Embedded Systems and Internet of Things

Mini Project Report

On

An Arduino-based Embedded System in Passenger Car for Road Safety

By

Rujuta Kulkarni (5)
Swamil Randive (6)
Kushagra Singh (23)
Jatin Patil (29)

Under the guidance of

Prof. Shiv H. Sutar
Assistant Professor

Department of Computer Engineering and Technology
School of Computer Science and Engineering
Dr. Vishwanath Karad MIT World Peace University

AY 2023-24 Semester IV

INDEX

Contents

PROBLEM STATEMENT

Traffic accidents are a leading cause of death and injury worldwide. According to the World Health Organization (WHO), over 1.3 million people die annually from road traffic crashes. Among the most vulnerable are children, which reports over a million child fatalities each year due to road accidents. The project aims to address this issue by implementing an intelligent system in passenger cars to enhance safety measures and mitigate risks associated with alcohol consumption while driving.

INTRODUCTION

In recent years, there has been an increasing focus on leveraging technology to improve road safety and reduce the incidence of accidents. This project proposes the development of an Arduino-based embedded system that integrates various sensors and actuators to monitor driver behaviour, detect alcohol presence, and reduce noise pollution in sensitive areas such as hospitals and schools. By employing real-time data processing and decision-making capabilities, the system aims to provide timely alerts and feedback to ensure safer journeys for passengers.

NEED OF THE PROJECT

The motivation behind this project lies in the urgent need to address the alarming rate of road accidents and fatalities, particularly among children. Despite various initiatives aimed at promoting road safety, the problem persists, highlighting the necessity for innovative solutions. By harnessing the power of embedded systems and Arduino microcontrollers, the project seeks to contribute to the ongoing efforts to improve road safety and save lives.

Existing car features may not adequately address these issues. Stock horns lack location awareness, and standard ignition systems cannot detect driver intoxication. This project aims to create a system that:

- Reduces horn usage in designated quiet zones using GPS data.
- Detects alcohol presence in the vehicle and issues warnings or takes preventive actions.

OBJECTIVES

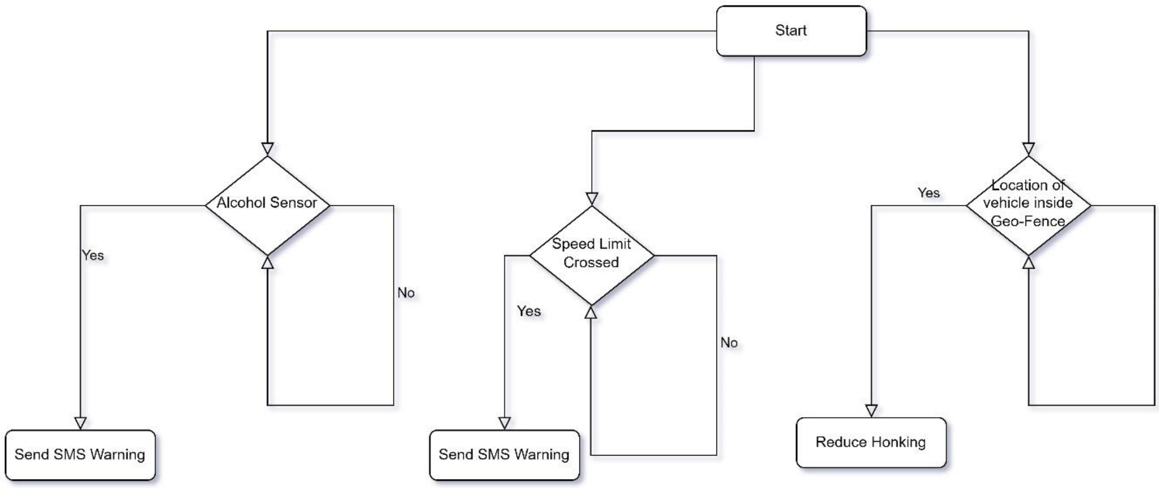
The primary objectives of this project are:

- To design and implement an Arduino-based embedded system for passenger cars.
- To develop a functionality that automatically reduces horn usage in pre-defined quiet zones using GPS data.
- To integrate an alcohol sensor for driver intoxication detection.
- To provide audio-visual alerts and potentially implement safety measures in case of alcohol detection.

PROJECT DESCRIPTION

The project involves the integration of various components, including an MQ3 alcohol sensor, buzzer, accelerometer/gyroscope, GPS module, LED screen, and GSM module, into the vehicle's onboard system. These components work in tandem to monitor driver behaviour, detect alcohol presence, and provide timely alerts and feedback to ensure safe driving practices.

Flowchart:



LIST OF COMPONENTS USED

- **Arduino Uno:** A microcontroller board that serves as the system's central processing unit, receiving sensor data, processing it, and controlling outputs.
- **MQ-3 Alcohol Sensor:** Detects the presence of alcohol vapours in the vehicle's cabin air. The sensor outputs an analog voltage signal proportional to the alcohol concentration.
- **Buzzer:** Provides audible alerts for horn reduction and potential alcohol warnings.
- **Accelerometer/Gyroscope (optional):** Can be used to detect driver fatigue or drowsiness based on unusual driving patterns (sudden changes in acceleration or erratic movements).
- **GPS Module:** Provides location data to identify pre-programmed quiet zones based on GPS coordinates.
- **GSM Module(SIM800A) :** Enables sending emergency alerts (e.g., SMS) to designated contacts in case of severe alcohol detection.

CODES

Code for GPS Module and Geofencing:

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include <math.h>

#define RX_PIN 4 // Arduino Pin connected to the TX of the GPS module
#define TX_PIN 3 // Arduino Pin connected to the RX of the GPS module
#define LED_PIN A0

TinyGPSPlus gps;                                    // the TinyGPS++ object
SoftwareSerial gpsSerial(RX_PIN, TX_PIN); // the serial interface to the GPS module

#define PI 3.14159265358979323846
#define GEOFENCE_LAT 18.56450 // replace with your latitude
#define GEOFENCE_LON 73.77314 // replace with your longitude
#define GEOFENCE_RADIUS 50    // in meters

double deg2rad(double degrees)
{
return degrees * (PI / 180.0);
}

double distance(double lat1, double lon1, double lat2, double lon2)
{
double R = 6371000; // Earth's radius in meters
double phi1 = deg2rad(lat1);
double phi2 = deg2rad(lat2);
double deltaPhi = deg2rad(lat2 - lat1);
double deltaLambda = deg2rad(lon2 - lon1);

double a = sin(deltaPhi / 2) * sin(deltaPhi / 2) +
  cos(phi1) * cos(phi2) *
  sin(deltaLambda / 2) * sin(deltaLambda / 2);
double c = 2 * atan2(sqrt(a), sqrt(1 - a));

double d = R * c;
return d;
}

void setup()
{
Serial.begin(9600);
gpsSerial.begin(9600); // Default baud of NEO-6M GPS module is 9600

Serial.println(F("Speed Detection System"));
}
```

```
void loop()
{
  //digitalWrite(9, HIGH);
  //digitalWrite(9, LOW);
if (gpsSerial.available() > 0)
{
if (gps.encode(gpsSerial.read()))
{
//geofencing code
double lat = gps.location.lat();
double lon = gps.location.lng();

if (distance(lat, lon, GEOFENCE_LAT, GEOFENCE_LON) > GEOFENCE_RADIUS)
{
Serial.print("OUTSIDE");
digitalWrite(LED_PIN, HIGH);
        //delay(200);
        digitalWrite(LED_PIN,LOW);
}
else
{
Serial.print("INSIDE");
digitalWrite(LED_PIN, LOW);

}

/*
Serial.print("Latitude: ");
Serial.print(lat, 6);
Serial.print(" Longitude: ");
Serial.print(lon, 6);
      */

Serial.print(" Distance: ");

Serial.println(distance(lat, lon, GEOFENCE_LAT, GEOFENCE_LON));


if (gps.location.isValid())
{
Serial.print(F("- latitude: "));
Serial.println(gps.location.lat());

Serial.print(F("- longitude: "));
Serial.println(gps.location.lng());

Serial.print(F("- altitude: "));
if (gps.altitude.isValid())
Serial.println(gps.altitude.meters());
```

```
else
Serial.println(F("INVALID"));
}
else
{
Serial.println(F("- location: INVALID"));
}

Serial.print(F("- speed: "));
if (gps.speed.isValid())
{
Serial.print(gps.speed.kmph());
Serial.println(F(" km/h"));
}
else
{
Serial.println(F("INVALID"));
}

Serial.println();
}
}

if (millis() > 5000 && gps.charsProcessed() < 10)
Serial.println(F("No GPS data received: check wiring"));
}
```

Code for GSM Module for sending SMS:

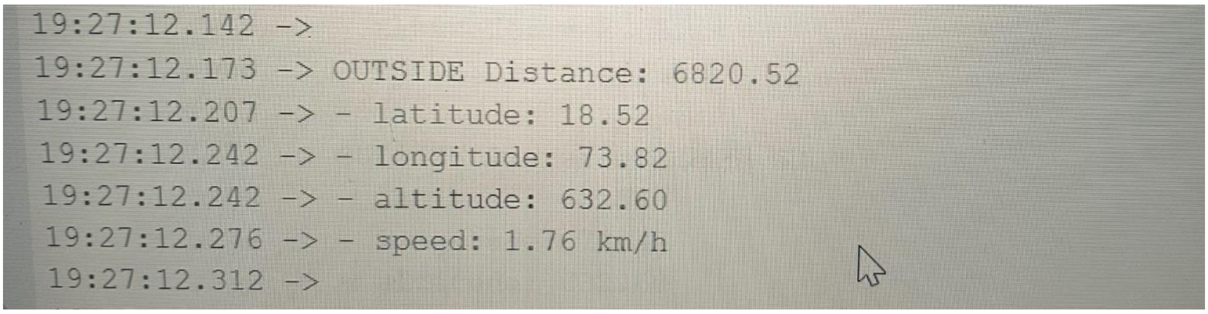
```
#include <SoftwareSerial.h>\
SoftwareSerial mySerial (1,0); //Rx Tx
void setup() {
 Serial.begin(9600);
 delay(100);
}
void loop() {
 Serial.println("AT+CMGF=1");
 delay(1000);
 Serial.println("AT+CMGS=\"+919699688519\"\r");
 delay(1000);
 Serial.println("Hello");
 delay(100);
 Serial.println((char)26);
 delay(1000);
}
```

Code for Alcohol Sensor:

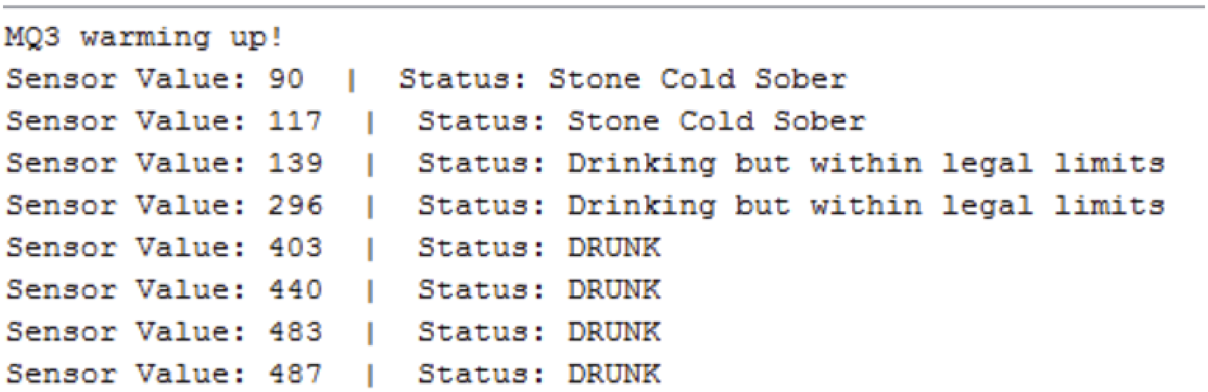
```
#define Sober 120   // Define max value that we consider sober
#define Drunk 400   // Define min value that we consider drunk

#define MQ3pin 0

float sensorValue;  //variable to store sensor value

void setup() {
	Serial.begin(9600); // sets the serial port to 9600
	Serial.println("MQ3 warming up!");
	delay(20000); // allow the MQ3 to warm up
}

void loop() {
	sensorValue = analogRead(MQ3pin); // read analog input pin 0

	Serial.print("Sensor Value: ");
	Serial.print(sensorValue);

	// Determine the status
	if (sensorValue < Sober) {
		Serial.println("  |  Status: Stone Cold Sober");
	} else if (sensorValue >= Sober && sensorValue < Drunk) {
		Serial.println("  |  Status: Drinking but within legal limits");
	} else {
		Serial.println("  |  Status: DRUNK");
	}

	delay(2000); // wait 2s for next reading
}
```

SCREENSHOTS

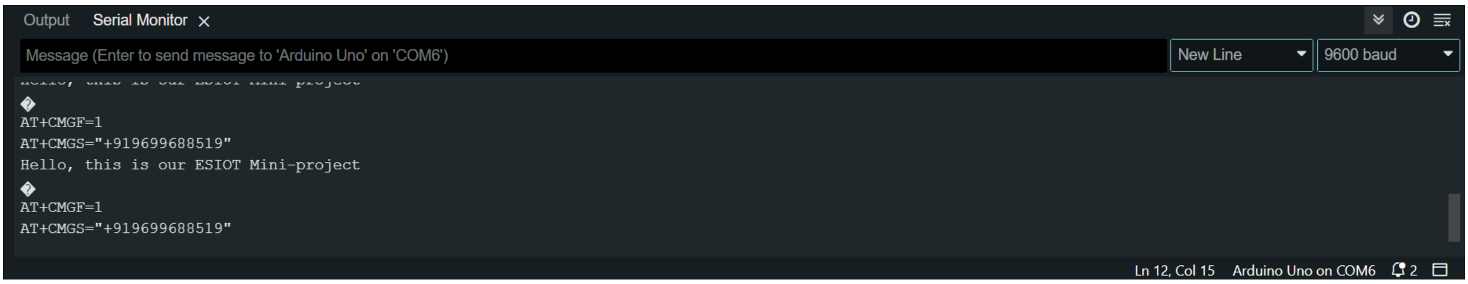
Geo-Fencing Output -



Alcohol Sensor Output -



GSM Module SMS Output -



Applications

- Road safety enhancement in passenger cars by promoting responsible driving behaviour and mitigating risks associated with alcohol consumption.
- Prevention of drunk driving incidents through the implementation of alcohol detection technology and real-time monitoring.
- Noise pollution reduction in sensitive areas such as hospitals and schools by implementing automatic horn reduction functionality.

Conclusion

This mini project report details the design and development of an Arduino-based embedded system for enhanced passenger car safety. The system addresses two key road safety concerns: excessive honking in noise-sensitive areas and drunk driving. The project successfully integrates functionalities like:

- Automatic horn reduction in pre-defined quiet zones using GPS data.
- Alcohol detection within the vehicle cabin using an MQ-3 sensor.
- Audio-visual alerts displayed on the LED screen for both horn reduction and alcohol detection.

The system can be further enhanced by:

- Implementing additional safety measures in case of alcohol detection (e.g. engine immobilization).
- Integrating an accelerometer/gyroscope for drowsiness detection.
- Utilizing a GSM module for sending emergency alerts to designated contacts.

By incorporating these advancements, the system can become a more comprehensive solution for promoting responsible driving behaviour and creating a safer driving environment.

References (IEEE format)

K. Seelam and C. J. Lakshmi, "An Arduino based embedded system in passenger car for road safety," 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 2017, pp. 268-271, doi: 10.1109/ICICCT.2017.7975201. keywords: {Pins;Relays;Automobiles;Radio frequency;Receivers;Decoding;Transmitters;Road safety;an Arduino Embedded System;speed control;horn control;Alcohol detection},

https://lastminuteengineers.com/mq3-alcohol-sensor-arduino-tutorial/

https://lastminuteengineers.com/neo6m-gps-arduino-tutorial/

CONTRIBUTION OF EACH TEAM MEMBER

Sr. No.	Name	Roll No.	Worked on
1.	Rujuta Kulkarni	5	GPS and Alcohol Module
2.	Swamil Randive	6	GSM Module Implementation and Documentation
3.	Kushagra Singh	23	GSM Module Implementation and Documentation
4.	Jatin Patil	29	GPS and Alcohol Module