JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

CREATING A MICROCONTROLLER-BASED SYSTEM FOR DETECTING ALCOHOL IN VEHICLES.

SANTHOSH KUMAR S[1], DEEPAK S[2], PRATHEEP KANNA S[3], RAVI RAJ V[4].

[1]*Department of Mechanical Engineering, Sri Sairam Engineering College, Chennai.*

[2]*Department of Mechanical Engineering, Sri Sairam Engineering College, Chennai.*

[3]*Department of Mechanical Engineering, Sri Sairam Engineering College, Chennai.*

[4]*Department of Mechanical Engineering, Sri Sairam Engineering College, Chennai.*

Abstract : The likelihood of road accidents is increasing sharply as the number of cars on the road grows rapidly. Drunk driving is often regarded as a major cause of road accidents across the world. The primary goal of this project is to create a system that detects the quantity of alcohol drunk by the vehicle's driver. The suggested system proposes to minimize the frequency of accidents caused by drunk driving by stopping the user from driving while intoxicated. The suggested concept is built around an Arduino-Uno and an alcohol detection sensor (MQ-3). When the amount of alcohol exceeds an allowed limit, the vehicle ignition system (DC Motor) is cut off and the appropriate authorities are notified via the buzzer.

IndexTerms - **Alcohol Detection, Micro-controller, Breathalyzer, Drunk Driving Prevention, Safety System, Workplace Safety, Real-time Monitoring, Sensor Technology, Immobilization, Calibration, Maintenance, Response Time, Environmental Conditions**

I. INTRODUCTION

In today's society, accident rates are rising due to increased vehicle usage resulting from employment opportunities. This surge in usage, particularly of cars and bikes, has led to more accidents, often due to speeding and driving under the influence. Lacking advanced safety measures exacerbates the problem. To address this, a microcontroller-based system has been developed to detect alcohol levels and potential accidents. This system, utilizing an Arduino Uno and an MQ-3 alcohol sensor, aims to discourage drunk driving by automatically disabling the vehicle's ignition system when alcohol levels exceed the limit and alerting relevant authorities. The proposed design is based on an Arduino Uno and an MQ-3 alcohol sensor. When the alcohol level exceeds the permissible limit, the starting system of the vehicle's DC motor is deactivated, and an alert is sent to the relevant authorities through a buzzer.

II. EXISTING SYSTEM:

The current method prioritizes passenger safety above quick assistance following a drunk and driving collision. India has the dubious distinction of having the highest number of fatalities from road driving accidents in the world. Road safety is becoming a big societal concern all over the world, particularly in India. The method built by us aimed at automatically identifying an alcohol and informing the buzzer

III. PROPOSED SYSTEM:

However, the utilization of a breathalyzer is manual and unlikely to detect most instances of drunk driving because by the time the events have occurred, alcohol detection typically happens retrospectively. In this research, the Arduino Uno is employed, which is an open-source microcontroller known for its user-friendly interface. Using the MQ-3 sensor, the system detects alcohol consumption by the driver. An ingenious aspect of this sensor is its adjustable range, which can be set to 5-10 cm to specifically detect alcohol intake from the driver alone. This sensor is then mounted on the vehicle's steering wheel. If the alcohol level exceeds the predetermined threshold, the vehicle's engine (DC Motor) is automatically shut off, and a prompt notification is sent to the relevant authorities via a buzzer. The proposed technology aims to enhance human safety.

IV. HARDWARE AND SOFTWARE USED:

1. MICROCONTROLLER
2. ALCOHOL SENSOR
3. LED LIGHT
4. MOTOR DRIVER
5. DC MOTOR
6. BUZZER
7. POWER SUPPLY UNIT
8. CONNECTING WIRES
9. SOLDERING KIT

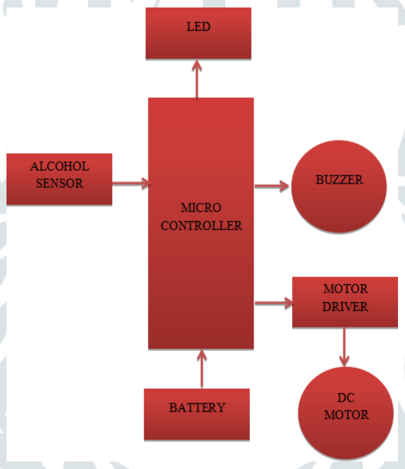
1. SOFTWARE REQUIRED

4. ARDUINO UNO IDE
5. EMBEDDED C

2. MODULES:

- SENSOR INTERFACING
- PREPARING POWER SUPPLY UNIT
- MICRO-CONTROLLER PROGRAMMING
- READING ANALOG DATA
- TEST AND DEBUG
- SUBMISSION

V. BLOCK DIAGRAM:



VI. ARDUINO IDE:

■ Arduino Software, encompasses an array of menus, a message area. It establishes connections with Arduino and Genuino hardware to upload programs and facilitate communication.

■ "Sketches," which are saved in files with the.ino extension, are codes created in the Arduino Software (IDE). Cut/paste and search/replace are functions of the text editor. When appropriate, the message section shows problems and offers feedback during exporting and saving. The Arduino Software (IDE) text output, including detailed error warnings, is shown on the console.In the bottom right corner, the setup board and serial port details are showcased, while toolbar buttons offer options to verify programs, upload files, create sketches, open, access, save, and utilize the serial monitor.

■ Before initiating a sketch upload, it is necessary to select the correct options from the Tools > Board and Tools > Port menus. Serial ports may vary in nomenclature based on the operating system; for instance, on Windows, they could be labeled as COM1 or COM2, on Mac as /dev/tty.usbmodem241, and on Linux as /dev/ttyACMx. The upload process commences once the appropriate serial port and board have been selected, followed by clicking the upload button or opting for Upload from the Sketch menu. For older boards, manual reset may be required, whereas Arduino boards equipped with auto-reset will perform this action automatically before uploading. Throughout the upload, the RX and TX LEDs typically flash intermittently, and the IDE will notify you of the upload's success or failure.

■ The Arduino bootloader is a little software on the microcontroller that makes it easier to transmit code without the need for extra hardware when uploading a sketch.During board reset, the bootloader briefly activates, launching the most recent uploaded sketch. The on-board LED (pin 13) blinks to indicate the start of the bootloader activation process.[4]

VII. INTEGRATED C:

- Embedded C is comprised of language extensions tailored for the C programming language, addressing compatibility concerns across various embedded systems. Historically, embedded C programming necessitated modifications to the standard C language to accommodate fixed-point arithmetic, handling multiple memory banks, and facilitating basic I/O operations.

- Real-time computing limitations are frequently faced by embedded systems, which are specialized systems inside larger mechanical or electrical systems. These systems include everything from tiny gadgets like digital watches to complex installations like traffic signals and hybrid cars. Programming these systems is made easier by Embedded C, which can adjust to their unique needs, such as low power consumption, compact size, ruggedness, and economical viability. Microcontrollers or microprocessors are the most common components used in modern embedded systems, while digital signal processors (DSPs) are frequently used in some applications. Design engineers use economies of scale to enable mass production while optimizing embedded systems for size, cost, reliability, and performance.[7]

- This code appears to be a mixture of C++ and Arduino code designed to read data from a gas sensor and control some outputs based on the sensor readings. However, it seems incomplete and lacks clarity in its purpose.

VIII.SOURCE CODE:

```
#include <OneWire.h>
#include <DallasTemperature.h>

const int SENSOR_PIN = A0;
int gas = 2;

const int INT5 = 5;
const int INT6 = 6;
const int INT9 = 9;
const int INT10 = 10;
int Speed;
OneWire oneWire(SENSOR_PIN);
DallasTemperature tempSensor(&oneWire);
float tempCelsius;
float tempFahrenheit;
int i;

void setup() {
  pinMode(A0, INPUT);
  pinMode(13, OUTPUT);
  pinMode(INT5, OUTPUT);
  pinMode(INT6, OUTPUT);
  pinMode(INT9, OUTPUT);
  pinMode(INT10, OUTPUT);

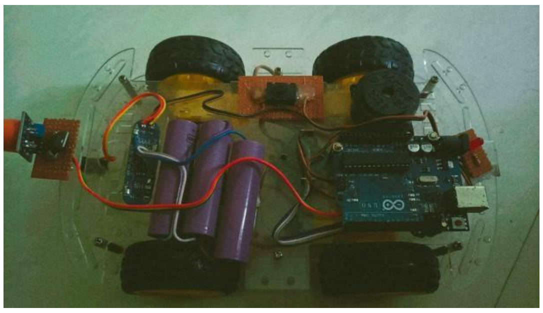
  analogWrite(INT5, 0);
  analogWrite(INT6, 153);
  analogWrite(INT9, 0);
  analogWrite(INT10, 153);

  delay(3000);
  Serial.begin(9600);
  analogWrite(INT5, 0);
  analogWrite(INT6, 0);
  analogWrite(INT9, 0);
  analogWrite(INT10, 0);
}
```

```
void loop() {
  gas = analogRead(A0);
  Serial.print(" gas: ");
  Serial.println(gas);

  if (gas < 500) {
    i = 0;
    digitalWrite(13, LOW);
    analogWrite(INT5, 0);
    analogWrite(INT6, 153);
    analogWrite(INT9, 0);
    analogWrite(INT10, 153);
  } else {
    digitalWrite(13, HIGH);
    while (i < 500) {
      Serial.print(" I: ");
      Serial.println(i);
      analogWrite(INT5, 0);
      analogWrite(INT6, 102);
      analogWrite(INT9, 0);
      analogWrite(INT10, 102);
      i++;
    }
    while (i < 1000) {
      analogWrite(INT5, 0);
      analogWrite(INT6, 51);
      analogWrite(INT9, 0);
      analogWrite(INT10, 51);
      Serial.print(" I: ");
      Serial.println(i);
      i++;
    }
    analogWrite(INT5, 0);
    analogWrite(INT6, 0);
    analogWrite(INT9, 0);
    analogWrite(INT10, 0);
  }
}
```
```

## IX. FINAL OUTPUT :


```

X. CONCLUSIONS

This study focuses on the detection of drunk driving by drivers utilizing a microcontroller-based system. The hardware system and the software system are the two primary components of the proposed system. The hardware is located within the truck or cab, and the software will be used by the owner, management, or driver. When an alcohol odor was identified, the car stopped automatically, and then a buzzer sounded to warn the driver. This technology is used in low visibility to prevent road accidents and save lives.

XI. REFERENCES

1. Nadu–India, Tamil. "Real Time Alcohol Detection and Accident Prevention System for Four Wheelers."Ajagbe, Sunday Adeola, et al. "An Alcohol Driver Detection System Examination Using Virtual Instruments." *Journal of Hunan University Natural Sciences* 50.11 (2023).
2. Jog, Pranjal, et al. "Automatic Alcohol Detection System."
3. Joshi, Thakare Bhagyashri S. VV. "Alcohol Detection using Pic Microcontroller."
4. Sarkar, Tamoghna, and Sukriti Shaw. "IoT Based Intelligent Alcohol Detection System for Vehicles." *Proceedings of the 4th International Conference on Big Data and Internet of Things*. 2019.
5. Lekha, P. Sree, and P. Venkata Prasad. "Alcohol Detection withAutomatic Engine Locking System."
6. Pathak, Vikrant, et al. "ALCOHOL DETECTION WITH PIC MICROCONTROLLER: A REVIEW."
7. Varma, Addepalli Jaya, A. Vijaya Durga, and Pagalla Bhavani Shankar. "Alcohol detection system in vehicle for human safety." *International Journal for Modern Trends in Science and Technology* 6.10 (2020): 80-83.
8. Anthony, Melanie, et al. "Alcohol detection system to reduce drunk driving." *International Journal Of Engineering Research & Technology* 9.3 (2021): 360-365.
9. Gowrishankar, J., et al. "Arduino-based alcohol sensing alert with engine locking system." *International Conference on Mobile Computing and Sustainable Informatics: ICMCSI 2020*. Springer International Publishing, 2021.
10. Joshua, Ighalo, et al. "Development of alcohol triggered vehicle engine lock system." *IAES International Journal of Robotics and Automation (IJRA)* 8.1 (2019): 68.