Proceedings of the 10th International Conference of Control Systems, and Robotics (CDSR'23)
Ottawa, Canada – June 01 - 03, 2023
Paper No. 213
DOI: 10.11159/cdsr23.213

IoT and Cloud Computing Integration to Minimze Drunk Driving Accidents

Harrison Carranza[1], Aparicio Carranza[2], Edwin Tito[2]
[1]Bronx Community College of the City University of New York
2155 University Avenue, Bronx, NY - USA
harrison.carranza@bcc.cuny.edu; acarranza@citytech.cuny.edu
[2]New York City Colleg of Technology of the City University of New York
186 Jay Street, Brooklyn, NY - USA
edwin.tito@mail.citytech.cuny.edu

***Abstract* -** Current studies demonstrate that there are more than 17,000 deaths annually due to drunk drivers in the United States, and in New York State it is estimated that around 30% of fatal crashes are due to alcohol intoxication. We have implemented an IoT embedded system with sensors that measure the blood alcohol level of the driver with a breath detecting system which uses a beam of infrared light to perform a calculation of the alcohol level present in the driver's blood. The hardware and software components have been programmed to control the behavior of the engine upon the data collected from the alcohol sensor. The IoT generated data is transmitted through a communication network using the IBM cloud to send alerts to public authorities and relatives of the intoxicated individual to take appropriate actions.

***Keywords*:** Alcohol Sensor, Breadth Detecting System, Crash Prevention, IBM Cloud, IoT

1. Introduction

Drunk driving is the leading cause of vehicle fatalities in the United States, with approximately more than 17,000 deaths annually. In fact, recent studies over the years have concluded that three in every ten Americans will be involved in an alcohol-related car accident at some instance of their lives [1]. According to the National Highway Traffic Safety Administration (NHTSA), there was a 14% increase in alcohol-impaired driving traffic deaths in 2020 compared to the data from 2019 (Drunk Driving | Statistics and Resources, 2022). Therefore, this reflects a major issue regarding road safety. One may ask: Are the roads getting more dangerous? Besides the unfortunate death and the grievances many families must go through, drunk driving accidents also contribute to large economic costs due to property damage, medical expenses, fines, and legal fees. According to a study done by NHTSA in 2010; an estimated $242 billion was the economic cost for all vehicle crashes; while $44 billion are due to alcohol-impaired accidents including intoxicated occupants [2].

Furthermore, the National Highway Traffic Safety Administration also estimates that in 2020 eighty-four percent of 13,105 drivers had a Blood Alcohol Content (BAC) of 0.08 g/dL, sixty-seven percent had a BAC level of 0.15 g/dL or higher [3]. These results indicate that a great number of intoxicated drivers are above the standard BAC limit of 0.05%. A BAC level of 0.08 percent is scientifically proven to be the level where the human body has become intoxicated and begins to go through the effects; this includes delayed reaction time and blurry vision, and the drivers begin to have less control over their vehicle [4]. Although drunk driving is a criminal offense in the United States, people continue to drive while under the influence. If the law cannot exactly stop drunk driving, then what can? A great solution to this would be technology; technology continues to advance to levels never seen before. There are now many problems that can be solved by using advancements in technology. Our implementation might be considered part of it as it will help to decrease the rate of drunk driving fatalities using a technological strategy to prevent an intoxicated driver from operating the vehicle.

The rest of this report is organized as follows: Section 1 shows the background information and a detailed objective of our effort. In section 2, a breakdown of the hardware, software, and cloud requirements are stated with an overall explanation for each of these components and how they relate to the benefits of our approach, Section 3 corresponds to the circuitry used by displaying the schematics and physical connections required. The results gathered on the experimental part are stated in Section 4. Lastly, our conclusion is included in Section 5.

2. Hardware, Software and Cloud Requirements

The entire system adopts the IoT module as we use an Internet-connected hardware microcontroller, Particle Argon is used to connect the hardware components, specially the MQ-3 Alcohol Sensor is used to identify the alcohol level in a driver and sends out an automated notification to law authorities and relatives. The core hardware modules are Argon Board, Alcohol Sensor Module (MQ-3), Servo Motor, and 16x2 LCD Display. The program to control our system is written in C++ using the online Particle IDE. Furthermore, to retain and share gathered data over the Internet we make use of a Cloud Service, in our case we have used the IBM Cloud Services that goes hand by hand with Node-RED which is a flow-based programming tool for wiring together hardware devices, APIs, and online services. With the use of Node-RED we set up the notification's settings to allow messages and alerts to be sent over the cloud to different end users based on the data triggered by the main hardware components of our system.

2.1. MQ-3 Alcohol Breathalizer

The use of this sensor is essential for our system, since it helps us detect through breathing if the driver is intoxicated. The sensor can detect the existence of alcohol in the air as well as the corresponding concentration which helps to determine the percentage of Blood Alcohol Content (BAC) present in the individual. The MQ3 sensor is very popular and it is considered a Metal Oxide Semiconductor (MOS) sensor due to its sandwich-like structure having a thin layer of silicon oxides in between the metal and semiconductor layers [5]. This sensor operates on a 5V Direct Current consuming approximately 800mW of power, additionally the design of this sensor was made to detect alcohol concentrations on the range from 25 to 500 parts-per-million (ppm) [6]. The unit measure ppm is mostly used for measuring gas concentration and it is a mass-to-mass or volume-to-volume ratio. The associated pins for this sensor are shown in Fig. 1.

- VCC is the Sensor's power supply pin which is connected to a 5V source.
- GND is the ground pin, which is connected to the Argon's ground pin.
- DOUT is the digital output pin; low output indicates no alcohol present; high output indicates presence of alcohol.
- AOUT is the analog output pin, provide us with an analog signal varying between VCC and ground depending on alcohol level detected.

Analysing the data sheet of the sensor provided by Hanwei Electronics CO. LTD [7], it provides data specifications regarding the sensitivity characteristics of the MQ-3 sensor in reference to the different gases it can detect, shown in Fig. 2.

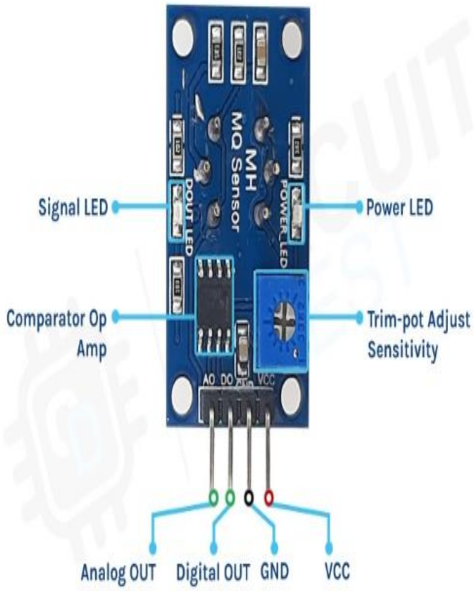

Fig. 1: MQ-3 Sensor Pins

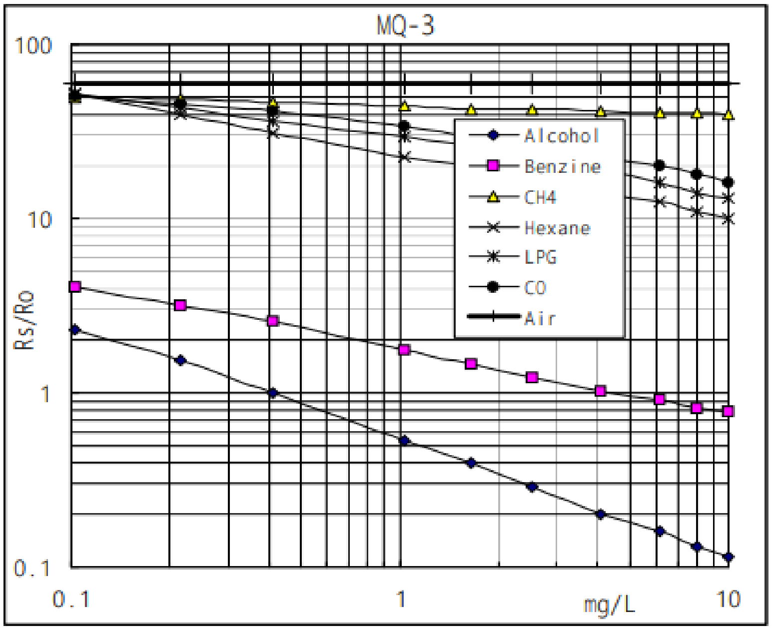

Fig. 2: Semi-log plot displays sensitivity characteristics for different gases

We were only interested in the alcohol characteristics for this sensor to correctly adjust our sensor and get the most accurate results. The plot in Fig. 2 displays the $\frac{R_S}{R_O}$ ratio where R_S corresponds to the sensor's internal resistance which constantly changes according to the amount of alcohol detected on its proximity. On the other hand, R_O corresponds to the resistance of the sensor in relation to clear air (non-alcohol detection).

Ultimately, as the individual blows over the MQ-3 sensor, the alcohol level detected on the breath by the sensor is sent to the Argon microcontroller board executing the software program to compare the detected value with the given values and conclude the BAC of the driver. Such data then triggers the pedal (servo motor) determining whether the driver can safely drive or not, which triggers the last component of our system by sending data over the cloud if the BAC is over the legal limit.

2.1.1 MQ-3 Equations Used to properly set up sensor

Our system uses the Analog output pin of the MQ3 sensor which gives us an output voltage from 0V to 5V, to measure that 0V – 5V output is displayed as it is shown in Fig. 3.

Table 1: Trials to convert Analog Values to BAC values in mg/dL

Trial #	Clean Air (No Alcohol Present) (Analog Readings)	Using Alcohol (Analog Readings)
1	1650	1695
2	1685	1710
Average	1667.5	1702.5

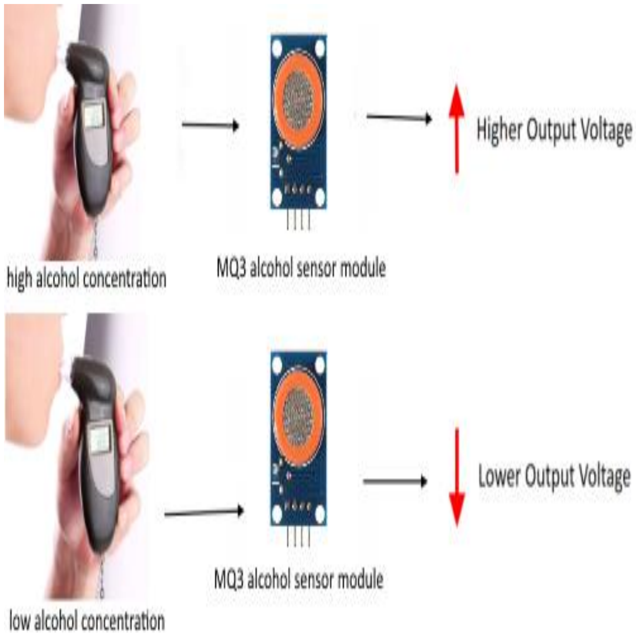

Fig. 3: Voltage Output for MQ3 Sensor

Fig. 4: Car Key Entry Mechanism

We connect the MQ-3 analog output pin to our particle Argon Analog input pin and using analogRead commands it returns an integer number from 0 to 4095 which corresponds to the input voltage value from 0V to 3.3V (max voltage in particle argon board). In our case, we had to find out the threshold value that indicates drunk driving. Unfortunately, the Voltage vs Alcohol Level is not linear, and the manufacturer doesn't provide a Voltage vs Alcohol Level, it only provides a chart. Thus, we used and equation to determine the threshold value to get an approximate BAC number. Even though the BAC value is an approximate number, the threshold value is accurate, which is the key factor for our system to work precisely. As displayed in the article How MQ3 Alcohol Sensor Works? & Interface it with Arduino [6]: it is clearly that "drunk" gives the MQ3 sensor a 2V output. Since we used the particle argon, the analogRead of 2V gives us a decimal number as follows:

$$\frac{2V}{3.3V} \times 4095 \cong 2482$$

meanwhile the legal BAC level is 80mg/dL, with this number, we can re-scale it to calculate an approximate BAC level before and after the threshold (the threshold value is accurate though). In our case to get the BAC level we run a testing trial on the MQ-3 sensor as shown in Table 1 above.

From this table we use the average values for the test with Alcohol to determine the value threshold required to determine whether the driver is sober or drunk, the following equation shows the procedure:

$$\frac{1702.5}{x} \geq 80 \qquad x \geq \frac{1702.5}{80} \qquad x = thresholdvalue \geq 21.28$$

2.2. Engine Trigger

An important part of our system focuses on controlling the operation of the vehicle depending on the BAC results of the driver's breath. Since current technology is advancing at greater steps and many car manufacturers have been taking into their consideration how to implement an anti-drunk driving technology to their vehicles, these studies have been in the works since around 2006. For instance, in an article by the Executive Vice President of Nissan, Mitsuhiko Yamashita, posted on 2007, he stated, "We are carrying out trials of alcohol interlock equipment in collaboration with the local government. This system prevents the engine from starting when a sensor that analyzes the breath indicates that the driver is drunk” [7]. In our solution the servo motor simulates the system to prevent the engine from starting. Fig. 4 shows the key entry mechanism used in a lot of cars, although nowadays Push-Button Start is more common.

The lock system in Fig. 4 (above represents the standard ignition of most cars. As with all other “lock and key” mechanisms, the lock is a mechanical component with several cuts created in unique patterns that prevents the mechanism from being engaged except when done with the key it has been uniquely cut for. In the case for a car, the lock can rotate into several positions:

1. The “OFF/LOCK” position which turns off power to the engine and electrical accessories.
2. The “ACC (Accessory position)” that provides power to electrical accessories only, not the engine.
3. The "ON" position, provides power to the engine and electrical accessories. The switch must be in this position for the engine to run and for the vehicle to be driven.
4. Finally, the “START” used only to start the engine.

For our system to prevent drivers from driving away while intoxicated is to have the motor tethered to the ignition lock of the car forcing the vehicle to stay in the lock/acc position unless the user blows an alcohol level below the limit, in which case the person will be able to turn on the car with no problem. It is worth to mention that there are lots of more research ongoing by manufacturing companies and government agencies to properly adhere the anti-drunk driving technology and as per the news it is stated that drunk driving detection technology will be required in new cars by 2026.

2.3. Software

A well-defined software program is an irreplaceable part of this project to ensure the reliability of the results. We have used Particle which provides both hardware and software tools to create and manage our IoT system. Particle works as a full-stack PaaS, which stands for Platform-as-a-Service and it is part of the Cloud Computing Services. As stated in their main website platform, their main goal is to be able to allow users to create and build projects in an easier manner as the service offered includes the combination of IoT software, connectivity via the Internet, and hardware to work together plus the data security and scalability required for any project [8]. Furthermore, Particle Web IDE was used to write the required programming part on of our solution. The platform in general is very user-friendly and makes it appropriate to write code, compile, and flash the devices using OTA (over-the-air) from any web browser. For users of Arduino IDE, it will be noticed that this IDE is very similar in looks, but the programming code is written in C++.

2.4 Cloud

For the Cloud part, we used MQTT protocols to connect our IoT device to the IBM Cloud. MQTT stands for Message Queuing Telemetry Transport, it is a lightweight messaging protocol that perfectly fits the needs of our requirements as we only send a small amount of data. MQTT requires low network bandwidth and its power saving, meanwhile it offers QoS options to guarantee the data is delivered. The IBM Cloud offers developers flexible solutions to manage a large number of IoT devices, we could use the IBM Watson IoT Platform to record and visualize the data uploaded from our IoT devices, with the Node-RED app we even have more options to handle the data, for example when the Blood Alcohol Content value the alcohol sensor detected exceed the limit, we have the options to send emails or text messages to an emergency contact number.

3. Circuit Schematic

The schematic circuitry for our system is very important and it is shown in Fig. 5. The schematic was the first task designed a drawn to understand all the required connections and hardware components needed to accomplish our planned objectives. This is a clear view of all the pins used in the Particle Argon board as well as the different pins from the MQ-3 sensor, servo motor, and LCD display. The physical connection of the different electronic components and the physical connections made to accomplish the goal of our solution is shown in Fig. 6.

3.1 Physical Hardware Circuit

Furthermore, Fig. 6 shows the hardware components connected in the Argon Board following the circuit schematic shown in Fig/ 5. The MQ-3 sensor is connected from the analog input to the A0 pin in the board to receive and send all the data for the alcohol level detected in the individual's breath. The servo motor serves as the representation of the car being operable or not depending on the BAC level detected, for instance if the BAC level exceeds the 0.8% legal limit the servo rotates 90 degrees to represent a locking system to the ignition of the car preventing it from being operable.

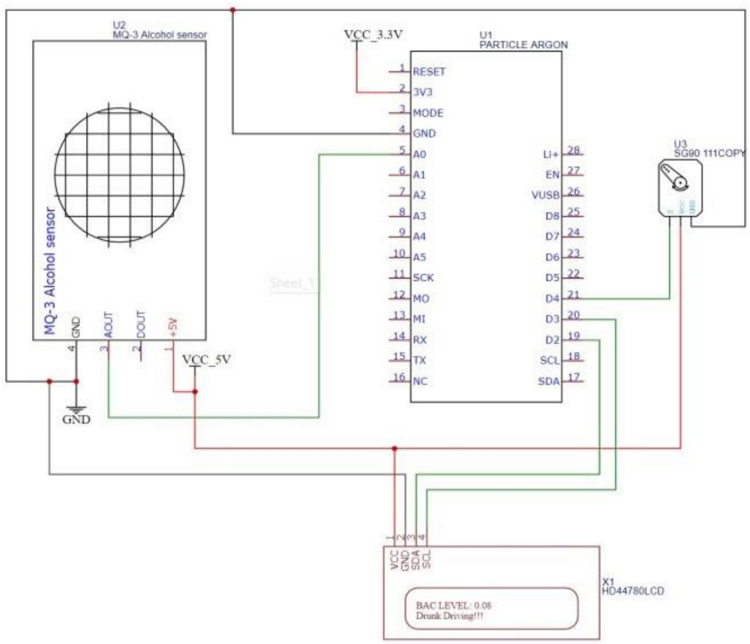

Fig. 5: Hardware Circuit Schematic

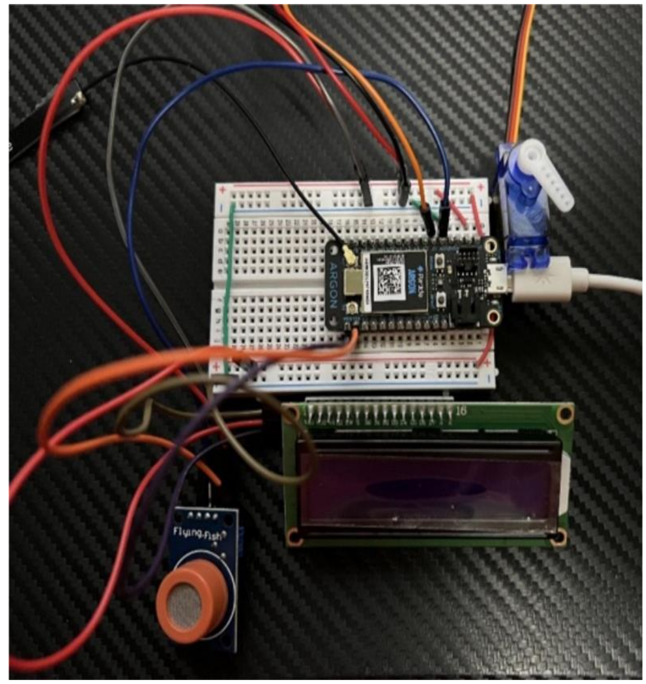

Fig. 6: Physical Circuit Design

4. Results

The results successfully demonstrates the use of the MQ-3 sensor in a vehicle to detect the level of alcohol in the user's blood. There are 4 pictures blow showing the two different states: Drunk and Sober. The code for our Arduino MQ-3 alcohol sensor is simple: when the readings from the MQ-3 sensor surpasses 80 mg/dL or a BAC of 0.08, the servo motor rotates, simulating the locking of the engine, the LCD screen also displays the BAC readings, and the state of the driver, lastly a notification is sent over the IBM cloud to relatives and law enforcement. Fig. 7 shows BAC readings of 64mg/dL and 79 mg/dL, which resulted after an individual blew into the sensor without having drunk alcohol and after drinking one shot of alcohol (79 mg/dL) within the legal limit but on the edge to being drunk and unable to drive. In Fig. 8, we used a hand sanitizer with an ethyl alcohol concentration of 70%, the first trial resulted in a reading of 84 mg/dL which goes over the limit by 5% and the driver is considered drunk, the locking system will begin and prevent the user from being able to start the engine, lastly an alert will be sent by using IBM cloud. The second trial, by applying more force to the bottle of hand sanitizer (creates a harder blew over the sensor) generated a higher reading of 93 mg/dL which implies that the driver is very drunk and with an even higher percentage of causing a car accident as this reading is 16.25% over the legal limit.

4.1 Cloud Notification Results

One big part of our project was to use IBM Cloud to send a notification when a driver is attempting to drive under the influence of alcohol. This was accomplished by using the MQTT which is to publish and subscribe messaging transport protocol used and designed for an efficient exchange of real-time data from a sensor and mobile devices. In our case we used the MQ-3 sensor to detect the alcohol level of the driver, in the case of the BAC being over the legal limit the MQTT oversees sending this real-time data to the IBM Cloud. Through IBM the solution we used the IBM Watson IoT Platform which is a cloud-hosted service designed to derive values from our Particle Argon device or any other Internet of Things device. Once this was set up, we were able to create "Cards" which is used to display our device's data as shown in Fig. 9 displaying a reading from the sensor with a BAC lower than the legal limit, thus driver is sober and is allowed to drive. On the other hand, Fig. 10 shows a reading of 96 being over the limit (80), thus the driver is drunk and unable to drive.

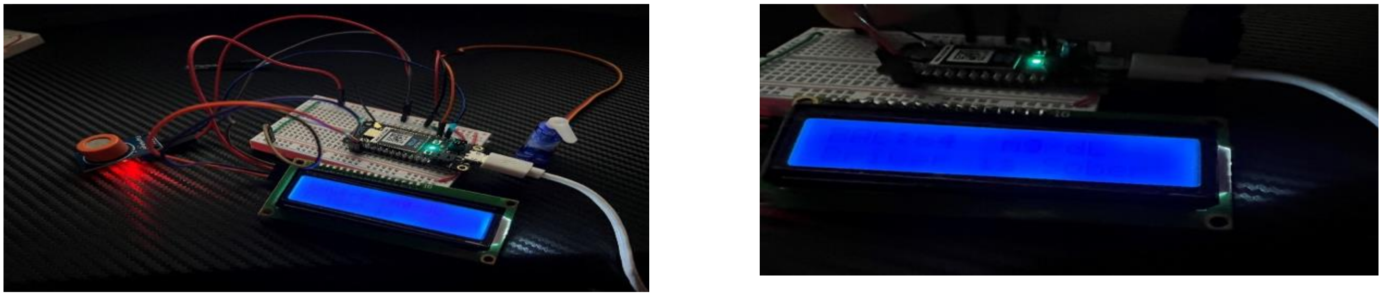

Fig. 7: Reading of 64 and 79 mg/dL within the legal limit.

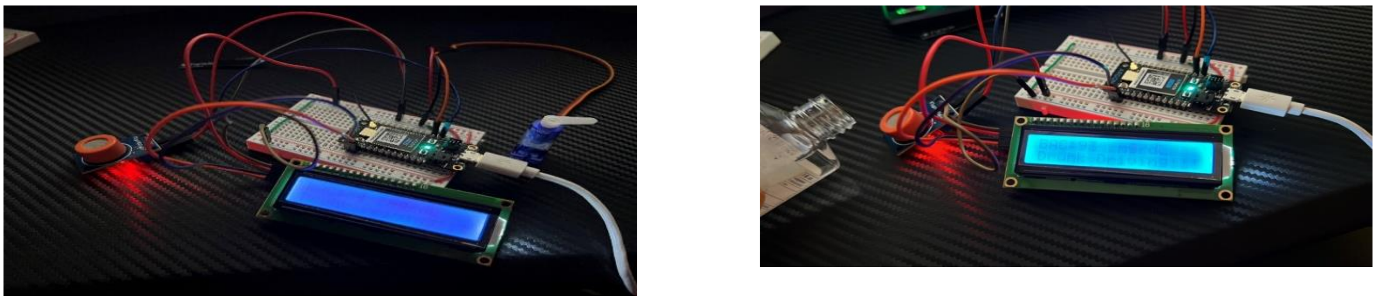

Fig.8: Reading of 84 and 93 mg/dL which is over the legal limit.

Moving forward, in order to set up a notification alert when drivers are drunk, we used Node-RED, another tool developed by IBM for visual programming and building workflows for IoT programs. We again made the use if MQTT-in to connect our Internet of Things board with the rest of the parameters. In Fig. 11 we show the finished Node-RED application the data from the sensor is input using the SensorMQ3In node where all the information in regard to the MQTT_HOST, MQTT_CLIENT, MQTT_USERNAME, and MQTT_PASSWORD is entered allowing access to the reading of the sensor. A function node is used right after to capture the reading of the MQ-3 sensor or the BAC level, then the switch node is in charge of comparing the sensor values to the limit value of 80. There are two cases created in the switch node: value over the limit (80) and value below the limit (80). When the value below the limit is triggered then a display message will show that the driver is sober and will include the BAC number recorded. Whereas, if the recorded value is over the limit, then it triggers the email notification to be sent to close relatives and alert to law enforcement with a message stating that the driver is attempting to drive drunk, the message will also display the BAC level. These results are shown in Fig. 12 which is the Node-RED serial port.

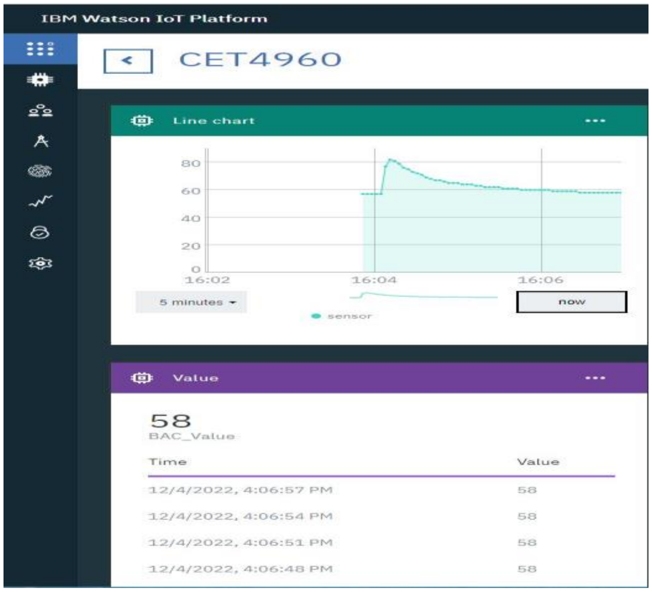

Fig. 9: MQ-3 sensor reading and BAC level below limit

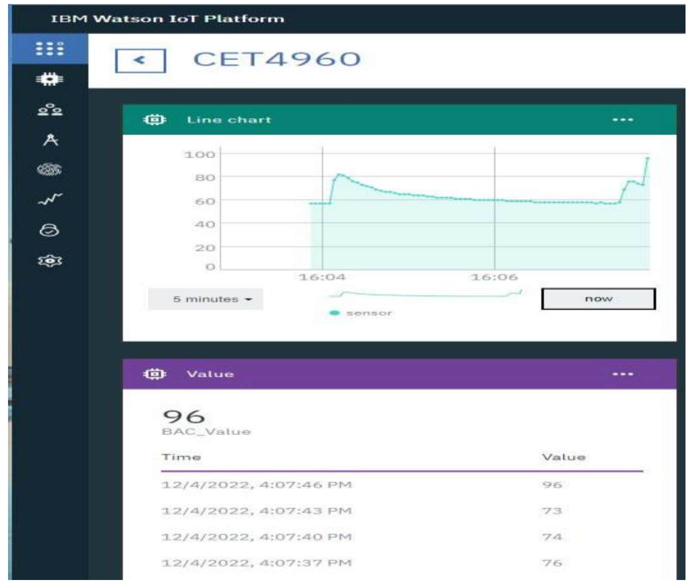

Fig. 10: MQ-3 sensor reading and BAC level over limit

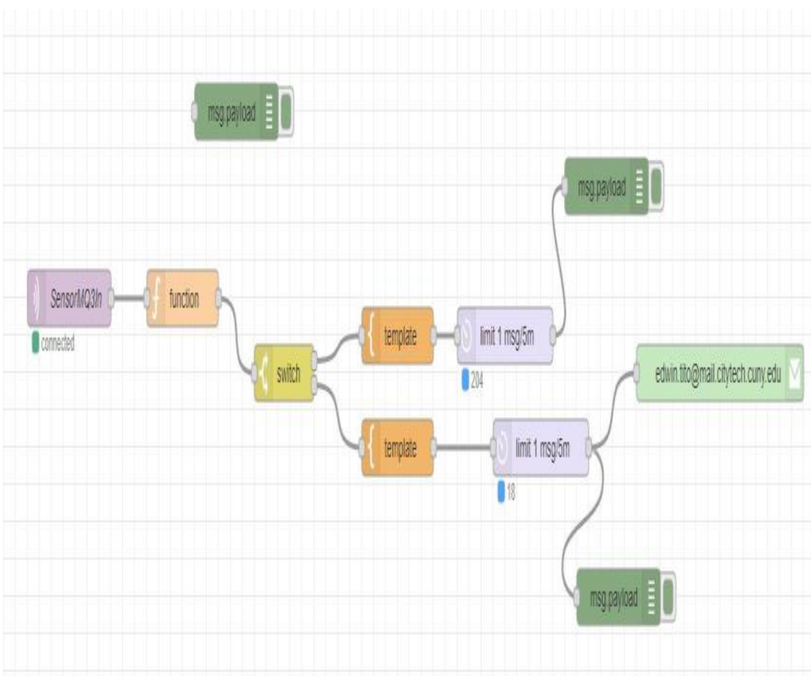

Fig. 11: Finished Node-RED App to trigger alert

Fig. 12: Node-RED Serial port showing alerts and BAC level of driver

Lastly, the only time an alert will be sent to law enforcement and relatives is when the BAC value is over the limit, we have limited the alerts to be send 1 alert every 5 minutes. In our case the email alert was successfully received as well as the BAC reading of the driver alerting that driver is drunk.

5. Conclusion

Overall, this project cannot be completely done as there are continuous studies by manufacturing companies on how to apply these mechanisms to newer car models to reduce and prevent car tragedies due to drunk drivers. Studies regarding this matter have been in place since around 2007. We believe this solution is a great advance in technology as it will benefit all of us, however there are many limitations that make this project an extraordinarily complex one. For example, adding a breathalyser in the car requires the correct correlation between the breathalyser and the car environment as it can detect any smell in the air that leads to alcohol such as a hand sanitizer, this will create a wrong data reading and therefore wrongfully prevent the user from operating the car. Another limitation is the implementation of the locking system to the car, this must be properly settled and go along with the rest of the car parts to ensure the reliability and safety of this system. Therefore, our system meets the objectives and as any other project there is a lot of room for improvement to finally start to put this in the real-world application.

References

[1] New York State Police (2015, October 15). Impaired driving [Online]. Available: https://troopers.ny.gov/impaireddriving

[2] Blincoe, L. J., Miller, T. R., Zaloshnja, E., & Lawrence, B. A. (2015, May). The economic and societal impact of motor vehicle crashes, 2010. (Revised) (Report No. DOT HS 812 013). Washington, DC: National Highway Traffic Safety Administration.

[3] NHTSA (2022, October 15). Drunk Driving [Online]. Available: https://www.nhtsa.gov/risky-driving/drunk-driving

[4] New, H. (2021, June 18). Chapter 9: Alcohol and Other Drugs. New York DMV [Online]. Available: https://dmv.ny.gov/about-dmv/chapter-9-alcohol-and-other-drugs

[5] Mo, X. (2021, September 8). Metal-oxide-semiconductor (MOS) fundamentals. Engineering LibreTexts. [Online]. Available: https://eng.libretexts.org/Bookshelves/Materials_Science/Supplemental_Modules_(Materials_Science)/Semiconductors/Metal-Oxide-Semiconductor_(MOS)_Fundamentals

[6] Last Minute Engineers. (2022, October 12). In-depth: How MQ3 alcohol sensor works? & interface it with Arduino. [Online]. Available: https://lastminuteengineers.com/mq3-alcohol-sensor-arduino-tutorial/

[7] Sparkfun Electronics (2022, December 3). Technical data MQ-3 gas sensor. [Online]: Available: https://www.sparkfun.com/datasheets/Sensors/MQ-3.pdf

[8] Yamashita, M. (2022, December 8). Annual report 2007 – Nissan. [Online]; Available: https://www.nissan-global.com/EN/DOCUMENT/PDF/AR/2006/AR06_e_all.pdf

[9] (2022, December 3). An integrated IOT platform-as-a-service. Particle. [Online]. Available: https://www.particle.io/

[10] Arora, S. (2021, November 11). IOT explained: What it is, how it works, Why it matters. [Online]. Available: https://www.simplilearn.com/what-is-iot-how-and-why-it-matters-article