DrillBit

The Report is Generated by DrillBit Plagiarism Detection Software

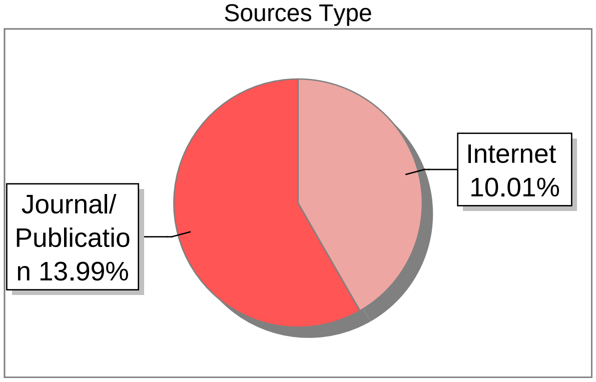
Submission Information

Author Name	Ravikumar Kodadal
Title	Survey On Accident Detection and Health Severity Assessment Using IoT
Paper/Submission ID	3576275
Submitted by	rkodadal@gmail.com
Submission Date	2025-05-05 12:06:59
Total Pages, Total Words	48, 7431
Document type	Project Work

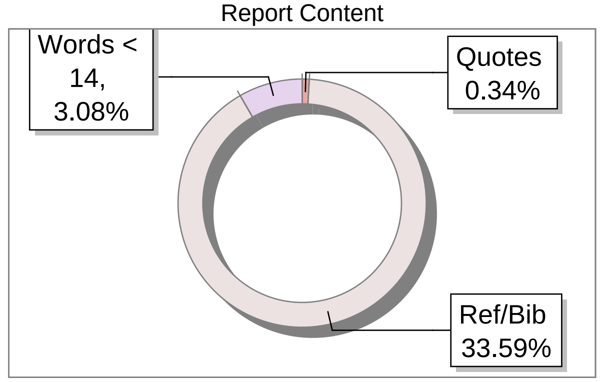
Result Information

Similarity **24 %**

Sources Type



Report Content



Exclude Information

Quotes	Not Excluded
References/Bibliography	Not Excluded
Source: Excluded < 14 Words	Not Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

24	58	B	A-Satisfactory (0-10%) B-Upgrade (11-40%) C-Poor (41-60%) D-Unacceptable (61-100%)
SIMILARITY %	MATCHED SOURCES	GRADE	

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	ece.anits.edu.in	4	Publication
2	www.ijfmr.com	3	Publication
3	moam.info	3	Internet Data
4	wgbis.ces.iisc.ac.in	2	Internet Data
5	www.ijfmr.com	1	Publication
6	ijrte.org	1	Publication
7	ijsrset.com	1	Publication
8	ece.anits.edu.in	1	Publication
9	qdoc.tips	1	Internet Data
10	docplayer.net	<1	Internet Data
11	www.srielectronics.com	<1	Internet Data
12	aarf.asia	<1	Publication
13	IEEE 2020 IEEE Region 10 Symposium (TENSYMP) - Dhaka, Bangladesh (202065-202	<1	Publication
14	mdpi.com	<1	Internet Data

15	www.mdpi.com	<1	Internet Data
16	frontiersin.org	<1	Internet Data
17	czasopisma.uni.lodz.pl	<1	Internet Data
18	translate.google.com	<1	Internet Data
19	Lowcost observations and experiments return a high value in plant phenology r by McDonoug-2020	<1	Publication
20	www.ijitee.org	<1	Publication
21	www.science.gov	<1	Internet Data
22	bura.brunel.ac.uk	<1	Publication
23	docplayer.net	<1	Internet Data
24	www.dx.doi.org	<1	Publication
25	www.dx.doi.org	<1	Publication
26	afflictor.com	<1	Internet Data
27	biomedcentral.com	<1	Internet Data
28	Brodart automation Building on an innovative and stable past by VanFleet-1995	<1	Publication
29	coek.info	<1	Internet Data
30	coek.info	<1	Internet Data
31	funpecrp.com.br	<1	Publication
32	icct.iacst.org	<1	Publication
33	link.springer.com	<1	Internet Data

34	Preparation and physicochemical properties of carboxymethyl Fritillar, by Xia Li Wen-yuan Ga- 2010	<1	Publication
35	sportdocbox.com	<1	Internet Data
36	Thesis Submitted to Shodhganga, shodhganga.inflibnet.ac.in	<1	Publication
37	www.ijnrd.org	<1	Publication
38	www.scribd.com	<1	Internet Data
39	30years of the Pacific and by Higgott-2017	<1	Publication
40	Adsorption- and Diffusion-Controlled Wettability Change in Modified Salinity Wat by Taheriotaghsara-2020	<1	Publication
41	darkmatter69.blogspot.com	<1	Internet Data
42	docplayer.net	<1	Internet Data
43	frontiersin.org	<1	Internet Data
44	ijarcce.com	<1	Publication
45	Investigating injury severity risk factors in automobile crashes with predictive by Delen-2017	<1	Publication
46	link.springer.com	<1	Internet Data
47	Mental health and itch in burns patients Potential associations, by McGarry, Sarah Bur- 2016	<1	Publication
48	Platinum carbon bond formation via Cu(, by Gundogan, Ali S.; M - 2015	<1	Publication
49	Thesis Submitted to Shodhganga, shodhganga.inflibnet.ac.in	<1	Publication
50	Treatment options for chronic pancreatitis by Issa-2014	<1	Publication
51	vdocuments.net	<1	Internet Data

52	www.academia.edu	<1	Internet Data
53	www.dx.doi.org	<1	Publication
54	www.forbes.com	<1	Internet Data
55	www.intechopen.com	<1	Internet Data
56	www.nice.com	<1	Internet Data
57	www.snowsoftware.com	<1	Internet Data
58	xdoc.es	<1	Internet Data

CHAPTER 1: INTRODUCTION

Motorcycles are a popular and efficient mode of transportation, particularly in densely populated urban areas and developing countries. They offer numerous advantages, including lower fuel consumption, easier navigation through traffic, and affordability. However, motorcycles come with a significant risk: due to their lower stability and lack of physical protection compared to vehicles like cars, motorcyclists are particularly vulnerable in the event of accidents. This vulnerability makes them more prone to serious injuries and fatalities, especially when accidents occur at high speeds or in remote areas where medical assistance may be delayed.

According to the World Health Organization (WHO), motorcycle accidents contribute significantly to global road traffic injuries and fatalities, with riders being among the most at-risk groups. A major challenge faced in such incidents is the delay in providing timely emergency assistance, which often leads to fatalities that could have been prevented or injuries that could have been mitigated with faster medical intervention. This delay can stem from several factors, including lack of awareness, traffic congestion, or delays in communication between the accident site and emergency responders.

To address this critical issue, the IoT-Driven Motorcycle Accident Detection System has been proposed. This system leverages real-time sensor data from various onboard devices, such as accelerometers, gyroscopes, and biometric sensors, to detect crashes and assess the severity of injuries. It also integrates GPS tracking to pinpoint the exact location of the accident. The data gathered from these sensors is processed by advanced artificial intelligence algorithms, such as Logistic Regression, which helps detect the occurrence of a crash, and Deep Neural Networks (DNN), which classify the severity of the rider's injuries.

In addition to crash detection, the system also evaluates the rider's physical condition by monitoring vital physiological signals like SpO2 (blood oxygen saturation), heart rate, and body temperature. This allows the system to determine the urgency of the situation and prioritize medical attention accordingly. Once the system detects a crash and classifies the injury severity, it immediately sends the rider's location and medical status to emergency services, as well as to pre-configured emergency contacts, such as family members or friends, ensuring a rapid response to the incident.

The integration of the Internet of Things (IoT) and Artificial Intelligence (AI) in this system brings several key benefits:

• **Minimizes response time**: By reducing the time between the accident and the dispatch of emergency services, the system significantly improves the chances of survival for the rider.

• **Enhances survival rates**: Faster dispatch of ambulances can mean the difference between life and death, as timely medical intervention can prevent fatalities or worsen injuries.

• **Reduces false positives**: Traditional crash detection methods based on simple thresholds can sometimes produce false alarms, leading to unnecessary medical responses. In contrast, the AI-powered system ensures higher accuracy by considering multiple factors, including the rider's physical condition.

• **Scalable and adaptable**: The system is designed to be deployed in both rural and urban areas with high accuracy and low latency, making it suitable for a wide range of geographical environments.

The technical architecture of this system involves the development of a smart embedded platform using low-power microcontrollers like the ESP32, which is known for its affordability and ability to handle IoT applications. The system also utilizes robust sensors that can operate in various environmental conditions, ensuring reliability and consistency in crash detection. To ensure efficient processing, lightweight AI models are employed, reducing computational requirements and enabling the system to function on low-power devices while maintaining high performance.

The primary objective of this project is to create a scalable and cost-effective solution that can be implemented on a large scale to enhance road safety for motorcyclists. By combining IoT and AI, this system aims to provide a practical and efficient solution to reduce motorcycle-related fatalities and injuries, ultimately contributing to safer roads and more efficient emergency response systems.

CHAPTER 2: LITERATURE SURVEY

1. Smart Helmets and Impact Detection Systems:

Smart helmets with sensors like accelerometers and gyroscopes detect impacts and sudden movements, such as those by **Forcite**. While effective in impact detection, they often lack integration with health monitoring and GPS systems for comprehensive safety.

2. Wearable Devices for Health Monitoring:

Devices like **Garmin** and **Apple Watch** track vital signs and detect falls, but they do not cater specifically to motorcyclist safety or provide automated emergency alerts in the event of an accident.

3. GPS-Based Tracking and Accident Location Sharing:

GPS systems, such as **Tracki** and **Life360**, enable real-time location sharing in emergencies but do not include features for assessing injury severity or health conditions, which limits their effectiveness in critical situations.

4. Artificial Intelligence in Injury Assessment:

AI tools, used in emergency care to assess injuries, are being adapted for accident situations. However, integrating AI for real-time injury assessment in wearable devices is still a challenge.

5. Integrated Safety Systems:

BMW's Motorrad Connected Ride integrates crash detection, GPS, and emergency calls but lacks health monitoring and AI-driven injury analysis, highlighting the gap for a more comprehensive safety solution.

6. Challenges in Real-Time Data Integration:

Real-time synchronization of multiple data streams (motion, health metrics, location) remains a challenge in ensuring effective emergency response. Many existing systems fail to provide instantaneous, seamless integration.

7. Privacy and Security Concerns:

Integrating sensitive data such as health metrics and location requires strong encryption and data protection. Many existing solutions do not fully comply with regulations like **GDPR**, exposing user data to security risks.

CHAPTER 3: PROBLEM STATEMENT

Problem Statement:

Despite the progress in individual safety technologies, the existing systems do not offer a holistic solution that combines accident detection, health monitoring, injury assessment, and emergency response into a single, easy-to-use framework. As a result, motorcycle riders often experience delayed responses from emergency services, which can exacerbate the severity of injuries and reduce the chances of survival.

CHAPTER 4: OBJECTIVES

- **Real-Time Accident Detection**:

  Implement a reliable system using IoT sensors (e.g., accelerometers, gyroscopes) to instantly detect motorcycle accidents based on sudden impact or abnormal riding behaviour.

- **Injury Severity Assessment:**

  Develop AI algorithms to analyse sensor data and assess the severity of injuries by evaluating the impact forces, rider's posture, and physiological data.

- **Precise Location Tracking:**

  Utilize GPS technology to accurately track the location of the accident and send real-time coordinates to emergency responders for a timely intervention.

- **Automated Emergency Notification:**

  Design a system that automatically alerts emergency services with accident details, injury severity, and location, reducing response time and ensuring faster medical assistance.

CHAPTER 5: SYSTEM REQUIREMENT SPECIFICATION

5.1 Hardware Requirements

Component	Specification
ESP32	Dual-core, Wi-Fi/Bluetooth
MAX30100	Pulse oximeter for heart rate & SpO2
DHT11/DHT22/AM2302	Temperature and humidity monitoring
GPS Module	Real-time location tracking
Vibration Sensor	Detects impact/accident eventst events

5.1.1 GPS:

Fig 5.1: GPS

The Global Positioning System (GPS) is a location system based on a constellation of about 24 satellites orbiting the earth at altitudes of approximately 11,000 miles. GPS was developed by the United States Department of Defense (DOD), for its tremendous application as a military locating utility. The DOD's investment in GPS is immense. Billions and billions of dollars have been invested in creating this technology for military uses. However, over the past several years, GPS has proven to be a useful tool in non-military mapping applications as well.

GPS satellites are orbited high enough to avoid the problems associated with land based systems, yet can provide accurate positioning 24 hours a day, anywhere in the world. Uncorrected positions determined from GPS satellite signals produce accuracies in the range of 50 to 100 meters. When using a technique called differential correction, users can get positions accurate to within 5 meters or less.

5.1.2 ESP32

Fig 5.2: ESP32

What is ESP32?

ESP32 is a low-cost System on Chip (SoC) Microcontroller from Espressif Systems, the developers of the famous ESP8266 SoC. It is a successor to ESP8266 SoC and comes in both single-core and dual-core variations of the Tensilica’s 32-bit Xtensa LX6 Microprocessor with integrated Wi-Fi and Bluetooth.

6 The good thing about ESP32, like ESP8266 is its integrated RF components like Power Amplifier, Low-Noise Receive Amplifier, Antenna Switch, Filters and RF Balun. This makes designing hardware around ESP32 very easy as you require very few external components.

Specification :

- 7 The ESP32 is dual core, this means it has 2 processors.
- It has Wi-Fi and bluetooth built-in.
- It runs 32 bit programs.
- The clock frequency can go up to 240MHz and it has a 512 kB RAM.
- This particular board has 30 or 36 pins, 15 in each row.
- It also has wide variety of peripherals available, like: capacitive touch, ADCs, DACs, UART, SPI, I2C and much more.

Some of the commonly used programming environments are:

- Arduino IDE
- PlatformIO IDE (VS Code)
- LUA
- MicroPython
- Espressif IDF (IoT Development Framework)
- JavaScript

As Arduino IDE is already a familiar environment, we will use the same to program ESP32 in our project.

5.1.3 Parts of ESP32

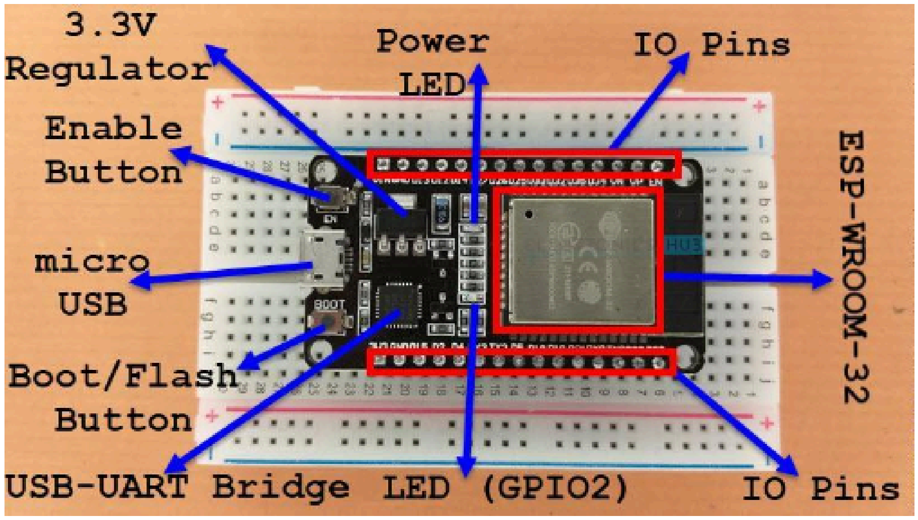


Fig 5.3: Parts of ESP32

The board has 30 Pins (15 pins on each side) layout, pinout and features vary from board to board. There are some board with 36 Pins and some with slightly less Pins.

As you can see from the image, the ESP32 Board consists of the following:

- ESP-WROOM-32 Module
- Two rows of IO Pins (with 15 pins on each side)
- CP2012 USB – UART Bridge IC
- micro–USB Connector (for power and programming)
- AMS1117 3.3V Regulator IC
- Enable Button (for Reset)
- Boot Button (for flashing)
- Power LED (Red)

- User LED (Blue – connected to GPIO2)
- Some passive components

An interesting point about the USB-to-UART IC is that its DTR and RTS pins are used to automatically set the ESP32 in to programming mode (whenever required) and also rest the board after programming.

5.1.4 DHT11

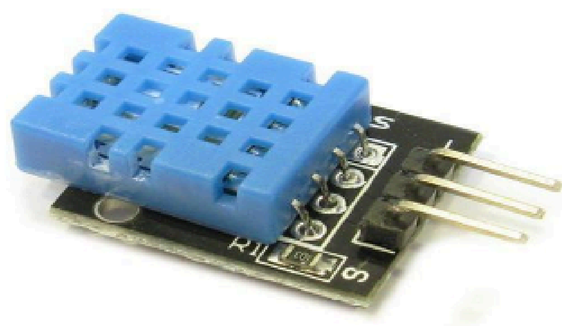


Fig 5.4: DHT11

The **DHT11** is a widely used, low-cost digital sensor designed to measure **temperature and relative humidity** in environmental monitoring systems. It is suitable for various electronics and IoT applications.

1. Sensing Elements

- The **humidity sensor** is a **resistive-type component** made of a substrate coated with a conductive polymer. As humidity changes, the resistance across the sensor varies accordingly.
- The **temperature sensor is an NTC (Negative Temperature Coefficient) thermistor**, which decreases its resistance as the temperature increases.

2. Internal Microcontroller

- The sensor module integrates a **high-performance 8-bit microcontroller** that reads data from the sensing elements.
- This microcontroller performs **digital signal processing**, calibration, and communication with external devices.

3. Calibrated Digital Output

- DHT11 provides a **calibrated digital signal output**, meaning it internally processes the analog signals from the sensors and transmits them in digital form.
- This reduces the need for external ADC (Analog-to-Digital Conversion) on the microcontroller side.

4. Communication Interface

- Uses a **single-wire serial interface**, which is simple and efficient.
- The sensor transmits data in a **40-bit format**: 8 bits each for humidity integer + decimal, temperature integer + decimal, and checksum.

5. Performance Characteristics

- **Humidity Range**: 20% to 90% RH with ±5% accuracy.
- **Temperature Range**: 0°C to 50°C with ±2°C accuracy.
- **Sampling Rate**: One reading every 1 second (1Hz).
- **Response Time**: Typically within 6 to 10 seconds.

6. Advantages

- **High reliability** due to built-in calibration and signal conditioning.
- **Excellent long-term stability** for consistent performance over time.
- **Fast response** to environmental changes.
- **Low power consumption**, making it ideal for portable and battery-powered devices.
- **Anti-interference design**, suitable for industrial and noisy environments.
- **Cost-effective** choice for hobbyists, students, and basic automation systems.

7. Applications

- Weather monitoring systems
- HVAC systems
- Smart agriculture
- IoT-based home automation
- Health and safety devices (like smart helmets)

5.1.5 Vibration Sensor

Fig 5.5: Vibration Sensor

Module Description

Can detect the vibration of the surrounding environment

Sensitivity adjustable the blue digital potentiometer adjustment (Figure)

Operating voltage 3.3V-5V

Output form: Digital switching output (0 and 1)

With fixed bolt hole for easy installation

Small board PCB size: 3cm * 1.6cm

Module interface specification (3-wire)

VCC external 3.3V-5V voltage (can be directly connected to the 5v microcontroller and 3.3v microcontroller)

GND external GND DO-board digital output interface (0 and 1)

Design and function

On account of its inertia, a mass exerts compressive forces on a ring-shaped piezo-ceramic element in time with the

oscillation which generates the excitation. Within the ceramic element, these forces result in charge transfer within the ceramic and a voltage is generated between the top and bottom of the ceramic element. This voltage is picked-off using contact discs – in many cases it is filtered and integrated – and made available as a measur-ing signal. In order to route the vibration directly into the sensor, vibration sensors are securely bolted to the object on which measurements take place.

Measurement sensitivity

Every vibration sensor has its own individual response characteristic which is closely linked to its measurement sensitivity. The measurement sensitivity is defined as the output voltage per unit of acceleration due to gravity (see characteristic curve). The production-related sensitivity scatter is acceptable for applications where the primary task is to record that vibration is occurring, and not so much to measure its severity. The low voltages generated by the sensor can be evaluated using a high-impedance AC amplifier

Applications

Vibration sensors of this type are suitable forthe detection of structure-borne acoustic oscillations as can occur for example in case of irregular combustion in engines and on machines. Thanks to their ruggedness, these vibration sensors can be used even under the most severe operating conditions.

Areas of application

- Knock control for internal-combustion engines
- Protection of machine tools
- Detection of cavitation
- Monitoring of bearings
- Theft-deterrent systems

5.1.6 MAX30100

Fig 5.6: MAX30100

What is Pulse Oximeter?

Pulse Oximeters are low cost non-Invasive medical sensors used to continuously measure the Oxygen saturation (SPO2) of haemoglobin in blood. It displays the percentage of blood that is loaded with oxygen.

Principle of Pulse oximeter

The principle of pulse oximetry is based on the differential absorption characteristics of oxygenated and the de-oxygenated hemoglobin. Oxygenated hemoglobin absorbs more infrared light and allows more red light to pass through. Whereas Deoxygenated hemoglobin absorbs more red light and allowing more infrared light to pass through.

What's inside the Sensor?

Each pulse oximeter sensor probe contains two light emitting diode one emitting red light and the other emitting near infrared light, it also has a photo-detector. The photo-detector measures the intensity of transmitted light at each wavelength. And using the differences in the reading the blood oxygen content is calculated. The probe is placed on a suitable part of the body, usually a fingertip or ear lobe.

Pulse Oximeter MAX30100 / MAX30102 is a cheap one and quite popular among hobbyists. Unfortunately, the cheapest module boards (which are sold by thousands on Aliexpress)

Everything I said below applies equally to the new MAX30102 modules and old modules with discontinued MAX30100 chip since they are assembled on the same boards.

Working of MAX30100 Pulse Oximeter and Heart-Rate Sensor:

The device has two LEDs, one emitting red light, another emitting infrared light. For pulse rate, only infrared light is needed. Both red light and infrared light are used to measure oxygen levels in the blood. When the heart pumps blood, there is an increase in oxygenated blood as a result of having more blood. As the heart relaxes, the volume of oxygenated blood also decreases. By knowing the time between the increase and decrease of oxygenated blood, the pulse rate is determined. It turns out, oxygenated blood absorbs more infrared light and passes more red light while deoxygenated blood absorbs red light and passes more infrared light. This is the main function of the MAX30100: it reads the absorption levels for both light sources and stores them in a buffer that can be read via I2C communication protocol.

5.1.7 Working of Pulse oximeter

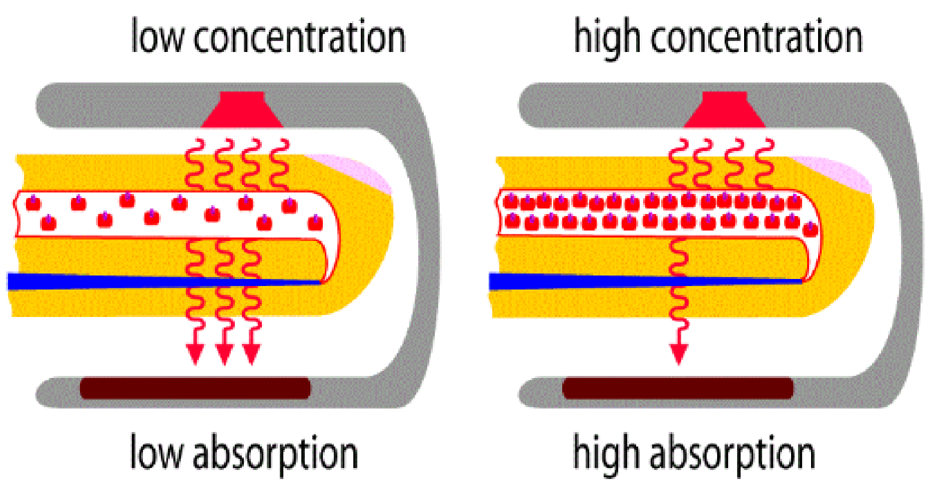


Fig 5.7: Working of Pulse oximeter

What is Pulse Oximeter?

Pulse Oximeters are low cost non-Invasive medical sensors used to continuously measure the Oxygen saturation (SPO2) of haemoglobin in blood. It displays the percentage of blood that is loaded with oxygen.

Principle of Pulse oximeter

The principle of pulse oximetry is based on the differential absorption characteristics of oxygenated and the de-oxygenated haemoglobin. Oxygenated haemoglobin absorbs more infrared light and allows more red light to pass through. Whereas Deoxygenated haemoglobin absorbs more red light and allowing more infrared light to pass through.

Working of Pulse oximeter

Oxygen enters the lungs and then is passed on into blood. The blood carries oxygen to the various organs in our body. The main way oxygen is carried in our blood is by means of haemoglobin. During a pulse oximetry reading, a small clamp-like device is placed on a finger, earlobe, or toe.

Small beams of light pass through the blood in the finger, measuring the amount of oxygen. It does this by measuring changes in light absorption in oxygenated or deoxygenated blood.

5.2 Software Requirements

- **Blynk IoT** for remote alerting
- **TensorFlow Lite** for on-device neural network inference

5.2.1 Programming Environment

- **IDE:** Arduino IDE
- **Board Selected:** ESP32 Dev Module
- **Programming Language:** C/C++ (Arduino-based)

5.2.2 Libraries Used

- MAX30100_PulseOximeter.h – For reading heart rate and SpO2 using the MAX30100 sensor.
- DHT.h – For reading temperature values from the DHT11 sensor.
- TinyGPS++.h – For parsing GPS data (latitude and longitude).
- WiFi.h and WiFiClient.h – For connecting ESP32 to Wi-Fi.
- Blynk.h – For connecting to the Blynk IoT platform and sending data.
- *(Optional)* LiquidCrystal_I2C.h – For displaying sensor values on an LCD (commented out in code).

5.2.3 Connectivity

- **Wi-Fi SSID & Password:** ESP32 connects to a local Wi-Fi network for internet access.
- **Blynk Cloud:** Used for real-time mobile app notifications and data display.
- **Virtual Pins (V0 to V4):** Used to display:
    - V0 → Latitude
    - V1 → Longitude
    - V2 → SpO2
    - V3 → Heart Rate
    - V4 → Temperature

5.2.4 Core Software Functionalities

- **Sensor Data Collection:**
    - Heart rate and SpO2 from MAX30100.
    - Temperature from DHT11.
    - Vibration and force values from analog sensors.
    - GPS location via TinyGPS++.

- **Accident Detection Logic:**
  - If vibration or force values drop below threshold, an accident is suspected.
  - If detected, a **notification** is sent via Blynk, including GPS and health data.
- **Data Uploading:**
  - Health and GPS data is regularly sent to the Blynk app.
  - Notifications are sent once per accident to avoid spamming (using oneTimeFlag).
- **Wi-Fi Status Handling:**
  - Checks Wi-Fi connectivity and restarts ESP32 if not connected.

5.2.5 Notification System

- **Blynk logEvent():** Sends a mobile alert titled **"Alert Accident Detected"** with health and location info.
- **Serial Monitor:** Used for debugging and displaying sensor readings.

Requirement	**Details**
IDE	Arduino IDE
Board	ESP32 Dev Module
Libraries Used	MAX30100_PulseOximeter.h – for pulse rate and SpO2DHT.h – for temperature sensorTinyGPS++.h – for GPS dataWiFi.h & WiFiClient.h – for Wi-Fi connection (ESP32)Blynk.h – for sending data to mobile app
Wi-Fi Credentials	SSID: ESP32, Password: 123456789
Blynk Template ID	TMPL3ktmAl35z
Blynk Auth Token	eLXvPy-SUnxOkGfoG16e3HnDPN3xiCJJ
Notification	Blynk log event: "notification" with health and location data

CHAPTER 6: SYSTEM DESIGN

6.1 System Architecture

Flow:

Sensors → ESP32 → Blynk Cloud → Hospital API

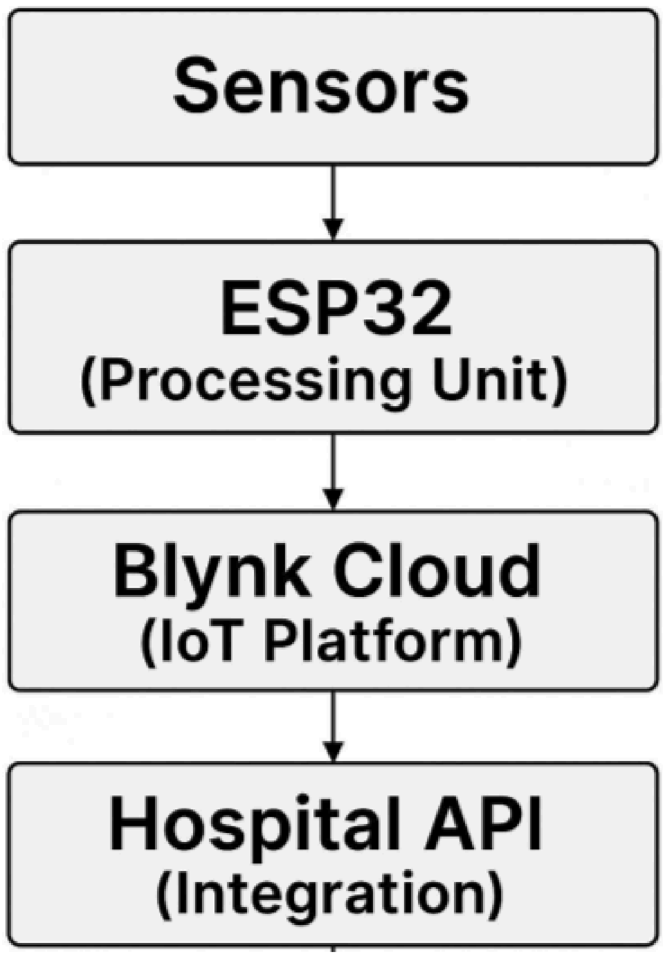


Fig 6.1: System Architecture

Process:

1. Detect crash
2. Assess severity using DNN
3. Capture GPS location
4. Send emergency alert

6.2 Data Flow Diagram

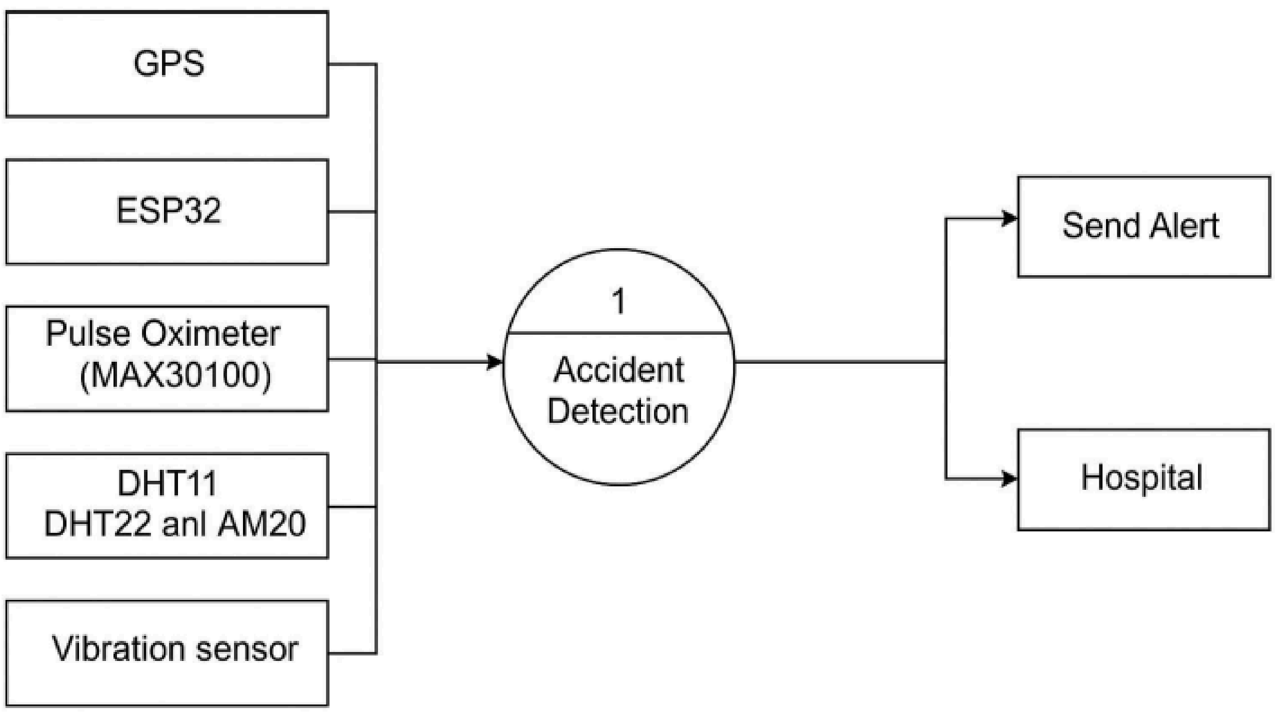


Fig 6.2: Data Flow

Inputs (Left Side - Sensors and Controller):

1. GPS:
   - Provides real-time location data of the individual.
2. ESP32:
   - Acts as the main controller that processes data from all sensors.
3. Pulse Oximeter (MAX30100):
   - Measures heart rate and SpO2 (blood oxygen level) to monitor the person's health.
4. DHT11, DHT22, and AM2302 Sensors:
   - Collect temperature and humidity data from the environment.
5. Vibration Sensor:
   - Detects any sudden shock or impact, which may indicate an accident.

Process (Center Circle - "1 Accident Detection"):

- This is the main function block where all sensor data is collected and analyzed.
- The system detects if an accident has occurred based on the vibration sensor and health data.
- It checks the condition of the person using pulse, temperature, and oxygen levels.

Outputs (Right Side):

1. Send Alert:
   - If an accident is detected, the system triggers an alert sound.
   - Sends a message with all data (location + health condition) to parents or guardians.
2. Hospital:

- Sends all necessary information (location and vitals) to the nearest hospital to help in providing immediate medical aid.

6.3 UML Diagrams

Use Case Diagram

Actors: Rider, EMT, System

Use Cases: Detect Crash, Transmit Location, Classify Injury

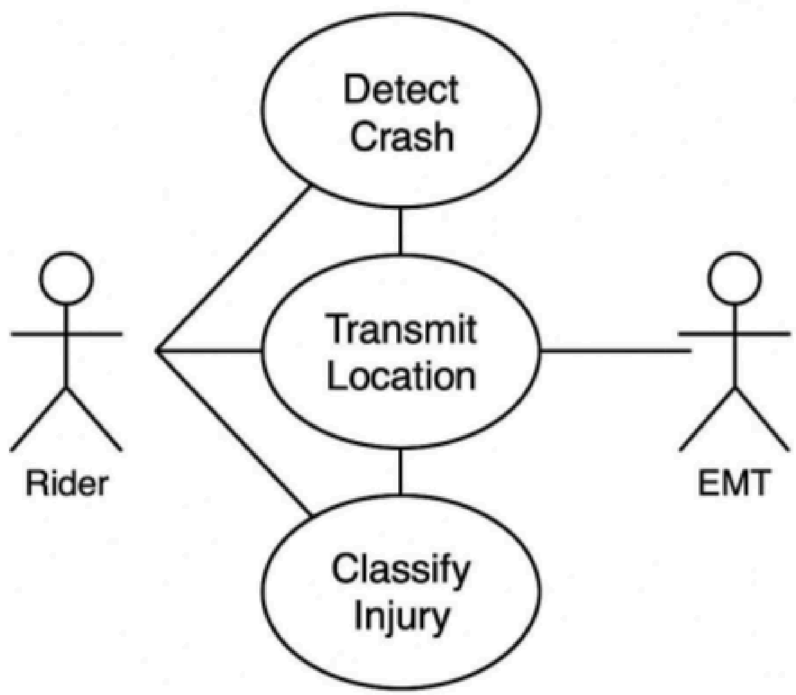


Fig 6.3: UML Diagrams

1. **Rider**: The individual wearing the smart helmet. This actor will interact with the system, like triggering alerts or using the device's features.

2. **EMT (Emergency Medical Technician)**: The emergency personnel who receives the alert and responds to the accident.
3. **System**: Represents the smart helmet system, which performs tasks like detecting crashes, transmitting location, and classifying injuries.

Use Cases:

1. **Detect Crash**: This use case represents the system detecting an impact or crash event. The system will use sensors like vibration sensors and force sensors to detect the event. The crash detection may trigger an alert and other processes.
2. **Transmit Location**: Once the crash is detected, the system transmits the **GPS location** to the EMT or the pre-configured emergency contacts. This can also involve sending other details like heart rate and oxygen levels.
3. **Classify Injury**: The system assesses the rider's physical condition based on data from the sensors (such as heart rate, oxygen levels, or even impact severity) and classifies the injury level. This could help EMTs prioritize their response.

CHAPTER 7: SYSTEM IMPLEMENTATION AND TESTING

7.1 Implementation

The system was implemented using a combination of hardware components and sensors to monitor a person's health status, detect accidents, and send immediate alerts.

7.1.1 key components

ESP32 Microcontroller:

- The **ESP32** microcontroller acts as the central hub in the system, managing all operations, data collection, and communication between the various sensors and external systems.
- It is responsible for **data acquisition** from sensors like the vibration sensor, Pulse Oximeter, and environmental sensors.
- The **ESP32** processes this data in real-time and makes decisions based on predefined algorithms. It uses these decisions to trigger actions such as sending alerts or updating the system with the current status of the rider.
- The microcontroller also handles **communication tasks** by sending alerts to emergency contacts, hospitals, or other systems (like GPS data or health metrics) through **wireless communication** (e.g., Wi-Fi or GSM).

• **GPS Module**:

- The **GPS module** continuously tracks the exact **geographic location** of the rider. It provides real-time updates on the **latitude** and **longitude** coordinates, allowing the system to pinpoint the rider's position accurately.
- This location data is essential for emergency services, ensuring they are directed to the correct accident site without delay. This feature is particularly crucial for incidents occurring in remote or rural areas where assistance might take longer to arrive.
- The GPS module provides precise information that helps **reduce the response time** and improve the effectiveness of rescue teams in locating the victim.

• **Vibration Sensor**:

- The **vibration sensor** plays a critical role in detecting sudden shocks or vibrations that may indicate an accident. It is sensitive enough to pick up the physical impact of events like **vehicle collisions**, **falls**, or any abnormal motion that occurs during a crash.
- Upon detecting an anomaly in movement, the vibration sensor sends a signal to the **ESP32**, triggering the system to activate emergency procedures.
- This sensor acts as the **first line of detection**, making it an essential component for accurate and prompt accident detection. The system's responsiveness depends on the effectiveness of this sensor in recognizing sudden impacts or abnormal motion patterns.

• **Pulse Oximeter (MAX30100)**:

- The **MAX30100 Pulse Oximeter** sensor monitors the **heart rate** (beats per minute) and **blood oxygen saturation (SpO2)** levels of the rider in real-time.
- These two key physiological parameters provide vital information about the rider's health status following an accident. A significant drop in **heart rate** or **SpO2 levels** may indicate a serious injury or a life-threatening condition, prompting the system to take immediate action.
- By continuously monitoring the rider's **cardiovascular** and **respiratory** status, the sensor helps evaluate the rider's physical condition after the crash and ensures that emergency responders are well-informed about the severity of the injuries.

• **Temperature and Humidity Sensors (DHT11, DHT22, AM2302)**:

- The **DHT series sensors** (DHT11, DHT22, AM2302) are used to measure the **ambient temperature** and **humidity** around the rider. These environmental conditions are important to monitor because they can affect the rider's health in the aftermath of an accident.
- If the temperature is too high or low, it could indicate shock, hyperthermia, or hypothermia. Extreme humidity can cause discomfort or even worsen the rider's physical state.
- Monitoring these environmental factors allows the system to assess **external stressors** that may influence the victim's condition and helps emergency responders to prepare appropriately for treatment.

• **Buzzer or Sound Module**:

- The **buzzer** or **sound module** serves as an **emergency alert** system, helping to attract attention immediately after an accident is detected. It emits a **loud, distinctive sound** when a crash or impact is detected, alerting people nearby to the emergency situation.
- The sound alerts **passersby, rescuers**, or other individuals who may be in proximity to the victim, encouraging them to provide assistance or inform emergency services.
- This feature ensures that the incident does not go unnoticed and prompts immediate action, especially in situations where the rider might be unconscious or unable to communicate.

• **Wireless Communication (e.g., Wi-Fi or GSM)**:

- **Wireless communication** is crucial for the real-time transmission of sensor data. The system relies on **Wi-Fi** or **GSM modules** to transmit collected information such as accident alerts, health data, and GPS coordinates.
- Once an accident is detected and verified, the system sends alerts, including the **victim's location**, **heart rate**, **SpO2 levels**, **temperature**, and **humidity** data, to the **nearest hospital** or emergency medical services.
- The system also communicates with **pre-configured emergency contacts**, such as the rider’s **parents or guardians**, notifying them of the incident and the rider's condition.
- The use of wireless communication ensures **real-time notifications**, enabling quicker responses from both medical personnel and family members. This significantly improves the chances of timely assistance and intervention, ultimately enhancing the survival rate in critical situations.

7.1.2 Detailed Process:

Step 1: Detect Crash

- **Sensors Used:** Vibration Sensor and Force Sensor (FSR).
- **Logic:**
    - Analog readings from the sensors are continuously monitored.
    - Thresholds are set (e.g., values below 500) to detect abnormal impact.
    - If vibration or force values drop suddenly (below threshold), the system assumes a potential crash.
- **Code Reference:**

```
if ((forceValue < 500) || (vibrationval < 500)) {
  Serial.println("Accident Detected");
  PostNotification(); // Triggers further
processing
}
```

Step 2: Assess Severity Using DNN (Deep Neural Network)

- **Inputs to DNN Model:**
    - Force of impact
    - Vibration intensity
    - Heart rate (BPM)
    - Blood oxygen (SpO_2)
    - Temperature
- **Process:**
    - These real-time sensor inputs are fed to a trained Deep Neural Network.
    - The model classifies the accident into categories (e.g., *mild*, *moderate*, *severe*).

- **Outcome:**
    - If **severe**, the system initiates an emergency alert.
    - If **mild**, it might notify only or wait for manual override.

Step 3: Capture GPS Location

- **GPS Module Used:** Based on TinyGPS++ library.
- **Process:**
    - Continuously listens to serial GPS data stream.
    - Validates the location signal.
    - Extracts latitude and longitude when available.
- **Code Reference:**

```
if (gps.location.isValid()) {
  latitude = gps.location.lat();
  longitude = gps.location.lng();
}
```

Step 4: Send Emergency Alert

- **Platform:** Blynk IoT Platform.
- **Trigger:** When accident is confirmed (based on sensor or DNN).
- **Actions:**
    - Sends real-time notification via Blynk.logEvent().
    - Sends sensor data (Temp, SpO_2, HR, GPS) to mobile app.
    - Visual pins (V0–V4) are updated with real-time values.

7.2 CODE:

```
#include "MAX30100_PulseOximeter.h"

#include <DHT.h>

#include<TinyGPS++.h>

//#include<LiquidCrystal_I2C.h>

#define DHTPIN 32

//#define BLYNK_TEMPLATE_ID "TMPL3vSbDnpao"

//#define BLYNK_TEMPLATE_NAME "SmartHelmate"

//#define BLYNK_AUTH_TOKEN "RC2qSwu2mVZZR8pfrx6YNa5YN1HKNXYU"

//#define BLYNK_TEMPLATE_ID "TMPL3HrAfWwkO"

//#define BLYNK_TEMPLATE_NAME "SmartHelmate"

//#define BLYNK_AUTH_TOKEN "njTRIUR7CCIMgvGsG4QVm-A36x-bDrFl"

#define BLYNK_TEMPLATE_ID "TMPL3ktmAl35z"

#define BLYNK_TEMPLATE_NAME "SmartHelmate"

#define BLYNK_AUTH_TOKEN "eLXvPy-SUnxOkGfoG16e3HnDPN3xiCJJ"

#define BLYNK_PRINT Serial

#ifdef ESP32

#include <WiFi.h>

#include <WiFiClient.h>

//===========================================================

char ssid[] = "ESP32";        // type your wifi name

char pass[] = "123456789";  // type your wifi password
```

```
String Message = "Accident detect";

//==============================================================

#define REPORTING_PERIOD_MS     1000

//LiquidCrystal_I2C lcd(0X27, 16, 2);

int buzzer = 27;

int vibration = 34;

int forceSensor = 35;

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

uint32_t tsLastReport = 0;

int count = 0;

char auth[] = BLYNK_AUTH_TOKEN;

PulseOximeter pox;

TinyGPSPlus gps;

unsigned long old = 0;

unsigned long current = 0;

int interval = 10;

int interval2 = 0;

int temp_val;

int BPM;

int SpO2;

int oneTimeFlag = 0;

double latitude, longitude;
```

```
String LAT, LONG;
void sendLocation() {
  LAT = "";
  LONG = "";
  if (gps.location.isValid()) {
    latitude = gps.location.lat();
    longitude = gps.location.lng();
    // Prin t to Serial Monitor
    Serial.print("Latitude: ");
    Serial.println(latitude, 6);
    Serial.print("Longitude: ");
    Serial.println(longitude, 6);
  } else {
    // Print to Serial Monitor
    Serial.println("Location not available.");
  }
}
void onBeatDetected()
{
  Serial.println("Beat!");
}
void PostData()
{
```

```
  Serial.println("Blynk notification Sent to Mobile");
  Blynk.virtualWrite(V0 , latitude);
  Blynk.virtualWrite(V1 , longitude);
  Blynk.virtualWrite(V2, SpO2);
  Blynk.virtualWrite(V3, BPM);
  Blynk.virtualWrite(V4, temp_val);
}
void PostNotification()
{
  while (Serial.available() > 0) {
    gps.encode(Serial.read());
  }
  //
  sendLocation();
  if (oneTimeFlag == 0)
  {
    String Buffer = "\r\nTemp:" + String(temp) + "\r\n";
    Buffer += "Lat:" + LAT + "\r\n";
    Buffer += "Long:" + LONG + "\r\n";
    Buffer += "SpO2:" + String(spo2) + "\r\n";
    Buffer += "HR:" + String(HR);
    current = millis();
```

```
    interval2 = (current - old) / 1000;
    Serial.println("Blynk notification Sent to Mobile");
    Blynk.logEvent("notification", ("Alert Accident Detected" + Buffer));
    //  Blynk.virtualWrite(V0 , Lat);
    //  Blynk.virtualWrite(V1 , Long);
    //  Blynk.virtualWrite(V2, spo2);
    //  Blynk.virtualWrite(V3, HR);
    //  Blynk.virtualWrite(V4, temp);
    oneTimeFlag = 1;
  }
}
void setup() {
  Serial.begin(9600);
  Serial.println("\nPlease wait for Blynk Server connection");
  //  lcd.init();
  //  lcd.backlight();
  //  lcd.clear();
  //  lcd.setCursor(0, 0);
  //  lcd.print("WEL-COME");
  pinMode(wifi_led, OUTPUT);
  //  pinMode(vibration, INPUT);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, pass);
```

```
  wifi_testing();
  Blynk.begin(auth, ssid, pass);
  pinMode(buzzer , OUTPUT);
  digitalWrite(buzzer , LOW);
  dht.begin(); // Initialize the PulseOximeter instance
  if (!pox.begin()) {
    Serial.println("FAILED");
    for (;;);
  } else {
    Serial.println("SUCCESS");
  }
  pox.setOnBeatDetectedCallback(onBeatDetected);
}
void loop() {
  pox.update();
  Blynk.run();
  int forceValue = analogRead(forceSensor);
  int vibrationval = analogRead(vibration);
   Serial.println(vibrationval);
   Serial.println(forceValue);
  if ((forceValue < 500) || (vibrationval < 500))
  {
    if (oneTimeFlag == 0)
```

```
  {
    Serial.println("Accident Detected");
    PostNotification();
  }
}
else
{
  if ((forceValue > 3000) && (vibrationval > 3000))
  {
    pox.update();
    oneTimeFlag = 0;
  }
  pox.update();
}
// wifi_testing();
if (millis() - tsLastReport > REPORTING_PERIOD_MS) {
  Serial.print("Heart rate:");
  Serial.print(pox.getHeartRate());
  Serial.print("bpm / SpO2:");
  Serial.print(pox.getSpO2());
  Serial.println("%");
  SpO2 = pox.getSpO2();
  BPM = pox.getHeartRate();
```

```
    temp_val = dht.readTemperature(false);
    //    lcd.clear();
    //    lcd.setCursor(0, 0);
    //    lcd.print("O: " + String(SpO2) + "% HB: " + String(BPM) + "/m");
    //    lcd.setCursor(0, 1);
    //    lcd.print("T: " + String(temp_val) + String((char)223) + "C");
    count++;
    if (count > 15)
    {
      PostData();
      count = 0;
    }
    tsLastReport = millis();
  }
}
void wifi_testing() {
  while (WiFi.status() != WL_CONNECTED) {
    delay(250);
    Serial.print(".");
    digitalWrite(wifi_led, LOW);
    delay(250);
    digitalWrite(wifi_led, HIGH);
    current = millis();
```

```
    if ((current - old) / 1000 > interval)
      ESP.restart();
  }
}
```

7.3 Testing:

- Each sensor was individually tested for accuracy and reliability.
- Multiple accident simulations (like jerks or vibrations) were conducted to ensure the vibration sensor triggers accurately.
- Data collected from the sensors was checked for consistency and correctness (e.g., heart rate, temperature).
- The GPS was tested for real-time tracking and accuracy of coordinates.
- The entire system was tested as a whole to ensure that upon detecting an accident:
    - A sound is activated.
    - All information (location, health data) is collected and sent correctly to the nearest hospital and guardians.

7.4 Results of Testing:

- The system responded correctly in all test scenarios.
- Sensors delivered reliable and accurate data.
- Information was successfully transmitted in real time.
- Alert mechanism worked as expected, ensuring quick response.

CHAPTER 8: RESULTS AND DISCUSSIONS

The developed system integrates various sensors and hardware components to build an effective accident detection and monitoring solution. The performance of each component was thoroughly tested, and the system showed consistent and reliable results across different test scenarios.

The vibration sensor successfully detected simulated accident scenarios, such as sudden impacts and abrupt movements. It served as the primary trigger to initiate the alert process. The GPS module consistently provided accurate real-time location coordinates, 49 which are essential for locating the victim and guiding emergency services.

The ESP32 microcontroller played a central role by handling inputs from all connected sensors, processing the data, and managing communication with external systems. It ensured seamless coordination between the components and reliable transmission of the collected information.

The Pulse Oximeter (MAX30100) effectively monitored the person's 32 vital signs, specifically heart rate and oxygen saturation (SpO2). 23 These parameters are important indicators of the victim's health status after an accident. Any abnormal values 21 can help medical professionals assess the severity of the situation.

The environmental sensors—DHT11, DHT22, and AM2302—provided accurate readings of temperature and humidity. These readings are important in outdoor or hazardous environments where 51 extreme weather conditions could affect the victim's well-being or influence the type of emergency response required.

Upon detecting an accident, the system activated a buzzer or sound alert to draw attention from people nearby, potentially leading to quicker local assistance. Simultaneously, all important data—such as the GPS location, heart rate, SpO2, temperature, and humidity—was transmitted to the nearest hospital and to the guardian or emergency contact of the individual involved.

This real-time response capability ensures that accurate and timely information reaches the right people, allowing emergency services to respond faster and more effectively. It also provides peace of mind to family members, knowing they will be immediately informed in case of an emergency.

The system's consistent performance under test conditions confirms its practicality and effectiveness. It is especially useful in scenarios such as road accidents, outdoor activities, or health monitoring for the elderly or individuals with medical conditions.

Key Points Discussed:

- The vibration sensor serves as a reliable accident detection mechanism.
- Real-time GPS tracking enhances emergency location accuracy.
- Continuous health monitoring using pulse and SpO2 improves medical response.
- Temperature and humidity data provide environmental awareness.
- Immediate alerts and automatic data sharing help in quick decision-making.

Conclusion

In this project, we have developed a comprehensive and innovative solution by integrating a range of sensors, including the GPS module, ESP32 microcontroller, Pulse Oximeter MAX30100, DHT series sensors (DHT11, DHT22, AM2302), and a vibration sensor. This combination of technologies forms the backbone of our smart accident detection and health monitoring system, designed to detect accidents, monitor the health status of the rider, and ensure rapid response during emergencies.

The **vibration sensor** plays a crucial role in detecting the occurrence of an accident. It is sensitive to sudden movements and shocks, which are indicative of an impact, allowing the system to identify when a crash has occurred. Once an accident is detected, the system is immediately triggered to take further actions to safeguard the victim's well-being.

To enhance the accuracy of the system, the **GPS module** is integrated to track the precise location of the accident. This real-time location data ensures that emergency services can be directed to the exact site without delay, a critical feature in ensuring that help arrives quickly. The GPS also allows the system to share the victim's location with their family members or guardians, keeping them informed and enabling them to provide necessary support.

The **MAX30100 Pulse Oximeter** sensor is used to monitor the **heart rate** and **SpO2 levels** (blood oxygen saturation) of the rider. These vital signs are crucial in assessing the rider's condition after an accident. A sudden drop in heart rate or SpO2 levels can indicate serious injuries, requiring immediate medical attention. The MAX30100 sensor continuously monitors these parameters and helps to identify the severity of the rider's condition, ensuring a more accurate and timely response.

Additionally, the **DHT series sensors**, including the **DHT11, DHT22, and AM2302**, are utilized to monitor the **temperature** and **humidity** around the rider. These environmental factors are important in assessing the health risks the rider may be facing after an accident. For example, a sudden change in temperature or extreme humidity levels could indicate shock, hypothermia, or other conditions that require immediate medical intervention.

Once the accident is confirmed, an **alert sound** is triggered by the system, providing an immediate audible warning to those nearby. This alert helps to raise awareness of the accident and potentially summon help from passersby if emergency services are not immediately reachable. At the same time, the system sends critical information—including the victim's location, health parameters (heart rate, SpO2, temperature), and any other relevant data—to emergency services, ensuring that medical responders are well-informed before they even reach the site.

The system also sends an **emergency alert** to the **victim's parents or guardians**, along with the same critical data. This notification keeps family members informed about the situation, reducing their anxiety and helping them make quick decisions regarding the next steps. The ability to notify both emergency services and family members in real time can significantly reduce response times and improve the chances of a positive outcome for the victim.

The core objective of this project is to ensure **timely assistance** in case of an accident, which directly enhances the chances of saving lives. By integrating multiple sensors and technologies into a single, cohesive system, this solution provides real-time health monitoring, precise location tracking, and rapid emergency notifications. This project represents a step forward in improving road safety and ensuring that timely medical assistance is provided when every second counts. The combination of **IoT and health monitoring** in this system has the potential to revolutionize the way motorcycle accidents are managed, ultimately saving lives and reducing the severity of injuries.

References

1. Helmet Guard: Engine Management and Accident Severity Assessment for Motorcycles SR Vanga, N Angadala… - 2024 15th International …, 2024 - ieeexplore.ieee.org
2. A vehicle occupant injury prediction algorithm based on road crash and emergency medical data
   A vehicle occupant injury prediction algorithm based on road crash and emergency medical data - ScienceDirect
   Gestures Publisher: ScienceDirect. Authors: Tetsuya Nishimoto, Kazuhiro Kubota, Giulio Ponte
   Date Added to ScienceDirect: Oct 2024.
3. Real-time **accident** detection and physiological signal monitoring to enhance **motorbike** safety and emergency response SM Siam, KI Sumaiya, MR Al-Amin, TH Turjo… - arXiv preprint arXiv …, 2024 - arxiv.org
4. IoT-Based Smart Accident Detection and Alert System CVS Babu, NS Akshayah, R Janapriyan - … Cloud-Based Industrial IoT, 2023 - igi-global.com
5. **IoT enabled HELMET** to safeguard the health of mine workers N **Singh**, VK Gunjan, G Chaudhary, **R** Kaluri… - Computer …, 2022 - Elsevier
6. Discovering **injury** **severity** risk factors in automobile crashes: a hybrid explainable **AI** framework **for** decision support
   M Amini, A Bagheri, D Delen - Reliability Engineering & System Safety, 2022 - Elsevier
7. IoT-based accident detection and emergency alert system for motorbikes
   SU Rehman, SA Khan, A Arif… - … Mechatronics Systems …, 2021 - ieeexplore.ieee.org
8. For papers – Design and Implementation of a Smart Bike Accident Detection System
   (PDF) Design and Implementation of a Smart Bike Accident Detection System Gestures Publisher: IEEE. Authors: Anika Saida, Fahim Siam, A.E M Ridwan Date Added to IEEE Xplore: June 2020
9. An IoT-based vehicle accident detection and classification system using sensor fusion N Kumar, D Acharya, D Lohani - IEEE Internet of Things …, 2020 - ieeexplore.ieee.org
10. IoT based car accident detection and notification algorithm for general road accidents. S Sharma, S Sebastian - International Journal of Electrical & Computer …, 2019 - core.ac.uk
11. Risk **assessment** of occupational **injuries using Accident Severity** Grade N Azadeh-Fard, A Schuh, E Rashedi, JA Camelio - Safety science, 2015 – Elsevier

12. A vehicle occupant injury prediction algorithm based on road crash and emergency medical data

A vehicle occupant injury prediction algorithm based on road crash and emergency medical data - ScienceDirect

Gestures Publisher: ScienceDirect. Authors: Tetsuya Nishimoto, Kazuhiro Kubota, Giulio Ponte

Date Added to ScienceDirect: Oct 2024

First International Conference on Multi-Disciplinary Research Trends in Engineering & Technology