

# Alcohol Detection System

**Description:** Develop a microcontroller based prototype which can detect the drink and drive issue of driver by detecting the consumption alcohol by the drivers.

**Problem Statement:** Develop a microcontroller-based prototype that can detect alcohol consumption by drivers to address the issue of drunk driving. The solution should provide a reliable and accurate method for determining whether a driver has consumed alcohol, with the potential for an interlock mechanism to prevent the vehicle from starting if the alcohol level is above a certain threshold.

## Scope of Solution:

The scope of the solution for a microcontroller-based prototype drunk driving detection system is developing and putting into place a complete system to identify when a driver has consumed alcohol and to take the necessary precautions to avoid driving while intoxicated. The system's goal is to reduce the dangers related to driving while intoxicated in order to improve road safety in an accurate and efficient manner.

1. Alcohol Detection: 1. Install an alcohol sensor to measure the amount of alcohol in the driver's breath or surrounding air. An example of such a sensor is the MQ-series gas sensor.

2. Data Processing: Employ a microcontroller specifically designed for processing sensor data, such as an Arduino, Raspberry Pi, or other similar device.

- Create algorithms for data filtering, averaging, and calibration for interpreting sensor data.

3. Threshold Detection: To differentiate between sober and intoxicated states, set predetermined alcohol concentration thresholds.

- Use logic to set off distinct reactions according to the amount of alcohol detected.

4. User Interface: • Give drivers an intuitive interface that includes:

- Visual display: Provide system status and alcohol level measurements.

- Audio alerts: When blood alcohol content surpasses the allowable threshold, audio alerts or cautions are played.

5. Interlock Mechanism:

- Install an interlock device that has the ability to stop the car from starting in the event that the alcohol content exceeds a certain threshold.

- To avoid unforeseen repercussions, make sure the interlock mechanism conforms with safety regulations.

#### 6. Power Supply:

Ensure that the system is built to run on an appropriate power source, such as the vehicle's power supply or a battery.

#### 7. Communication

- If required, incorporate communication modules such as Wi-Fi, Bluetooth, or cellular connectivity to provide data to the appropriate authorities.

#### 8. Enclosure and Integration:

- Construct an enclosure to safeguard the system against external influences and guarantee a sturdy integration with automobiles.

#### 9. Safety Measures:

- Make sure the gadget is dependable and impenetrable by putting safety mechanisms in place to stop system malfunctions or tampering.

#### 10. Evaluation and Adjustment:

Test the prototype thoroughly in a variety of scenarios to make sure it is accurate and dependable.

- To keep the measurement accurate, periodically calibrate the alcohol sensor.

#### 11. Legal and Compliance Considerations:

- Verify that the system conforms with national and international laws and guidelines concerning the use of alcohol detection devices and the prevention of drunk driving.

#### 12. Data recording

- If necessary, use data recording features to keep track of alcohol content for later use.

#### 13. Data Analytics:

Examine gathered information to find patterns and trends pertaining to instances involving drunk drivers.

#### 14. Maintenance and Support:

- Provide maintenance and support services to guarantee the system's continuous accuracy and functionality.

15. Scalability: Build the prototype with scalability in mind to enable integration into different vehicle types and possible mass manufacturing.

16. Public Education and Awareness • Take into account launching public education initiatives to let drivers know why and where the system is there.

## **REQUIRED COMPONENTS TO DEVELOP SOLUTIONS (Including IDE name, software and hardware)**

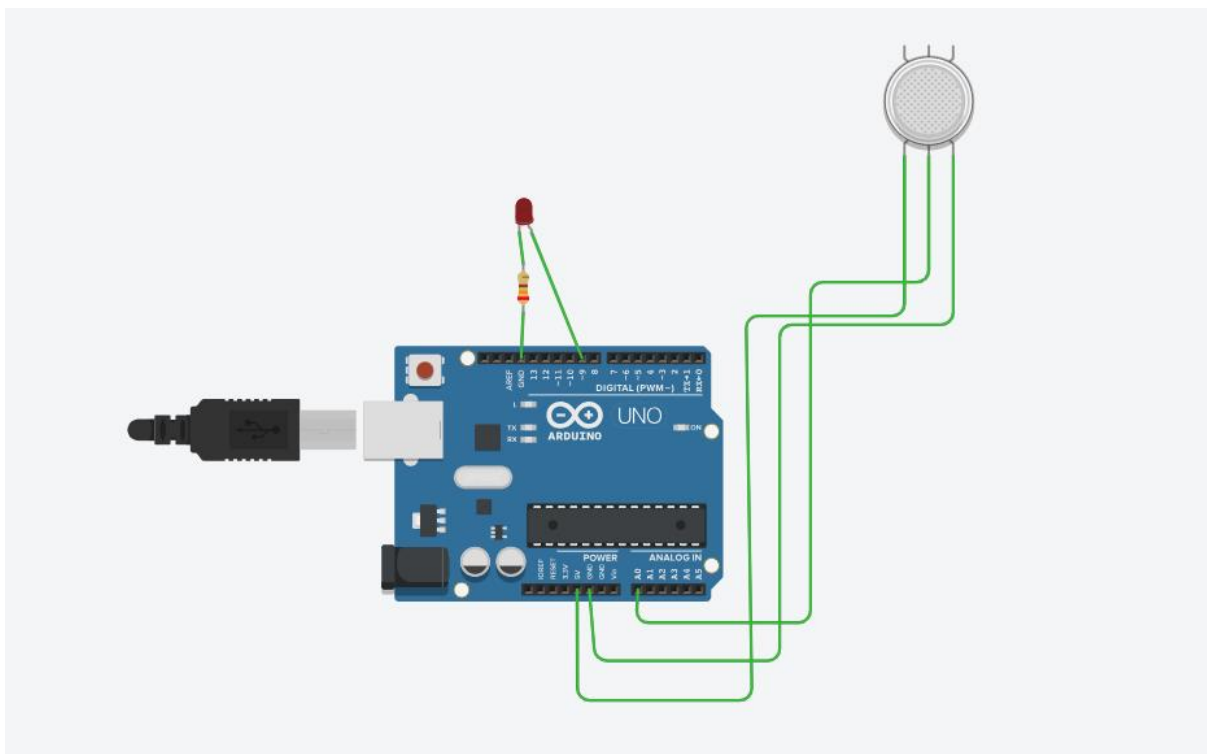
### **Hardware:**

- MQ-3 or MQ-5 alcohol sensors
- Microcontroller: Arduino Uno
- LCD Display
- Mechanism of Interlock
- Power Source (automotive or battery)
- User interface: touchscreen, LEDs, or buttons
- Module for communication (Bluetooth, Wi-Fi, etc.)
- Connectors and wiring

### **Software:**

- Arduino IDE (for microcontroller programming)
- Programming Skills in C/C++
- Tinkercad (for circuit simulation)

### **Simulated Circuit (TinkerCad)**



**Code for the Solution:**

```
const int gasSensorPin = A0;
const int alarmPin = 9; // LED connected to pin 9
void setup()
{
  pinMode(alarmPin, OUTPUT);
  Serial.begin(9600); // Initialize serial communication for debugging
}
void loop()
{
  int gasLevel = analogRead(gasSensorPin);
  Serial.print("Gas Level: ");
  Serial.println(gasLevel);
  if (gasLevel >= 00)
  {
    // Adjust the threshold as needed
    digitalWrite(alarmPin, HIGH); // Turn on the alarm (LED)
    Serial.println("Warning: Alcohol detected!");
  }
  else
  {
    digitalWrite(alarmPin, LOW); // Turn off the alarm
  }

  delay(1000); // Read gas level every second (adjust as needed)
}
```