Intelligent Inebriation Interception: A Smart Breathalyzer Leveraging IoT and SMTP Protocols for Enhanced Safety

Suraj Ramesh
MEng Systems and Technology,
McMaster University,
Hamilton, ON, Canada
ramess18@mcmaster.ca

Vipin Chandran Muthirikkaparambil
MEng Systems and Technology,
McMaster University,
Hamilton, ON, Canada
muthiriv@mcmaster.ca

Nidhi Relan Gupta
MEng Engineering Design,
McMaster University,
Hamilton, ON, Canada
relann@mcmaster.ca

Kaustuv Roy
MEng Systems and Technology,
McMaster University,
Hamilton, ON, Canada
royk7@mcmaster.ca

Mahadev K
MEng Systems and Technology,
McMaster University,
Hamilton, ON, Canada
km@mcmaster.ca

Abstract - This report introduces a transformative Smart Breathalyzer system designed to redefine responsible drinking practices and enhance road safety. Utilizing an MQ3 alcohol sensor, the system accurately finds intoxication levels in individuals. Upon detecting intoxication, an integrated camera captures the vehicle's number plate (tesseract), sends it to a centralized Google Sheets (via Google Cloud Console) triggering a comprehensive database search (Atlas – MongoDB) for the driver's details, including name, phone number, and license number. This information is then used to penalize the intoxicated individual for potential drink and drive offenses. The system automates the penalty process and issues notifications via email (Simple Mail Transfer Protocol - SMTP) to the offender, ensuring effective intervention. Our project encompasses the development, integration, and evaluation of this innovative system, presenting a novel approach to discourage drunk driving and promote a safer community.

Keywords – MQ3 Alcohol Sensor, tesseract, Google Cloud Console, Atlas – MongoDB, Simple Mail Transfer Protocol (SMTP)

I. INTRODUCTION

In recent years, the growing concern over road safety and the consequences of driving under the influence of alcohol has prompted the exploration of innovative technologies to mitigate the risks associated with drunk driving. This project aims to address this critical issue by introducing a Smart Breathalyzer System, employing cutting-edge technologies such as Raspberry Pi, MQ3 alcohol sensor, buttons, an LCD Display, and a Raspberry Pi camera. The integration of these components creates a comprehensive solution capable of accurately detecting alcohol levels, capturing vehicle number plates, and seamlessly transmitting data to cloud platforms for efficient monitoring and enforcement.

The primary objective of this project is to contribute to the ongoing efforts in enhancing public safety on roads. By leveraging the capabilities of the Raspberry Pi platform and various sensors, we aim to develop a reliable and intelligent system that can identify individuals driving under the influence and promptly notify authorities for appropriate action. The integration of a Pi camera further enhances the system's capabilities by capturing vehicle number plates, enabling authorities to track and penalize offenders effectively.

The project architecture involves the use of an MQ3 alcohol sensor to measure the alcohol content in the exhaled breath of individuals undergoing the test. This data is then processed by the Raspberry Pi, which utilizes a combination of buttons and an LCD Display to facilitate user interaction and display relevant information. The captured data, along with details of vehicle number plates (image processing), is transmitted to both Google Sheets and the Ubidots cloud, ensuring centralized and accessible storage of information for monitoring and analysis. Additionally, the system incorporates a notification feature to alert users of their penalty status via mobile email notifications, fostering increased accountability and awareness among potential offenders.

This project aims to contribute to the ongoing discourse on road safety and drunk driving by providing a comprehensive, technologically advanced solution. The integration of hardware components, cloud connectivity, and mobile notifications establishes a robust and efficient system for monitoring and enforcing alcohol limits on the roads, ultimately fostering a safer and more responsible driving environment.

II. ARCHITECHTURE AND BLOCK DIAGRAM

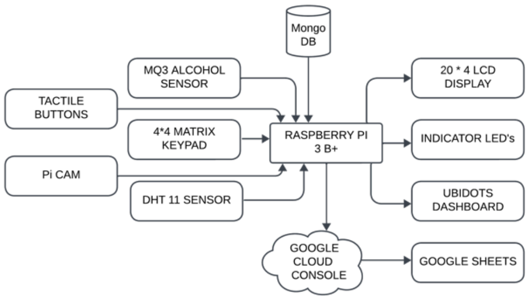

Fig. 1. Block Diagram.

The system comprises of five key components - the MQ3 alcohol sensor, Raspberry Pi microcontroller, LCD display module, buttons, and Pi camera module.

The MQ3 sensor forms the fundamental detection unit of the system. It is responsible for sensing the alcohol content from an individual's breath sample. The sensitive material within the sensor can detect presence of alcohol. The Raspberry Pi acts as the brain of the system - responsible for processing sensor data, controlling the LCD display, integrating the buttons, and communicating with the cloud platforms. The Pi receives digital data from the MQ3 sensor, processes it to determine the breath alcohol content, and conveys relevant information on the LCD display. The 20x4 LCD display module and five buttons provide a seamless user interface for drivers undergoing the smart breathalyzer test. The display presents useful information like test status, alcohol level detected, and penalty details. The buttons allow users to navigate across menu options.

The Pi camera module is a critical component for capturing images of the vehicle registration plate number. It enables identifying offenders for appropriate penal action. The camera gets activated automatically during the driver alcohol testing process, if the driver is found to be intoxicated.

Finally, the Raspberry Pi facilitates connectivity with Google Sheets and Ubidots cloud - allowing remote storage and access of captured breath alcohol and vehicle number plate data. The cloud integration enables effective data analytics for long-term monitoring and enforcement.

Overall, the strategic integration of these hardware and software components creates a robust and intelligent breathalyzer system for enhanced drunk driving identification and prevention.

III. CIRCUIT DIAGRAM

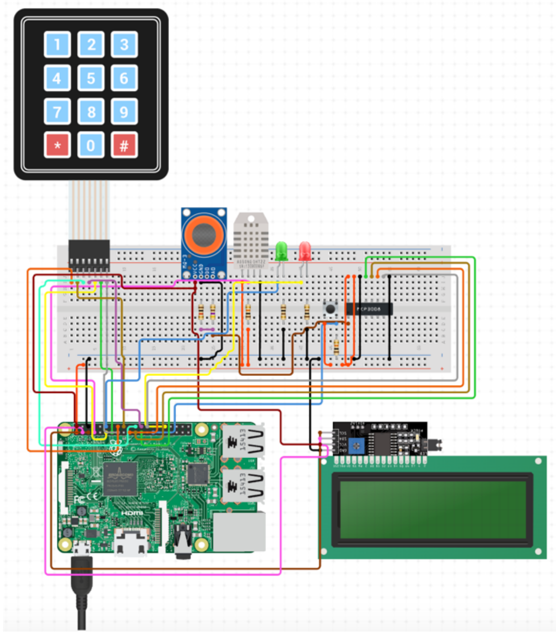

Fig. 2. Circuit Diagram

The Smart Breathalyzer System now utilizes a digital MQ3 alcohol sensor, interfaced to the Raspberry Pi board. The digital output eliminates the need for analog to digital conversion on the Pi.

A 20x4 I2C enabled LCD display provides more screen real estate for status messages and results. The I2C communication protocol simplifies wiring, with just four connections to facilitate display control from the Raspberry Pi. A single push button is now dedicated to trigger capturing images of vehicle registration plates using the Pi Camera Module during the breath analysis cycle. This simplifies user interaction. The primary user interface is now a 4x4 matrix keypad, allowing numeric input for menu navigation on the LCD. The keypad is wired to dedicated GPIO pins with row-column scanning to encode the button presses to desired commands for the LCD menu.

IV. INPUTS AND OUTPUTS

A. Inputs:

- Breath Sample Input (From User): User-provided breath samples containing volatile compounds, particularly alcohol.

- Digital Signal from MQ3 Alcohol Sensor: A digital signal indicating the alcohol content in the breath sample.

- User Interactions (Button Presses): Input from 4*4 matrix keypad facilitating user interaction during the breathalyzer test.

- Pi Camera Module Activation Signal: Signal to activate the Pi camera module when the driver is found to be intoxicated, capturing images of the vehicle registration plate number.

B. *Outputs:*

- Digital Sensor Data: Converted digital data representing the detected alcohol level from the MQ3 sensor.

- LCD Display Output: Information displayed on the 20x4 LCD-I2C module, including test status, alcohol level detected, and penalty details.

- Captured Images from Pi Camera: Images of the vehicle registration plate number captured by the Pi camera module during the testing process.

- Cloud Data Output: Transmitted data to Google Sheets and Ubidots cloud platforms, storing sensor data and vehicle number plate data for remote access and analysis.

- Mobile Email Notifications: Notification feature to alert users of their penalty status, fostering increased accountability and awareness among potential offenders.

V. Project Breakdown

A. *Alcohol Detection*

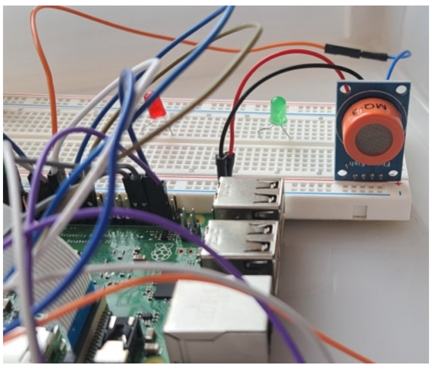

Fig. 3. MQ3 sensor interfaced with Raspberry Pi 3B+.

The MQ3 sensor is a metal oxide semiconductor (MOS) gas sensor The sensor is made of aluminium oxide coating with a Tin Dioxide (SnO2) sensitive layer.[2] The MQ3 alcohol sensor is interfaced with the Raspberry Pi, utilizing both its analog and digital capabilities for detecting alcohol levels in the breath sample. The MQ3 sensor outputs an analog signal proportional to the alcohol concentration, and this analog signal is converted to digital format by the Raspberry Pi for further processing. The digital pin of the sensor is used to provide a simplified, binary output that is more convenient for interfacing with the Raspberry Pi.

1) Communication Protocol:

The MQ3 alcohol sensor typically has four pins: A0 (analog output), D0 (digital output), and GND (ground) and 5V (power supply). For interfacing with the Raspberry Pi, the digital pin (D0) is used.

a. *Digital Output (D0):* The D0 pin of the MQ3 sensor provides a digital output signal. This signal becomes HIGH (logic level 1) when the alcohol concentration exceeds a certain threshold, and LOW (logic level 0) when it is below the threshold. This binary output simplifies interfacing with the Raspberry Pi.

b. *Analog Output (A0):* The A0 pin provides an analog voltage signal proportional to the alcohol concentration. However, in this setup, the focus is on using the digital output for simplicity.

The library used in the program is: *pip install Rpi.GPIO*

B. *License Plate Detection and Extraction*

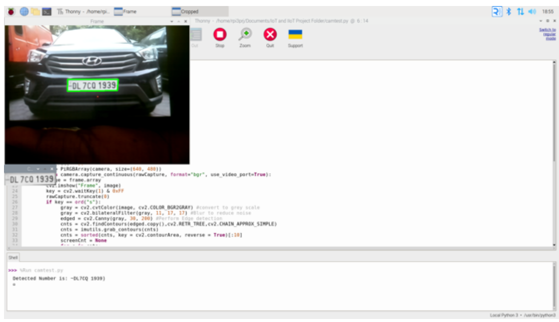

Fig. 4. License Plate Detected and information extracted on button trigger.

The provided Python code implements license plate detection using the Raspberry Pi camera and Tesseract Optical Character Recognition (OCR). The code captures frames from the Pi camera, processes the images, and extracts license plate information using OCR. Additionally, the extracted license plate information is sent to Google Sheets via the Google Cloud Console.

The code initializes the Raspberry Pi camera with a resolution of 640x480 and a framerate of 30 frames per second. The program continuously captures frames from the camera, converts them to grayscale, and displays the original and processed frames using OpenCV. License plate detection is triggered when a button connected to the GPIO pin is pressed. This is achieved using the "GPIO.input()" function.

The captured frame is converted to grayscale and then subjected to bilateral filtering and edge detection (Canny) to enhance the license plate's visibility. Contours are detected in the processed image, and the largest contour, assumed to be the license plate, is identified. In order to identify the outline of the image, namely to detect the edge of the image, the relevant differential operator will be used.[3] The script then creates a mask based on the detected contour and applies it to the original image to isolate the license plate. The isolated license plate region is then cropped and saved as a separate image.

Tesseract OCR is applied to the cropped license plate image to extract the alphanumeric characters. Tesseract is available for Linux, Windows and Mac OS X, however, due to limited resources only Windows and Ubuntu are rigorously tested by developers.[1]

a. Sending Extracted Data to Google Sheets

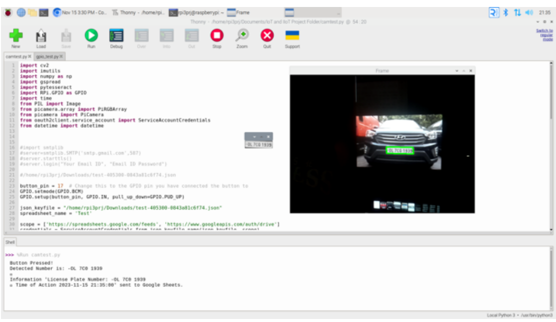

Fig. 5. License Plate detected and sent to Google Sheets

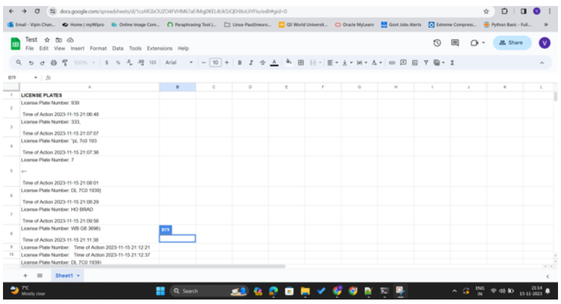

Fig. 6. Google Sheets displaying the time of action and the license plate number

The first step is to enable the Google Sheets API in the Google Cloud Console and create a project to obtain the necessary credentials (client ID and client secret). The next step is to install the required Python libraries for working with Google Sheets and authentication. Commonly used libraries include `google-api-python-client`, `google-auth`, and `google-auth-oauthlib`.

Use the obtained client ID and client secret to authenticate your application. This typically involves creating and storing credentials in a JSON file that your Python script can use for authentication. The script uses the authenticated credentials to access the Google Sheets API. This includes loading the credentials from the JSON file, creating a service object, and obtaining the Google Sheets API service.

Specify the spreadsheet ID of the target Google Sheets document. This ID is usually present in the URL of the spreadsheet and uniquely identifies it. Use the Google Sheets API to append the extracted license plate information to the specified spreadsheet. The script formats the data into rows and columns, and the API appends this data as a new row in the spreadsheet.

C. Climate Parameters Dashboarding

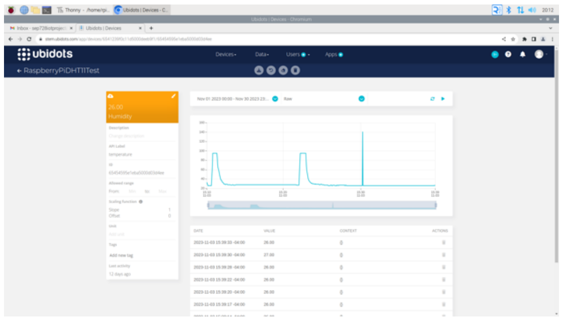

Fig. 7. Values of Humidity from DHT 11 sensor as seen in UBIDOTS Dashboard

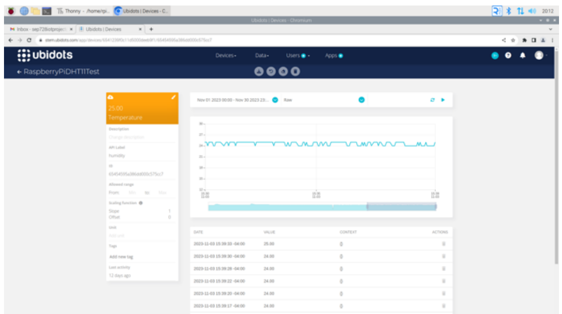

Fig. 8. Values of Temperature from DHT 11 sensor as seen in UBIDOTS Dashboard

The DHT11 sensor is interfaced with the Raspberry Pi to monitor temperature and humidity levels. Physically connecting the sensor involves wiring its data pin to a GPIO pin on the Raspberry Pi. For communication, the sensor employs a single-wire digital protocol. The Python script reads data from the sensor using the `Adafruit_DHT` library, which simplifies communication with DHT sensors on the Raspberry Pi. The library allows the Raspberry Pi to send a signal to the sensor, receive the sensor's response, and interpret the digital signal to obtain temperature and humidity readings. The acquired data is then processed for display on the LCD module and transmitted to Ubidots for remote monitoring and analysis.

The Raspberry Pi communicates with Ubidots, an IoT platform, to securely transmit the temperature and humidity data. Ubidots employs the HTTP protocol for data exchange. The Python script utilizes the `requests` library to send an HTTP POST request to the Ubidots API, including the temperature and humidity values. Ubidots processes this data and updates a predefined dashboard. In Ubidots, a dashboard is configured to visually represent the real-time data received from the Raspberry Pi. Users can access this dashboard through the Ubidots web interface, providing a comprehensive view of temperature and humidity trends over time. This integration allows for efficient remote monitoring and analysis, enabling users to make informed decisions based on the environmental conditions captured by the DHT11 sensor connected to the Raspberry Pi.

D. *Email Notification via SMTP Protocol*

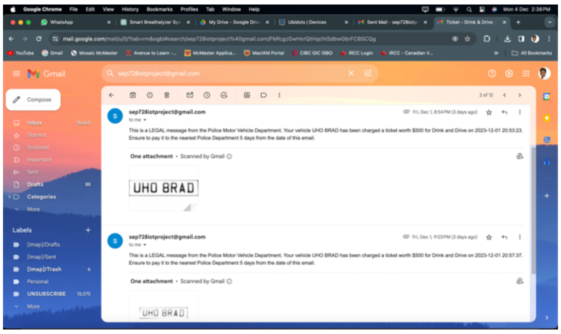

Fig. 9. Email Notification sent to the intoxicated driver

Upon detecting an intoxicated driver, the system initiates a series of actions to notify the driver and relevant authorities. Once the license plate information is successfully extracted using image processing techniques, the system queries a MongoDB database to retrieve personal details of the driver. This information includes the driver's name, mobile number, and email address. Subsequently, these details are displayed on the LCD display, providing immediate feedback to law enforcement personnel or any on-site authorities. Simultaneously, a comprehensive email notification is prepared, addressing the driver directly. The email content notifies the driver of the penalization, indicating the vehicle's license plate number, the specific charge, and the prescribed penalty amount. This personalized communication aims to raise awareness and ensure that the driver is promptly informed of the consequences of driving under the influence. The email notification is sent using the Simple Mail Transfer Protocol (SMTP), with the Python `smtplib` library facilitating the communication between the Raspberry Pi and the email server. The system configures the email subject, body, and recipient based on the extracted details from the MongoDB database. Additionally, an image attachment of the cropped license plate region serves as a visual record of the incident. This attachment enhances the communication by providing concrete evidence of the detected violation. The use of email notifications not only informs the driver but also serves as a formal communication channel, ensuring accountability and compliance with legal procedures. It reinforces the severity of the offense and emphasizes the need for responsible driving behaviour. Incorporating multifaceted notifications, including LCD display updates and personalized email communication, contributes to an efficient and comprehensive smart breathalyzer system. Displaying driver details on the LCD screen in real-time enhances on-site decision-making for law enforcement, while the email notification adds a layer of accountability and awareness for the driver. By leveraging the capabilities of both local display and remote communication, the system maximizes its impact in addressing the critical issue of drunk driving. This holistic approach aims not only to enforce penalties but also to educate and promote responsible driving behaviour for improved road safety.

E. *LCD Display Information at each stage*

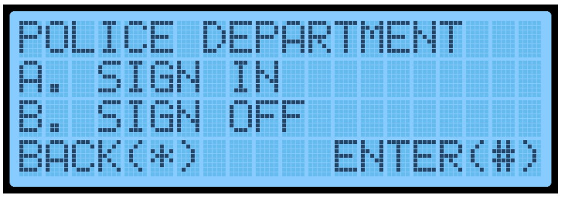

Fig. 10. Home Screen on the LCD Display

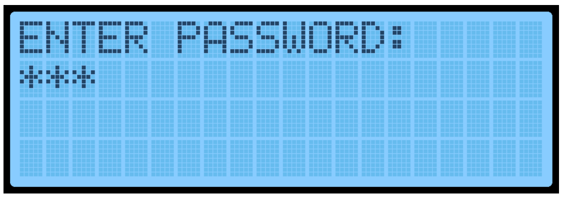

Fig. 11. Enter Password Screen as soon as ‘A’ is triggered from the keypad

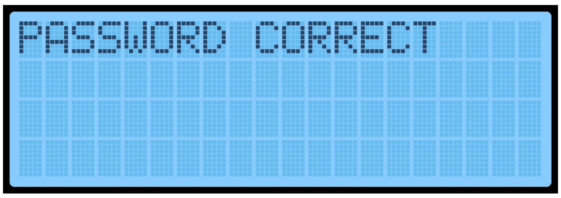

Fig. 12. Screen shows after authentication is passed

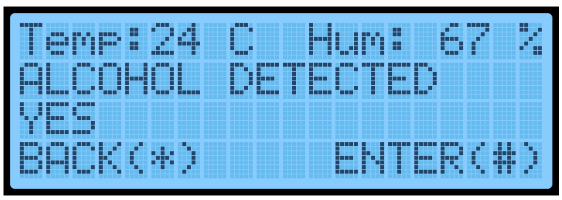

Fig. 13. Alcohol Detection Screen in the LCD Display

Fig. 14. Screen showing information related to the driver retrieved from MongoDB database

VI. ACKNOWLEDGMENT

We would like to express our sincere gratitude to all individuals and institutions whose contributions and support have been instrumental in the successful completion of this research. Special thanks to Prof. Pedro Tondo, McMaster University and Jithin Rajan Varghese (TA) for their valuable insights, guidance, and resources. Their unwavering assistance has significantly enriched the quality of this work, and we are truly thankful for their collaborative efforts.

VII. REFERENCES

[1] R. R. Palekar, S. U. Parab, D. P. Parikh and V. N. Kamble, "Real time license plate detection using openCV and tesseract," 2017 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2017, pp. 2111-2115, doi: 10.1109/ICCSP.2017.8286778.

[2] A. Prasad, J. Pavankalyan, A. S. Ganesh and K. M. Krishna, "Drowsiness And Alcohol Detection with Engine Locking," 2022 3rd International Conference on Communication, Computing and Industry 4.0 (C2I4), Bangalore, India, 2022, pp. 1-5, doi: 10.1109/C2I456876.2022.10051464.

[3] Z. Shuaishuai and P. Chen, "Research on License Plate Recognition Algorithm Based on OpenCV," 2019 Chinese Automation Congress (CAC), Hangzhou, China, 2019, pp. 68-72, doi: 10.1109/CAC48633.2019.8996599.