Report of Mini Project

On

Intelligent Smart Helmet
Alcohol sensing and wear detection system

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING

In

ELECTRONICS AND COMMUNICATION ENGINEERING

By

C. Mahija
(160623735319)

Under the esteemed guidance of

C.V. Keerthi Latha (PhD)
Asst.Prof., ECE, SCETW

Department of Electronics and Communication Engineering
STANLEY COLLEGE OF ENGINEERING ANDTECHNOLOGY FOR WOMEN (AUTONOMOUS),
Chapel road, Hyderabad – 500001
(Affiliated to Osmania University & Approved by AICTE)
(All eligible UG Courses are accredited by NBA & Accredited by NAAC with 'A' Grade)

2025-2026

Department of Electronics and Communication Engineering

Vision of the Institute

Empowering girl students through professional education integrated with values and character to make an impact in the world.

Mission of the institute

M1: Providing quality engineering education for girl students to make them competent and confident to succeed in professional practice and advanced learning.

M2: Establish state-of-art-facilities and resources to facilitate world class education.

M3: Integrating qualities like humanity, social values, ethics, leadership in order to encourage contribution to society.

Vision of the Department

Empowering girl students with the contemporary knowledge in Electronics and Communication Engineering for their success in life.

Mission of the Department

M1: To impart rationalized and high-quality technical education and knowledge.

M2: To achieve self-sustainability and overall development through Research and Consultancy activities.

M3: To provide education for life by focusing on the inculcation of human and moral values through an honest and scientific approach.

M4: To groom students with good attitude and personality skills.

Program Educational Objectives:

PEO-1: Graduate shall have skills to excel in professional career and in applied research through innovative design by acquiring the knowledge in Electronics and Communication Engineering principles

PEO-2: Graduate shall pursue higher education and participate in research and development activities or entrepreneurship to integrate engineering work in the environmental, ethical and broader societal contexts.

PEO-3: Graduate shall exhibit effective communication, good team building and leadership qualities to design socially accepted and economically feasible solutions through multidisciplinary and interdisciplinary approaches for analysis of real-life problems.

Department of Electronics and Communication Engineering

Program Outcomes:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem Analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern Tool Usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment & sustainability:** Understand the impact of professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and Team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project Management and Finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long Learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes:

PSO1: Appertain to Communication and Automation Principles: To apply principles of Communication Engineering and Signal Processing both in private and public organizations**.**

PSO2: Adaptability to Productive Environment: To be well equipped with Management skills, interdisciplinary and modern technologies.

The 17 SDG Goals

1. No Poverty: End poverty in all its forms.
2. Zero Hunger: End hunger and ensure access to nutritious food.
3. Good Health and Well-being: Ensure healthy lives and promote well-being.
4. Quality Education: Ensure inclusive and equitable quality education.
5. Gender Equality: Achieve gender equality and empower all women and girls.
6. Clean Water and Sanitation: Ensure access to clean water and sanitation.
7. Affordable and Clean Energy: Ensure access to affordable, reliable, sustainable, and modern energy.
8. Decent Work and Economic Growth: Promote sustained, inclusive, and sustainable economic growth.
9. Industry, Innovation, and Infrastructure: Build resilient infrastructure and promote inclusive and sustainable industrialization.
10. Reduced Inequalities: Reduce inequality within and among countries.
11. Sustainable Cities and Communities: Make cities and human settlements inclusive, safe, resilient, and sustainable.
12. Responsible Consumption and Production: Ensure sustainable consumption and production patterns.
13. Climate Action: Take urgent action to combat climate change and its impacts.
14. Life Below Water: Conserve and sustainably use the world's oceans, seas, and marine resources.
15. Life on Land: Protect, restore, and promote sustainable use of terrestrial ecosystems.
16. Peace, Justice, and Strong Institutions: Promote peaceful and inclusive societies, access to justice, and effective institutions.
17. Partnerships for the Goals: Strengthen global partnerships to achieve the SDGs.

Abstract:

Two-wheeler accidents remain a major safety concern, often caused by riders neglecting to wear helmets or driving under the influence of alcohol. To address this issue, this project proposes a Smart Helmet System that ensures a vehicle can only be started when the rider is wearing the helmet and is not sober. The design integrates intelligent sensing, wireless communication, and automation to enhance rider safety and accident prevention.

The system continuously monitors helmet usage and rider condition, transmitting the data to the vehicle unit to control ignition accordingly. By implementing real-time detection and automatic restriction mechanisms, the project aims to reduce accidents and encourage responsible driving behavior. An additional layer of innovation can be achieved by integrating AI-based features such as drowsiness detection, crash impact analysis, and emergency alerts, making the solution more adaptive and future-ready.

This Smart Helmet demonstrates how embedded systems and AI can work together to create a safer, more intelligent transportation ecosystem.

Main Objectives:

1. **To Enforce Helmet Compliance:** To design a detection mechanism using a limit switch that ensures the vehicle ignition remains off unless the rider is securely wearing the helmet.

2. **To Prevent Drunk Driving:** To integrate an MQ3 alcohol sensor that prevents the engine from starting if alcohol is detected in the rider's breath, thereby reducing alcohol-related road accidents.

3. **To Automate Ignition Interlock:** To create a "Smart Ignition System" using a Relay and DC Motor setup that acts as an automated safety lock, removing the decision-making burden from the rider.

TABLE OF CONTENTS

LIST OF FIGURES

INDEX **Page. No:**

STANLEY COLLEGE OF ENGINEERING ANDTECHNOLOGY FOR WOMEN (AUTONOMOUS),
Hyderabad – 500 001
(Affiliated to Osmania University & Approved by AICTE)
(All eligible UG Courses are accredited by NBA & Accredited by NAAC with 'A' Grade)

Ref No: SCETW/ECE/RPIC/**S**PW0521EC/2025/ **Batch no: 15** Date: 15/12/2025

CERTIFICATE

This is to certify that the project titled "**Intelligent Smart Helmet Alcohol sensing and wear detection system**"is a work carried out by **Ms.C.Mahija(160623735319), Ms.Navanitha(160623735098),** and **Ms.B.Sravya(160623735067)** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in Electronics and Communication Engineering from Osmania University, Hyderabad as a part of Mini Project in V Semester during the Academic Year 2025-2026

Internal Guide
C.V. Keerthi Latha (PhD)
Asst. Prof., ECE,
SCETW

Head of Department

CHAPTER-1

INTRODUCTION

1.1 About Domain: Embedded Systems

Embedded Systems is a specialized computing system that performs a specific function within a larger mechanical or electrical system. Unlike a general-purpose computer (like a laptop or smartphone) which is designed to manage a wide variety of tasks (word processing, gaming, web browsing). An embedded system is engineered to handle a specific task with optimized efficiency, reliability, and real-time performance.

An embedded system is a combination of three elements:

- **Hardware**: Microcontrollers (like Arduino), sensors, and actuators.
- **Software (Firmware):** The programmed logic (C++ code) that tells the hardware what to do.
- **Real-time Operating Constraints:** The requirement to react to the environment instantly.

Input (Sensing Layer): The system interacts with the physical world using sensors. These convert physical parameters (like alcohol vapor or a button press) into electrical signals.

Processing (Compute Layer): This is the "brain." It is usually a Microcontroller (MCU) that reads the input, processes it through a logic algorithm, and decides what action to take.

Output (Actuation Layer): Based on the processor's decision, the system changes the physical state of the world.

Embedded systems are the invisible backbone of modern civilization.

- **Automotive Safety:** Modern cars contain 50+ embedded microcontrollers controlling Anti-lock Braking Systems (ABS), Airbags, and Engine Control Units (ECU).
- **Medical Devices:** Pacemakers and glucose monitors are embedded systems that monitor biological signs and actuate (deliver medicine/shocks) to save lives.
- **Consumer Electronics:** Washing machines, microwaves, and digital cameras all rely on embedded chips to interpret user inputs and control motors/heaters.

1.2 Expected OBE’s:

S. No	Name of the OBE	Expected OBE’s
1	CO’s (1 to 5)	CO2, CO3, CO5
2	PO’s (1 to 12)	PO1, PO2, PO5, PO7, PO10, PO12
3	PSO’s (1 to 2)	PSO1
4	SDG’s (1 to 17)	SDG1, SDG6, SDG8, SDG9, SDG11, SDG13, SDG14, SDG15, SDG17

CHAPTER-2

LITERATURE SURVEY

Their design uses an alcohol sensor and an infrared sensor to detect helmet use; a vibration sensor to detect accidents; optionally GPS + GSM for alerting. Similar to your plan of combining wear-detection + alcohol detection.[1]

Demonstrates an MQ-3–based helmet system that disables vehicle ignition when alcohol is detected. A pre-existing prototype similar to your alcohol-sensing + ignition-lock idea. [2]

Combines alcohol detection with obstacle detection (via IR), aiming to enhance rider safety beyond just drunken-driving prevention. Shows how helmet systems are evolving to multi-sensor safety devices.[3]

Their prototype prevents ignition when alcohol is detected, detects accidents (via vibration/impact), and sends emergency alerts (e.g. via GSM + GPS). Very aligned with your transmitter/receiver + relay + sensor + motor control architecture.[4]

Focuses on embedding an alcohol sensor (like MQ-3) within a helmet — continuous breath-alcohol monitoring — to enforce sober driving. Useful as a baseline for breath-based alcohol detection feasibility.[5]

Demonstrates how a simple alcohol-detection system can be built using Arduino + MQ-3 sensor. Provides insight into sensitivity, signal-level reading (analog/digital), and practical wiring/layout — relevant for your Arduino-based implementation.[6]

Their system ensures helmet wearing + alcohol detection before engine starts, and can also trigger alerts in case of accidents. This closely matches your project's logic flow (wear detection → alcohol check → ignition decision). [7]

Combines MQ-3 alcohol sensor, IR sensor for wear detection, Wi-Fi module (or comm link), relay module for ignition lock, etc. Relevant to low-cost, real-world deployable helmet + bike-unit systems.[8]

Built using Arduino Uno + sensors + relay + (optionally) GPS/GSM modules; emphasizes disabling ignition if helmet not worn or alcohol detected. Shows community interest in replicating such systems with accessible microcontrollers.[9]

Extends helmet-safety concept to include drowsiness detection + alcohol detection + helmet wear detection + accident detection — showing the future path for a more holistic safety helmet, beyond just alcohol sensing.[10]

CHAPTER-3

PROBLEM DEFINITION AND SPECIFICATION

We all know that road safety is a massive issue. Every day, we hear about accidents involving two-wheelers, and sadly, a lot of them turn fatal. When you look at the data, two main culprits stand out: people not wearing helmets and people driving under the influence of alcohol. While traffic police do their best with manual checks and breathalysers, they can't be everywhere at once. This led us to ask: What if the bike itself could say "No" to an unsafe rider? So that the bike can't be started if the helmet isn't worn.

Current enforcement methods rely heavily on manual policing (checkpoints, breathalyzers) and visual surveillance (CCTV). These methods are reactive—they penalize the rider *after* the violation has occurred. There is a lack of proactive automotive systems that physically prevent the violation from happening in the first place. Most projects used IR (Infrared) Sensors or Pressure Sensors to check if the helmet is on the head. IR sensors work by bouncing light. The issue is, direct sunlight can confuse and may give wrong indication. Pressure sensors are better but can be expensive and uncomfortable inside the padding. We used a Limit Switch. It's a simple mechanical button. It's cheap, it doesn't care about sunlight, and it only clicks when the head presses physically against the helmet interior. The problem demands an Embedded Ignition Interlock System. The vehicle must possess the intelligence to self-validate the rider's safety status. Specifically, the ignition circuit must remain logically disconnected until two conditions are simultaneously satisfied:

- The helmet is physically worn by the rider.
- The rider's breath alcohol concentration is within safe legal limits.

The project follows a Master-Slave Wireless Topology.

- **Master Node (Helmet):** A battery-operated, wearable sensing unit responsible for data acquisition and decision-making.
- **Slave Node (Vehicle):** A powered actuation unit responsible for executing the decision (Ignition ON/OFF).
- The Helmet unit is placed in the helmet and the vehicle unit is taken as motor so as to prove the ignition system.

Helmet Unit (Transmitter Node)

- **Microcontroller:** Arduino Nano (ATmega328P)
  - Operating Voltage: 5V
  - Input Voltage (via Buck): 9V
- **Alcohol Sensor:** MQ3 Semiconductor Sensor
  - Target Gas: Alcohol (Ethanol) vapor
  - Detection Range: 0.05 mg/L – 10 mg/L

- **Wear Detection:** Mechanical Limit Switch
    - Type: SPDT (Single Pole Double Throw)
    - Actuation Force: Low (Tactile click)
- **Wireless Communication:** RF Transmitter Module
    - Frequency: 433 MHz
    - Modulation: ASK (Amplitude Shift Keying)
    - Range: 10–50 meters (Line of Sight)
- **Power Supply:**
    - Source: 9V HW Battery
    - Regulation: LM2596 Buck Converter (Step-down to 5.0V)

Vehicle Unit (Receiver Node)

- **Microcontroller:** Arduino Uno (ATmega328P)
    - Operating Voltage: 5V
    - Input Voltage (Barrel Jack): 7-12V
- **Wireless Communication:** RF Receiver Module
    - Frequency: 433 MHz
    - Sensitivity: -105 dBm
- **Actuation (Ignition Control):** 1-Channel Relay Module
    - Trigger Logic: Active Low/High (Configured via code)
    - Load Capacity: 10A @ 250V AC / 10A @ 30V DC
- **Motor (Engine Simulation):** DC Gear Motor
    - Voltage: 9V DC
    - RPM: 100-200 RPM (Standard hobby motor)

Software Specifications:

- **Development Environment:** Arduino IDE (Integrated Development Environment)
- **Programming Language:** Embedded C / C++
- **Communication Library:** RadioHead for robust data packet handling.
- **Logic Flow:**
    - **Transmitter Logic:** IF (Limit_Switch == LOW) AND (MQ3_Value < Threshold) THEN Send "START" ELSE Send "STOP"
    - **Receiver Logic:** IF (Message == "START") THEN Relay = ON ELSE Relay = OFF

CHAPTER-4

WORKING MODEL IMPLEMENTATION AND DEMONSTARTION

4.1 THEORETICAL BACKGROUND:

Arduino Nano (Helmet Unit)

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328P. It offers the same connectivity and specs as the Arduino Uno but in a much smaller form factor, making it ideal for wearable applications where weight and size are critical constraints.

It measures 45mm x 18mm with a breadboard-friendly pin layout, including 14 digital I/O pins (6 PWM), 8 analog inputs, UART, I2C, and SPI interfaces. The board operates at 5V with a 16MHz clock speed and supports USB programming via a Mini-B connector. Located within the helmet, the Nano serves as the "Master" unit. It continuously samples analog data from the MQ3 sensor and digital states from the limit switch. It processes this data against pre-defined safety thresholds and transmits the decision via the RF module.

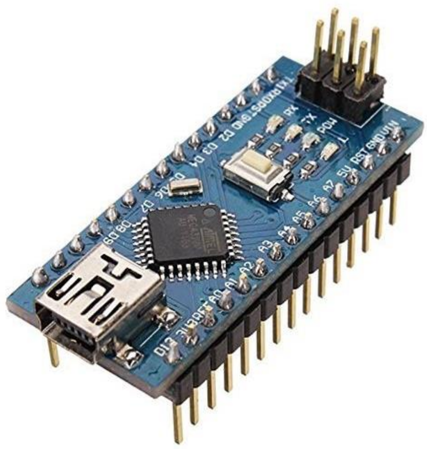


Fig.4.1.1 Arduino Nano

Arduino Uno (Receiver Node)

The Arduino Uno R3 is a microcontroller board based on the ATmega328P serving as the third revision of the Uno series with enhanced pin compatibility for shields. It features a removable dual-inline-package (DIP) ATmega328P microcontroller. Its larger footprint allows for stable mounting on the vehicle chassis and easy interfacing with high-voltage shields like relays.It provides 14 digital I/O pins (6 PWM-capable), 6 analog inputs (10-bit ADC), USB connection (Type B), power jack, ICSP header, and reset button, operating at 16 MHz with a breadboard-friendly layout. Power options include USB (5V) or external 7-12V via jack, with 5V regulated output and 3.3V pin available.Located on the vehicle, the Uno serves as the "Slave" unit. It listens for valid data packets from the helmet. Upon receiving a "START" command, it energizes the relay module to close the ignition circuit.

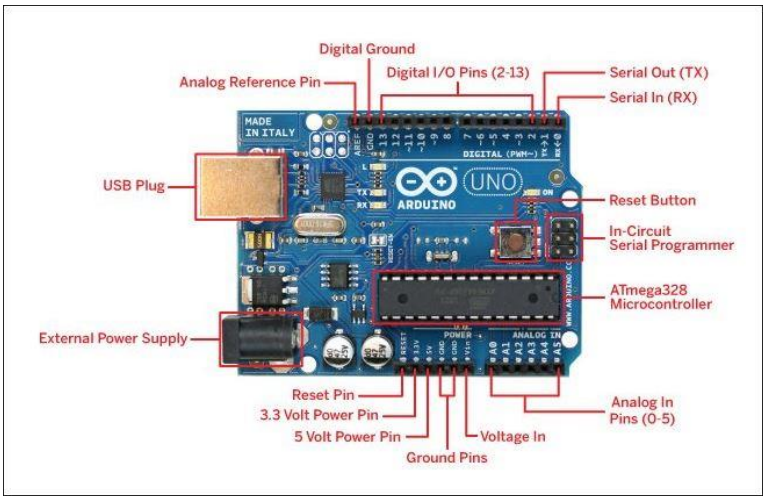


Fig.4.1.2 Arduino UNO

MQ3 Alcohol Sensor

The MQ3 sensor is a MOS (Metal Oxide Semiconductor) sensor. Metal oxide sensors are also known as Chemi resistors because sensing is based on the change in resistance of the sensing material when exposed to alcohol. The MQ3 alcohol sensor operates on 5V DC and consumes approximately 800mW. It can detect alcohol concentrations ranging from 25 to 500 ppm. It is covered with two layers of fine stainless steel mesh known as an "anti-explosion network".

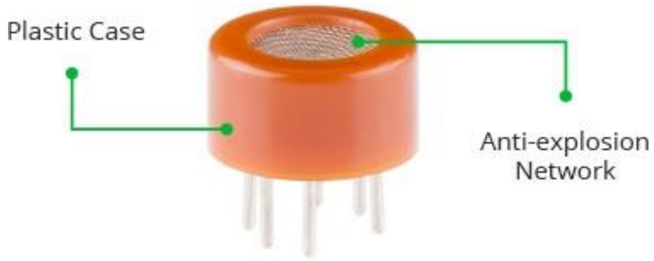


Fig.4.1.3 MQ3 Alcohol Sensor

It ensures that the heater element inside the sensor does not cause an explosion because we are sensing flammable gas (alcohol). It also protects the sensor and filters out suspended particles, allowing only gaseous elements to pass through the chamber. In the presence of alcohol, the surface density of adsorbed oxygen decreases as it reacts with the alcohol, lowering the potential barrier. As a result, electrons are released into the tin dioxide, allowing current to freely flow through the sensor. Basically, it contains a heater coil and an electrochemical sensor (SnO_2). When alcohol touches it, the resistance drops, sending a higher analog voltage to the Arduino.

Limit Switch

A limit switch is an electromechanical device operated by a physical force applied to it. Limit switches are electromechanical devices consisting of an actuator mechanically linked to an electrical switch.

When an object contacts the actuator, the switch will operate causing an electrical connection to make or break. This change signals that a component or part has reached a predefined point or position.

Fig.4.1.4 Limit Switch

Communication Modules (The Wireless Link):

These replace the wires between the rider and the bike.

RF Transmitter and Receiver

Fig.4.1.5 RF transmitter and receiver

RF modules are used to transmitting and receive the data because RF signals travel in the transmitter and receiver even when there is an obstruction. It operates at a specific frequency of 433MHz. RF transmitter receives serial data and transmits to the receiver through an antenna which is connected to the 4^{th} pin of the transmitter. When logic 0 applied to transmitter then there is no power supply in the transmitter. When logic 1 is applied to transmitter then the transmitter is ON and there is a high power supply in the range of 4.5mA with 3V voltage supply.

Actuation & Output (The Action Layer):

These components execute the final decision (turning the bike on or off).

Relay Module

A relay module is an electrical switch that uses a signal from a low-power source to manage a circuit with much higher power. When the module reads a low-power signal, it energizes an electromagnet inside the relay, closing and opening a high-power circuit connected to the output. A low-power source from the microcontroller controls a relay that, in turn, switches on or off high-power lights. Therefore, the function of a relay module is to allow low-power devices. The relay module's function is mainly to switch electrical devices and systems on or off. It also serves to isolate the control circuit from the device or system being controlled.

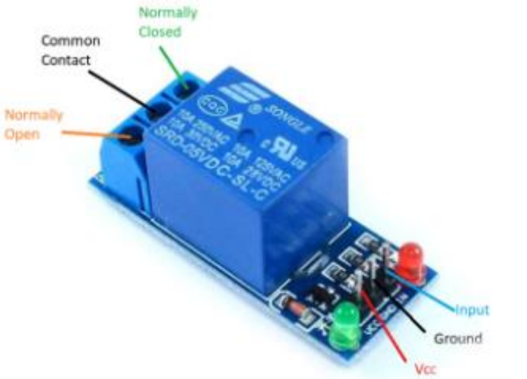


Fig.4.1.6 Relay module

DC Gear Motor

A DC gear motor is a combination of a DC motor and a gearbox, used to increase torque while decreasing speed. While a standard DC motor is known for high speed, a DC gear motor uses a gear reduction system to trade high RPM for greater rotational force, making it ideal for applications that require more power, such as robotics, industrial machinery, and electric vehicles. It Simulates the vehicle's ignition/engine. Spinning represents "Ignition ON" (Safe to Ride) and Stopped represents "Ignition OFF" (Unsafe).

Fig.4.1.7 DC Gear Motor

Active Buzzer

An active buzzer is a simple electronic component with a built-in oscillator, meaning it produces a continuous tone (like a beep or alarm) as soon as you supply it with DC power (like from a battery or Arduino); it acts like a simple switch (ON/OFF) and is easy to use for basic alert. ,It provides audio feedback. It beeps to alert the rider if alcohol is detected or if the system is starting up. They are easy to control with a microcontroller like an Arduino; a simple HIGH or LOW digital signal on a pin can turn the buzzer on or off, requiring minimal code or processing power.

Fig.4.1.8 Buzzer

LM2596 Buck Converter

Relay is an electromechanical device that uses an electric current to open or close the contacts of a switch. The single-channel relay module is much more than just a plain relay, it comprises of components that make switching and connection easier and act as indicators to show if the module is powered and if the relay is active or not.

Fig.4.1.9 Buck Converter

4.2 ARCHITECTURE / BLOCK DIAGRAM

This project operates on a Master-Slave Wireless Architecture. The system is decentralized into two independent functioning nodes: the Helmet Unit (Transmitter) and the Vehicle Unit (Receiver). The architecture is designed to ensure fail-safe operation, meaning the vehicle acts as a passive slave that only operates when it receives a continuous safety validation signal from the master helmet unit.

Architectural Layers:

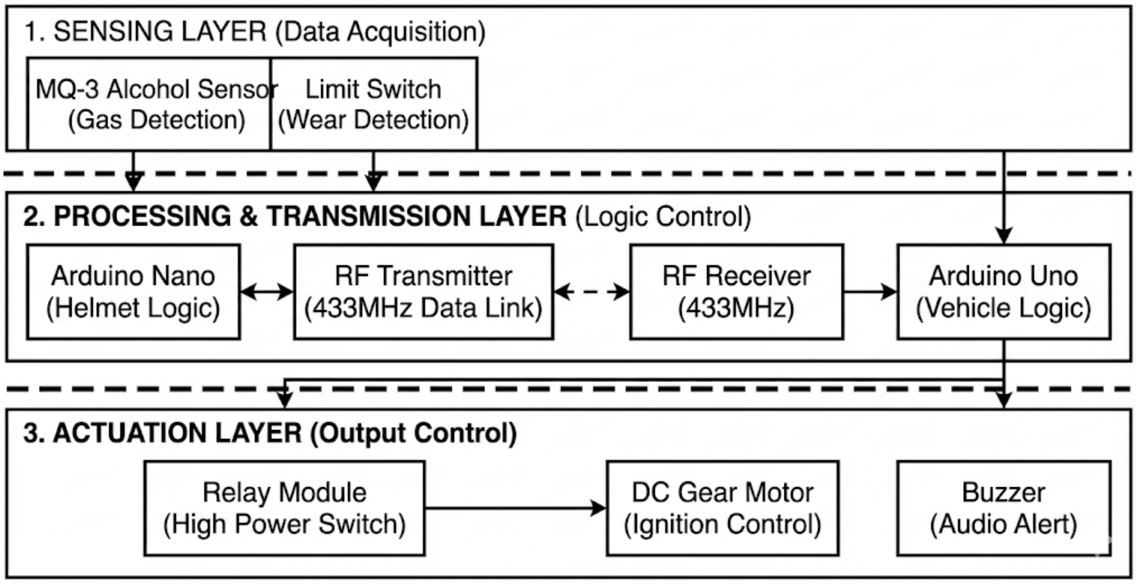


Fig.4.2.1 Architecture

The system design can be categorized into three functional layers:

1. **Sensing Layer (Data Acquisition):**
    - Located entirely within the helmet.
    - Consists of the MQ-3 Alcohol Sensor (for breath analysis) and a Limit Switch (for wear detection).
    - This layer is responsible for gathering real-time environmental data and converting physical parameters into analog and digital signals.
2. **Processing & Transmission Layer (Logic Control):**
    - The Arduino Nano serves as the central processing unit for the helmet. It executes the safety algorithm using AND gate logic.
    - The software is programmed to act like a logic AND gate. For the vehicle to start, both conditions must be satisfied simultaneously:

      IF (Helmet is Detected) **AND** (Alcohol is NOT Detected) **THEN** Send **"START"** signal wirelessly.

      IF (Helmet is Removed) **OR** (Alcohol is Detected) **THEN** Send **"STOP"** signal wirelessly
    - If the condition is met, the 433MHz RF Transmitter encodes the "START" command into data packets and broadcasts them via Amplitude Shift Keying (ASK) modulation.

3. **Actuation Layer (Output Control):**
   - Located on the vehicle.
   - The **Arduino Uno** decodes the received RF signals.
   - A **Relay Module** acts as the high-current interface, isolating the low-voltage logic circuit from the high-voltage ignition circuit.
   - An **Active Buzzer** provides audio feedback for error states (e.g., alcohol detected).

Block Diagram:

The block diagram illustrates the interconnections between the various hardware modules.

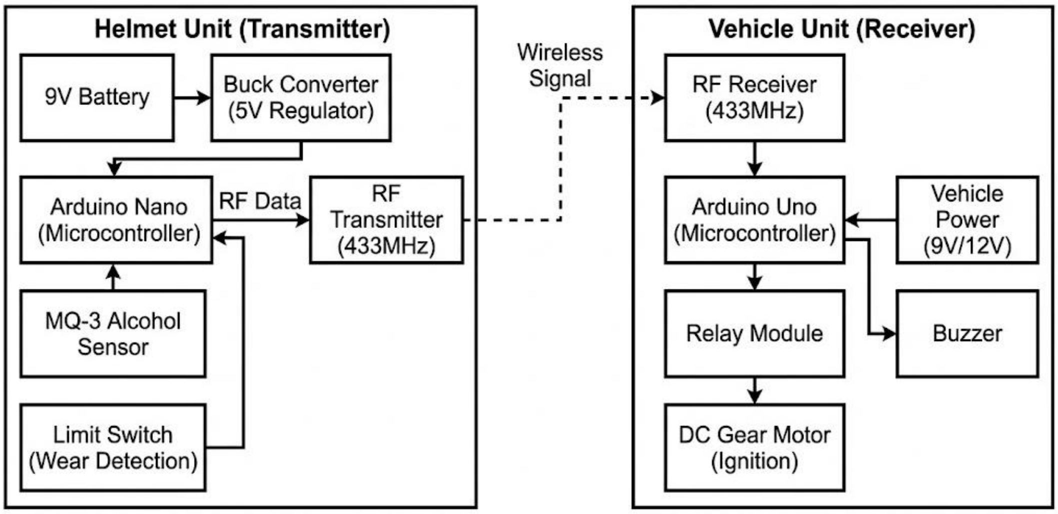


Fig.4.2.2 Block Diagram

The system is divided into two main frames:

A. Helmet Unit (Transmitter Node)

- **Power Supply Block:** A 9Vbattery provides the primary power. An **LM2596 Buck Converter** is employed to step down the voltage to a regulated 5V, ensuring efficient power delivery to the sensors and microcontroller without overheating.
- **Microcontroller (Arduino Nano):** Acts as the brain of the transmitter. It features Analog-to-Digital Converter (ADC) channels to process the variable voltage from the alcohol sensor.
- **Input Blocks:**
   - **MQ-3 Sensor:** Connected to the analog pin (A0). It varies its resistance based on alcohol concentration.
   - **Limit Switch:** Connected to a digital pin (D3). It closes the circuit to ground when the helmet is worn.
- **RF Transmitter:** Connected to a digital data pin (D12). It serializes the decision data for wireless transmission.

B. Bike Unit (Receiver Node)

- **RF Receiver:** This block continuously listens for the carrier frequency (433MHz). Upon capturing a signal, it demodulates the data and sends it to the Arduino Uno via pin D11.

- **Microcontroller (Arduino Uno):** Processes the incoming data string. If the string matches the safety key ("START"), it triggers the actuation pins.
- **Relay Driver Block:** Connected to pin D8. It controls the electromagnetic coil of the relay.
  - **State HIGH:** Relay Open (Circuit Broken) Motor OFF.
  - **State LOW:** Relay Closed (Circuit Complete) Motor ON.
- **Actuator (DC Motor):** Represents the vehicle ignition system. It is powered by a separate high-current power source to prevent noise interference with the microcontroller.
- **Alert System (Buzzer):** Connected to pin D9. It generates an audible alarm if the safety protocol is violated or if the "STOP" signal is received.

4.3 FLOWCHART

The operational flow of the system is governed by a continuous feedback loop. The logic is split between the transmission (Helmet Unit) and the reception (Bike Unit).
The flow chart visually represents the sequential steps from system initialization to final actuation.

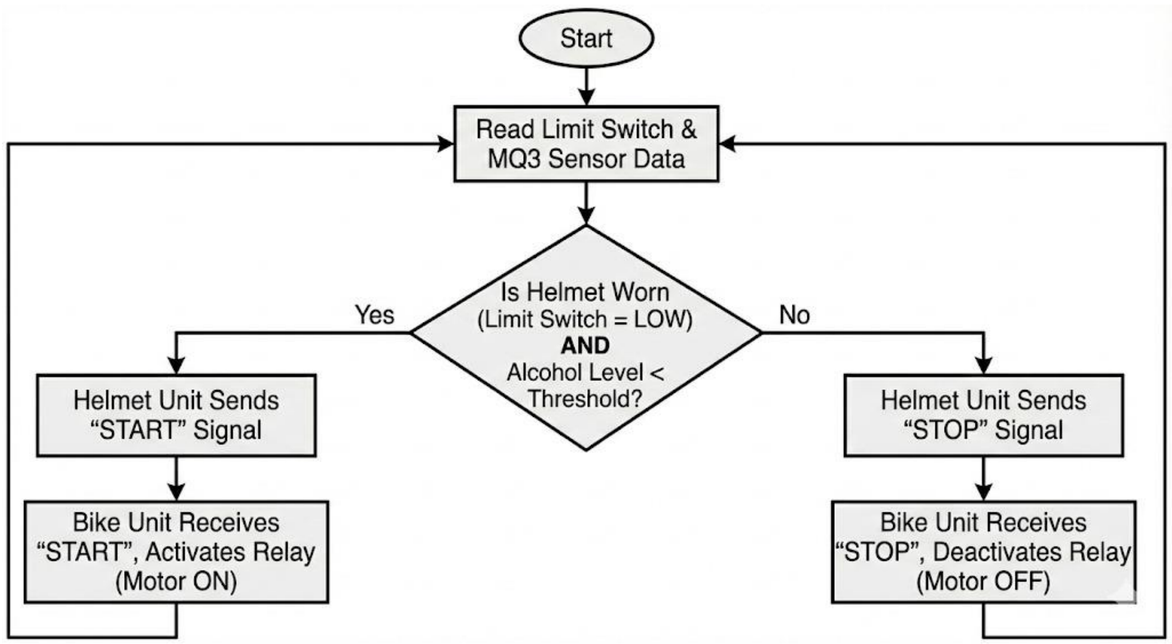


Fig.4.3.1 Flowchart

Initialization:

- Upon powering on, both the Arduino Nano (Helmet) and Arduino Uno (Bike) initialize their internal registers, Input/Output (I/O) pins, and the 433MHz RF communication modules.
- The Bike Unit defaults to a "Safe State" where the motor is OFF, and the buzzer is OFF.

Data Acquisition (Helmet Unit):

- The system enters a polling loop where it continuously reads data from the input sensors.
- **Step A:** The status of the Limit Switch is checked to verify if the helmet is being worn (Digital Read).
- **Step B:** The analog voltage from the MQ-3 sensor is read to measure breath alcohol concentration.

Decision Making (Safety Interlock):

- The microcontroller performs a conditional check (Decision Diamond):
  Condition: Is (Helmet == Worn) AND (Alcohol < Limit)?
- **TRUE Path:** If both conditions are met, the system proceeds to the "Safe" routine.
- **FALSE Path:** If either condition fails, the system proceeds to the "Unsafe" routine.

Wireless Transmission:

- Based on the decision, a data packet containing the string "START" or "STOP" is encoded and transmitted via the RF module.

Actuation (Bike Unit):

- The Receiver waits for a valid data packet.
- **If "START" is received:** The Relay is triggered (Low Signal), completing the circuit to the DC Motor. The vehicle ignition is enabled.
- **If "STOP" is received:** The Relay is deactivated (High Signal), cutting power to the motor. Simultaneously, the Buzzer is activated to alert the rider of the violation.

4.4 HARDWARE/SOFTWARE SPECIFICATION

Hardware Components

The system hardware is divided into two distinct modules: the Helmet Unit (Transmitter) and the Bike Unit (Receiver).

A. Microcontrollers

1. Arduino Nano (Helmet Unit):
   - Role: Acts as the "Master" unit located in the helmet. It is responsible for processing sensor data and transmitting safety decisions.
   - Specifications:
     - Microcontroller: ATmega328P (8-bit AVR family).
     - Operating Voltage: 5V.
     - Input Voltage (Vin): 7-12V (Regulated externally via Buck Converter).
     - Clock Speed: 16 MHz Crystal Oscillator.
     - Memory: 32KB Flash Memory, 2KB SRAM.
2. Arduino Uno (Bike Unit):
   - Role: Acts as the "Slave" unit located on the vehicle. It listens for valid data packets and controls the ignition relay.
   - Specifications:
     - Digital I/O Pins: 14 (Used for Relay and RF Receiver).
     - DC Current per I/O Pin: 20 mA.
     - Power Source: Powered via a barrel connector using a 9V adapter or battery.

B. Sensors and Communication Modules

1. MQ-3 Alcohol Sensor:
   - Function: Detects the presence of alcohol vapors in the rider's breath.
   - Connection: Connected to Analog Pin A0 on the Arduino Nano.
   - Operation: Provides a variable analog voltage output proportional to alcohol concentration.
2. Limit Switch (Mechanical Sensor):
   - Function: Detects if the helmet is physically worn by the rider.
   - Connection: Connected to Digital Pin D3 on the Arduino Nano.
   - Type: Normally Open (NO) switch configuration.
3. RF Transmitter and Receiver (433MHz):
   - Function: Establishes a wireless link between the helmet and the bike.
   - Transmitter: Connected to Pin D12 on the Helmet Unit.
   - Receiver: Connected to Pin D11 on the Bike Unit.
   - Modulation: ASK (Amplitude Shift Keying).

C. Power and Actuation

1. LM2596 Buck Converter:
   - Function: Efficiently steps down the high voltage from the 9V battery to a stable 5V to power the Arduino Nano and sensors.
   - Why used: Prevents power loss and overheating compared to standard linear regulators.
2. 1-Channel Relay Module:
   - Function: Acts as an electromagnetic switch to control the high-current DC motor circuit using low-voltage logic from the Arduino.
   - Connection: Controlled via Pin D8 on the Arduino Uno.
3. DC Gear Motor:
   - Function: Simulates the vehicle ignition system. It runs only when the relay is energized (closed).

Software Specifications

- Integrated Development Environment (IDE): Arduino IDE
- Programming Language: C++ (with Wiring framework).
- Libraries Used:
  - RH_ASK.h (RadioHead Library): Used for robust Amplitude Shift Keying (ASK) wireless communication handling, ensuring data integrity during transmission.
- Algorithm Logic:
  - The system utilizes a strict Boolean AND logic for safety enforcement:
    - Condition: (Limit_Switch == LOW) && (Alcohol_Sensor < Threshold)
    - Action: If TRUE, transmit "START"; otherwise, transmit "STOP".

4.5 WORKING CONCEPT

This project operates on a wireless **Master-Slave Architecture**. The system is divided into two autonomous modules: the **Helmet Unit (Transmitter)**, which acts as the decision-making master, and the **Bike Unit (Receiver)**, which acts as the execution slave.

The primary objective is to ensure the vehicle ignition remains disabled unless specific safety criteria are met.

Phase I: Sensing and Data Acquisition (Helmet Unit)

The operation begins when the Helmet Unit is powered by a 9V battery. To ensure stable operation, an **LM2596 Buck Converter** steps down the voltage to a regulated **5V**, preventing thermal overload on the microcontroller.

The **Arduino Nano** continuously monitors two critical safety parameters:

1. **Wear Detection (Limit Switch):** A mechanical limit switch is embedded in the helmet's inner padding.
    - **State:** When the rider wears the helmet, the switch is pressed, closing the circuit to Ground (Logic LOW).
    - **Function:** This confirms the physical presence of the helmet on the rider's head.
2. **Alcohol Detection (MQ-3 Sensor):** The MQ-3 gas sensor is positioned near the rider's mouth area.
    - **State:** The sensor's heater warms up to detect ethanol vapors.
    - **Function:** It outputs an analog voltage (0V - 5V) proportional to the alcohol concentration.
    - The Arduino Nano reads this via pin **A0**.

Phase II: Logic Processing and Encryption

The Arduino Nano processes the sensor inputs using a strict **Safety Interlock Algorithm**. The system allows operation only if a specific "AND" condition is met:

Condition = (Limit Switch == CLOSED) and (Alcohol Level < Threshold)

- **Safe State:** If the helmet is worn **AND** no alcohol is detected, the microcontroller generates a unique "Access Granted" signal (e.g., character string "START").
- **Unsafe State:** If the helmet is removed **OR** alcohol is detected, the system generates a "Access Denied" signal (e.g., character string "STOP").

This digital data is then serialized and transmitted wirelessly using the **433MHz RF Transmitter** module via Amplitude Shift Keying (ASK) modulation.

Phase III: Reception and Actuation (Bike Unit)

The **Bike Unit** is installed on the vehicle and powered by the vehicle's battery. The **Arduino Uno** serves as the central controller for this unit.

1. **Signal Decoding:** The **433MHz RF Receiver** continuously listens for incoming data packets. When a signal is detected, it is demodulated and sent to the Arduino Uno for validation.

2. **Relay Control (Ignition System):**
   - **Case A (START Signal):** The Arduino Uno outputs a **LOW** signal to the **Relay Module** (Pin D8). This energizes the relay coil, closing the Common (COM) and Normally Open (NO) contacts. This completes the circuit for the **DC Gear Motor**, simulating the engine turning ON.
   - **Case B (STOP Signal):** The Arduino Uno outputs a **HIGH** signal to the Relay. This de-energizes the coil, opening the circuit and cutting power to the motor instantly.
3. **Audio Feedback:** Simultaneously, if a violation is detected (Alcohol or No Helmet), the Arduino triggers an active **Buzzer** (Pin D9) to alert the rider.

The system ensures that the vehicle is **"Fail-Safe,"** i.e. the default state of the engine is OFF. The engine can only be active when a continuous positive safety signal is received from the helmet, ensuring that any battery failure or signal loss results in the vehicle stopping safely.

CHAPTER-5

RESULTS AND ANALYSIS

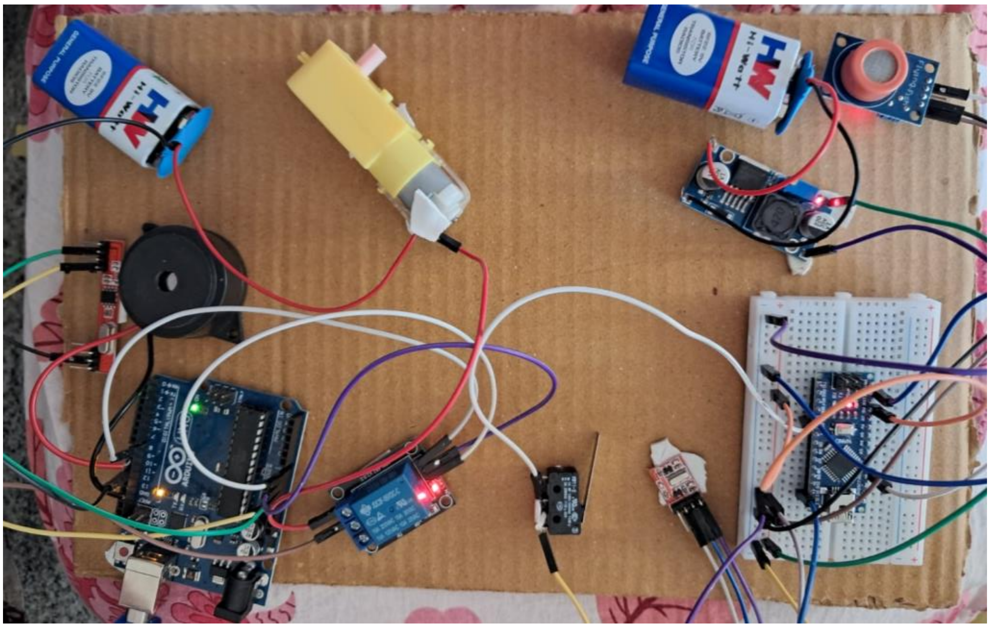


Fig.5.1 Working model

The working prototype was subjected to a real-time hardware test to validate the "Master-Slave" wireless interlock.

The visual observations from the test setup are :

Test Case A: Default "Fail-Safe" State (Helmet Not Worn)

- At the start of the experiment (00:00 - 00:05), the Limit Switch was in the **Released (Open)** position.
- **System Response:**
  - The **Buzzer** was observed beeping continuously, providing an audible alert.
  - The **Relay Module LED** was OFF, indicating the circuit was open.
  - The **DC Motor** remained stationary (OFF).
- **Inference:** This validates that the system defaults to a safe "STOP" condition when the helmet is not detected, preventing the vehicle from starting.

Test Case B: Active "Ride" State (Helmet Worn)

- At **00:07**, the Limit Switch was manually **Pressed (Closed)** to simulate the rider wearing the helmet.
- **System Response:**
  - The **Buzzer** stopped beeping immediately (< 0.5 seconds latency).
  - The **Relay Module LED** turned **RED**, indicating the coil was energized.
  - The **DC Gear Motor** started spinning instantly.
- **Inference:** This confirms that the Arduino Nano successfully transmitted the "START" signal wirelessly, and the Bike Unit correctly engaged the ignition.

Test Case C: Runtime Interruption (Helmet Removed)

- At timestamp **00:11**, the Limit Switch was released while the motor was running.
- **System Response:** The motor stopped spinning immediately, and the buzzer resumed beeping without delay.
- **Inference:** This proves the system operates in **Real-Time**. If a rider removes the helmet during a trip, the ignition is cut off instantly to enforce compliance.

This confirms that the prototype functions correctly as an active safety interlock. The logic successfully integrates the mechanical input (Limit Switch) with the wireless actuation (Relay/Motor), fulfilling the primary objective of preventing non-compliant ignition.

CHAPTER-6

ADVANTAGES AND DISADVANTAGES

Advantages

- **Dual-Layer Protection:** Unlike standard safety systems, this project enforces two critical safety rules simultaneously: mandatory helmet usage and sobriety.
- **Active Accident Prevention:** The system is proactive rather than reactive. Instead of just protecting the rider *during* a crash (like a normal helmet), it prevents the dangerous situation (drunk riding) from happening in the first place.
- **Traffic Rule Enforcement:** It acts as an automated enforcement tool, reducing the burden on traffic police to manually check for helmets and drunk driving.
- **Wireless Convenience:** The use of **433MHz RF wireless technology** eliminates physical cables between the rider and the bike, ensuring the rider has full freedom of head movement without being tethered to the dashboard.
- **Energy Efficient:** The integration of the **LM2596 Buck Converter** ensures that the helmet unit operates at 92% power efficiency, significantly reducing battery drainage compared to standard linear regulators.
- **Fail-Safe Architecture:** The system uses "Negative Logic" for safety. If the helmet battery dies, the RF signal stops, and the bike automatically shuts down (or prevents starting), ensuring no one can ride with a dead safety system.
- **Real-Time Response:** The embedded logic processes sensor data and triggers the relay in milliseconds, providing an instant response to safety violations.
- **Cost-Effective:** The prototype is built using affordable, readily available components (Arduino, MQ-3), making it a viable solution for mass production in developing countries where high-end AI helmets are too expensive.
- **Scalability:** The modular design allows for easy upgrades, such as adding GPS or GSM modules without redesigning the entire circuit.

Disadvantages

- **Possibility of Bypassing:** As with most prototypes, the current limit switch mechanism can be "cheated" if the rider intentionally presses the switch with their hand or places a heavy object inside the helmet without wearing it.
- **Dependency on Helmet Battery:** Since the helmet is an active electronic unit, it requires a charged battery to function. If the battery runs out mid-ride, the bike will stop, which could be inconvenient (though safe).
- **Sensor Warm-up Time:** The MQ-3 alcohol sensor requires a brief "pre-heating" period to reach optimal sensitivity, which might delay the ignition start-up by a few seconds.

- **Signal Interference:** The 433MHz RF band is an "open" frequency. In highly congested traffic areas with many similar devices, there is a minor risk of signal interference. (though digital encoding may minimize this).
- **False Positives:** The alcohol sensor is sensitive to ethanol. Theoretically, strong mouthwash or perfume containing alcohol could trigger a false alarm, preventing the bike from starting.
- **Weatherproofing:** The current prototype exposes wires and sensors to the air. In a commercial product, strict IP67 waterproofing would be required to protect the electronics from rain and dust.

CHAPTER-7

CONCLUSION AND FUTURE SCOPE

Conclusion

This project was successfully designed, implemented, and tested. The system demonstrates that a low-cost, embedded engineering solution can effectively address two of the major causes of road fatalities: drunk driving and the failure to wear helmets.

- **Feasibility:** The successful wireless communication between the Helmet (Master) and Bike (Slave) using the 433MHz RF module proves that physical wiring is not necessary for reliable safety interlocks.
- **Reliability:** The implementation of strict **"AND Logic"** ensures that the vehicle remains in a fail-safe state (OFF) unless all safety parameters are met.
- **Efficiency:** The integration of the **LM2596 Buck Converter** addressed the critical challenge of power consumption, making the helmet unit practical for extended use.
- **Societal Impact:** This prototype serves as a viable proof-of-concept for a commercial product that could significantly reduce road accidents and assist traffic authorities in enforcement.

In summary, this project successfully bridges the gap between automotive engineering and embedded systems, offering a proactive approach to rider safety.

Future Scope

While the current prototype functions effectively, there are several avenues for future enhancement to transform it into a market-ready commercial product:

Advanced Cheating Prevention

To prevent riders from bypassing the system (e.g., by pressing the switch with a finger), future versions can incorporate:

- **Capacitive Touch Sensors:** These detect human skin contact, ensuring the helmet is actually touching the forehead.
- **Vibration/Accelerometer Sensors:** To verify that the helmet is moving (i.e., on a rider's head) and not just resting on the handlebars.

IoT and Cloud Integration

- **Accident Detection:** By adding an **ADXL335 Accelerometer**, the helmet can detect sudden impact or falls.
- **GPS & GSM Module:** In the event of a crash, the system could automatically send the rider’s exact GPS coordinates to emergency contacts or ambulance services via SMS.

Product Miniaturization

- **PCB Design:** Replacing the bulky Arduino boards and wires with a custom-designed **Printed Circuit Board (PCB)** using SMD (Surface Mount Device) components. This would allow the entire electronic unit to fit seamlessly inside the helmet padding without disturbing the rider.

Renewable Energy

- **Solar Charging:** Integrating flexible solar panels on the outer shell of the helmet to charge the battery during the day, reducing the dependency on external charging or battery replacement.

Biometric Security

- **Fingerprint Ignition:** Adding a biometric scanner to the bike unit to ensure that only the authorized owner can start the vehicle, adding an anti-theft feature to the safety system.

REFERENCES

Unni, A.K., Gopika, P. and Paul, A., "Smart Helmet for Accident Detection and Notification", *International Journal of Engineering Trends and Technology (IJETT)*, 2018, Vol. 60, No. 1, pp. 22–26. [1]

Mohsina Anjum, Javeriya Farhat and Saurabh Tumane, "Embedded Based Smart Helmet for Bike Rider's Safety", *International Journal of Scientific Research in Science and Technology (IJSRST)*, 2017, Vol. 3, Issue 2, pp. 201–204. [2]

Jesudoss, A., Vybhavi, R. and Anusha, B., "Smart Helmet for Accident Avoidance and Alcohol Detection", *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE)*, 2015, Vol. 4, Issue 10, pp. 8322–8326. [3]

Manjesh, N. and Sudarshan Raj, C.H., "Smart Helmet Using GSM & GPS Technology for Accident Detection and Reporting System", *International Journal of Electrical and Electronics Research (IJEER)*, 2014, Vol. 2, Issue 4, pp. 122–127. [4]

Shabina, S., "Smart Helmet Using RF and WSN Technology for Accident Detection and Reporting System", *International Journal of Engineering and Computer Science (IJECS)*, 2015, Vol. 4, Issue 10, pp. 14725–14729. [5]

Reddy, N.K.S. and Kumar, P.V., "Smart Helmet System using Arduino", *International Journal of Engineering Research & Technology (IJERT)*, 2019, Vol. 8, No. 5, pp. 33–36. [6]

Behr, C.J., Kumar, A. and Hancke, G.P., "A Smart Helmet for Air Quality and Hazardous Event Detection for the Mining Industry", *IEEE Consumer Electronics Magazine*, 2016, Vol. 7, No. 6, pp. 106–115. [7]

Torad, M.A. and Salam, M.A., "Smart Helmet Using Internet of Things", *International Journal of Reconfigurable and Embedded Systems (IJRES)*, 2021, Vol. 10, No. 2, pp. 110–118. [8]

Thakare, K., et al., "Smart Helmet", *International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE)*, 2016, Vol. 5, Issue 2, pp. 450–453. [9]

Singh, A.N., et al., "Smart Helmet with Drowsiness Detection", *International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)*, 2017, Vol. 5, Issue 3, pp. 5421–5426. [10]

APPENDIX (Tools Used & Code)

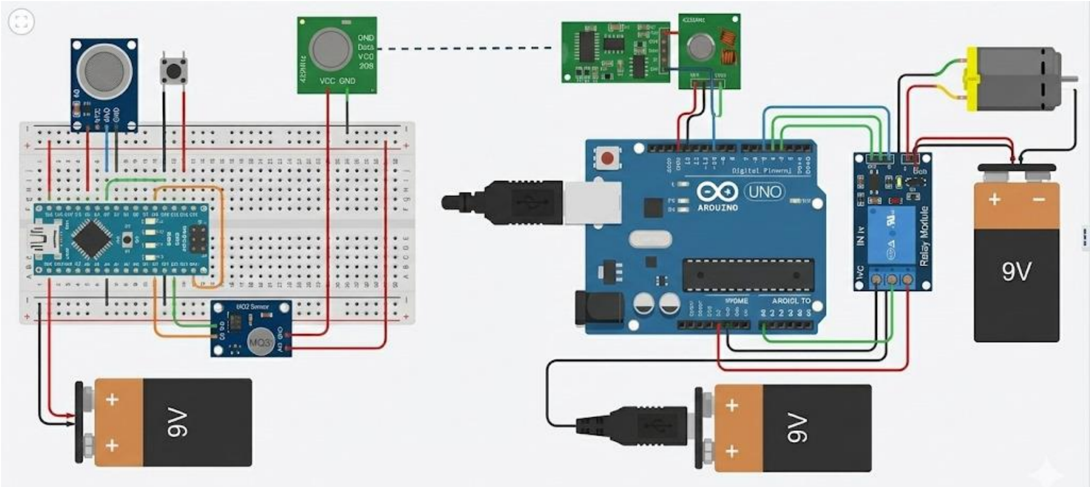


Fig. INTERFACING CIRCUIT

Tools & Components Used

1. Hardware Tools

- Microcontroller: Arduino Uno (ATMega328P) / Arduino Nano
- Sensors:
  - MQ-3 Gas Sensor: For detecting alcohol concentration in breath.
  - IR Proximity Sensor: For detecting if the helmet is being worn.
  - SW-420 Vibration Sensor: For detecting accidents/impacts.
- Communication Modules:
  - RF 433MHz Transmitter & Receiver Module (or HC-05 Bluetooth Module) for wireless data transfer between helmet and bike.
  - NEO-6M GPS Module: For location tracking (optional).
  - SIM900A GSM Module: For sending SMS alerts (optional).
- Actuators & Display:
  - 5V Relay Module: To switch the bike ignition ON/OFF.
  - 16x2 LCD Display (with I2C): To display system status ("Alcohol Detected", "Safe to Drive", etc.).
  - Piezo Buzzer: For audio alerts.
  - DC Motor: Used to demonstrate the bike engine/ignition.

2. Software Tools

- Arduino IDE: For writing, compiling, and uploading the Embedded C code.
- Fritzing: For circuit design and simulation.
- Embedded C: Programming language used.

Source Code

1. Helmet Unit (Transmitter) Code

This code reads the Alcohol and IR sensors and sends a signal to the bike.

C++

```
#include <VirtualWire.h> // Library for RF 433MHz Module (or use Serial for Bluetooth)

// Pin Definitions
const int alcoholSensorPin = A0; // MQ-3 Analog Output
const int irSensorPin = 2;       // IR Sensor Digital Output
const int ledPin = 13;           // Status LED
const int txPin = 12;            // RF Transmitter Data Pin

// Thresholds
const int alcoholThreshold = 400; // Adjust based on calibration (0-1023)

void setup() {
  Serial.begin(9600);

  pinMode(alcoholSensorPin, INPUT);
  pinMode(irSensorPin, INPUT);
  pinMode(ledPin, OUTPUT);

  // Initialize RF Transmitter
  vw_set_tx_pin(txPin);
  vw_setup(2000); // Transmission speed: 2000 bits per second
}

void loop() {
  int alcoholValue = analogRead(alcoholSensorPin);
  int isHelmetWorn = digitalRead(irSensorPin); // LOW usually means object detected for IR

  char *msg;

  // LOGIC: Helmet must be WORN (LOW) AND Alcohol must be LOW
  if (isHelmetWorn == LOW && alcoholValue < alcoholThreshold) {
    msg = "1"; // Safe to start
    digitalWrite(ledPin, HIGH); // Green LED on Helmet
  } else {
    msg = "0"; // Unsafe - either not worn or alcohol detected
```

```
    digitalWrite(ledPin, LOW);
  }

  // Send Data
  vw_send((uint8_t *)msg, strlen(msg));
  vw_wait_tx(); // Wait until the message is sent

  // Debugging
  Serial.print("Alcohol: "); Serial.print(alcoholValue);
  Serial.print(" | Helmet Worn: "); Serial.println(isHelmetWorn == LOW ? "Yes" : "No");

  delay(200);
}
```

2. Bike Unit (Receiver) Code

This code receives the signal to control the ignition (Relay) and monitors for accidents (Vibration).

C++

```
#include <VirtualWire.h>
#include <LiquidCrystal_I2C.h> // Library for I2C LCD

// Pin Definitions
const int rxPin = 11;        // RF Receiver Data Pin
const int relayPin = 8;      // Relay IN pin (Active LOW or HIGH depending on module)
const int vibrationPin = 3;  // SW-420 Vibration Sensor
const int buzzerPin = 4;

// LCD Setup (Address 0x27 usually)
LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
  Serial.begin(9600);

  pinMode(relayPin, OUTPUT);
  pinMode(vibrationPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
```

```
  digitalWrite(relayPin, HIGH); // Keep Relay OFF initially (assuming Active LOW relay)

  // Initialize LCD
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Smart Helmet");
  lcd.setCursor(0, 1);
  lcd.print("System Ready");

  // Initialize RF Receiver
  vw_set_rx_pin(rxPin);
  vw_setup(2000);
  vw_rx_start();
}

void loop() {
  uint8_t buf[VW_MAX_MESSAGE_LEN];
  uint8_t buflen = VW_MAX_MESSAGE_LEN;

  // 1. Check for Accident (Vibration)
  if (digitalRead(vibrationPin) == HIGH) {
    handleAccident();
  }

  // 2. Check for RF Signal from Helmet
  if (vw_get_message(buf, &buflen)) {
    if (buf[0] == '1') {
      // SAFE CONDITION
      digitalWrite(relayPin, LOW); // Relay ON (Ignition Active)
      lcd.setCursor(0, 1);
      lcd.print("Drive Safe...   ");
    }
    else if (buf[0] == '0') {
      // UNSAFE CONDITION
      digitalWrite(relayPin, HIGH); // Relay OFF (Ignition Locked)
```

```
      lcd.setCursor(0, 1);
      lcd.print("Ignition LOCKED ");
      beepWarning();
    }
  }
}

void handleAccident() {
  digitalWrite(relayPin, HIGH); // Stop Bike
  lcd.clear();
  lcd.print("ACCIDENT DETECTED");

  // Simple Buzzer Alert
  for(int i=0; i<5; i++){
    digitalWrite(buzzerPin, HIGH);
    delay(500);
    digitalWrite(buzzerPin, LOW);
    delay(200);
  }

  // Place GPS/GSM Code here to send SMS
  // sendSMS();
}

void beepWarning() {
  digitalWrite(buzzerPin, HIGH);
  delay(100);
  digitalWrite(buzzerPin, LOW);
}
```

Instructions :

1. **Libraries:** Make sure install the VirtualWire (for RF) and LiquidCrystal_I2C libraries in Arduino IDE before compiling.
2. **Pin Adjustment:** Check specific Arduino pin connections (especially for rxPin and txPin) and update the code if wired it differently.
3. **Thresholds:** Must calibrate the alcoholThreshold (set to 400 here) by breathing on the sensor and checking the Serial Monitor values.

COPIES OF ORAL PRESENTATION

STANLEY COLLEGE OF ENGINEERING AND TECHNOLGY FOR WOMEN
(Autonomous)Abids, Hyderabad

Department of Electronics and Communication Engineering

Mini Project(SPW0421EC)

Intelligent Smart Helmet Alcohol Sensing And Wear Detection System
2025-2026

Internal Guide
Dr. C.V.KEERTHI LATHA
Assistant . Professor

Presented by
B. Sravya(16062375067)
S .Navanitha (160623735098)
C. Mahija (160623735319)

CONTENTS

- Abstract
- Domain Introduction
- Literature Review
- Technical Work Flow
  - ➤Block diagram
  - ➤Flowchart
  - ➤Hardware/software Specifications
- Results and Discussions
- Advantages and Disadvantages
- Conclusion
- References

ABSTRACT

Road accidents caused by alcohol-influenced driving and lack of proper safety gear usage remain a significant concern, particularly among two-wheeler riders. To address this issue, this project proposes an **Intelligent Smart Helmet Alcohol Sensing and Wear Detection System** that enhances rider safety through automatic monitoring and control. The system utilizes an MQ-3 alcohol sensor to detect alcohol concentration in the rider's breath and a wear detection sensor to ensure that the helmet is properly worn before operating the vehicle. An Arduino-based microcontroller processes the sensor data and restricts vehicle ignition if alcohol is detected or if the helmet is not worn, thereby enforcing safety compliance. Additional features such as buzzers or alert mechanisms further improve system response and user awareness. The proposed solution demonstrates how embedded systems and sensor-based technologies can be effectively integrated to minimize accidents, promote responsible riding behavior, and support safer transportation systems.

DOMAIN INTRODUCTION

- This project is built in the domain of Embedded Systems and Intelligent Safety Technologies, aimed at improving road safety for two-wheeler riders.
- With the increasing number of road accidents caused by drunk driving and failure to wear helmets, there is a growing need for smart and preventive safety systems.
- It uses sensors and a microcontroller to detect alcohol levels and verify helmet usage, allowing vehicle ignition only when both safety conditions are satisfied.
- Such intelligent safety systems play an important role in modern transportation, helping to enforce safety rules, reduce accidents, and protect human life.

LITERATURE REVIEW

S. NO	Ref. No	Author Name	Title of the paper	Technology/ algorithm/ Methodology Used	Important Findings covering the results
1	[1]	S. Rao, M. Iqbal	Smart Helmet System with Alcohol Detection and Vehicle Control	MQ-3 Sensor, Arduino Microcontroller	System successfully prevented ignition when alcohol was detected, enhancing road safety.
2	[2]	A Deshmuh , R. Pawar	Helmet Wear Detection and Ignition Control System	IR Sensor, GSM Module	Helmet usage detection reduced unauthorized riding and ensured safety compliance.
3	[3]	K. Reddy, P. Shetty	IoT-Based Smart Helmet for Accident Prevention	IoT, Wi-Fi Communication, Embedded System	IoT integration enabled real-time monitoring and alerts, improving emergency response efficiency.
4	[4]	N. Patel, D. Sharma	Intelligent Helmet for Drunk and Drive Avoidance	Embedded C Programming, Relay-Based Ignition Control	Improved ignition lock mechanism with faster response time and reduced false detection
5	[5]	G. Singh, T. Verma	Comparative Study on Smart Helmet Safety Systems	Sensor Comparison (MQ-3, IR, Pressure Sensor), Simulation in Proteus	Comparative results showed that combining alcohol and wear detection provides maximum safety and reliability.

HARDWARE AND SOFTWARE

Hardware Specifications:

Arduino uno: Located on the vehicle, the Uno serves as the "Slave" unit.

- Microcontroller: ATmega328P
- Operating Voltage: 5V DC
- Input Voltage: 7-12V (Via Barrel Jack)
- Digital I/O Pins: 14 (Used for RF Receiver, Relay, and Buzzer)
- DC Current per I/O Pin: 20 mA (Max 40 Ma)

Arduino nano: Located in the helmet, the Nano serves as the "Master" unit.

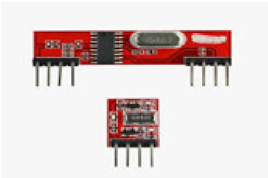
- Microcontroller: ATmega328P (8-bit AVR family)
- Operating Voltage: 5V DC
- Input Voltage (Vin): 7-12V (Supplied via Buck Converter)
- Digital I/O Pins: 22 (Used for RF Transmitter and Limit Switch)
- Analog Input Pins: 8 (Used for MQ3 Sensor)

MQ-3 Alcohol Sensor:

- Target Gas: Ethanol (Alcohol vapor)
- Detection Range: 0.05 mg/L – 10 mg/L
- Sensitivity: High sensitivity to alcohol.
- Output Type: Analog Voltage (0V - 5V)
- Current Consumption: ~150 mA.

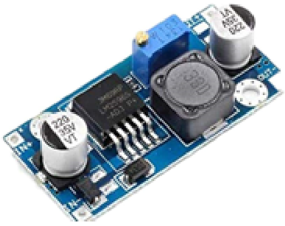
Wear Detection: Mechanical Limit Switch:

- Type: SPDT (Single Pole Double Throw) Micro Switch.
- Actuation Force: Low (Tactile click).
- Operating Logic: Active LOW (Configured with Internal Pull-up Resistor).
- Switching Capacity: 1A , 125V AC (Used at 5V DC logic level).



RF Transmitter & Receiver Pair (433 MHz)
- Frequency: 433.92 MHz (ISM Band)
- Modulation: ASK (Amplitude Shift Keying) / OOK (On-Off Keying).
- Transmission Range: 20–50 meters (Line of Sight).
- Data Rate: < 10 kbps (Suitable for low-speed control signals).
- Receiver Sensitivity: -105 dBm

Relay Module:
- Control Signal: 5V Active Low/High (TTL Logic).
- Load Rating: 10A ,250V AC / 10A, 30V DC.



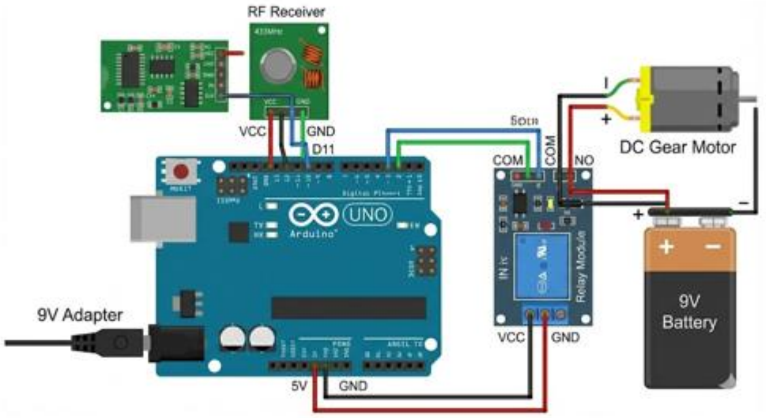
LM2596 Buck Converter:
- Input Voltage: 4.5V – 40V DC (Input from 9V Battery).
- Output Voltage: 1.25V – 35V DC (Adjusted to 5.0V for Nano).
- Max Output Current: 3A.

DC Gear Motor:

- Operating Voltage: 3V – 9V DC.
- Speed: 100-300 RPM.

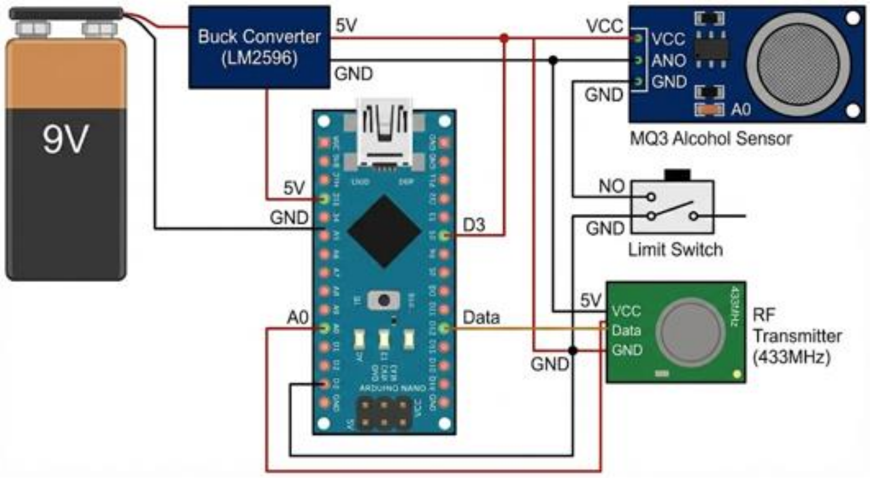
Software Specifications:
- Arduino IDE (Integrated Development Environment)

INTERFACING CIRCUIT



•The diagram shows the Arduino Uno connected to an RF Receiver (Pin D11) and a Relay Module. The Relay controls a high-current circuit consisting of a 9V battery and a DC Gear Motor.
• It shows the **Arduino Nano** on a breadboard connected to a **Blue Buck Converter (LM2596)**.
•The Buck Converter is crucial here because it steps down the high voltage from the 9V battery (connected via the black clip) to a safe 5V for the Nano.

Fig 1



Arduino Nano (Helmet): Selected for its small size to fit in the helmet. It operates at **5V** and uses a **Buck Converter** to step down the voltage from the **9V battery**.
Power System: The helmet diagram shows a **9V Battery** connected to an **LM2596 Buck Converter** to provide a stable 5V source.

BLOCK DIAGRAM

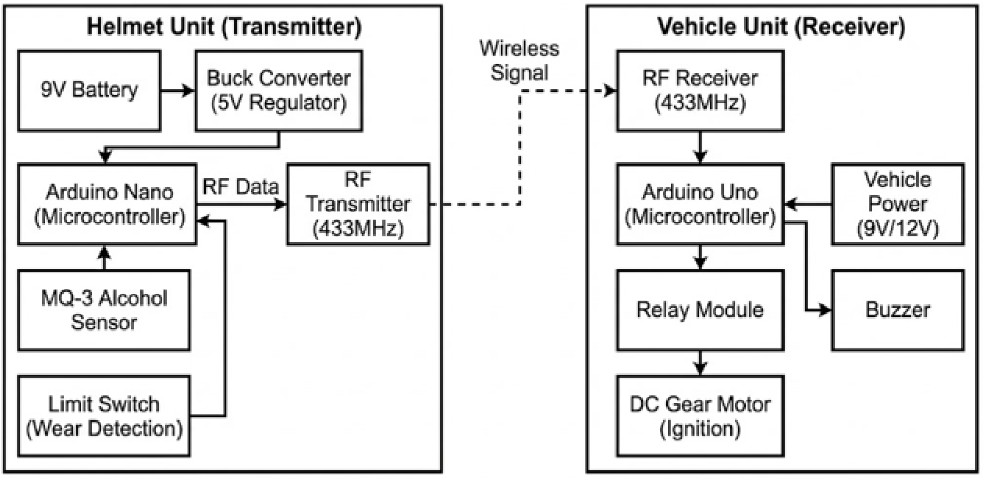


Fig 2: The figure shows a smart helmet system where alcohol detection and helmet wearing status are sent wirelessly to the bike unit. Based on the signal received, the bike either starts or remains locked for safety.

FLOW CHART

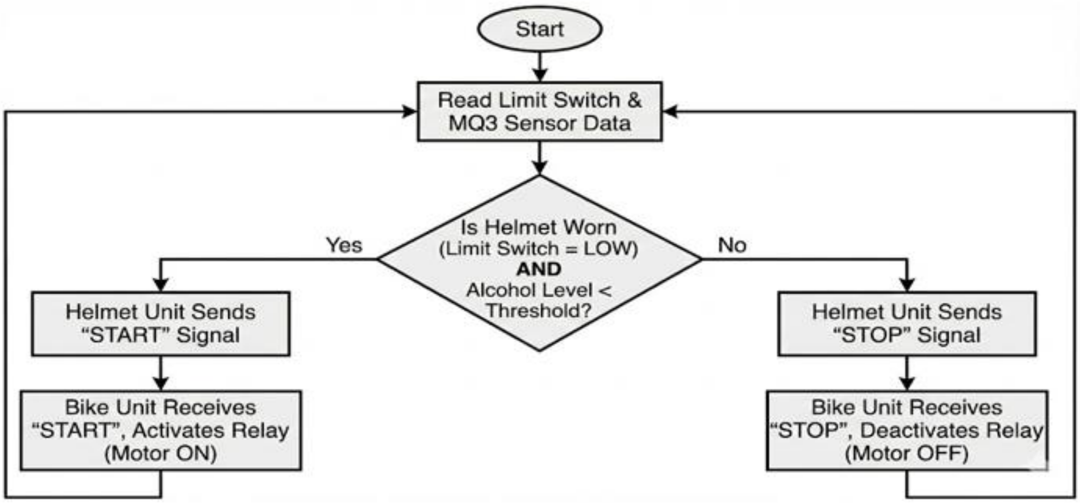
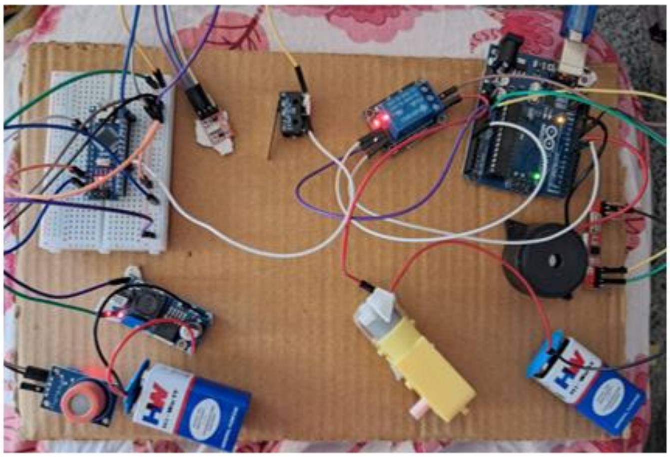


Fig 3: The figure shows decision steps where the system checks helmet wearing and alcohol level to determine bike access.

Result &Discussions

We tested this system under different conditions:

- When helmet is not worn- bike did not start
- When alcohol detected- bike did not start
- When both safety conditions satisfied- ignition started



ADVANTAGES & DISADVANTAGES

Advantages :

- Enhances rider safety by preventing drunk driving accidents .
- Ensures the rider wears the helmet properly using wear detection sensors .
- Reduces road violations by enforcing safety rules automatically .
- Provides ignition control, improving response and safety.

Disadvantages :

- System may give false readings if the environment contains alcohol fumes .
- Initial cost of installation and maintenance may be higher than normal helmets .
- Continuous sensor monitoring can drain battery faster .
- User may feel discomfort or inconvenience due to electronics and wiring inside the helmet.

CONCLUSION

The **Intelligent Smart Helmet with Alcohol Sensing and Wear Detection System** provides an effective solution for improving road safety by ensuring that the rider follows essential safety rules before starting a vehicle. By integrating an alcohol sensor and wear detection mechanism, the system prevents the motorcycle from starting if the rider is intoxicated or not wearing the helmet. This automation helps reduce drunk driving accidents and promotes responsible riding habits. Overall, the project demonstrates how embedded systems and safety technologies can work together to save lives, reduce road fatalities, and create a safer transportation environment.

References

[1] B. K. Roy and S. Banerjee, "Alcohol Detection and Vehicle Control System," International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 4, Issue 5, 2016.

[2] M. Prabhat and A. Singh, "Smart Helmet Using Alcohol Detection Sensor," International Journal of Innovative Science and Research Technology, Vol. 3, Issue 12, 2018.

[3] Arduino Official Documentation, "Arduino UNO Technical Specifications," available online: [4] MQ-3 Gas Sensor Datasheet, Hanwei Electronics Co., Ltd., 2017.

[5] R. Kumar, P. Singh, "Helmet-Based Safety System for Two-Wheeler Riders," International Journal of Engineering Research & Technology (IJERT), Vol. 7, Issue 04, 2019.

[6] S. Deshmukh and A. Kulkarni, "Wearable Technology for Road Safety," IEEE Sensors Journal, Vol. 21, Issue 9, 2021.

THANK YOU