International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

Accident Detection & Patient Health Monitoring System Using ESP32 & IoT

Nagesh H B
Assistant Professor Department of Electronics and Communications
ACS College of Engineering,
Bangalore,India
durgaindira@acsce.edu.in

Gurudatta MR
UG Scholar
Department of Electronics and Communication
ACS College of Engineering,
Bangalore,India
gurudatta2111@gmail.com

Navneet Kishore Pandey
UG Scholar
Department of Electronics and Communication
ACS College of Engineering,
Bangalore,India
navneetkishorepandey186@gmail.com

Sanjay R
UG Scholar
Department of Electronics and Communication
ACS College of Engineering,
Bangalore,India
sanjay12sam2004@gmail.com

Shreyas L
UG Scholar
Department of Electronics and Communication
ACS College of Engineering,
Bangalore,India
shreyushreyas2020@gmail.com

Abstract*— Road accidents remain a serious global issue, often leading to fatalities due to delayed emergency response. The project presents an IoT-based Accident Detection and Patient Health Monitoring System using the ESP32 microcontroller. The system automatically detects accidents using an accelerometer and monitors vital health parameters such as heart rate, body temperature, and SpO_2. Upon detecting an accident, the system retrieves the victim is location through a GPS module and sends real-time alerts to emergency contacts via Telegram. An LCD displays live health data, while a manual override reduces false alerts. The proposed system offers a reliable, low-cost solution to improve emergency response and enhance road safety.

Keywords*—Arduino UNO, IR sensor, Accelerometer, Zigbee, Rain Sensor, Buzzer etc.

I. INTRODUCTION

Road transportation has become an essential part of modern life, enabling millions of people to travel efficiently for work, education, and daily activities. However, with the continuous rise in the number of vehicles on highways and city roads, the rate of road accidents has also increased significantly. These accidents often lead to severe injuries, long-term disabilities, and in many cases, loss of life. One of the major reasons for high fatality rates is the delay in providing medical attention during the crucial "golden hour" where quick treatment can make the difference between survival and death. In most accident scenarios, victims are unable to call for help due to unconsciousness or physical trauma, and bystanders may not always be present to report the incident. This results in delayed response time and poorly coordinated rescue efforts. Traditional systems used for accident reporting rely heavily on manual communication, which is slow, unreliable, and dependent on human presence. Although high-end vehicles have embedded safety systems, these solutions are expensive and not accessible to the majority of users. Smartphone-based accident detection apps also suffer from limitations such as battery dependency, hardware inaccuracy, and the possibility that the device gets damaged during impact. Therefore, there is a strong need for an automated, low-cost, and reliable accident detection system that can operate independently and respond instantly.

II. LITERATURE SURVEY

Fatal road accidents can be easily avoided by understanding the psychological state of a driver. Majority of road accidents occur during night driving due to the drowsiness state of a vehicle driver. The paper provides mechanism to reduce accidents to a large extend by monitoring eye blinking of the driver which indicates the drowsiness, obstacles located in the road and the drunken state of the drivers. Automatic pre cautionary system is activated based on the above alarming condition. Accident and its probable location are also generated at the nearby police station that helps initiating medical help. In normal cases no medical help is received due to the non-availability of accident information. This happens mainly at night and in roads where the traffic is low.

The researchers used data from the Fatality Analysis Reporting System (FARS) of the National Highway Traffic Safety Administration (NHTSA). This database contains detailed records of every fatal road accident in the U.S. The authors studied the data from to understand how different vehicle-related conditions influence the severity of accidents. The results show that vehicles with poor mechanical health or malfunctioning safety features are much more likely to be involved in deadly crashes. By identifying these components, the study highlights the importance of regular maintenance, effective safety inspections, and better vehicle design standards.

The authors developed KD-CVIS, an image dataset including various accident scenarios. A YOLO-CA convolutional network is used for detecting collision events by analyzing

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

camera feeds. To improve accuracy and speed of accident detection using advanced image processing and deep learning models.: Intelligent roadside cameras, YOLO-CA neural network, Multiscale Feature Fusion (MSFF), Dynamic loss-weight adjustment. The model achieved a high detection speed of 21.6 FPS and an average precision of 90%, showing promising results for smart-city integration. Requires advanced infrastructure, high-cost cameras, and IoT gateways — not suitable for low-cost vehicle-mounted systems like ESP32-based solutions.

III. PROPOSED SYSTEM

.

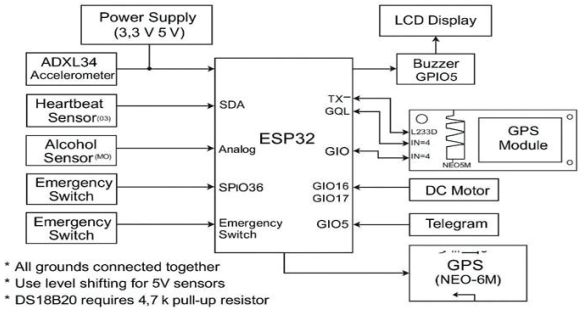

Fig 3.1: Proposed System

This block diagram represents an accident detection and health-monitoring system built using an ESP32 microcontroller. Various sensors such as the ADXL accelerometer, heartbeat sensor, alcohol sensor, temperature sensor, and an emergency switch are connected to the ESP32 to continuously monitor vehicle movement and the driver's physical condition. The ESP32 receives all sensor readings, processes the data, and decides whether an accident has occurred. It also communicates with the GPS module, which provides exact location details. Output devices like the LCD display, buzzer, H bridge, and DC motor are controlled by the ESP32 for visual alerts, sound warnings, and motor control in emergency situations. Finally, the ESP32 uses its Wi-Fi capability to send accident alerts and location data through Telegram, ensuring quick notification to emergency contacts.

IV. HARDWARE REQUIREMENTS

ESP32: The ESP32 is a powerful, low-cost microcontroller developed by Espressif Systems, widely used in IoT (Internet of Things) applications because of its high processing speed, built-in communication features, and rich set of peripherals. It is an upgraded version of the ESP8266 and includes a dual-core Tensilica processor, which allows it to handle multiple tasks simultaneously, making it suitable for real-time monitoring systems like accident detection and health monitoring. The ESP32 supports both 3.3V logic and offers excellent power-saving modes, making it suitable for battery-operated systems. It also has built-in timers, watchdogs, touch sensors, hall sensors, and a hardware encryption engine, which ensures secure communication. The device can run Arduino code, Micro Python, or the native ESP-IDF framework, giving developers flexibility in programming. This fig shown in figure 4.1.

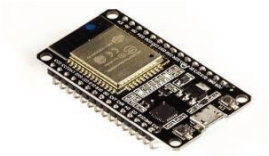

Fig 4.1: ESP32

DC Motor: DC motors convert electrical into mechanical energy and they consist of permanent magnets and loops of wire inside, when current is applied, the wire loops generate a magnetic field, which reacts against the outside field of the static magnets. The interaction of the fields produces the movement of the shaft/armature. Thus, electromagnetic energy becomes motion. A DC Motor is shown in fig 4.2.

Fig 4.2: DC Motor

Buzzer: A buzzer is a simple sound-producing device used in electronic systems to give audio alerts or warnings. It converts electrical signals into sound using either a piezoelectric element or an electromagnetic coil. In your project, the buzzer is connected to a GPIO pin of the ESP32 and is activated whenever an accident is detected or when the emergency alert countdown starts. When the ESP32 sends a HIGH or LOW signal to the buzzer, it produces a beeping sound to alert the user or nearby people. Buzzers are commonly used because they are small, require very little power, and give immediate audible feedback in emergency situations. Buzzer is shown in fig 4.3.

Fig 4.3: Buzzer

Battery: A battery is a device consisting of one or more electrochemical cells with external connections provided to power electrical devices such as flashlights, smartphones, and electric cars. When a battery is supplying electric power, its positive terminal is the cathode and its negative terminal is the anode. The terminal marked negative is the source of electrons that will flow through an external electric circuit to the positive terminal. When a battery is connected to an external electric load, a redox reaction converts high-energy reactants to lower-energy products, thin cells used in smartphones, to large lead acid batteries or lithium-ion batteries in vehicles, and at the large battery banks the size of rooms that provide

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

standby or emergency power for telephone exchanges and computer datacenters. Battery is shown in fig 4.4.

Fig 4.4: Battery

LCD Display: A liquid-crystal display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals. Liquid crystals do not emit light directly, instead using a backlight or reflector to produce images in color or monochrome. LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images with low information content, which can be displayed or hidden, such as preset words, digits, and 7-segment displays, as in a digital clock. They use the same basic technology, except that arbitrary images are made up of a large number of small pixels, while other displays have larger elements. The figure 4.5 shows the LCD Display.

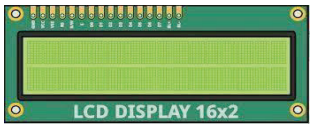

Fig 4.5: LCD Display

Pulse Sensor: Pulse Sensor is a well-designed plug-and-play heart-rate sensor for Arduino. It can be used by students, artists, athletes, makers, and game & mobile developers who want to easily incorporate live heart rate data into their projects. The sensor clips onto a fingertip or earlobe and plugs right into Arduino. The fig 4.6 shows the Pulse Rate Sensor.

Fig 4.6: Pulse Sensor

Temperature Sensor: The digital temperature sensor like DS18B20 follows single wire protocol and it can be used to measure temperature in the range of -67oF to +257oF or - 55oC to +125oC with +-5% accuracy. The range of received data from the 1-wire can range from 9-bit to 12- bit. Because, this sensor follows the single wire protocol, and the controlling of this can be done through an only pin of Microcontroller. This article discusses an overview of a DS18B20 temperature sensor.

Fig 4.7: Temperature Sensor

Alcohol Sensor: An alcohol sensor detects the attentiveness of alcohol gas in the air and an analog voltage is an output reading. The sensor can activate at temperatures ranging from -10 to 50° C with a power supply is less than 150 Ma to 5V. The sensing range is from 0.04 mg/L to 4 mg/L, which is suitable for breathalyzers. The passive alcohol sensor (PAS) is a device developed to assist police in identifying drinking drivers. The PAS draws in mixed expired and environmental air from in front of the subject's face and passes it into a fuel cell sensor that can detect very small amounts of alcohol. The fig 4.10 shows Alcohol Sensor detection.

Fig 4.8: Alcohol Sensor

Push Button: A Pushbutton Switch is a switch designed so that its contacts are opened and closed by depressing and releasing a pushbutton on the Switch in the direction of its axis. Used to arise the emergency condition.

Fig 4.9: Push Button

ADXL Sensor (Accelerometer): An accelerometer is a tool that measures the vibration, motion, or acceleration of a structure. Cameras and smartphones these days use an accelerometer consisting of an axis-based motion sensor. It is an electromechanical device that measures either static or dynamic acceleration. Acceleration, as we know, is the measure of change in velocity upon a given time.

Fig 4.10: ADXL Sensor

V. SOFTWARE REQUIREMENTS Arduino IDE:

Fig 5.1: Arduino IDE

A program for Arduino may be written in any programming language for a compiler that produces binary machine code for the target processor. Atmel provides a development environment for their microcontrollers, AVR Studio and the newer Atmel Studio. The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

language Java. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. A program written with the IDE for Arduino is called a sketch. Sketches are saved on the development computer as text files with the file extension. ino. Arduino Software (IDE) pre-1.0 saved sketches with the extension. pde. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution.

A minimal Arduino C/C++ sketch, as seen by the Arduino IDE programmer, consist of only two functions: setup(): This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch. loop(): After setup() has been called, function loop() is executed repeatedly in the main program. It controls the board until the board is powered off or is reset

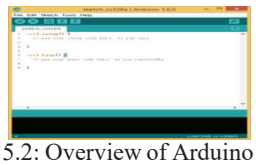

Fig 5.2: Overview of Arduino IDE

Embedded C:

• Embedded C is a specialized version of the C programming language designed specifically for programming embedded systems, such as microcontrollers and IoT boards. It follows the same basic syntax and structure as standard C but includes additional features that allow direct interaction with hardware components. In your project, Embedded C plays the central role of controlling how the ESP32 communicates with sensors, processes data, and performs real-time actions during an accident.

• One of the main strengths of Embedded C is its ability to work very close to the hardware. This means the programmer can directly access pins, ports, timers, ADC channels, communication protocols, and internal registers of the microcontroller. This ability is essential in your project because the ESP32 needs to read analog signals from the heartbeat sensor, take digital readings from the accelerometer, read one-wire data from the temperature sensor, and receive GPS coordinates through UART communication. Each of these tasks requires precise hardware-level control, which Embedded C provides.

• Embedded C also supports real-time operation, which is extremely important for life-saving systems like accident detection. When an accident occurs, the ESP32 must quickly detect the sudden change in acceleration, start a countdown, trigger the buzzer, fetch GPS location, and send an emergency alert through Telegram. These tasks must happen within seconds, and the timing accuracy provided by Embedded C enables the system to respond immediately without delays.

• Another important feature of Embedded C is its portability. Code written in C can easily be adapted to different microcontrollers with minor changes. In your case, although you are using the ESP32, the same Embedded C logic can be reused for Arduino, STM32, AT mega, or PIC microcontrollers. This makes it easier to upgrade or extend your project in the future.

• Embedded C allows modular programming, meaning the program can be divided into separate functions such as reading sensors, processing data, sending messages, and handling communication. This modularity makes your project easier to understand, debug, and maintain. Each function can be tested independently, ensuring more reliable operation.

• The ESP32 also supports multiple communication protocols such as I²C, SPI, UART, Wi-Fi, and 1-Wire. Embedded C makes it possible to implement these protocols through libraries and direct register manipulation. For example, I²C is used for the accelerometer and LCD, UART is used for GPS, analog input is used for the alcohol and pulse sensor, and Wi-Fi is used for Telegram alerts. All these communication tasks are handled through Embedded C programming.

• Memory management is another critical aspect of Embedded C. Embedded systems often have limited RAM and flash memory, so the programmer must write efficient code. The ESP32 has more memory than many microcontrollers, but still, proper management ensures your program runs smoothly, especially when handling Wi-Fi operations and multiple sensor readings simultaneously.

VI. RESULT & DISCUSSIONS

Result:

The developed accident detection and health monitoring system successfully performed all major functions during testing. The ESP32 effectively collected real-time data from the accelerometer, heartbeat sensor, temperature sensor, alcohol sensor, and GPS module. The accident detection mechanism worked reliably, identifying sudden impacts and abnormal movements with good accuracy. The buzzer alert and countdown timer helped avoid false alarms by giving the user time to cancel unnecessary triggers. If no cancellation occurred, the system confirmed an accident and collected accurate GPS coordinates.

The Telegram bot integration worked smoothly for emergency communication. Whenever an accident was detected, the ESP32 immediately formatted the alert message and transmitted it over Wi-Fi. The received message contained the user's location, heartbeat, temperature, alcohol level, and system status, proving that data packaging was successful. The GPS coordinates appeared correctly as a clickable Google Maps link, enabling fast identification of the accident spot. The message delivery time was short, taking only a few seconds, which supports quick emergency response. The system continued working even after repeated tests, showing consistent performance.

The hardware components also performed effectively under continuous operation. The motor driver responded correctly to ESP32 signals, confirming compatibility with additional

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

safety functionalities. The alcohol sensor showed proper detection sensitivity, and the temperature and heartbeat sensors produced stable readings throughout the test. Power consumption remained within expected limits, and the connections were stable without data loss. The complete system demonstrated smooth integration of sensing, processing, communication, and alerting units.

Hardware Prototype:

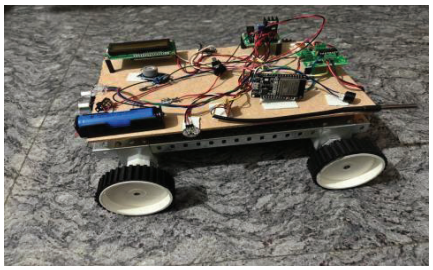

Fig 6.1: Hardware Prototype Of The Project

The figure 6.1 shows the Hardware prototype of the project without power Supply and the connection of Hardware components and the required program code is dumped into the ESP32.

Telegram Message Alert:

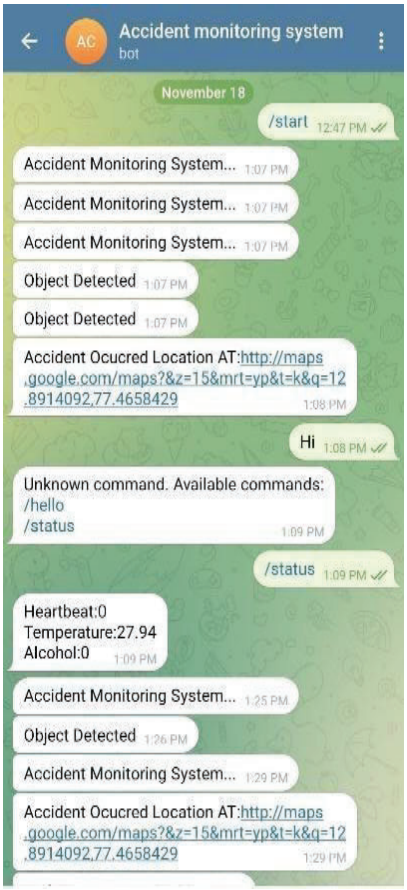

Fig 6.2: Telegram Message Alert

Sensor Readings:

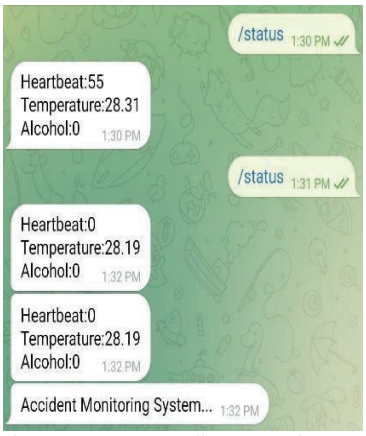

Fig 6.3: Sensor Readings In Telegram

APPLICATIONS
- Vehicle Accident Detection
- Emergency Alert Systems
- Patient Health Monitoring
- Smart Transportation Safety
- Ambulance Tracking Systems
- Fleet Management
- Public Transport Safety
- Rider Safety Systems
- IoT-Based Medical Assistance
- Remote Health Reporting

ADVANTAGES
- Real-time Accident Detection
- Automatic Emergency Alerting
- Accurate Location Tracking
- Continuous Health Monitoring
- False Alarm Prevention
- Low-Cost and Scalable
- Easy IoT Integration
- Improves Survival Chances
- User-Friendly and Portable
- High Reliability

LIMITATIONS
- Dependence on Internet Connectivity
- GPS Inaccuracy in Indoor or Covered Areas

VII. CONCLUSION

The Accident Detection and Patient Health Monitoring System using IoT and ESP32 provides an efficient, low-cost, and scalable solution for improving road safety and

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

emergency medical response. By automatically detecting accidents, retrieving GPS coordinates, monitoring vital signs, and transmitting real-time data through Telegram, the system significantly reduces the delay between the accident occurrence and medical assistance. The system operates autonomously without requiring the victim or bystanders to manually report the incident, making it especially useful in remote regions or night-time travel. Continuous health monitoring gives hospitals critical patient information ahead of time, enabling faster diagnosis and treatment. The modular design allows for future expansions such as cloud data storage, GSM communication, additional sensors (ECG, BP), ambulance dispatch automation, and smart-city integration. Overall, the project successfully demonstrates how IoT, sensor technologies, and embedded systems can be combined to create a reliable solution that enhances emergency responsiveness and potentially saves countless lives.

VIII. REFERENCES

[1] Abhirup Das et al., "Vehicle Accident Prevention and Location Monitoring System," International Journal of Emerging Engineering Research and Technology.

[2] Tariq Jamil et al., "Design and Implementation of Eye-Blink Detector System for Automobile Accident Prevention," Journal of Electronics and Safety Systems.

[3] Tang Youming et al., "PCA Analysis of Fatal Road Accidents Based on Vehicle Condition Factors," Traffic Safety and Analysis Reports.

[4] Daxin Tian et al., "Automatic Car Accident Detection Using Cooperative Vehicle Infrastructure Systems," International Conference on Intelligent Transportation.

[5] Omar Kassem Khalil, "Road Accident Analysis Using Telematics," Abu Dhabi Safety Research Review.

[6] Rowley, J. et al., "Analysis of Human-Caused Vehicle Accidents and Autonomous Communication Systems," Automated Mobility Journal.

[7] ESP32 Technical Documentation, Espressif Systems.

[8] MAX30100 Pulse Oximeter Sensor Datasheet.

[9] Arduino IDE Reference Documentation.

[10] GPS NEO-6M Module User Manual.

[11] Kiran Sawant, Imran Bhole, Prashant Kokane, Piraji Doiphode, Yogesh Thorat, "Accident Alert and Vehicle Tracking System," ISSN 2348-120X, Vol. 3, Issue 4, pp. 259–263, Oct–Dec 2015.

[12] Benjamin Coifman, "A Real-Time Computer Vision System for Vehicle Tracking and Traffic Surveillance," Transportation Research Part C, 2003.

[13] R. Ramani, S. Valar Mathy, N. Suthanthira Vanitha, S. Selvaraju, M. Thirupathi, R. Thangam, "Vehicle Tracking and Locking System Based on GSM and GPS," I.J. Intelligent Systems and Applications, 2013.

[14] Kunal Maurya, Mandeep Singh, Neelu Jain, "Real-Time Vehicle Tracking System Using GSM and GPS Technology—An Anti-Theft Tracking System," International Journal of Electronics and Computer Science Engineering, ISSN 2277-1956, V1N3, pp. 1103–1107. [15] V. Ramya, B. Palaniappan, K. Karthick, "Embedded Controller for Vehicle In-Front Obstacle Detection and Cabin Safety Alert System," International Journal of Computer Science & Information Technology (IJCSIT), Vol. 4, No. 2, April 2012.