinks they can consume.

2.6.2 Future Work Detailed Breakdown

The detailed breakdown for our future work setup would be as follows:

1. User presses the setup switch.
2. LED prompts user for necessary information (height, weight, gender).
3. Information is entered on keypad (wait for user) and then saved to the EEPROM of the device.
4. User flips back setup switch.

Subsequently, should the user choose to see how many remaining drinks they have remaining, the process would be as follows:

1. User presses remaining drinks switch.
2. User blows into a breathalyzer.
3. Microcontroller, upon calculating blood alcohol content, uses the saved information to output the number of remaining drinks the user can consume on the LED.

The keypad we have selected uses a series of outputs. A connection between two of these outputs is considered closed when a key is pressed and depending on which two outputs is closed, we know which row and column (i.e. which key) the user pressed. Utilizing this device would require 7 open lines on our Arduino Pro Mini 328, which we expect to have ample room for.

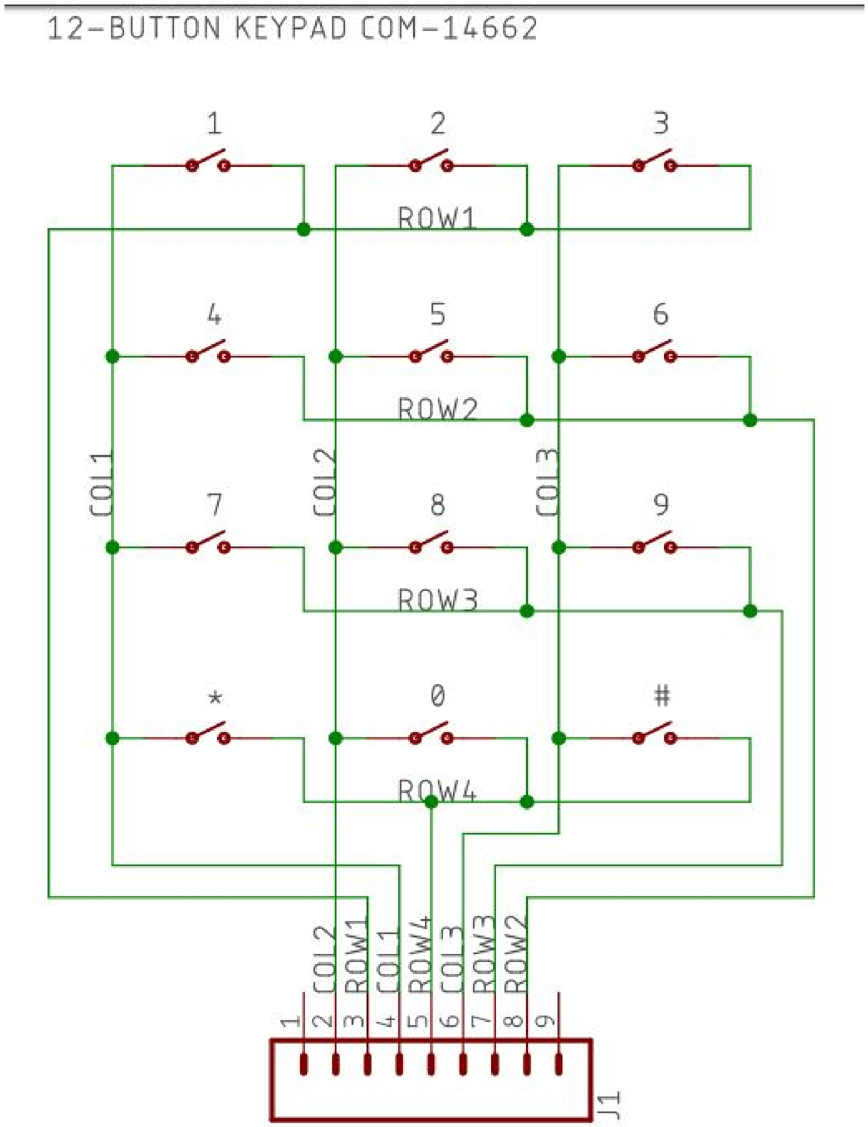

Fig 15. 12-Button Keypad Internal Schematic [21]

2.6.3 Future Work Cost and Schedule

Should we have additional time, during the final week before mock demo week, we plan to attempt to work on our goal for future work. As a result, the maximum time we will contribute to this goal will be 36 hours (1 week times 12 hours / each of work). Should we not have the time to complete it, we hope to at least be able to gather additional information about how to incorporate this aspect of the project in the future, which we can present in our final presentation.

For additional parts we need the following:

Item	Quantity	Total Price
12 Button Keypad	1	$4.50
Rocker Switch	2	$1.90

Fig 16. Future Work Cost & Item Table

Total Cost: $6.40

2.6.5 Future Work Ethics

This future work does have a significant ethics issue that should be discussed. Even at a blood alcohol content of 0.07%, which is just below the legal limit, symptoms are described as, "Speech, memory, attention, and coordination impairments are probably more noticeable at this point, and you may have difficulty concentrating." [22]

This extension, while useful for someone seeking to go out and then be able to retrieve their keys and drive home, could potentially encourage operating a motor vehicle as close to the legal limit as possible. At this level, there could still be unintended consequences when operating a motor vehicle, over remaining sober, and we feel it could be unethical to encourage a user to drive at this level over driving closer to sober, even if it is technically legal.

Another ACM Code of Ethics, we need to guarantee that our product will "do no harm." [23]. Being under the influence can cause a lot of damage to the person drinking but also to the other people, for example victims of drunk drivers. The mental difference between 0.07% and 0.08% blood alcohol contest is minimal, and encouraging those going out to stay close to this limit as possible could violate this clause.

3. Cost and Schedule

3.1 Cost Analysis

To calculate the cost of the project, we have two components, labor and equipment. For labor we have 3 people on our team and 16 weeks across the course of the semester. Based on the 2017 - 2018 new grad report [24], the 50th percentile of Computer Engineer new graduates (everyone in our group) was $93,000, which comes out to $44.7 per hour. We plan to work 12 hours / week on our project. That comes out to a labor cost of $25,747.

Additionally our component costs are as follows:

Item	**Part # / Manufacturer**	**Quantity**	**Total Price**
RFID Case	N/A	1	$10.00 * balsa wood / aluminum & copper*
Seven Segment LED Display	COM-11441 / Sparkfun	1	$13.95
Alcohol Gas Sensor	SEN-08880 ROHS / Hanwei Electronics Co. Ltd.	1	$4.95
Relay	G5LE / Omron	1	$1.95
DC / DC Converter	R-78E / RECOM	1	$4.95
Voltage Regulator 5V	L7805 / STMicroelectronics	1	$0.95
Voltage Regulator 12V	L7812 / STMicroelectronics	1	$1.50
Rechargeable 12V Battery	YB1206000-USB / TalentCell	1	$33.99
DIP Switch - 8 position (passcode)	Switch DIP8 / 4UCON Technology	1	$1.50
Rocker Switch	R1966A / E-Switch	2	$1.90
Arduino Pro Mini 328	Arduino-Pro-Mini 5V / Arduino LLC	1	$9.95
Solenoid 12V	ROB-15324 / Sparkfun	1	$9.95
Quad 2-Input NAND Gate	74LS00 / Electronics Salon	1	$1.06
Quad 2-Input AND Gate	74LS08 / Electronics Salon	1	$1.06
Quad 2-Input OR Gate	74LS32 / Electronics Salon	1	$1.06

Fig 17. Cost and Part Table

Total Part Cost: $98.72
Total Labor Cost: $25,747.00
Combined Total Cost: $25,845.72

3.2 Schedule

Week	Cameron	Kush	Stanley
10/7	Design Review & Incorporate Feedback. Begin to order parts.	Design Review & Incorporate Feedback. Begin to order parts.	Design Review & Incorporate Feedback. Begin to order parts.
10/14	Design PCB for Early Bird PCBWay order.	Design PCB for Early Bird PCBWay order.	Construct case with space for solenoid and test RFID blocking with car keys.
10/21	Continue PCB if required. Otherwise construct and test passcode circuit. Work on individual report. Learn how to work with microcontroller.	Construct breathalyzer circuit and test to determine cutoffs and calculations for blood alcohol content. Work on individual report.	Construct case with space for solenoid and test RFID blocking with car keys. Work on individual report.
10/28	Connect passcode circuit / switch output to microcontroller and test output to lock system.	Construction power circuit with 5V converter and 12V regulator. Test voltage output requirements met.	Construct and test solenoid circuit with relay.
11/4	Work on interfacing LED screen with microcontroller and displaying calculated blood alcohol content value.	Incorporate with rest of the project and test.	Incorporate with rest of the project and test.
11/11	Work on integration and testing. Work on future work if	Work on integration and testing. Work on future work if	Work on integration and testing. Work on future work if

	additional time.	additional time.	additional time.
11/18	Mock Demo / Final Report / Final Demo / Final Presentation Preparation	Mock Demo / Final Report / Final Demo / Final Presentation Preparation	Mock Demo / Final Report / Final Demo / Final Presentation Preparation
11/25 (Thanksgiving Break)	Mock Demo / Final Report / Final Demo / Final Presentation Preparation	Mock Demo / Final Report / Final Demo / Final Presentation Preparation	Mock Demo / Final Report / Final Demo / Final Presentation Preparation
12/2	Mock Demo / Final Report / Final Demo / Final Presentation Preparation	Mock Demo / Final Report / Final Demo / Final Presentation Preparation	Mock Demo / Final Report / Final Demo / Final Presentation Preparation

Fig 18. Schedule

As a team, this schedule is movable and we plan to move team members around as required. For example, no one our team has worked with or designed PCBs outside of the Eagle Cad assignment (our team as a whole has minimal experience with electronic work) so we anticipate this might require additional work. We also anticipate because of the complicated nature of working with it that getting accurate values from the breathalyzer and using it might present additional difficulties. Given our belief in the potential value of this project, our overall goal is to have a fully functioning prototype by the end of the semester.

4. Discussion of Ethics and Safety

Our project is much smaller and doesn't deal with many substantial safety or ethical dilemmas, but there are still some considerations that we need to account for. These come in a myriad of legal, ethical, and health based issues.

First and foremost, since our project is intertwined with alcohol consumption we need to make sure that our product is tested with people who are of legal age in the country of testing. In our case, we are only using people who are over the age of 21 to test our product. This ensures we comply with all legal legislature in the area of consumption to prevent any sort of legal issues with regards to underage drinking. We also want to make sure all alcohol consumption in regards to testing is controlled and completed responsibility to ensure no dangerous health conditions on the tester.

Since our consumer product will require a breathing apparatus, we want to make sure that the module in which consumers breathe is clean or easy to clean. Having a clean breathing module will help protect consumers from any sort of bacteria development and potential health effects. From our testing, using a clean straw provided us better results that reusing the same dirty straw.

To provide the most accurate reading, we will ensure that the module is able to be separated and cleaned.

Our product will deal with some electronic components, so it needs to be enclosed in a tight manner to prevent it from getting damaged from the elements. Some potential elements that could cause problems are water/drinks or powders like salt and sugar. Making our product element-proof well help provide more longevity to the product as well as prevent any sort of untimely malfunctions.

We will be using an internal battery to power this product throughout the duration of its usage. We have opted to buy a rechargeable battery so we can allow this product to be taken on the go. Because of this, we want to make sure that the battery and voltage supplied to the rest of the circuit is of safe amounts. We can test this by measuring voltage values at various positions within the circuit to and validating that with the expected values.

Referring to the ACM Code of Ethics, all products developed using modern technology need "Honor Confidentiality." [23] Our device does deal with some sensitive consumer data, and to honor confidentiality, we will need to use the passcode to verify the information of the user. If we do decide to pursue our future work, we will keep all biometrics saved and safely away from display so that comply by that rule. All information collected will not be used outside the scope of this project.

Another ACM Code of Ethics, we need to guarantee that our product will "do no harm." [23]. Being under the influence can cause a lot of damage to the person drinking but also to the other people, for example victims of drunk drivers. While the legal limit is a blood alcohol content of 0.8, we will have a safety tolerance to account for the potential errors in calculation by the sensor. The mental difference between .079 and .08 is almost non-existent, and to prevent any bad decision being made especially during the use of our product, we will err on the side of caution. As an extension, in the testing phase of this product, if we are consuming any drinks ourselves we need to make sure we take the necessary precautions to stay safe from the potential negative effects of alcohol consumption.

5. Citations

[1] *Fight back against misinformation. Get the facts*, M.A.D.D. No More Victims, May 2014. Available at: https://madd.org/statistics/

[2] *Impaired Driving: Get the Facts*, Center for Disease Control and Prevention, 2019. Available at: https://www.cdc.gov/motorvehiclesafety/impaired_driving/impaired-drv_factsheet.html

[3] *Lowered Inhibitions and Bad Decisions from Alcohol*, Alcohol.org, An American Addiction Centers Resource, 2019. Available at: https://www.alcohol.org/effects/inhibitions/

[4] *How to install an ignition interlock device,* IgnitionInterlockInfo, 2019. Available at: https://ignitioninterlockinfo.com/installation-removal/how-to-install-an-iid/

[5] *40% of Americans don't have $400 in the bank for emergency expenses: Federal Reserve*, ABC News, May 24, 2019. Available at: https://abcnews.go.com/US/10-americans-struggle-cover-400-emergency-expense-federal/story?id=63253846

[6] *Volunteer Ignition Interlock Breathalyzer Device (IID) Reviews,* Amazon, August 2018. Available at: https://www.amazon.com/Volunteer-Ignition-Interlock-Breathalyzer-Device/product-reviews/B01LZO2OPC/ref=cm_cr_dp_d_show_all_btm?ie=UTF8&reviewerType=all_reviews

[7] *A brief history on ignition interlock & breathalyzer devices*, IgnitionInterlockInfo, 2019. Available at: https://ignitioninterlockinfo.com/ignition-interlock-breathalyzer-devices/device-history/

[8]. Kelley-Baker, T., Taylor, E., Berning, A., Yao, J., Lauer, L., & Watson, D. (2017, June). *The feasibility of voluntary ignition interlocks as a prevention strategy for young drivers.* (Report No. DOT HS 812 425). Washington, DC: National Highway Traffic Safety Administration.

[9] *Arduino-Pro-Mini Schematic*, Sparkfun, April 9, 2014. Available at: https://cdn.sparkfun.com/datasheets/Dev/Arduino/Boards/Arduino-Pro-Mini-v14.pdf

[10] *Arduino Pro Mini: Power Consumption*, IOT Experiments, 25 August 2016. Available at: https://www.iot-experiments.com/arduino-pro-mini-power-consumption/

[11] *Arduino Pro Mini (DEV-11113): Graphical Datasheet*, Sparkfun, 2019. Available at: https://cdn.sparkfun.com/assets/d/5/2/f/0/ProMini16MHzv2.pdf

[12] *Technical Data, MQ-3 Gas Sensor*: Sparkfun / Hanwei Electronics Co. Available at: https://www.sparkfun.com/datasheets/Sensors/MQ-3.pdf

[13] *Cabinet Lock11, Cabinet Deadlock Bold*, Sparkfun, 2019. Available at: https://cdn.sparkfun.com/assets/9/b/9/e/0/NI-11_instruction.pdf

[14] *L7800 Series. Positive Voltage Regulators*, Sparkfun, November 2004. Available at: https://cdn.sparkfun.com/datasheets/Components/General/TO-220.pdf

[15] *TPS63070 2-V to 16-V Buck-Boost Converter with 16A Switching Current*, Texas Instruments / Sparkfun, June 2016, Revised March 2019. Available at: https://cdn.sparkfun.com/assets/e/4/a/1/4/TPS63070_DataSheet.pdf

[16] *R-78E-1.0 DC/DC Converter RECOM*, Sparkfun, April 2017. Available at: https://cdn.sparkfun.com/assets/0/f/7/3/e/R-78Exx-1.0.pdf

[17] *Serial 7 Segment Display Datasheet*, Sparkfun Github, November 2017. Available at: https://github.com/sparkfun/Serial7SegmentDisplay/wiki/Serial-7-Segment-Display-Datasheet

[18] *Content: The Breathalyzer Assumes a Specific Blood-to-Breath Ratio to Calculate the BAC*, The Alcohol Pharmacology Education Partnership. 2019. Available at: https://sites.duke.edu/apep/module-4-alcohol-and-the-breathalyzer-test/content-the-breathalyzer-assumes-a-specific-blood-to-breath-ratio-to-calculate-the-bac/

[19] *Arduino analogRead()*, Arduino Analog I/O, 2019. Available at: https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/

[20] Alobaidi TA, Hill DW, Payne JP (1976). "Significance of variations in blood: breath partition coefficient of alcohol". Br Med J. 2 (6050): 1479–81. doi:10.1136/bmj.2.6050.1479. PMC 1689868. PMID 793681. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1689868/

[21] *12-Button-Keypad COM-14662*, Sparkfun Electronics, 2019. Available at: https://cdn.sparkfun.com/assets/7/e/f/6/f/sparkfun_keypad.pdf

[22] *Blood Alcohol Content (BAC) Calculation*, Alcohol.org. Available at: https://www.alcohol.org/bac-calculator/

[23] *ACM Code of Ethics and Professional Conduct*, Advanced Computing as a Science & Profession, 2018. Available at: https://www.acm.org/code-of-ethics

[24] *IlliniSuccess_AnnualReport_2017-2018_FINAL,* University of Illinois at Urbana-Champaign, 2018. Available at: https://uofi.app.box.com/s/ml9oh48vawrhm7e019kw5ix4j91j605p