ECE 445 Final Presentation
Breathalyzer Box

Team #1 : Cameron Palte, Kush Patel, Stanley Zhao

TA: Shaoyu Meng | Professor: Wei He

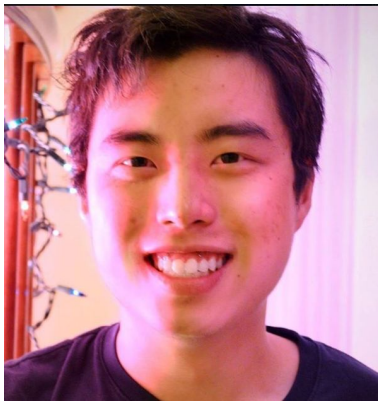
ECE ILLINOIS

ILLINOIS

Team Members

Kush Patel
Computer Engineering
Graduating: December 2019

Cameron Palte
Computer Engineering
Graduating: December 2019



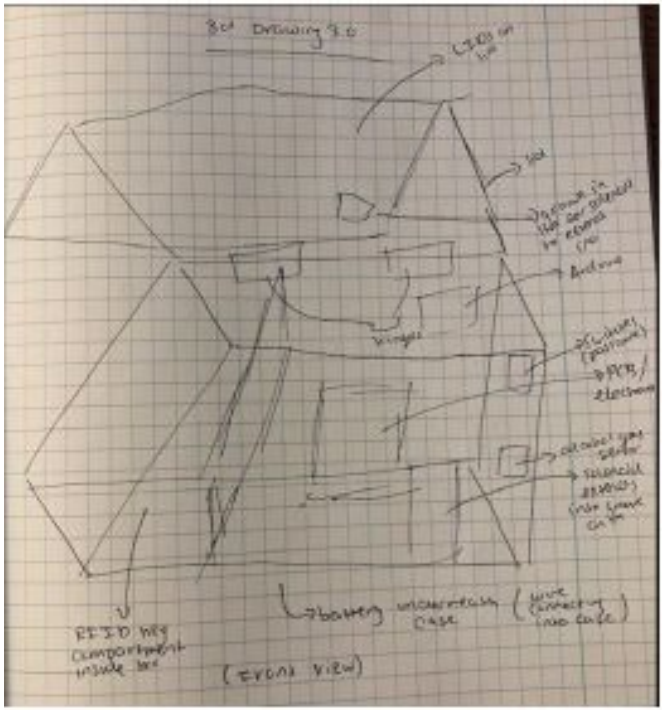
Stanley Zhao
Computer Engineering
Graduating: December 2019

Problem Overview

- Drunk driving kills more than 10 thousand Americans / year.
- Current ignition lock systems are car-based in an increasing age of shared vehicle ownership.
- Volunteer breathalyzers need to be professionally installed and are expensive.
- Can be used by an individual or by a bar or drinking establishment for their customers.

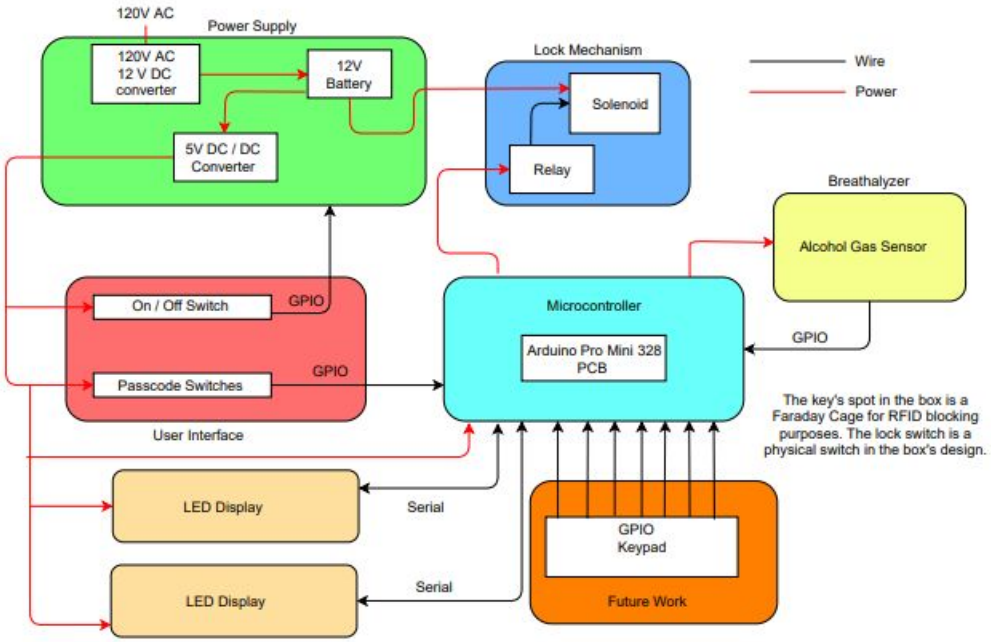
Our Solution

- Box with Faraday cage inside to prevent push-to-start keys.
- Key-based solution. Users place their keys in box and can only retrieve them when sober.
- LED Display on top to notify users of their blood alcohol content.



Original Design - Overview

- Passcode switches to prevent different user from retrieving key.
- Arduino Pro Mini 328 receives alcohol gas sensor and passcode input, sends trigger to relay.
- Power distribution circuit consists of 12V battery and 5V DC / DC converter.
- PCB holds power distribution and passcode circuitry.



Original Design - High-Level Requirements

- 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.
  Result: Met Requirement 70% of the time vs. commercial breathalyzer.
  Critical error rate <4% (detect <0.08 when not or vice-versa).

- An inaccurate passcode will prevent the user from unlocking the breathalyzer box regardless of their blood alcohol content.
  Result: Met Requirement.

- The breathalyzer box can prevent a push-to-start key from being used, and be manufacturable from less than $100 of parts.
  Result: Met Requirement.

Project Build Overview - Stages

Stage 1: Construction and test the separate individual components.

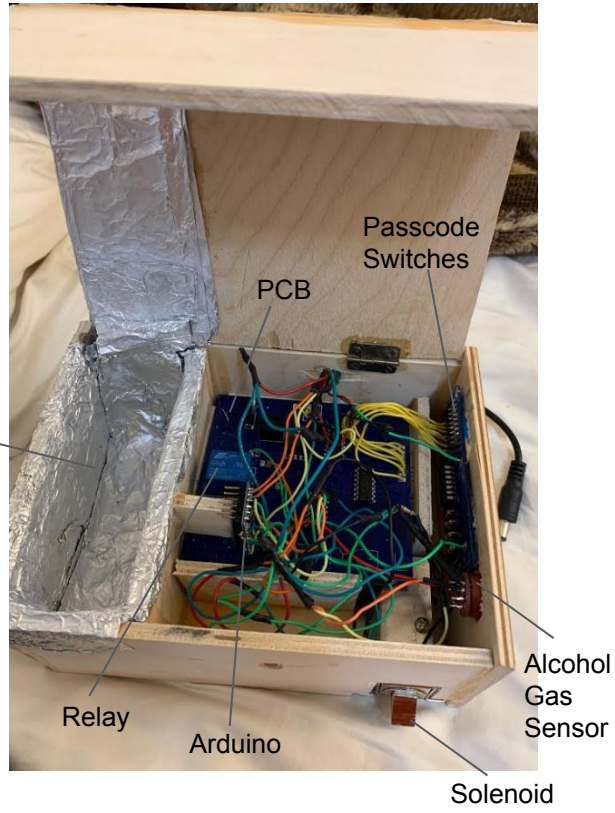
- Passcode Circuit, Lock Mechanism, RFID / Case, Arduino Code, Alcohol Gas Sensor, Power Circuit, LED

Stage 2: Integrate related circuits.

- Lock Mechanism with RFID / Case, Power Circuit with Arduino and LED

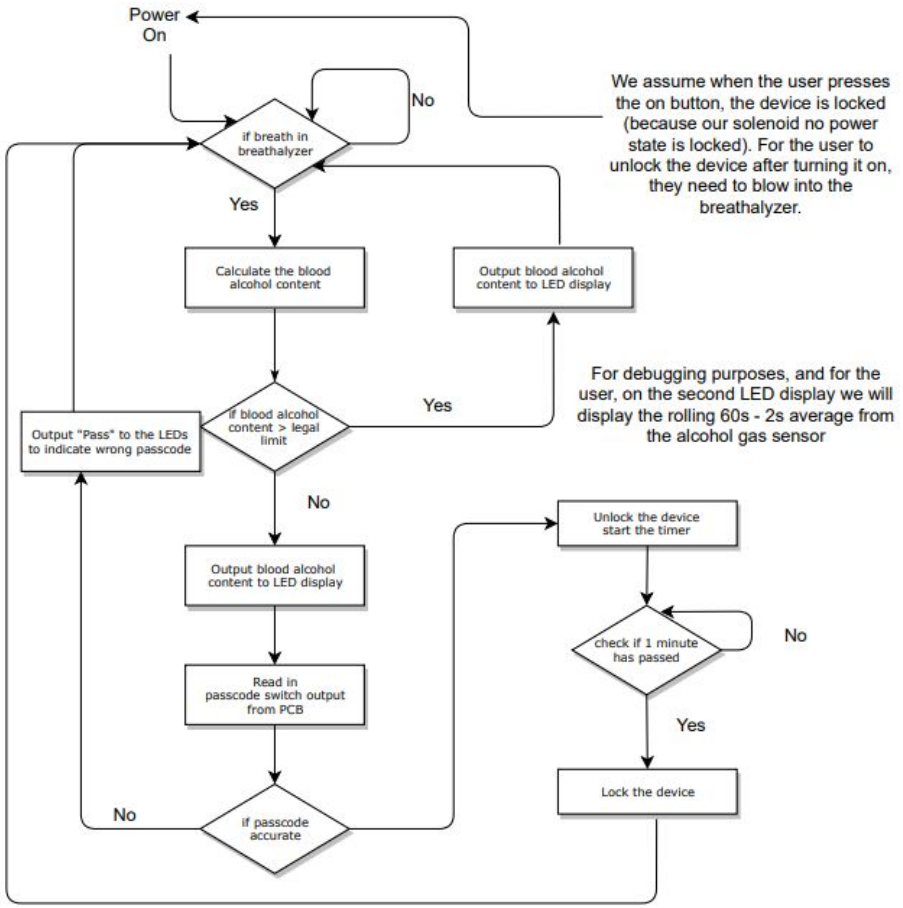
Stage 3: Full integration and testing.

- Gather data of box fully together, and investigate future work.



Project Build (Software)

- Alcohol gas sensor data converted to a blood alcohol content with the 60 - 2s rolling average.
- Thresholds set by comparing to commercial breathalyzer.
- Assume the box starts in locked state when turned on (i.e. always need to breath in to unlock).
- When a breath is detected, blood alcohol content is calculated and sent to LEDs.
- If user blood alcohol content < the legal limit, and passcode correct, box unlocks.
- Solenoid re-extends after 1 minute, box locked when lid is closed.



Project Build (Physical Design)

S

Physical Case constructed in several stages.

- Wood cut from 0.25” balsa wood using a dremel with wood blade.
- Various components screwed together where reinforcement needed. (i.e. two lid pieces).
- Wood glue with super glue used for all piece attachments.
- Copper and aluminum foil super glued along with PCB, alcohol gas sensor, and and switch.
- Solenoid screwed down.
- Groove cut in the lid for solenoid to extend into and lock case.

Requirement & Verification Tables

Power

12V Battery / Wall Charger Requirement	12V Battery / Wall Charger Verification	Results
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.	Verified successfully.
Battery can output 12V +/- 0.5V when fully charged.	Fully charge battery and confirm the required voltage is outputted with a voltmeter.	Verified successfully.
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.	Verified successfully.
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.	Verified successfully.

- 5V DC/DC Output at >40% battery data on next slide.

5V DC/DC Converter Requirement	5V DC/DC Converter Verification	Results
Outputs voltage at 5V +/- 0.25V (based on datasheet) assuming battery at >40% capacity.	Connect to 12V input from 12V regulator and measure output with voltmeter.	Verified successfully.
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.	Verified successfully.

- Passed all Requirement & Verification tests from 5V DC/DC converter and 12V battery.
- 12V battery drop-off as charge decreased - discussed next slide.
- Used resistor network to test current.

Requirement & Verification Tables - Microcontroller and Passcode

- Passed all Requirement & Verification tests for Microcontroller and Passcode.
- Passcode Circuit PCB output = ~4.47 V for correct passcode, ~0.05V for incorrect.

Passcode Requirement	Passcode Verification	Results
User can input their passcode on the switches, while the box is closed.	Close the box and use a voltmeter to confirm that the switches can be set to low or high.	Verified successfully.
The passcode switch logic gates output a 1 if the passcode matches the hardcoded passcode, otherwise a 0 is outputted.	Construct the passcode circuit using the logic gates for the hardcoded [11110000] passcode. Verify that the passcode circuit output is a 1 when the correct passcode is in switches.	Verified successfully.

Microcontroller Requirement	Microcontroller Verification	Results
Based on the inputs into the microcontroller, the correct output is sent to the relay.	Set passcode input and breathalyzer resistance to 1 and <0.08% BAC respectively and confirm that a 1 output is sent to relay. Confirm a 0 output is sent for any other combination of inputs.	Verified successfully.
Microcontroller runs code to read and analyze blood alcohol content automatically upon being turned on.	Provide 5V from regulator and confirm that the Arduino boots and correct code executes, by having the code turn on and off LEDs on the board as it reaches certain stages.	Verified successfully.
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.	Verified successfully.
Microcontroller can adequately receive input from the switches.	Use a multimeter to verify that the microcontroller receives the correct input from switches and PCB.	Verified successfully.

Requirement & Verification Tables Physical Design

- Passed all Requirement & Verification tests for Physical Design.
- Lock switch changed from physical switch to part of container design.
- Added aluminum to original RFID plan of just copper sheet.

Lock Switch Requirement	Lock Switch Verification	Results
Pressing switch causes container to lock.	Take an unlocked container version and press the lock switch. Confirm that the container is subsequently locked.	Verified successfully. Physical switch. Changed to closing lid locks container.
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.	Verified successfully. Once the lid is closed cannot unlock.

RFID Case Requirement	RFID Case Verification	Results
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.	Verified successfully.
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.	Verified successfully.

Requirement & Verification Tables
Alcohol Gas Sensor

- Passed all Requirement & Verification tests for Alcohol Gas Sensor.
- 60-2s difference for 0.07% on commercial breathalyzer avg ~-402.5
- 60-2s difference for 0.09% on commercial breathalyzer avg ~-450.0
- 0.24V difference between 0.07% and 0.09%.
- Average difference between each 0.02% = ~0.35V.
- 60-2s difference for sober breath = ~+58.8 (or +0.294V vs. no difference for no breath.

Alcohol Gas Sensor Requirement	Alcohol Gas Sensor Verification	Results
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.	Verified successfully.
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.	Verified successfully.
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.	Verified successfully.

Requirement & Verification Tables
User Interface

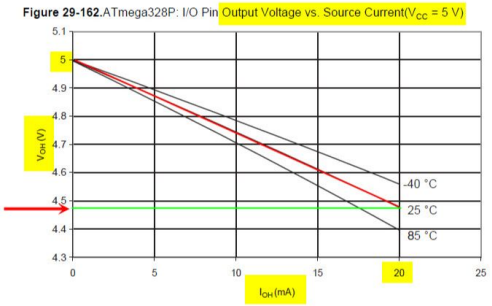
- Passed all Requirement & Verification tests for User Interface.
- On / off switch part of battery.
- LED Display supplied with 5V instead of 3.3V.
  - Still within datasheet limitations, and based off of powering with 5V DC/DC converter.
- Added a second LED Display to container.
  - Currently used for debugging information.
  - Plan to take utilize as part of future work.
  - Passed all requirements of first display.

On / Off Switch Requirement	On / Off Switch Verification	Results
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.	Verified successfully.
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 without needing to execute code.	Verified successfully. (based on requirement - verification was a typo, code needs to execute to verify)

LED Display Requirement	LED Display Verification	Results
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.	Verified successfully. Supplied 5V instead (within LED voltage range)
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.	Verified successfully. Didn't use oscilloscope, simply tested through displaying numbers on LED.
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.	Verified successfully.

Requirement & Verification Tables
Lock Mechanism

- Passed all Requirement & Verification tests for Lock Mechanism.
- Due to battery limitations and regulator issues discussed previously, solenoid can receive as little as 8V.
  - Still functions as expected.



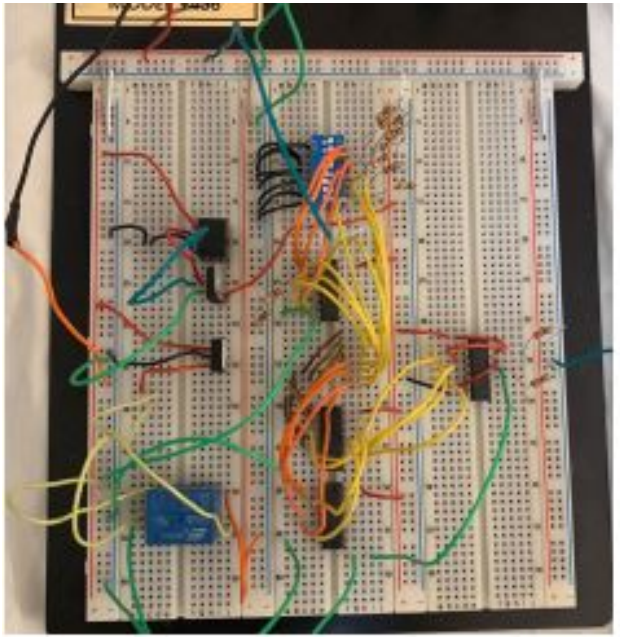
Relay receives ~4.5V from Arduino under load. (needs minimum 3.5V to trigger)

Solenoid Requirement	Solenoid Verification	Results
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.	Verified successfully.
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.	Verified successfully.
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.	Verified successfully.

Relay Requirement	Relay Verification	Results
Upon receiving a signal from the microcontroller, relay can activate solenoid.	Power the relay with 5V from Arduino output and confirm that the solenoid retracts.	Verified successfully.
Relay prevents solenoid from activating, even with device on, if no signal from microcontroller.	Power Arduino while keeping output from microcontroller at 0V and confirm that the solenoid remains unactivated.	Verified successfully.

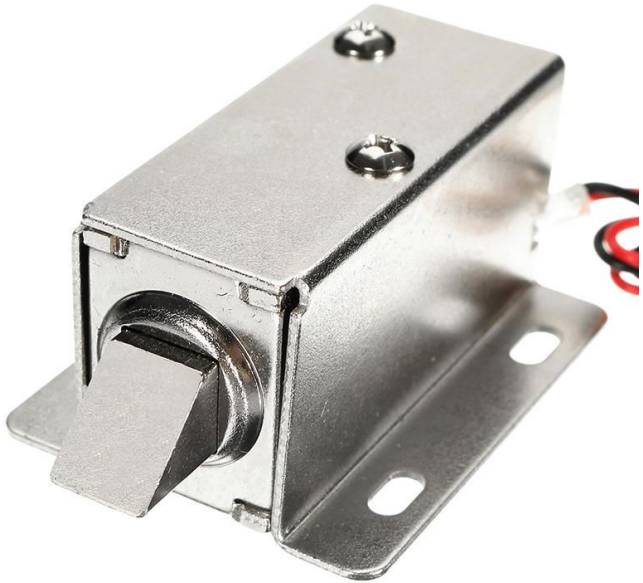
Functional Tests (Arduino, Relay, and Lock, Passcode)

- Two separate test circuits.
  - Passcode
  - Arduino, Relay, and Lock
- Passcode test circuit was used to model what would eventually be on PCBs.
  - Valid passcode = ~4.47V output = strong logical high.
- Arduino, Relay, Lock circuit was to test the ability or Arduino to trigger lock.
  - 100% success rate after resolving current issues (discussed later in challenges).



Challenges
(RFID Blocking & Solenoid Overheat)

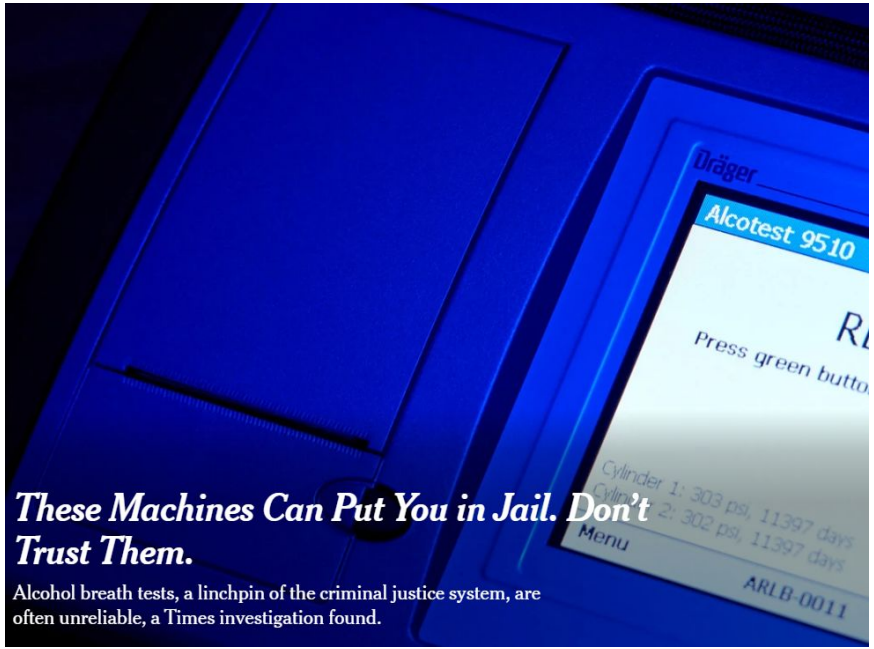
- Initial plan involved RFID blocking using copper sheeting for Faraday cage.
  - Only worked ⅔ times.
  - Gaps too large especially around hinge.
  - Resolved using aluminum foil layers superglued to cover gaps.
- Solenoid became extremely warm to the touch after being on for >few min.
  - Solenoid draws 7.2 watt-hours (0.6 A x 12 V).
  - Resolved by changing code to only unlock for 1 minute - user then relocks by closing the box anytime after the 1 minute has passed.



Challenges (Alcohol Gas Sensor)

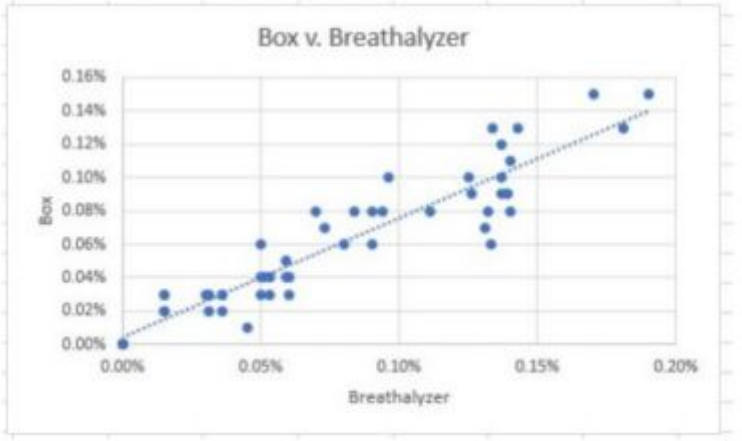
Breathalyzers are inherently inaccurate:

- fuel vapors, humidity, temperature, and ratio assumptions.
- Dramatic improvements after 24 hour warm up (difference between individual BAC levels 50x).
- 70% of values within 0.02% of commercial breathalyzer - room for improvement.



Overall Success and Comparison Data

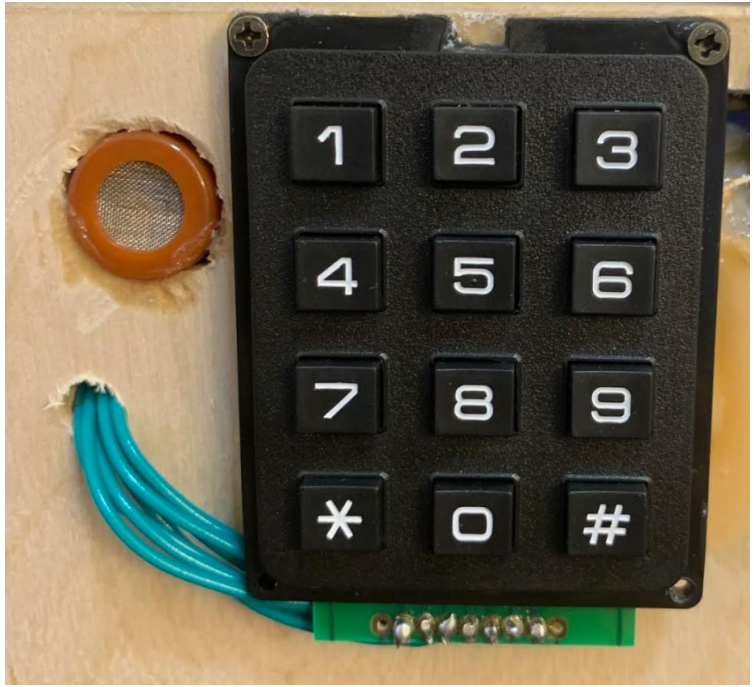
- Overall, project fully functioned as expected and originally envisioned.
- Error rate for blood alcohol content > 0.02% from commercial breathalyzer high
  - Due to lack of data at upper blood alcohol ranges.
- Critical error rate low.



Error Rate (box determines blood alcohol content > 0.02% outside commercial breathalyzer).	30.90%
Critical Error 1 (box determines user is sober when user is above the legal limit)	1.82%
Critical Error 2 (box determines person below the legal limit is in fact above)	1.82%

Future Work

- Functionality augmentation.
- Keypad has already been attached to box to test for future work.
- Plan is to allow user to input their height, weight, and gender.
- They can then press a switch on the box that will take in their next breath and calculate how approximately long until they can retrieve their keys.
- Other future work around gathering more data from alcohol gas sensor, trying new sensors, and improving reliability.



Ethics and Safety

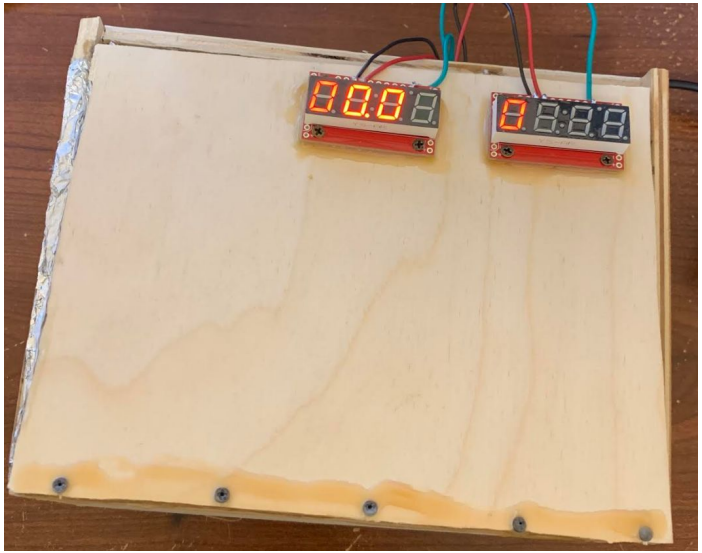
- Most effective way for us to collect data and test project is to test ourselves with drinking and compare versus a commercial breathalyzer.
  - All group members 21.
- Need to make sure breathing apparatus and breathing into box is sanitary.
  - Designed to use straws, so user's mouth does not make contact with device.
- Electronic components in moving box.
  - Superglued PCB and all electronic components to box to prevent from moving around.
  - Used heat shrink to isolate exposed wires.
  - Whenever possible, all electronic parts were kept within box in a separate area from key chamber.
- No consumer data saved about blood alcohol content.
- Attempted to remain conservative with blood alcohol content to reduce risk of Critical Error 1.
- Tested output voltage from battery to make sure it stayed at safe levels.

Conclusion

Questions?

Thank you to our TA, Shaoyu Meng!

Thank you to our Professor, Wei He!



Citations

https://cbsnews1.cbsistatic.com/hub/i/2014/06/16/9dc6188b-e7fe-4000-90fd-404836ad76ee/drunk-driving-police-car.jpg
https://www.nytimes.com/2019/11/03/business/drunk-driving-breathalyzer.html
https://vancouvercriminallaw.com/wp-content/uploads/2017/08/breathalyzers.jpg
https://images-na.ssl-images-amazon.com/images/I/41GAUS%2BVBUL._SX342_.jpg
https://imgaz.staticbg.com/thumb/large/oaupload/banggood/images/CF/51/f9d91d1e-93c0-4dac-a762-c6c5cff70857.jpeg
https://forum.arduino.cc/index.php?topic=275280.0