SMART HELMET USING ESP-NOW PROTOCOL

A PROJECT REPORT

Submitted by

KAVIPRIYA B	**21ECR072**
MICHAEL RICHARD D	**21ECR095**
MUKESH P	**21ECR108**
NANDHAKISHORE K	**21ECR110**

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

VELALAR COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Chennai)

ERODE-638012

MAY 2024

VELALAR COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Chennai)

ERODE-638012

BONAFIDE CERTIFICATE

Certified that this project report "**SMART HELMET USING ESP-NOW PROTOCOL**" is the bonafide work of **"KAVIPRIYA B (21ECR072), MICHAEL RICHARD D (21ECR095), MUKESH P (21ECR108), NANDHAKISHORE K (21ECR110)"** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
Ms.C.Santhini M.E.,	Dr.M.Nisha Angeline M.E., PhD.,
SUPERVISOR	**HEAD OF THE DEPARTMENT**
Assistant Professor,	Professor,
Department of ECE,	Department of ECE,
Velalar College of Engineering	Velalar College of Engineering
and Technology,	and Technology,
Thindal, Erode-638012.	Thindal, Erode-638012.

Submitted for Semester Project viva-voce examination held on __________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We have a privilege to express our heartfelt thanks to our honorable Secretary and correspondent **Thiru. S. D. CHANDRASEKAR B.A.,** who provide all facilities to build this Project.

We express our deep sense of gratitude and sincere thanks to our Principal **Dr. M. JAYARAMAN M.E., Ph.D.,** for permitting us to undertake this Project.

We were indeed grateful to our Dean **Prof. P. JAYACHANDAR M.E.,** forgiving us the opportunity and continuous inspiration to carry out this Project.

We express our profound gratitude to our beloved Controller of Examination **Prof. Dr. K. R. VALLUVAN B.Tech., M.E., Ph.D.,** and Head of the Department **Prof. Dr. M. NISHA ANGELINE M.E., Ph.D.,** for their valuable guidance and immensely profited by their comments and reviews.

We express our sincere thanks to our project coordinator **Mr. R. Gopalan M.E., (Ph.D.,) Assistant Professor** and project supervisor **Ms. C. Santhini M.E., Assistant Professor** for their valuable guidance to do our Project successfully.

We express our sincere thanks to our parents for their support to do our project successfully.

ABSTRACT

Road accidents in our country are increasing day by day. Most of the accidents occur due to not wearing helmet, which can cause severe head injuries or even fatality of the rider. So it is necessary to make it mandatory to wear helmet while riding on a bike. In this project we have made a prototype of smart helmet. It has an Infrared sensor inside the helmet, which will detect whether the rider is wearing helmet or not. The bike will not start until the rider will wear the helmet. There are two modules on is mounted on helmet and another is mounted on vehicle.

The development of a Smart Helmet utilizing the ESP-NOW protocol, aimed at enhancing safety and connectivity for individuals engaged in activities such as biking, motorcycling. The Smart Helmet integrates various sensors to detect environmental conditions, monitor vital signs of the wearer, and assess potential hazards in real-time. Leveraging the ESP-NOW wireless communication protocol ensures low-power, high-speed, and robust connectivity between the helmet and a central monitoring system, such as a smartphone or a dedicated receiver. The Smart Helmet offers a comprehensive solution for promoting safety, situational awareness, and rapid response in dynamic environments, contributing to injury prevention and overall well-being.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	ARTIFICIAL INTELLIGENCE
GND	GROUND
GPS	GLOBAL POSITIONING SYSTEMS
HD	HIGH DEFINITION
IDE	INTEGRATED DEVELOPMENT ENVIRONMENT
IOT	INTERNET OF THINGS
IT	INFORMATION TECHNOLOGY
LED	LIGHT EMITTING DIODE
NODEMCU	NODE MICRO CONTROLLER UNIT
PC	PERSONAL COMPUTER
RFID	RADIO FREQUENCY IDENTIFICATION
USB	UNIVERSAL SERIAL BUS
VCC	VOLTAGE COMMON COLLECTOR
WIFI	WIRELESS FIDELITY

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION TO THE PROJECT

In recent years, the integration of technology into safety equipment has become increasingly prevalent, aiming to enhance both convenience and security for users. One such innovation is the Smart Helmet, a revolutionary concept that combines traditional head protection with advanced electronic features. This project focuses on developing a Smart Helmet equipped with a unique security mechanism: the ability to activate the associated vehicle only when the helmet is worn. Leveraging the ESP-NOW protocol, a robust and efficient wireless communication standard, this system ensures seamless connectivity between the helmet and the vehicle, enabling secure and reliable operation.

The primary objective of this project is to address the critical issue of unauthorized vehicle usage, a common concern for vehicle owners, particularly in environments where theft or unauthorized access is prevalent. By integrating the Smart Helmet with the vehicle's ignition system, we aim to create a seamless and secure solution that prevents the vehicle from starting unless the designated helmet is worn by the user. This not only deters theft but also promotes responsible usage by ensuring that only authorized individuals can operate the vehicle.

The Smart Helmet's functionality extends beyond mere vehicle activation Moreover, the ESP-NOW protocol offers low-latency communication, ensuring rapid response times and reliable connectivity even in dynamic environments.

This project represents a significant advancement in the realm of wearable technology and vehicle security, offering a comprehensive solution that prioritizes both safety and convenience for users. By harnessing the power of the ESP-NOW protocol, we aim to develop a Smart Helmet that not only revolutionizes the way we think about head protection but also sets a new standard for integrated safety systems in the automotive industry.

1.2 INTERNET OF THINGS

IoT stands for Internet of Things. It depicts the network of connections between physical items, such as cars and appliances, that have sensors, software, and connectivity built in to enable data exchange and connection. This technology makes it possible to collect and share data from a variety of devices, which paves the way for the creation of automated and more efficient systems.

A network of interconnected items, such as computers, mechanical and digital equipment, goods, animals, and individuals with unique IDs, is known as the Internet of Things (IOT). and the ability to transmit the information over a network that facilitates human-to-human or human-to-computer communication.

The Internet of Things (IoT) is a network of physically connected objects, such as vehicles, appliances, and other things that have sensors, software, and network connectivity built right into them. These devices may collect and exchange data, creating a platform that enables everyday things to interact and communicate with one another. The ultimate goal of the Internet of Things is to enable these devices to gather and communicate data in order to improve decision-making, automate processes, and increase productivity.

1.3 COMPONENTS OF IOT

1.3.1 Devices and Sensors

Actual objects or devices have sensors built in to collect data. Examples include smart thermostats, NodeMCU, Temperature sensors, Heart rate sensors, industrial equipment with built-in sensors, and wearable fitness trackers.

1.3.2 Connectivity

The ability of a gadget to connect to the internet or share data with other devices. This can be accomplished with the aid of wired and wireless communication technologies such as Bluetooth, Wi-Fi, and cellular networks.

1.3.3 Data Processing

The collected data is analyzed and processed, either locally on the device or on the cloud. This stage is critical in order to extract meaningful insights and make informed judgments.

1.3.4 Cloud Computing

Many IoT apps employ cloud systems to store and process data. Cloud computing provides scalable and reliable infrastructure for handling enormous data volumes generated by Internet of Things devices.

1.3.5 Actuators

Components that enable devices to act in response to processed data. The ESP2866 module wait for the signal for connection the both module by using ESP-NOW protocol.

1.3.6 User Interface

These interfaces enable users to control and operate Internet of Things gadgets. This can include mobile apps and online interfaces.

1.4 APPLICATION OF IOT

1.4.1 Smart Home

Home automation: Control and monitor your heating, ventilation, lighting, security systems, and appliances from a distance. Smart Appliances: Washing machines, refrigerators, and other appliances with Internet of Things capabilities can communicate with one another and be controlled from a single location.

1.4.2 Industrial IOT

By utilizing information obtained from sensors on machines to predict when maintenance is necessary, predictive maintenance reduces downtime and optimizes productivity. Supply chain optimization: IoT makes it easier to track and manage shipments, inventories, and logistics.

1.4.3 Retail

Inventory control: By employing RFID tags and sensors to track inventory levels, stockouts can be prevented. Personalized Marketing: With the use of IoT data, it is possible to enhance the shopping experience and offer tailored promotions.

1.4.4 Healthcare

Remote Patient Monitoring: Internet of Things (IoT) devices can assist healthcare providers in providing prompt, remote patient care by collecting and transmitting health data. Wearable Technology: Fitness trackers and smartwatches monitor health metrics and provide users with information.

1.4.5 Smart Cities

Drivers are guided to available parking spaces by intelligent parking systems. stations that monitor air quality to reduce pollution. Intelligent street lighting that modifies brightness in response to weather and traffic patterns.

1.4.6 Agriculture

Autonomous vehicles and drones for managing and monitoring crops. Sensors in the soil that give current information on the amounts of nutrients and moisture. Using precision farming methods, fertilizer and pesticides are applied precisely where needed.

1.4.7 Logistics and Supply Chain

Monitoring the cold chain while transporting perishable item. RFID and GPS tracking to view shipments in real time. Integration of blockchain technology for transparent and safe supply chain management.

1.4.8 Connected Vehicles

Communication between vehicles (V2V) to increase road safety. Autonomous cars that avoid collisions by using Internet of Things sensors for navigation. Fleet management systems for scheduled maintenance and real-time tracking.

1.4.9 Energy Management

Smart meters give users the ability to track and modify how much energy they use, technologies for adjusting demand that maximize energy use during peak hours. Integration of renewable energy sources to generate power in an efficient manner.

1.5 ADVANTAGES OF IOT

- **Improved Efficiency:** Increased operational efficiency results from automating repetitive jobs and processes, and quick decision-making and response are made possible by real-time monitoring and data collection.
- **Cost saving:** Condition-based monitoring and predictive maintenance save downtime and maintenance expenses, while energy management systems optimize resource utilization and save money.

- **Data Collection and Analysis:** Real-time monitoring and data analytics provide better decisions and insights.
- **Enhanced Productivity:** Workflows are streamlined by automation and intelligent systems, increasing total productivity. Having real-time data at hand facilitates speedier problem solving and decision making.
- **Improved the quality of life:** More comfort, convenience, and energy savings are available in smart homes. IoT devices for healthcare help provide remote and individualized patient care.
- **Good Decision Making:** Making educated decisions is made possible by having access to real-time data. Organizations can proactively foresee and resolve problems with the aid of predictive analytics.
- **Benefits For Environment:** IoT technology aid in environmental protection and monitoring. Overall energy usage is decreased through smart energy management.
- **Innovation And New Business Models**: IoT fosters originality and inventiveness by supporting the development of fresh products and business strategies.
- **Remote Monitoring and Controlling:** Accessibility and convenience are increased by remote device monitoring and control. There's less need for in-person presence when troubleshooting and maintenance are done remotely.

CHAPTER 2
LITERATURE SURVEY

2.1 INTRODUCTION

Pranav Pathak,et al. “IoT based Smart Helmet with Motorbike Unit for Enhanced Safety” 2020 Institute of Electrical and Electronics Engineers ACCESS (IEEE ACCESS).India is a densely populated nation with a huge amount of traffic on the road so a two-wheeler proves to be one of the most effective modes of road transport. But, two-wheelers are also one of the most unsafe means of road transport. A helmet is a must for any rider on the road but still many people avoid it. This leads to fatal injuries in case of accidents. Over speeding and rash driving are not only dangerous for the rider but also for other people in the vicinity. Many drivers are unaware of vehicles approaching from behind due to several reasons and this leads to accidents. Also, many young riders drive carelessly jeopardizing their as well as others’ life. 1.35 million people died in road accidents according to The Global Status report on Road Safety 2018[1] report.

Developing nations accounted for 90% of these casualties. As per the Ministry of Road Transport & Highways published report ROAD ACCIDENTS IN INDIA – 2018[2] two-wheeler accidents are 35.2% of total accidents. This makes two wheelers the most unsafe means of road transport. Also, the number of accidents in 2018 increased by 4.2% compared to 2017. In 2108, 47,560 people lost their lives in two-wheeler accidents. In 2017 44,092 people were killed by two-wheeler accidents, this shows that the number of casualties increased by 7.9%. People in the age group of 18-60 constituted 84.7% of the total death. This age group is also referred to as the work ingage. group. In 2017, 35,975 due to non-wearing of a helmet. In 2018, the number rose to 43,614. 97,588 people died in 2018 due to over speeding. In 2018, 4,188 people died due to drunk and driving accidents and 25,801 people died due to hit from the back collision. This shows how fatal road accidents and particularly two-wheeler accidents are.

Despite the constant efforts from the authorities, two-wheeler accidents are increasing day by day. The main aim of the project is the reduction of accidents by making rider aware of any risk that can lead to an accident. The helmet unit (HU) and motorbike unit (MU) have many sensors that ensure the safety of the rider. These units not only ensure the safety of the rider but also inform concerned authorities in case of mishaps. Due to IoT concerned authorities can monitor the activity of bikes in real-time and can take precautionary measures. The alert system comprised of GPS and GSM module is quick and informs the concerned authorities in case of a mishap. presence and directions during night rides.

2.2 EXISTING SYSTEM

In the existing system, we can start the bike by the using as a key and we can stop the bike when the rider doesn’t wear the helmet. Even though we can’t lock the ignition when the rider is drinking mode or during an accident, but we can locate the bike. The helmet is the protective gear to prevent and avoid head injuries in an environment and saves the life of bike riders. This existing idea addresses three main objectives are To ensure the safety of the rider,To the rising modern needs of technological innovations, and To overcome the accident rate. The smart helmet allows vehicle ignition only when the helmet is worn only. An RF Module can be used as a wireless link for communication between transmitter and receiver. The RF module has a frequency that is easy being hacked or interference by other network signals. If the rider is drunk the ignition gets locked and sends a message to the previously saved number with his location. In case of an accident, it will send a message through GSM along with location with the help of the GPS module, but can’t find the exact accident spot as accurate. The message sending is not quick and efficient in the existing system.

2.3 ADVANTAGES

1. Ignition control

The smart helmet features ignition control, allowing the rider to conveniently turn the motorcycle's ignition on and off directly from the helmet.

2.Alcohol Detection

It incorporates an MQ-3 sensor to detect whether the rider has consumed alcohol, promoting safety by preventing intoxicated individuals from riding.

3.LED Strips for Visibility

LED strips are integrated into the helmet, enhancing visibility for both the rider and other motorists, particularly in low-light conditions such as nighttime riding. These LED strips serve as indicators of the rider's presence, improving overall safety on the road..

4. Fall Detection

Utilizing a vibrating sensor, the helmet detects falls or accidents. This feature alerts emergency services or designated contacts in the event of a crash, enabling prompt assistance to the rider.

5. Accident Location

In the unfortunate event of an accident, the smart helmet aids in locating both the rider and the motorcycle. This capability can be crucial for emergency responders, ensuring swift assistance and potentially saving lives.

2.4 APPLICATIONS

- The helmet employs IoT sensors to monitor parameters like speed, location, and environmental conditions in real-time, enhancing safety.
- IoT-enabled crash sensors detect accidents and promptly notify emergency contacts or authorities, enabling swift assistance.
- Features like alcohol detection, GPS tracking, and visibility enhancements improve rider safety and provide crucial assistance during emergencies.

CHAPTER 3
PROPOSED SYSTEM

In the previous section, an extensive literature review is conducted to identify the limitations of the existing systems. The proposed smart helmet utilizes the ESP Now protocol for low-power, peer-to-peer communication between the helmet and other IoT devices, ensuring efficient data transmission and minimal latency. It provides comprehensive safety measures for riders.

3.1 OBJECTIVE

The objective of the proposed project is to develop a Smart Helmet system leveraging the ESP-8266 module and ESP Now protocol for enhanced safety and convenience for bike riders. This system integrates components including a relay module, battery, helmet, ignition set, and momentary switch. The primary goals include real-time monitoring of rider behavior and environmental conditions, automatic accident detection and improved visibility through LED indicators. By utilizing ESP-8266 and ESP Now protocol, the system ensures seamless communication between the helmet and other IoT devices with minimal latency. Additionally, the project aims to provide immediate alerts and assistance in case of emergencies, thus promoting rider safety and overall riding experience. Through the implementation of over-the-air updates and customization options, the system will be scalable, adaptable, and continuously improved to meet evolving user needs.

3.2 WORKING PRINCIPLE

The helmet incorporates the sensors to monitor wheather rider is wearing helmet or not complete Its hardware setup is shown below.

Sensor data is collected and processed within the helmet using the ESP-8266 module. Algorithms analyze this data to detect anomalies, such as sudden impacts or alcohol levels above a threshold. The processed data is transmitted wirelessly to other IoT devices or a central monitoring system using the ESP Now protocol. This protocol ensures low-latency, peer-to-peer communication between the helmet and designated receivers Instead of ESP Now, the NodeMCU communicates with other IoT devices or a central monitoring system using Wi-Fi connectivity. It can send data packets containing sensor information to a designated server or receiver over a Wi-Fi network The system may incorporate a momentary switch or other user interface elements for manual control or interaction. For example, the rider may use the switch to acknowledge alerts or initiate specific actions The system is powered by a battery, and efficient power management techniques are employed to ensure prolonged operation without frequent recharging or replacement The NodeMCU continuously monitors sensor data and communicates with other devices over Wi-Fi. Over-the-air updates enable continuous improvement of functionalities and adaptation to changing user requirements, similar to the ESP-8266 setup.

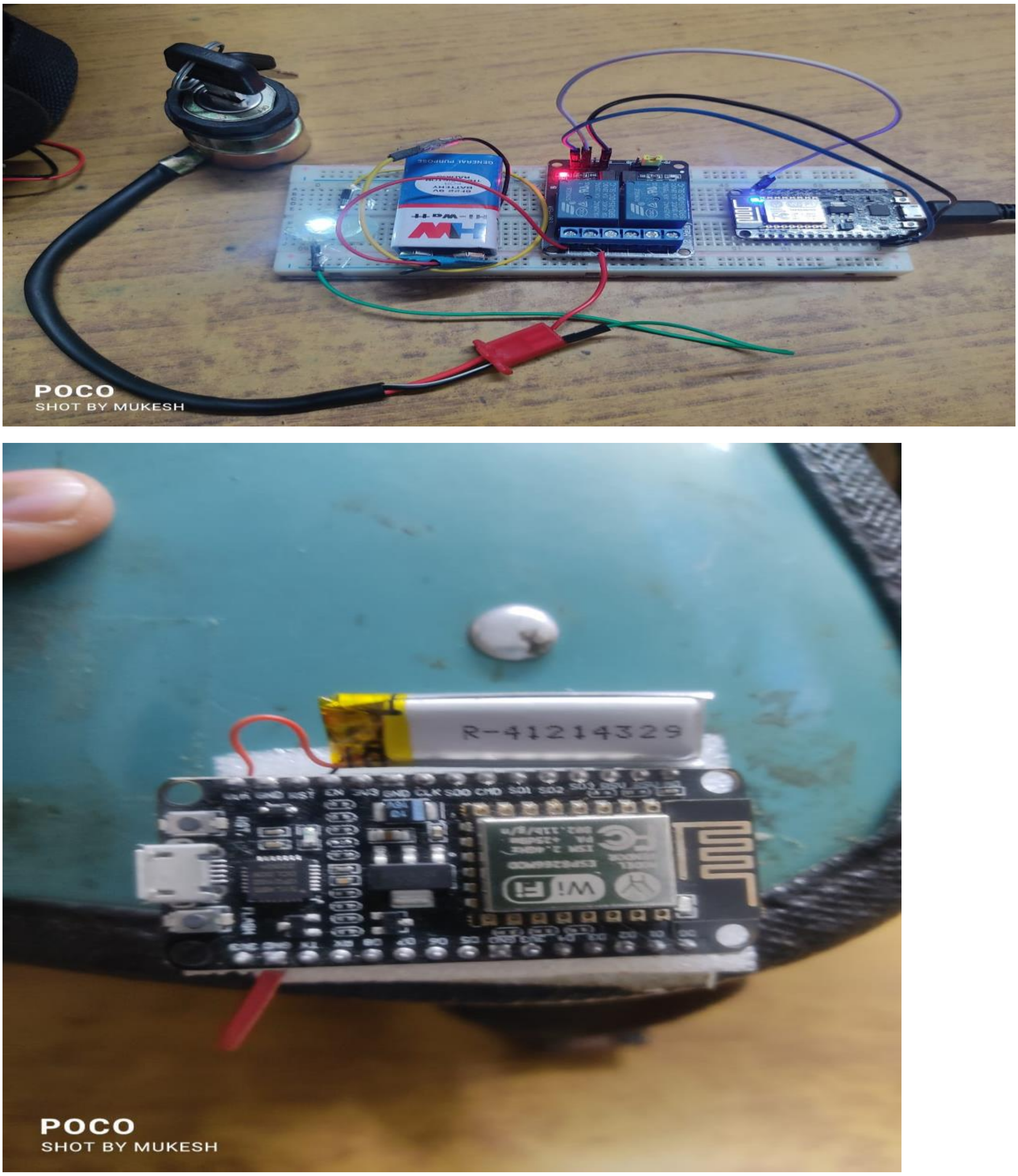

Figure 3.1 System design and connection of the hardware components

The momentary switch on the transmitter unit (helmet) is wired to send signals to the ESP8266/ESP32 module.The ESP8266/ESP32 module on the receiver unit (bike) communicates with the transmitter (helmet) over Wi-Fi.When the ESP8266/ESP32 module on the receiver unit receives a signal from the transmitter (helmet), indicating a specific action (e.g., enabling or disabling ignition), it triggers the relay module accordingly.The relay module controls the ignition set of the bike based on signals received from the ESP8266/ESP32 module, effectively managing the bike's ignition system.This block diagram illustrates the basic setup of the Smart Helmet system, where the transmitter unit (located in the helmet) sends signals to the receiver unit (installed on the bike), allowing the rider to control the bike's ignition system remotely.

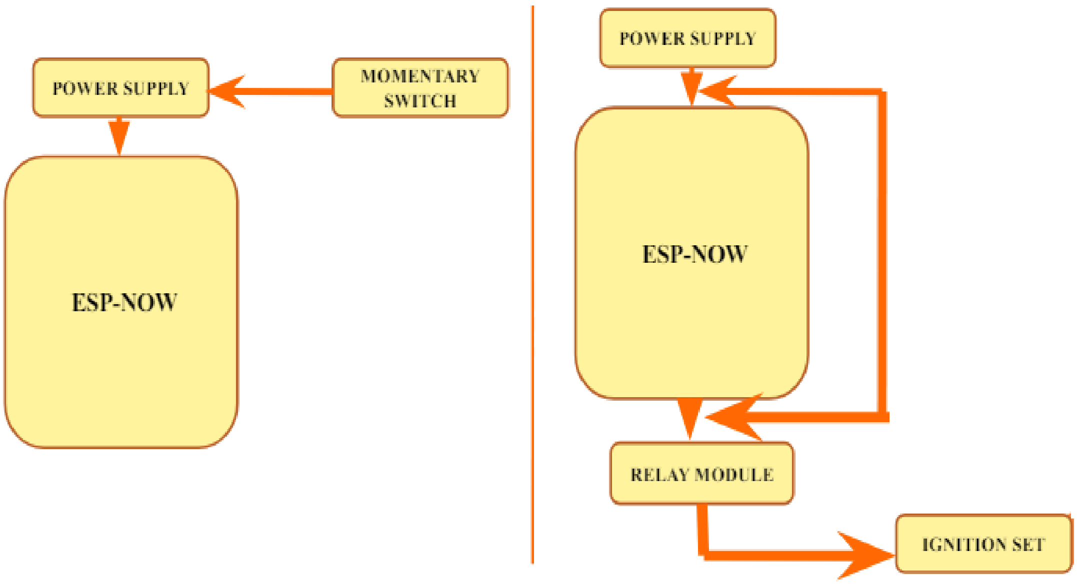

Figure3.2 Block Diagram of Smart helmet using ESP-NOW Protocol (Transmitter-helmet& Receiver-bike)

In a system utilizing the ESP-NOW protocol, a NodeMCU transmitter module communicates with specific NodeMCU receiver modules through their unique MAC addresses. Here's how the process generally works:

- Initialization: Both the transmitter and receiver modules need to be initialized and configured to communicate with each other using the ESP-NOW protocol. This typically involves setting up the necessary Wi-Fi parameters and establishing the ESP-NOW communication protocol.

- Pairing: During the initialization phase, the transmitter module needs to know the MAC addresses of the receiver modules it intends to communicate with. These MAC addresses serve as unique identifiers for each NodeMCU device. The transmitter module is programmed with the MAC addresses of the intended receiver modules.

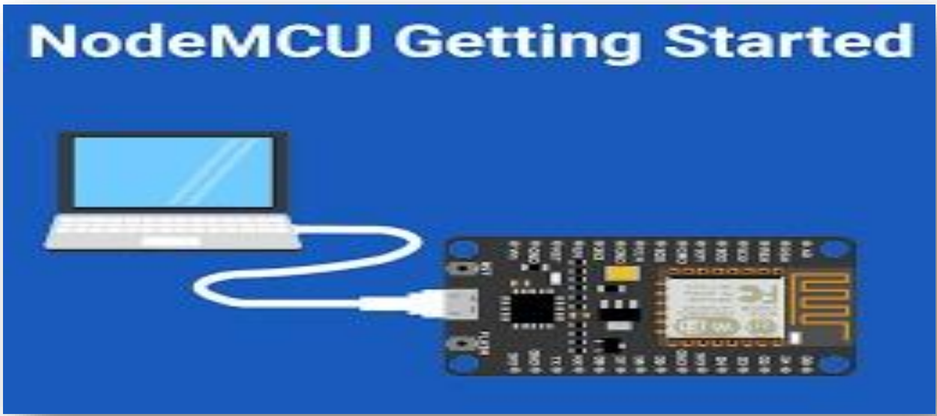

Figure 3.3 Wi-Fi module communication setup

- Addressing: When the transmitter module wants to communicate with a specific receiver module, it includes the MAC address of that receiver module in the data packet it sends out. This ensures that the data packet is only received by the intended recipient.

Figure 3.4 Wi-Fi module specific MAC address

- Transmission: The transmitter module sends out data packets containing the MAC address of the intended receiver module along with the actual data to be transmitted. The ESP-NOW protocol handles the transmission of these packets over the Wi-Fi network.

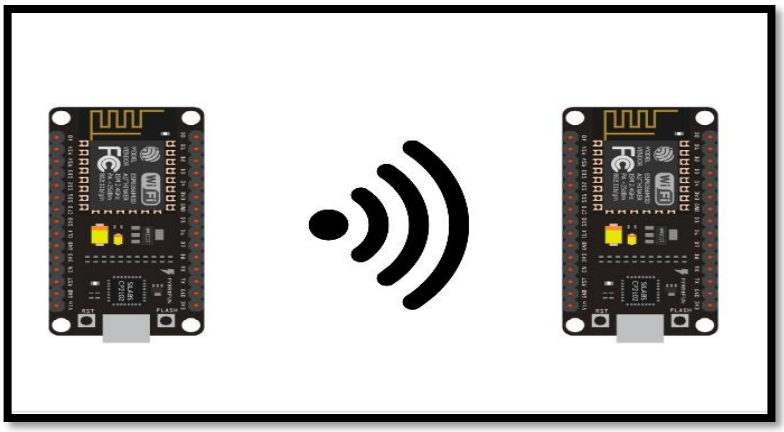

Figure 3.5 Message transmission from one module to another viaWi-Fi communication

- Reception: On the receiving end, the NodeMCU receiver modules are constantly listening for incoming data packets. When a receiver module receives a packet, it checks the MAC address included in the packet to determine if it's addressed to itself. If the MAC address matches its own, it processes the data contained in the packet.

- Response (optional): The receiver module can optionally send a response back to the transmitter module to acknowledge receipt of the data or to provide additional information. This response can also be addressed using the MAC address of the transmitter module.

By utilizing MAC addresses as unique identifiers and including them in the data packets, the transmitter module can effectively target specific receiver modules in the ESP-NOW network, ensuring that the data is delivered only to the intended recipients.

3.3 CONNECTION OF CONTROLLERS WITH THE SENSORS

The transmitter code starts with including necessary libraries, such as ESP8266WiFi.h and espnow.h. It defines a broadcast address (broadcastAddress) which is the MAC address of the receiver device.The OnDataSent function is defined, which will be called when data is successfully sent to the receive.In the setup function,these process are followed,

- Serial communication is initialized.

- Pins are configured for controlling a relay and led.
- WIFI mode is set to station mode.
- ESP-NOW is initialized and configured with the transmitter role.
- A callback function(OnDataSent) is registered to handle data transmission events.
- The receiver device is added as a peer with its MAC address and role.

In the loop function these process are followed,

- It sends data (integer value 1) to the receiver device using esp_now_send
- Data is send every 2 seconds

Similar to the transmitter, the receiver code includes necessary libraries and defines the OnDataRecv function to handle received data. In the setup function these process are followed,

- Serial communication is initialized.
- WiFi mode is set to station mode.
- ESP_NOW is initialized an configured with the receiver role.
- A callback function (OnDataRecv) is registered to handle received data.

In the loop function,

- There's no specific action in the loop because the receiver mainly listens for incoming data.

Their connection are shown as below:The transmitter and receiver are configured with the ESP-NOW role:controller and Slave,respectively. They use the same WiFi network. The transmitter sends data to the Receiver using its MAC address. The receiver listens for incoming data and process it using the registered callback function(OnDataRecv).This setup enables communication between two ESP8266 devices using ESP-NOW protocol, where one devices acts as the transmitter sending data, and the other device acts as the receiver receiving and processing the data.

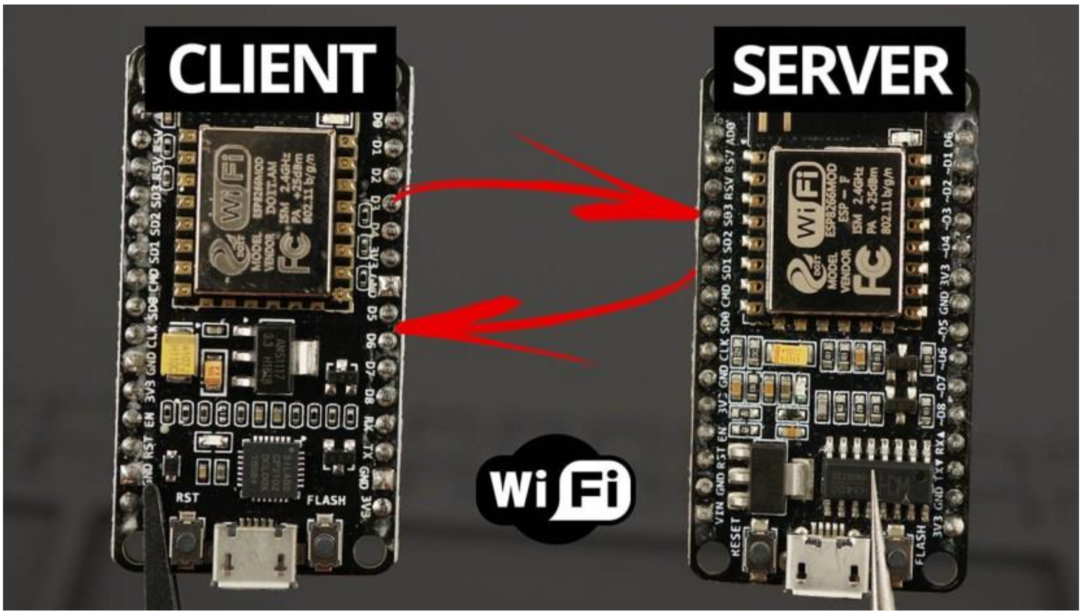

Figure 3.6 Wi-Fi communication (HTTP) between two ESP8266 NodeMCU boards

Setting up a client-server Wi-Fi communication between two NodeMCU boards (ESP8266) involves configuring one board as the server and the other as the client. Here's a general guide on how to achieve this, To Setting Up the Server NodeMCU Connect one NodeMCU board to your computer via USB then Open the Arduino IDE and ensure that you have the ESP8266 board package installed. Next Replace `"Your WiFi Network"` and `"Your WiFi Password"` with your Wi-Fi network credentials. Fianlly upload this transmitter code to one of the NodeMCU boards. To Setting Up the Client NodeMCU,Connect the other NodeMCU board to your computer via USB.Open the Arduino IDE and ensure that you have the ESP8266 board package installed. Replace `"YourWiFiNetwork"` and "YourWiFiPassword"` with your Wi-Fi network credentials.Replace `"ServerIPAddress"` with the IP address of the NodeMCU board acting as the server (you can find this in the Serial Monitor after uploading the server code).Upload this code to the other client NodeMCU board. Open the Serial Monitor for both NodeMCU boards to view the debug messages. This setup establishes a basic HTTP client-server communication between the two NodeMCU boards over Wi-Fi. You can extend this example to send more complex data or perform other actions based on the received data.

3.3.1 Transmitter (Helmet - Rider's Control Unit):

The transmitter unit, located within the helmet, features a momentary switch that the rider can press to trigger specific actions or send signals.This provides the necessary power to operate the transmitter unit, ensuring its functionality during rides.

3.3.2 Receiver (Bike)

The receiver unit, installed on the bike, communicates with the transmitter (helmet) over Wi-Fi, receives signals, and processes them.Integrated into the bike's system, the relay module is responsible for controlling the bike's ignition system. It receives signals from the transmitter and acts accordingly to enable or disable the ignition. The receiver unit on the bike also requires power to operate, typically provided by the bike's battery or another power source. Connected to the relay module, the ignition set of the bike is the component responsible for starting and stopping the engine. It responds to signals from the transmitter, either allowing or preventing the flow of current to the ignition system.

CHAPTER 4
SYSTEM DESCRIPTION

4.1 HARDWARE REQUIREMENTS

- ESP8266 MODULE
- RELAY
- MOMENTARY SWITCH
- IGNITION SET
- LED
- POWER SUPPLY
- BREADBOARD
- CONNECTING WIRES
- USB CABLE

4.2 SOFTWARE REQUIREMENTS

- Environment : Arduino IDE
- Operating System : Windows 11
- Programming Language : Arduino Sketch (variant of C++)

4.3 HARDWARE DESCRIPTION

4.3.1 ESP8266 Microcontroller

The ESP8266 is a versatile and widely used Wi-Fi microcontroller module developed by Espressif Systems. The hardware description of the ESP8266 module are given below

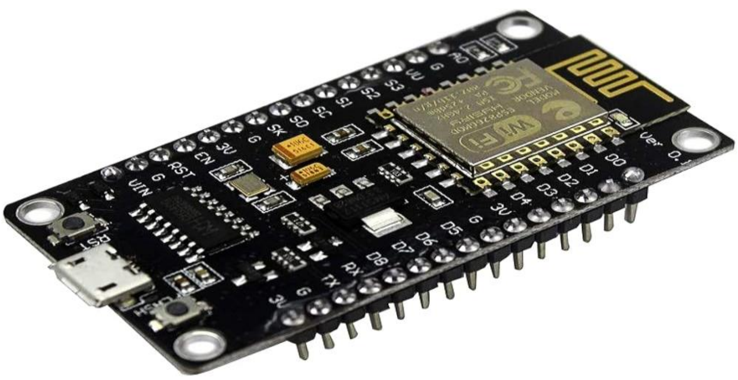

Figure 4.1 ESP8266 Microcontroller

1. Microcontroller Core: The ESP8266 module typically features a Tensilica Xtensa LX106 32-bit microcontroller core running at frequencies up to 160 MHz. This microcontroller core handles the program execution and interacts with peripherals and external components.

2. Wi-Fi Connectivity: One of the key features of the ESP8266 is its integrated Wi-Fi connectivity, which allows it to connect to Wi-Fi networks and communicate over the internet. It supports the 802.11 b/g/n standards for wireless communication.

3. Flash Memory: The ESP8266 module includes built-in Flash memory for storing firmware, program code, and data. The amount of Flash memory varies between different versions of the ESP8266 module, typically ranging from 512 KB to 4 MB.

4. RAM (Random Access Memory): The ESP8266 also contains RAM for temporary data storage during program execution. The amount of RAM available on the module can vary, typically ranging from 32 KB to 128 KB.

5. GPIO (General Purpose Input/Output) Pins: The ESP8266 module features multiple GPIO pins that can be configured as digital inputs or outputs. These pins can be used to interface with external sensors, actuators, displays, and other peripheral devices.

6. Analog Input Pins: Some variants of the ESP8266 module include analog input pins, allowing it to read analog sensor values using its built-in ADC (Analog-to-Digital Converter).

7. Interfaces: The ESP8266 module includes interfaces such as SPI (Serial Peripheral Interface), I2C (Inter-Integrated Circuit), UART (Universal Asynchronous Receiver-Transmitter), and PWM (Pulse Width Modulation). These interfaces enable communication with external devices and peripherals.

8. Power Supply: The ESP8266 module typically operates at 3.3 volts (Vcc). It may include onboard voltage regulators to accept a wider range of input voltages.

9. Form Factor: The ESP8266 is available in various form factors, including surface-mount modules, modules with pins for breadboarding, and integrated development boards (e.g., NodeMCU, Wemos D1 mini) that include USB-to-serial converters for easy programming.

10. Programming: The ESP8266 can be programmed using the Arduino IDE, PlatformIO, or other development environments. It supports the C and C++ programming languages, with a rich set of libraries available for various functionalities.

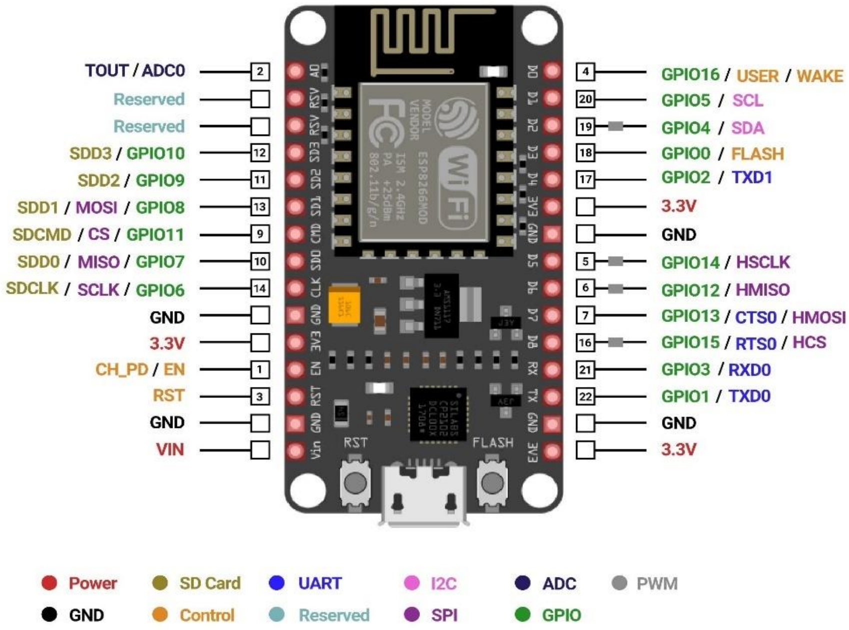

Figure 4.2 Pin Diagram of ESP8266 Microcontroller

4.3.2 RELAY

A relay is an electromechanical device used to control the flow of electricity in a circuit by using a smaller electrical signal to control a larger one. Here's a breakdown of its components and how it working.

Figure 4.3 Relay

1. Coil: The coil is the primary input of the relay. It consists of wire wound around a core, typically made of ferromagnetic material. When current flows through the coil, it generates a magnetic field.

2. Armature: The armature is a movable component within the relay that is attracted to the magnetic field produced by the coil. It is usually connected to one or more contacts.

3. Contacts: The contacts are the switch-like elements of the relay. They are typically made of conductive materials such as silver or copper. When the relay is activated (i.e., when current flows through the coil), the armature moves and either makes or breaks electrical connections between the contacts, depending on the relay's configuration.

- **Normally Open (NO) Contacts:** These contacts are open when the relay is not energized. When the relay is energized, the contacts close, allowing current to flow through the circuit.
- **Normally Closed (NC) Contacts:** These contacts are closed when the relay is not energized. When the relay is energized, the contacts open, interrupting the current flow.

4. Frame: The frame provides mechanical support and housing for the coil, armature, and contacts. It often includes mounting points for installation and may have terminals for connecting wires.

WORKING OF RELAY:

When a voltage is applied to the coil of the relay, it creates a magnetic field that attracts the armature. The movement of the armature then causes the contacts to change position, completing or breaking the circuit. This allows the relay to control the flow of electricity in another part of the circuit without directly coming into contact with it.

Relays are commonly used in various applications where electrical isolation, high-power switching, or control from a low-power signal is required. They are found in industrial control systems, automotive electronics, home automation, and many other fields. In the context of the provided code, the relay is likely being used to control the power supply to a device or component based on the status of the ESP8266 microcontroller**.**

4.3.3 MOMENTARY SWITCH

A momentary switch, also known as a momentary push button or tactile switch, is a type of electrical switch that is only activated while it is being pressed. Here's a breakdown of its hardware description:

Figure 4.4 Momentary Switch

1. Button: The most visible part of a momentary switch is the button itself. This is the part that a user presses to activate the switch. The button is typically made of plastic or metal and may have a variety of shapes, sizes, and colors depending on the specific application and design requirements.

2. Actuator: The actuator is the part of the switch that physically moves when the button is pressed. It is usually connected to a spring mechanism that provides tactile feedback to the user, giving a click or tactile sensation when the switch is activated.

3. Contacts: Inside the switch housing, there are electrical contacts that make or break a circuit when the switch is activated. These contacts are usually made of metal and are designed to provide a reliable electrical connection. In a momentary switch, the contacts return to their original position when the button is released, breaking the circuit.

4. Terminals: The terminals are the metal pins or leads that extend from the bottom of the switch housing. These terminals are used to connect the switch to an electrical circuit, typically by soldering wires or attaching connectors.

5. Housing: The housing is the outer casing or enclosure that contains all the internal components of the switch. It is usually made of plastic, metal, or a combination of both materials and provides protection and mechanical support for the switch.

6. Mounting Mechanism: Momentary switches may include a mounting mechanism, such as screw holes or snap-in tabs, that allows them to be securely attached to a panel or enclosure.

7. Rating: Momentary switches come in various sizes and configurations to accommodate different voltage and current ratings. It's essential to select a switch with the appropriate rating for the intended application to ensure safe and reliable operation.

4.3.4 LED

An LED (Light Emitting Diode) is a semiconductor device that emits light when current flows through it. LEDs are highly efficient, durable, and versatile light sources used in a wide range of applications, including displays, indicators, signage, automotive lighting, general illumination, and more. Their compact size, low power consumption, and long lifespan make them a popular choice for various lighting needs. Here's a breakdown of its hardware description:

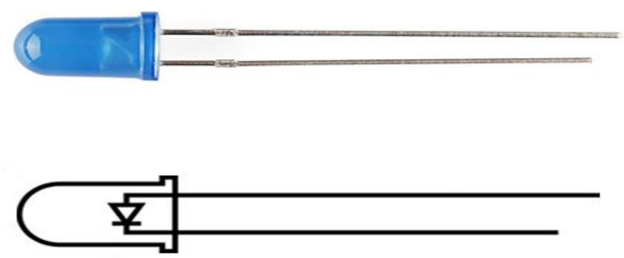

Figure 4.5 LED

1. Semiconductor Chip: At the heart of an LED is a semiconductor chip, typically made of gallium arsenide (GaAs) or gallium phosphide (GaP). The chip is doped with impurities to create a p-n junction, similar to a diode.

2. Anode (+): The anode is the positive terminal of the LED. It is connected to the p-type semiconductor region of the chip.

3. Cathode (-): The cathode is the negative terminal of the LED. It is connected to the n-type semiconductor region of the chip.

4. Encapsulation: The semiconductor chip is encapsulated in a transparent or translucent material, such as epoxy resin or silicone, to protect it from environmental factors and to provide mechanical support. This encapsulation material also acts as a lens, focusing and diffusing the light emitted by the chip.

5. Lead Wires: The anode and cathode of the LED are connected to external electrical circuits via lead wires. These wires are typically made of metal and are attached to the semiconductor chip during the manufacturing process.

6. Package: The encapsulated LED chip, along with its lead wires, is housed in a package that provides additional protection and facilitates handling and mounting. LED packages come in various shapes and sizes, such as through-hole, surface-mount, and chip-on-board (COB) packages.

7. Color: The color of light emitted by an LED depends on the semiconductor materials used in the chip and the manufacturing process. LEDs are available in a wide range of colors, including red, green, blue, yellow, amber, white, and various shades in between.

8. Forward Voltage (Vf): The forward voltage is the voltage required to turn on the LED and make it emit light. It varies depending on the LED's color and semiconductor materials.

9. Forward Current (If): The forward current is the current flowing through the LED when it is illuminated. Exceeding the maximum forward current rating can damage the LED.

10. Brightness (Luminous Intensity): The brightness of an LED is measured in lumens or millicandela (mcd) and indicates the amount of light emitted by the LED. Brightness can be controlled by varying the forward current or by using pulse-width modulation (PWM) techniques.

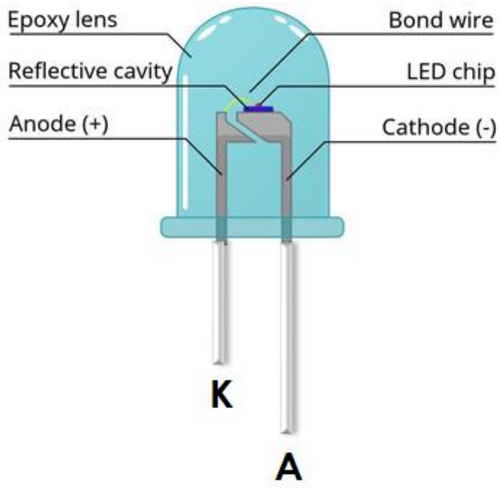

Figure 4.6 PIN OF LED

4.3.5 IGNITION SYSTEM

Bike key set, also known as the ignition system, is a crucial component of a motorcycle's hardware. the bike key set (ignition system) is a combination of mechanical and electrical components that work together to start and operate the motorcycle. It provides security and control over the vehicle's electrical systems, allowing the rider to start the engine and operate various functions safely and efficiently. Here's a detailed description of its components:

1.Key Switch: The key switch is the primary component of the ignition system. It's a mechanical switch that allows the rider to turn the ignition on and off using a key. When the key is turned, it completes or interrupts the electrical circuit, allowing or preventing the flow of electricity to the bike's various systems.

2. Ignition Lock Cylinder: This is the part of the key set where the key is inserted. It contains tumblers and pins that match the specific key's pattern. When the correct key is inserted and turned, it aligns the tumblers and allows the ignition switch to turn.

3. Wiring Harness: The wiring harness connects the key switch to the motorcycle's electrical system. It consists of wires, connectors, and sometimes fuses or relays. The wiring harness carries electrical signals from the key switch to the various components of the motorcycle, such as the ignition coil, fuel pump, and starter motor.

4. Ignition Coil: The ignition coil is responsible for converting low-voltage electricity from the battery into high-voltage electricity needed to create a spark at the spark plugs. It is activated when the key switch is turned on, sending a signal to the coil to begin the ignition process.

5. Starter Solenoid: In motorcycles with electric starters, the starter solenoid is part of the ignition system. It is a relay switch that connects the battery to the starter motor when the key switch is turned to the start position. This engages the starter motor, which turns the engine over to start it.

6. Kill Switch: Some motorcycles have a kill switch as part of the ignition system. This is a secondary switch that allows the rider to quickly shut off the engine in case of an emergency or to prevent theft. It typically interrupts the flow of electricity to the ignition system when activated.

7. Handlebar Lock: In addition to the key switch, some motorcycles have a handlebar lock as part of the ignition system. This is a mechanical lock that prevents the handlebars from being turned, making it more difficult for thieves to steal the bike.

Figure 4.7 Ignition System

4.3.6 Power Supply

The power supply for an ESP8266 board converts input voltage to the required level (typically 3.3 volts) using a voltage regulator. It includes input and output capacitors to filter out noise and ensure stable power delivery. Some models feature input protection components like diodes or fuses. Output connectors enable easy connection to the ESP8266 board. The power supply's current rating should meet the demands of the board and peripherals.

Figure 4.8 Power Supply

4.3.7 BreadBoard

A breadboard is a versatile prototyping tool used in electronics to create temporary circuits without soldering. It consists of a plastic board with numerous holes arranged in a grid pattern. These holes are interconnected by metal strips underneath the surface, allowing components to be easily inserted and connected together. Breadboards are commonly used for experimenting with and testing electronic circuits, as components can be quickly rearranged and swapped out. They are often divided into sections for organizing circuits, such as power rails and component areas. Breadboards come in various sizes and configurations, with larger ones accommodating more complex circuits.

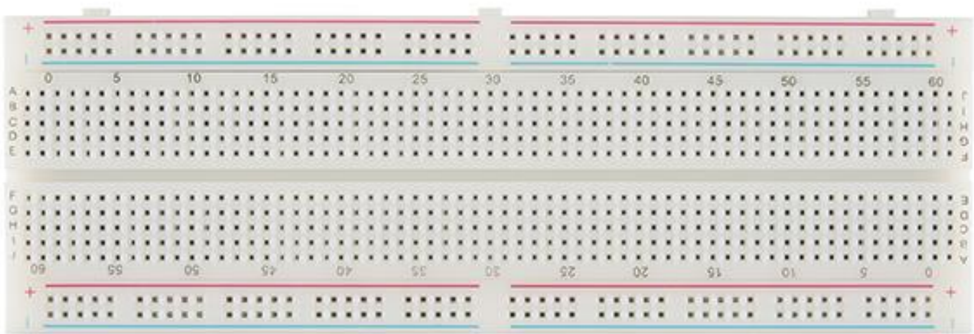

Figure 4.9 BREADBOARD

4.3.7 USB Cable

In the context of the Internet of Things (IoT), a USB cable, also known as a Universal Serial Bus cable, is used to transmit data and provide power between IoT devices and other hardware components.

It facilitates data and power interchange by serving as a physical link between IoT modules, sensors, and devices and the wider IoT system.

USB cables are widely used to link various devices in an Internet of things network in order to facilitate communication and power the connected devices.

Figure 4.10 USB cable

4.3.8 Connecting Wires

Jumper wires, another name for connecting wires, are used to join hardware components so they can exchange data or communicate with one another.

Figure 4.11 Connecting Wires

4.4 SOFTWARE DESCRIPTION

4.4.1 Arduino

The Arduino software, also known as the Arduino IDE (Integrated Development Environment), is a user-friendly platform used for writing, compiling, and uploading code to Arduino microcontroller boards. Here's a description:

1. Code Editor: The Arduino IDE features a simple code editor with syntax highlighting, making it easy to write and edit code in the Arduino programming language (based on C/C++).

2. Sketches: Arduino programs are referred to as "sketches." The IDE provides templates for creating new sketches and organizes them into tabs for easy navigation.

3. Library Manager: The IDE includes a library manager that allows users to easily install and manage libraries of pre-written code, expanding the capabilities of their projects without having to write everything from scratch.

4. Serial Monitor: The Serial Monitor tool enables real-time communication between the Arduino board and the computer, allowing users to send and receive data for debugging and monitoring purposes.

5. Board Manager: Arduino supports various microcontroller boards, each with its own specifications and capabilities. The Board Manager allows users to select the appropriate board and configure settings such as processor type and port.

6. Upload Tool: Once the code is written, users can upload it to the Arduino board via USB using the built-in upload tool. The IDE compiles the code, converts it into a format that the Arduino board can understand, and transfers it to the board's memory.

7. Integrated Examples: The IDE comes with a collection of example sketches covering a wide range of functionalities, providing beginners with useful starting points and reference materials for learning and experimenting.

Overall, the Arduino software simplifies the process of programming Arduino boards, making it accessible to users of all skill levels, from beginners to experienced developers. It provides essential tools and features for writing, compiling, and uploading code, as well as debugging and monitoring capabilities to aid in the development process.

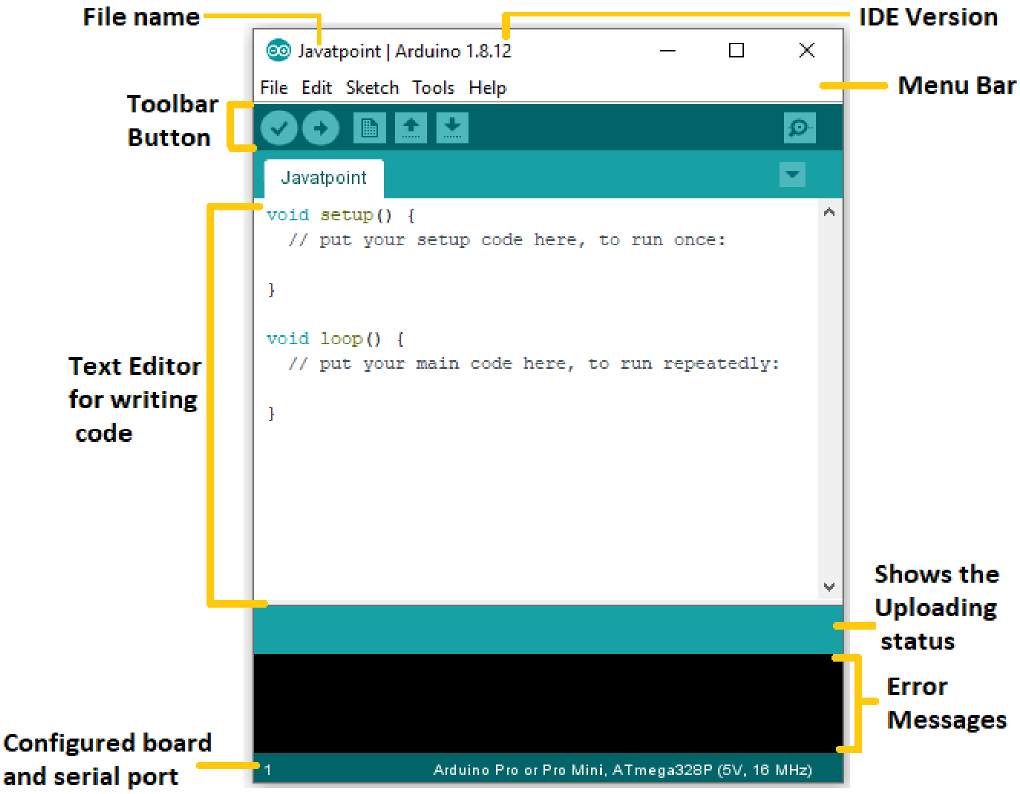

Figure 4.12 Arduino App

4.4.2 Arduino Sketch

An Arduino sketch is a piece of code, written in the Arduino programming language, that controls the behavior of an Arduino microcontroller. It is basically a series of instructions that outline the tasks that the Arduino board is supposed to perform.

Important details regarding Arduino sketches

- **Setup Function:** Every Arduino program includes the setup() function, which is called after the board is powered on or reset. This part is where you configure pin modes, initialize variables, and carry out any other one-time setup.
- **Loop Function:** An Arduino sketch's loop() method, which is its primary function, begins running continually after the setup() function is complete. The code inside the loop () function, which manages repetitive operations, defines the continuous behavior of the Arduino.
- **Basic Structure:** The two sections of an Arduino sketch are usually setup () for initialization and loop () for main execution. This format simplifies programming and works well for a wide range of applications.

- **Variables:** Variables are utilized in the sketch to store and alter data. Arduino supports a wide range of data types, including floating, integers, and characters. Variables can be used to represent sensor readings, control parameters, and any other values needed by the software.
- **Functions and control structure:** Unique functions can be incorporated into Arduino sketches to organize and modularize code. Control structures that perform conditional and repeating operations include while, for, else, and if.
- **Code comments:** Since comments are a component of the code, they need to be recorded. If they are multi-line comments, they are enclosed between "/" and "/" and signified by "//". They explain the goals and purposes of particular code segments to you and other people**.**

4.5 HARDWARE METHODOLOGY

The hardware methodology for implementing a smart helmet system using the ESP-Now protocol, ESP8266 module, and relay to enable bike ignition only when the helmet is worn:

1. Hardware Components:

- ESP8266 Module: Use an ESP8266 module (such as ESP-12E or NodeMCU) as the main controller.
- Relay Module: Connect a relay module to the ESP8266 to control the bike's ignition system.
- Helmet Detection Sensor: Employ a sensor (like an accelerometer or proximity sensor) to detect when the helmet is worn.
- Power Supply: Provide power to the ESP8266 module and other components.

2. Connection Setup:

- Connect the ESP8266 module to the relay module. Typically, this involves connecting the digital output pin of the ESP8266 to the control pin of the relay module.
- Connect the helmet detection sensor to the ESP8266 module. This sensor should be able to detect when the helmet is worn by the user.

3. Testing and Integration:

- Test the system thoroughly to ensure proper communication between the ESP8266 modules and accurate detection of helmet status.

- Integrate the hardware components into the bike's ignition system, ensuring proper connections and secure mounting of components.
- Conduct real-world testing to validate the functionality of the smart helmet system. Ensure that the bike ignition is only enabled when the helmet is worn and properly detected.

4. Safety Considerations:

- Ensure that the relay setup does not interfere with the normal operation of the bike's ignition system and follows safety standards.
- Perform adequate insulation and protection of electronic components to prevent damage from environmental factors or vibrations during bike operation.

4.6 PROGRAMMING METHODOLOGY

- Writing code to collect data from sensors, such as soil moisture, temperature, motion sensor, sound sensor is necessary part of programming IoT with NodeMCU ESP32.
- After that, this data is sent to an internet of things platform for analysis and visualization.
- In addition, the code contains logic to regulate irrigation pumps and other devices according to sensor readings.
- The Arduino IDE is used to program the NodeMCU ESP32, and libraries are utilized to interface with various sensors.
- Convolutional Neural Network algorithm is used for the emotion prediction. It is a image classification algorithm the model is trained with images which represent the emotion of the human.

4.7 PIN CONFIGURATION

4.7.1 NodeMCU

- GPIO0 is utilized to determine the boot mode during launch. Therefore, in order to avoid flash mode booting during startup, it shouldn't be pulled LOW. It is still possible to utilize this as an output pin.
- Although GPIO is an acronym for "general purpose input output," it is not possible to use GPIO34 through GPIO39 as outputs.

- GPIO32–GPIO39: These pins and the Analog to Digital Sensor can be used to measure voltages.
- GPIO5: The relay on the board is connected to this pin, as seen in the figure above. It supports the bike ignition to start the bike.

4.7.2 Relay Connecting with NodeMCU

Relay Pins

- VCC (Power): Connect to the Vin pin of the ESP8266 NodeMCU.
- GND (Ground): Connect the GND pin of the ESP8266 NodeMCU.
- Signal (Analog Output): Connect to GPIO5 analog input pin on the ESP8266 NodeMCU.

Pins on the NodeMCU ESP8266

- VCC: Connect to the 3.3V pin of the ESP8266 NodeMCU.
- GND: Connect to the GND pin on the ESP8266 NodeMCU
- Connect the analog input, such as GPIO5, to the signal pin of the relay

4.7.3 Momentary Switch Connecting Battery

Momentary switch pin

- VCC: Connect to the positive pin of battery.
- GND: Connect the negative pin of battery.

Pins on battery

- Positive (+): Connect to the positive pin of the Momentary switch.
- Negative (-): Connect to the negative pin of the Momentary switch.

4.7.4 Momentary Switch connecting to NodeMCU

Momentary switch

- VCC: Connect to the Vin pin of ESP8266 NodeMCU.
- GND: Connect to the GND pin of ESP8266 NodeMCU.

Pins on NodeMCU ESP8266

- VCC: Connect to the 3.3V pin of the ESP8266 NodeMCU.
- GND: Connect to the GND pin on the ESP8266 NodeMCU

CHAPTER 5
RESULTS AND DISCUSSION

5.1 HARDWARE MODULE

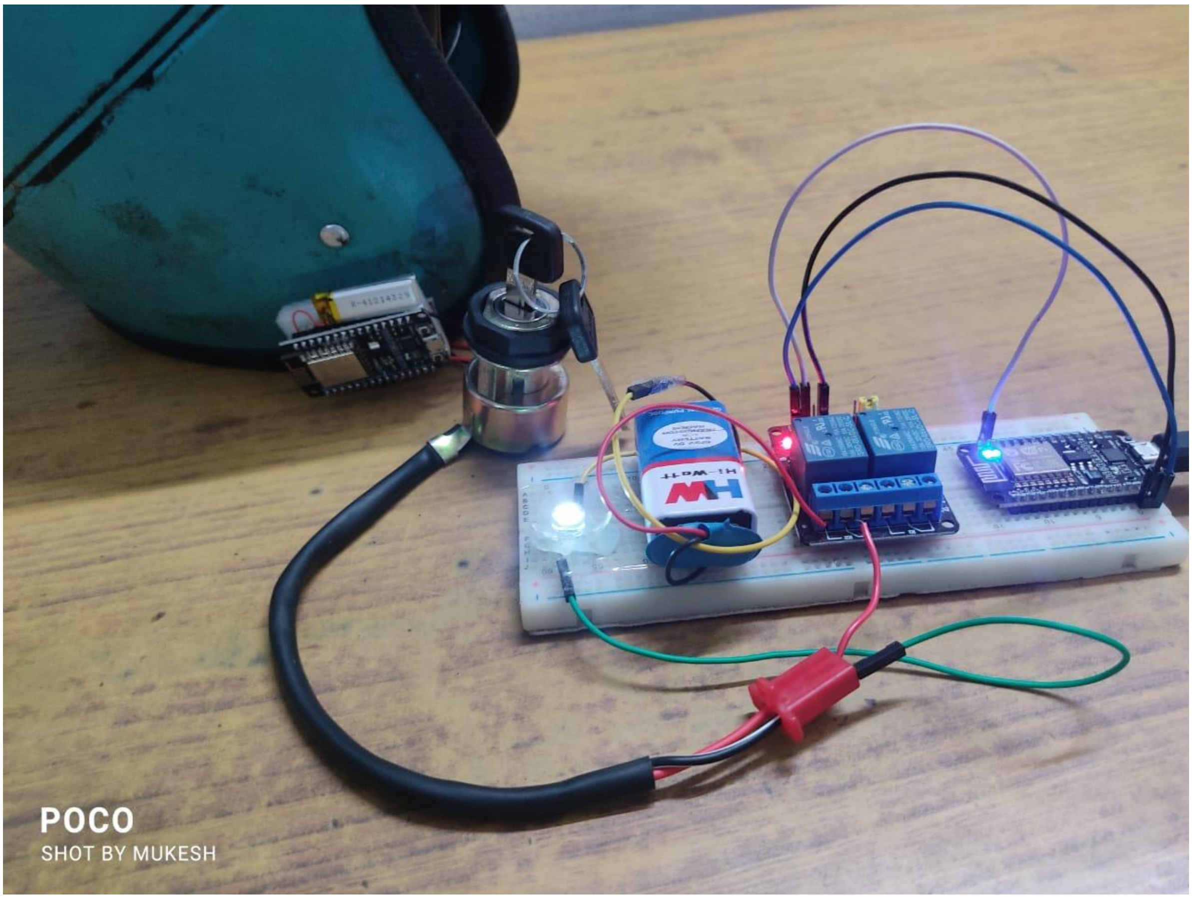

Figure 5.1 Hardware Module

In figure 5.1, it shows that the experimental hardware setup of our proposed system are shown. We have performed various experiments to test the feasibility of our smart system design. We have successfully connected all the sensors with the controllers and the server. The ESP8266 serves as the brain of the smart helmet. It controls all the other hardware modules and processes the data. It communicates wirelessly using the ESP-NOW protocol to send and receive data to and from other ESP8266 modules. The relay module is connected to the ESP8266 and various electrical components within the helmet, such as lights, indicators, or a buzzer. The ESP8266 sends signals to the relay module to control the activation or deactivation of these electrical components based on certain conditions. For example, it may turn on the helmet's lights when it gets dark or activate indicators when the rider signals a turn. The battery powers all the electronic components within the helmet, including the ESP8266 and relay module. It provides the necessary voltage and current to keep the system operational for an extended period. The battery should be rechargeable for convenience. The helmet serves as the physical housing for all the electronic components. It must be designed to accommodate the modules securely and safely while ensuring comfort for the user.The ignition set may include sensors or switches that detect when the helmet is worn or when the motorcycle is started. When the helmet is worn or the motorcycle is started, the ignition set triggers the ESP8266 to initialize and start monitoring various parameters. The momentary switch is a user input device that allows the rider to interact with the smart helmet. Depending on its configuration, the momentary switch can be used to activate specific functions or modes, such as turning on emergency lights or sending distress signals. Based on the sensor data and predefined logic programmed into the ESP8266, it sends commands to the relay module to control the activation or deactivation of electrical components. This coordinated operation of the hardware modules enables the smart helmet to enhance safety and provide additional functionalities to the rider while on the road.

CHAPTER 6
CONCLUSION AND SCOPE FOR FUTURE WORK

6.1 CONCLUSION

- In conclusion, the development of a smart helmet utilizing the ESP-NOW protocol represents a significant advancement in motorcycle safety and security. By integrating an ESP8266 board into the helmet, we have successfully implemented a system where the motorcycle only starts when the rider is wearing the helmet. This not only promotes responsible riding behavior but also enhances the overall safety of the rider.
- Through the use of wireless communication technology, specifically ESP-NOW, the helmet is able to send a signal to the motorcycle's relay system, allowing it to start only when the helmet is detected. This innovative approach ensures that riders prioritize their safety by wearing a helmet before operating the motorcycle, thereby reducing the risk of accidents and potential injuries.
- Moreover, by incorporating the ESP8266 board, we have demonstrated the feasibility and effectiveness of leveraging IoT devices for enhancing motorcycle safety. This project serves as a testament to the potential of IoT solutions in addressing real-world challenges and improving the overall experience of motorcycle riders.
- Overall, the implementation of the smart helmet using the ESP-NOW protocol underscores the significance of innovation in the realm of motorcycle safety.

6.2 FUTURE SCOPE

Future scope for the smart helmet project includes integrating advanced sensors for collision detection, expanding wireless communication options for extended range, developing a companion mobile app for added functionality, incorporating cloud connectivity for remote monitoring and updates, exploring partnerships with motorcycle manufacturers for seamless integration, implementing real-time GPS tracking for enhanced rider safety, enhancing user interface for intuitive operation, incorporating voice commands for hands-free control, integrating biometric sensors for health monitoring, exploring AI algorithms for predictive maintenance and hazard detection, developing a robust security system to prevent tampering or unauthorized access, optimizing power management for extended battery life, implementing machine learning algorithms for personalized safety recommendations, exploring potential integration with vehicle-to-vehicle communication systems, and conducting user feedback studies to continuously improve usability and effectiveness.

APPENDIX

SENDER CODING

```
#include <ESP8266WiFi.h>
#include <espnow.h>
// REPLACE WITH RECEIVER MAC Address
uint8_t broadcastAddress[] = {0x24, 0xd7, 0xeb, 0xcc, 0xcb, 0xca};//24:d7:eb:cc:cb:ca
#define relay 5
// Structure example to send data
unsigned long lastTime = 0;
unsigned long timerDelay = 2000;  // send readings timer
// Callback when data is sent
void OnDataSent(uint8_t *mac_addr, uint8_t sendStatus) {
  Serial.print("Last Packet Send Status: ");
  if (sendStatus == 0){
    Serial.println("Delivery success");
    digitalWrite(relay,LOW);
    digitalWrite(LED_BUILTIN,LOW);
  }
  else {
    Serial.println("Delivery fail");
    Serial.println(sendStatus);
    digitalWrite(relay,HIGH);
    digitalWrite(LED_BUILTIN,HIGH);
  }
}
void setup() {
  // Init Serial Monitor
  Serial.begin(115200);
```

```
 pinMode(relay,OUTPUT);
 pinMode(LED_BUILTIN,OUTPUT);
  // Set device as a Wi-Fi Station
  WiFi.mode(WIFI_STA);
  // Init ESP-NOW
  if (esp_now_init() != 0) {
    Serial.println("Error initializing ESP-NOW");
    return;
  }
  // Once ESPNow is successfully Init, we will register for Send CB to
  // get the status of Trasnmitted packet
  esp_now_set_self_role(ESP_NOW_ROLE_CONTROLLER);
  esp_now_register_send_cb(OnDataSent);
  // Register peer
  esp_now_add_peer(broadcastAddress, ESP_NOW_ROLE_SLAVE, 1, NULL, 0);
}
void loop() {
  if ((millis() - lastTime) > timerDelay) {
    // Set values to send
    int myData = 1;
    // Send message via ESP-NOW
    esp_now_send(broadcastAddress, (uint8_t *) &myData, sizeof(myData));
    lastTime = millis();
  }
}
```

RECEIVER CODING

```
#include <ESP8266WiFi.h>
#include <espnow.h>
int myData=0;
// Callback function that will be executed when data is received
void OnDataRecv(uint8_t * mac, uint8_t *incomingData, uint8_t len) {
  memcpy(&myData, incomingData, sizeof(myData));
 Serial.println(myData);
}
void setup() {
  // Initialize Serial Monitor
  Serial.begin(115200);
  // Set device as a Wi-Fi Station
  WiFi.mode(WIFI_STA);
  // Init ESP-NOW
  if (esp_now_init() != 0) {
    Serial.println("Error initializing ESP-NOW");
    return;
  }
  // Once ESPNow is successfully Init, we will register for recv CB to
  // get recv packer info
  esp_now_set_self_role(ESP_NOW_ROLE_SLAVE);
  esp_now_register_recv_cb(OnDataRecv);
}
void loop() {
}
```

REFERENCES

[1] N.K.Madzhi,M.K.A.M.Rasli(2016)Smart Safety Helmet for Bike Riders using IOT.Institute of Electrical and Electronics Engineers(IEEE)

[2] J Joy mathavan,V K d ijesekara,N Satheeskanth,Smart helmet to start the motorbike and to prevent accidents.International Conference on Materials Science and Manufacturing Technology ,ICMSMT-2022

[3] Abhinav Anand, Alcoholic detection, Department of Electronics and Telecommunications,IJEETC, 2015.

[4] Mr. Vivek A. Patel, Mr. Akash Mishra, Mr. RanaHiten, Mr. Kautik Prajapati, Smart Helmet (IRJET)

[5] Deekshitha K J, Pushpalatha S2, Implementation of Smart Helmet (IRJET).

[6] N. Manjesh., Sudarshan Raj, Smart Helmet using GSM & GPS Technology for Accidentdetection and Reporting System, International Journal of Electrical and ElectronicsResearch, 2 (2014) 122-127.

[7] S. A. Shabbeer and M. Meleet, "Smart Helmet for Accident Detection and Notification," 2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS), Bangalore, 2017, pp. 1-5, doi: 10.1109/CSITSS.2017.8447702.

[8] N. Nataraja, K. S. Mamatha, Keshavamurthy and Shivashankar, "SMART HELMET," 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, India, 2018, pp. 2338-2341, doi: 10.1109/RTEICT42901.2018.9012338.

[9] N. V. Joshi, S. P. Joshi, M. S. Jojare, N. S. Joshi and A. R. Askhedkar, "Design and Finite Element Analysis of IoT based Smart Helmet," 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Vancouver,BC,Canada,2020,pp.1-8,doi:10.1109/IEMTRONICS51293.2020.9216393.

[10] M. Jeong, H. Lee, M. Bae, D. Shin, S. Lim and K. B. Lee, "Development and Application of the Smart Helmet for Disaster and Safety," 2018 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, 2018, pp. 1084-1089, doi: 10.1109/ICTC.2018.8539625.

[11] "IoT Based Smart Helmet and Accident Identification System," 2020 IEEE Region 10 Symposium (TENSYMP), Dhaka, Bangladesh, 2020, pp. 14-17, doi: 10.1109/TENSYMP50017.2020.9230823.

[12] R. Budiman, D. W. Sudiharto and T. Brotoharsono, "The Prototype of SmartHelmet with Safety Riding Sends Notification for Motorcycle Rider," 2018 3rd International Conference on Information Technology, Information System and Electrical Engineering, Computer Engineering(ICITISEE), Yogyakarta, Indonesia, 2018, pp. 362-367, doi: 10.1109/ICITISEE.2018.8721027