A Real Time Research Project Report

on

ACCIDENT ALARMING SYSTEM

Submitted in partial fulfilment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING – (AI&ML)

Submitted By

K.SAI MANASA	**(228R1A6630)**
K.SWATHI POORNIMA	**(228R1A6633)**
M.RISHIKA SHIVANI	**(228R1A6637)**
G.BHARGAVA SAI	**(228R1A6625)**

Under the Esteemed guidance of

Mr. G. Venkateswarlu

Assistant Professor, Department of CSE (AI & ML)

Department of Computer Science & Engineering (AI&ML)

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Approved by AICTE, NEW DELHI, Affiliated to JNTU, Hyderabad, Kandlakoya,
Medchal Road, R.R. Dist. Hyderabad-501 401)

2023-2024

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Approved by AICTE, Affiliated to JNTUH, Accredited by NBA, NAAC)

Kandlakoya (v), Medchal, Telangana.

Department of Computer Science & Engineering (AI & ML)

CERTIFICATE

This is to certify that the project entitled **"ACCIDENT ALARMING SYSTEM"** is a bonafide work carried out by

M.RISHIKA SHIVANI	**(228R1A6637)**
K.SAI MANASA	**(228R1A6630)**
K.SWATHI POORNIMA	**(228R1A6633)**
G.BHARGAVA SAI	**(228R1A6625)**

in partial fulfilment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING (AI&ML)** from CMR Engineering College, under our guidance and supervision. The results presented in this project have been verified and are found to be satisfactory. The results embodied in this project have not been submitted to any other university for the award of any other degree or diploma.

Internal Guide	**Head of the Department**
Mr.G. Venkateswarlu	**Dr. M. Kumara Swamy**
Assistant Professor	Professor & HOD
CSE (AI&ML)	CSE (AI&ML)

Project Coordinator
Ms E. Parvathi
Assistant Professor, ECE, CMREC

ACKNOWLEDGEMENT

We sincerely thank the management of our college **CMR ENGINEERING COLLEGE** for providing during our project work.

We derive great pleasure in expressing our sincere gratitude to our principal **Dr**. **A.SRINIVASULA REDDY** for his timely suggestions, which helped us to complete the project work successfully.

It is the very auspicious moment we would like to express our gratitude to **Dr. M.KUMARASWAMY** , Head of the Department, CSE-AIML for his consistent encouragement during the progress of this project.

We take it as a privilege to thank our project coordinator **Mrs. E.PARVATHI.** Assistant professor, Department of ECE for his ideas that led to complete the project work and we also thank his continuous guidance, support and unfailing patience, throughout the course of this work.

We sincerely thank our project internal guide **Mr. G. VENKATESHWARLU,** Assistant professor, Department of CSE-AI&ML for his guidance and encouragement in carrying out this project.

.

DECLARATION

This is to certify that the work reported in the present Major project entitled **ACCIDENT ALARMING SYSTEM** is a record of bonafide work done by us inthe Department of Computer Science and Engineering (AI&ML) , CMR Engineering College. The reports are based on the project work done entirely by us and not copied from any other source. We submit our project for further development by any interested students who share similar interests to improve the project in the future.

The results embodied in this Major project report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

M.RISHIKA SHIVANI	**(228R1A6637)**
K.SAI MANASA	**(228R1A6630)**
K.SWATHI POORNIMA	**(228R1A6633)**
G.BHARGAVA SAI	**(228R1A6625)**

CONTENTS

CHAPTER-8 CONCLUSION

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

In recent years, the rapid increase in the number of vehicles on the road has led to a corresponding rise in traffic accidents, which often result in significant human and economic losses. To address this critical issue, an Accident Alarming System (AAS) has been developed, aiming to enhance road safety by promptly detecting accidents and alerting emergency services and nearby vehicles. The AAS leverages a combination of advanced sensors, GPS technology, and communication networks to provide a comprehensive solution for accident detection and notification.

The system is equipped with accelerometers, gyroscopes, and other sensors that continuously monitor the vehicle's dynamics and detect abnormal patterns indicative of an accident, such as sudden deceleration, impact, or rollover. Upon detecting such an event, the system automatically transmits the vehicle's location, the severity of the accident, and other relevant data to a central server. This information is then relayed to emergency response teams, enabling them to reach the accident site promptly and provide timely assistance.

Additionally, the AAS integrates with vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication technologies, ensuring that nearby drivers are alerted to potential hazards ahead, thereby reducing the risk of secondary accidents. The system also includes a manual alert feature, allowing drivers to manually trigger an alarm in case of a medical emergency or other critical situations.

This Accident Alarming System offers a robust framework for enhancing road safety, reducing response times for emergency services, and ultimately saving lives. By leveraging cutting-edge technologies and fostering real-time communication between vehicles and infrastructure, the AAS represents a significant step forward in the quest for safer roads.

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

The Accident Alarming System is designed to improve road safety by promptly detecting accidents and alerting emergency services through the integration of Arduino, GPS, and GSM modules. This system leverages the capabilities of these technologies to ensure quick response times, thereby mitigating the severity of accidents and potentially saving lives.

At the heart of the system is the Arduino microcontroller, which serves as the central processing unit. The Arduino integrates data from various sensors and modules, including the GPS and GSM modules. The GPS module tracks the vehicle's real-time location, providing accurate coordinates that are crucial in the event of an accident. These coordinates are sent to the Arduino for processing and inclusion in the alert messages. The GSM module is responsible for communication, enabling the system to send SMS alerts containing the location and details of the accident to pre-configured emergency contacts and services.

Accident detection is primarily handled by an accelerometer sensor, which continuously monitors the vehicle's motion. In the event of a sudden change in motion, such as abrupt deceleration or impact, the sensor signals the Arduino, triggering the accident detection process. The Arduino then requests the current location from the GPS module, processes the data, and generates an alert message. This message includes essential details such as the accident's coordinates and the time of occurrence, and is sent out via the GSM module to ensure that help is dispatched promptly.

The system's automated alerts minimize the need for human intervention, which is crucial in severe accidents where the driver may be incapacitated. The inclusion of precise GPS coordinates ensures that emergency services can quickly locate the accident site, significantly reducing the time taken for rescuers to arrive. The system's design focuses on cost-effectiveness and ease of implementation, utilizing readily available components to make it affordable for widespread use.

In conclusion, the Accident Alarming System using Arduino, GPS, and GSM modules offers a robust and effective solution for enhancing road safety. By automating the detection and notification process, the system not only improves emergency response times but also increases the chances of survival and recovery for accident victims. This innovative approach represents a significant advancement in the quest for safer roads.

1.2 OBJECTIVE OF THE PROJECT

- **Innovation**: Designing a Bluetooth-controlled shopping cart is a novel idea that can showcase your innovation and creativity. You'll be at the forefront of integrating technology into everyday tasks.
- **Convenience**: Imagine the convenience of being able to control your shopping cart remotely with your smartphone. This innovation could potentially make shopping easier and more enjoyable for many people, especially those with mobility issues or parents with young children.
- **Learning Experience**: Building a Bluetooth-controlled shopping cart provides an excellent opportunity to learn new skills. You'll gain experience in electronics, programming, and mechanical design, among other things. This project can help you develop expertise that's valuable in various fields, including engineering and product development.
- **Problem-Solving:** Overcoming the challenges involved in creating a Bluetooth-controlled shopping cart will sharpen your problem-solving skills. You'll encounter obstacles along the way, such as ensuring reliable communication between the cart and the smartphone, designing a user-friendly interface, and integrating the necessary components seamlessly.
- **Practical Application**: While this project is undoubtedly fun and innovative, it also has practical applications. Beyond the retail setting, a Bluetooth-controlled cart could be adapted for use in warehouses, hospitals, or other environments where efficient transportation of goods is essential.
- **Community Impact**: Your project could have a positive impact on your community by making shopping more accessible to individuals with disabilities or elderly shoppers. It could also attract attention as a unique and inclusive solution to common challenges.

1.3 ORGANIZATION OF THE PROJECT

In the chapter 1, we discussed about the introduction, overveiw, objective of the project.

In the chapter 2, we discussed about the existing system, proposed system, embedded system.

In the chapter 3, we discussed about the block diagram description, Hardware description.

In the chapter 4, we discussed about the ArduinoIDE, Arduino setup and initialization, developing the code.

In the chapter 5, we discussed about the working of the project.

In the chapter 6, we discussed about the result of the project.

In the chapter 7, we discussed about the advantages, applications of the project.

In the chapter 8, we discussed about the conclusion, future scope of the project

CHAPTER-2

LITERATURE SURVEY

2.1EXISTINGSYSTEM

Several existing systems are designed to enhance road safety by detecting vehicle accidents and alerting emergency services. These systems leverage various technologies, including sensors, microcontrollers, GPS, and GSM modules, to provide real-time monitoring and reporting of accidents. This section provides an overview of some notable existing systems and their functionalities.

The OnStar system, developed by General Motors, is one of the most well-known in-vehicle safety systems. It uses a combination of GPS and cellular technology to provide automatic crash response, turn-by-turn navigation, and emergency services. When an airbag deploys, OnStar automatically connects to an advisor who can assess the situation and contact emergency services with the vehicle's exact location. OnStar's robust infrastructure and reliable service make it a leader in the field, although it is primarily available in GM vehicles.

Fleet management systems often include accident detection and reporting features. Companies like Verizon Connect offer fleet management solutions that monitor vehicle health, driver behavior, and accident detection. These systems use telematics devices installed in the vehicles to collect data on speed, location, and impact. In the event of an accident, the system automatically sends an alert to the fleet manager and emergency services, providing real-time information to facilitate a quick response.

While these existing systems offer significant benefits, they also have limitations. High installation and subscription costs can be a barrier for some users. Additionally, many systems are proprietary and only available for specific vehicle makes and models, limiting their widespread adoption. There are also concerns regarding privacy and data security, as these systems continuously track vehicle location and other sensitive information.

PROPOSED SYSTEM

The proposed system aims to provide a cost-effective, reliable, and easily implementable solution for accident detection and emergency notification. By utilizing readily available components such as Arduino microcontrollers, GPS modules, and GSM modules, this system offers a flexible and scalable approach to enhancing road safety and ensuring timely emergency response.

The accelerometer sensor continuously monitors the vehicle's motion.
Upon detecting a sudden change indicative of an accident, the sensor sends a signal to the Arduino.
The Arduino requests the current location from the GPS module once an accident is detected.
The GPS module provides the vehicle's real-time coordinates.
The Arduino processes the sensor and location data to generate an alert message.
The GSM module sends the alert message to pre-configured emergency contacts and services.
Optionally, the system can initiate a call to emergency services if required.

Key Features and Functionality

Remote Control: The system allows remote control via a mobile interface, enabling users to manually trigger alerts, check system status, and monitor vehicle location in real-time, enhancing safety and control.

Benefits:
Automated alerts with precise GPS coordinates enable quick dispatch of emergency services, potentially saving lives.
Utilizes affordable components, making the system accessible and easy to implement.
Continuous tracking of vehicle motion and location ensures immediate accident detection.

Convenience:
Offers an intuitive mobile interface for remote monitoring and control, providing convenience in managing alerts and system status.
Enables users to trigger alerts and check vehicle status from anywhere, enhancing convenience and peace of mind.
Automatically sends SMS alerts with accurate location details, ensuring prompt emergency response without requiring manual intervention.

Efficiency:
The system ensures efficiency through automated accident detection, real-time GPS tracking, and reliable GSM communication. Continuous monitoring provides immediate accident alerts, while optimized power management ensures prolonged operation. This combination reduces response times, enhances coordination, and maintains consistent performance, all with low power consumption.

Cost-effectiveness:
The system is cost-effective, using affordable and readily available components like Arduino, GPS, and GSM modules. Its low installation and maintenance costs make it accessible for a wide range of users, providing an economical solution for enhancing road safety and ensuring timely emergency response without financial burden.

2.3 INTRODUCTION TO EMBEDDED SYSTEM

An embedded system is a computer system—a combination of a computer processor, computer memory, and input/output peripheral devices—that has a dedicated function within a larger mechanical or electrical system.

It is embedded as part of a complete device often including electrical or electronic hardware and mechanical parts. Because an embedded system typically controls physical operations of the machine that it is embedded within, it often has real-time computing constraints

Embedded systems control many devices in common use today. Ninety-eight percent of all microprocessors manufactured are used in embedded systems.

Modern embedded systems are often based on microcontrollers (i.e. microprocessors with integrated memory and peripheral interfaces), but ordinary microprocessors (using external chips for memory and peripheral interface circuits) are also common, especially in more complex systems.

In either case, the processor(s) used may be types ranging from general purpose to those specialized in a certain class of computations, or even custom designed for the application at hand. A common standard class of dedicated processors is the digital signal processor (DSP).

Since the embedded system is dedicated to specific tasks, design engineers can optimize it to reduce the size and cost of the product and increase the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.

Embedded systems range from portable devices such as digital watches and MP3players, to large stationary installations like traffic light controllers, programmable logic controllers, and large complex systems like hybrid vehicles, medical imaging systems, and avionics.

Structure of the embedded systems architecture:

The below diagrams how the basic structure of the embedded systems architecture.

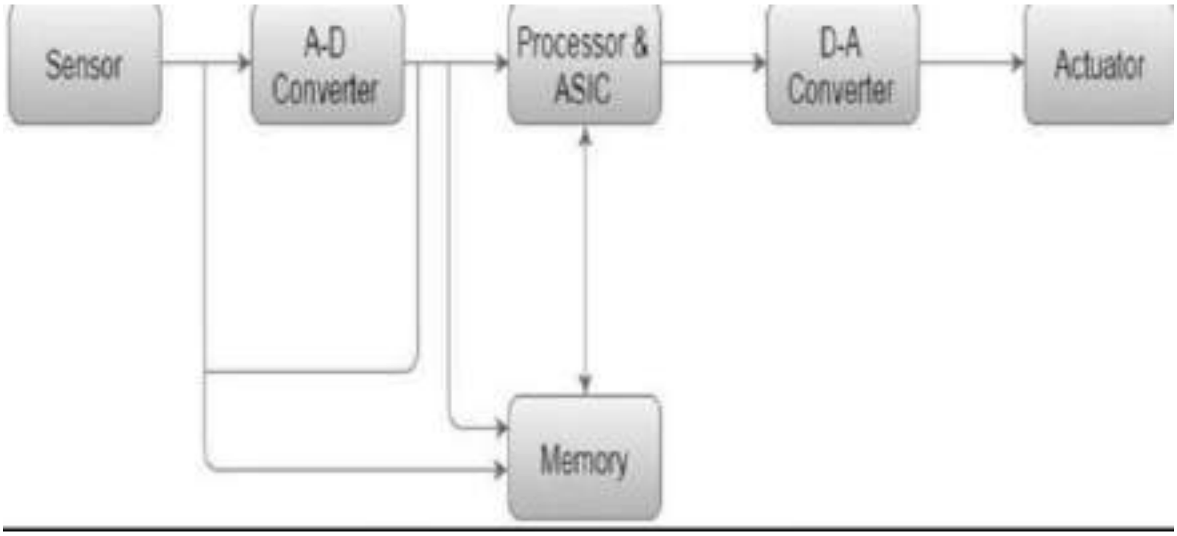


Fig.2.1basicarchitectureofembeddedsystem

It measures the quantities that are physical and converts it to an electrical signal which may be read by an observer or through any electronic tool like an A-D converter. A sensor shops the measured amount to the memory.

A-D Conventer :

An analog-to-digital converter that is used converts the analog signal sent by using the sensor right into a digital signal.

Processor:

Processors process there cords to degree the output and keep it to the memory.

D-A Conventer :

A virtual-to-analog converter converts the virtual records fed by using the processor to analog information.

Actuator: An actuator compares the output given by means of the D-A converter to the actual (anticipated) output saved in it and stores the authorized output.

Embedded system architecture:

This article makes use of an architectural structures engineering method to embedded systems due to the fact it's far one of the maximum powerful gear that can be used to recognize an embedded structures layout or to clear up demanding situations faced while designing a new device.

Whatmakesthearchitecturaltechniquesoeffectiveisitscapacitytoinformallyandquick speak a layout to a spread of people with or without technical backgrounds, even acting as a basis in planning the assignment or certainly designing a device.

Because it truly outlines the requirements of the system, architecture can act as a solid basis for studying and testing the quality of a device and its performance below various situations.

Eventually, the diverse systems of an architecture can then be leveraged for designing destiny merchandise with comparable traits, as a result allowing design understanding to be reused, and leading to a decrease of destiny design and development charges.

By the use of the architectural technique in this article, I'm hoping to relay to the reader that defining and expertise the architecture of an embedded gadget is an important aspect of precise gadget design. This is due to the fact, similarly to the benefits listed above:

Each embedded gadget has an architecture, whether or not it's miles or isn't documented, because every embedded system consists of interacting elements (whether or not hard-ware or software program).

An architecture by way of definition is a fixed of representations of these factors and the irrelationships. In place of having a faulty and steeply-priced architecture force done you through no longer taking the time to define an structure earlier than beginning improvement, take control of the design via defining the architecture first.

Because an embedded architecture captures diverse views, which can be representations of the system, it is a beneficial device in understanding all of the major factors, why every aspect is there, and why the factors behave the way they do. None of the factors within an embedded device works in a vacuum.

Each detail inside a device interacts with some different detail in a few fashions. Without know-how the "whys" at the back of an element's provided functionality, overall performance, and so forth, it would be difficult to determine how the gadget could behave underneath a spread of instances in there al global.

Applications

Embedded systems are commonly found in consumer, industrial, automotive, home appliances, medical, commercial and military applications.

Tele communications systems employ numerous embedded systems from telephone switches for the network to cell phones at the end user. Computer networking uses dedicated routers and network bridges to route data.

Consumer electronics include MP3 players, television sets, mobile phones, video game consoles, digital cameras, GPS receivers, and printers

House hold appliances, such as microwave ovens, washing machines and dish washers, include embedded systems to provide flexibility, efficiency and features.

Applications
of
Embedded Sys

Fig.2.2 applications of embedded systems

Advanced HVAC systems use networked thermostats to more accurately and efficiently control temperature that can change by time of day and season.

Home automation uses wired-and wireless-networking that can be used to control lights, climate, security, audio/visual, surveillance, etc., all of which use embedded devices forsensing and controlling.

Transportation systems from flight to automobiles increasingly use embedded systems. New airplanes contain advanced avionics such as inertial guidance systems and GPS receivers that also have considerable safety requirements.

Various electric motors —brushless DC motors, induction motors and DC motors
— use electronic motor controllers. Automobiles, electric vehicles, and hybrid vehicles increasingly use embedded systems to maximize efficiency and reduce pollution.

Other automotive safety systems using embedded systems include anti-lock braking system (ABS), Electronic Stability Control (ESC/ESP), traction control (TCS) and automatic four-wheel drive.

Medical equipment uses embedded systems for monitoring, and various medical imaging (PET, SPECT, CT, and MRI) for non-invasive internal inspections. Embedded systems within medical equipment are often powered by industrial computers.

Embedded systems are used in transportation, fire safety, safety and security, medical applications and life-critical systems.

Unless connected to wired or wireless networks via on-chip 3G cellular or other methods for IoT monitoring and control purposes, these systems can be isolated from hacking and thus be more secure.

For fire safety, the systems can be designed to have a greater ability to handle higher temperatures and continue to operate. In dealing with security, the embedded systems can be self-sufficient and be able to deal with cut electrical and communication systems.

A new class of miniature wireless devices called motes are networked wireless sensors. Wireless sensor networking, WSN, makes use of miniaturization made possible by advanced IC design to couple full wireless subsystems to sophisticated sensors, enabling people and companies to measure a myriad of things in the physical world and act on this information through IT monitoring and control systems.

These motes are completely self-contained, and will typically run off a battery source for years before the batteries need to be changed or charged.

Characteristics of embedded system

Embedded systems are designed to do some specific task, rather than be a general-purpose computer for multiple tasks. Some also have real-time performance constraints that must be met, for reasons such as safety and usability; others may have low or no performance requirements, allowing the system hardware to be simplified to reduce costs.

Embedded systems are not always standalone devices. Many embedded systems consist of small parts within a larger device that serves a more general purpose.

For example, the Gibson Robot Guitar features an embedded system fortuning the strings, but the overall purpose of the Robot Guitar is, of course, to play music. Similarly, an embedded system in an automobile provides a specific function as a subsystem of the carit self.

The program instructions written for embedded systems are referred to as firm ware, and a rest or edin read-only memory or flash memory chips. They run with limited computer hardware resources: little memory, small or non-existent keyboard or screen.

User interface in embedded system

systems range from no user interface at all, in systems dedicated only to one task, to complex graphical user interfaces that resemble modern computer desktop operating systems. Simple embedded devices use buttons, LEDs, graphic or character LCDs (HD44780LCD for example) with a simple menu system.

More sophisticated devices that use a graphical screen with touch sensing or screen-edge buttons provide flexibility while minimizing space used: the meaning of the button scan change with the screen, and selection involves the natural behavior of pointing at what is desired.

Systems often have a screen with a "joy stick button" for a pointing device.

Some systems provide user interface remotely with the help of a serial (e.g. RS-232, USB, I²C, etc.) or network (e.g. Ethernet)connection.

This approach gives several advantages: extends the capabilities of embedded system, avoids the cost of a display, simplifies BSP and allows one to build a rich user interface on the PC.

A good example of this is the combination of an embedded web server running on an embedded device (such as an IP camera) or a network router. The user interface is displayed in a web browser on a PC connected to the device, there for needing no software to be installed.

Processors in embedded systems

Examples of properties of typical embedded computers, when compared with general-purpose counterparts, are low power consumption, small size, rugged operating ranges, and low per-unit cost.

This comes at the price of limited processing resources, which make them significantly more difficult to program and to interact with.

However, by building intelligence mechanisms on top of the hardware, taking advantage of possible existing sensors and the existence of a network of embedded units, one can both optimally manage available resources at the unit and network levels as well as provide augmented functions, well beyond those available.

For example, intelligent technique scan be designed to manage power consumption of embedded systems.

Embedded processors can be broken into two broad categories. Ordinary microprocessors (μP) use separate integrated circuits for memory and peripherals.

Microcontrollers (μC) have on-chip peripherals, thus reducing power consumption, size and cost.

In contrast to the personal computer market, many different basic CPU architectures are used since the software is custom-developed for an application and is not a commodity product installed by the end user. Both Von Neumann, as well as various degrees of Harvard architectures, is used.

RISC as well as non-RISC processors are found. Word lengths vary from 4-bit to 64-bitsand beyond, although the most typical remain 8/16-bit.

Most architecture comes in a large number of different variants and shapes, many of which are also manufactured by several different companies.

Numerous microcontrollers have been developed for embedded systems use. General-purpose microprocessors are also used in embedded systems; but generally, require more support circuitry than microcontrollers.

Tools of embedded system

As with other software, embedded system designers use compilers, assemblers, and debuggers to develop embedded system software. However, they may also use some more specific tools: Utilities to add a checksum or CRC to a program, so the embedded system can check if the program is valid.

For systems using digital signal processing, developers may use a math work bench to simulate the mathematics.

System-level modeling and simulation tools help designers to construct simulation models of a system with hardware components such as processors, memories, DMA, interfaces, buses and software behavior flow as a state diagram or flow diagram using configurable library blocks.

Simulation is conducted to select the right components by performing power vs. performance trade-off, reliability analysis and bottleneck analysis.

Typical reports that help a designer to make architecture decisions includes application latency, device throughput, device utilization, power consumption of the full system as well as device-level power consumption.

A model-based development tool creates and simulates graphical data flow and UML state chart diagrams of components like digital filters, motor controllers, communication protocol decoding and multi-rate tasks. Custom compilers and linkers may be used to optimize specialized hardware.

An embedded system may have its own special language or design tool, or add enhancements to an existing language such as Forthor Basic. Another alternative is to add a real-time operating system or embedded operating system Modeling and code generating tools often based on state machines Software tool scan come from several sources:

Software companies that specialize in the embedded market Ported from the GNU software development tools Sometimes, development tools for a personal computer can be used if the embedded processor is a close relative to a common PC processor As the complexity of embedded systems grows, higher-level tools and operating systems are migrating into machinery where it makes sense.

For example, cell phones, personal digital assistants and other consumer computers often need significant software that is purchased or provided by a person other than the manufacturer of the electronics.

In these systems, an open programming environment such as Linux, NetBSD, OSGi or Embedded Java is required so that the third-party software provider can sell to alarge market.

Embedded systems are commonly found in consumer, cooking, industrial, automotive, and medical applications.

Some examples of embedded systems are MP3 players, mobile phones, video game consoles, digital cameras, DVD players, and GPS.

Household appliances, such as microwave ovens, washing machines and dishwashers, Include embedded systems to provide flexibility and efficiency.

Advantages

1.It is easy for bulk production.

2.This system is highly reliable for everyday life.

3. It has very few interconnections.

4. This system is tiny in size.

5.They are cheap.

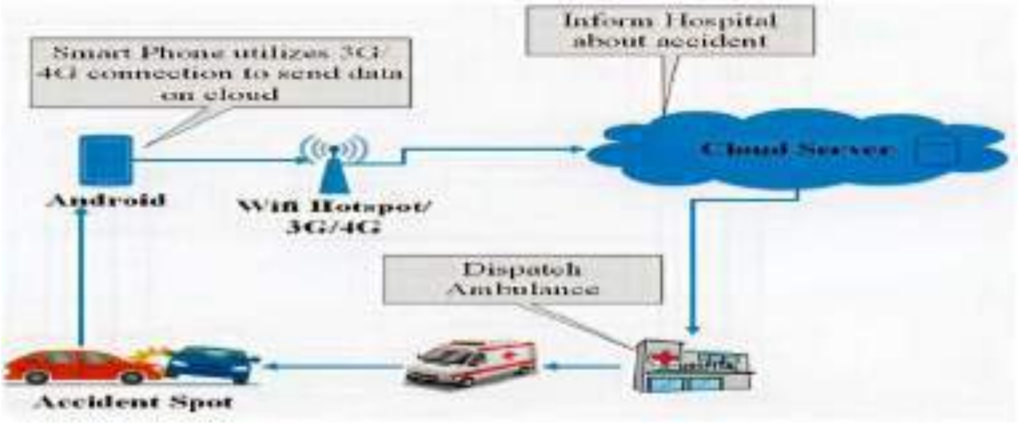
6.It has a quick operation.

7.It has improved product quality and better performance.

8.It optimizes available system resources.

9.It has low power operation and efficient

10.They are less error-prone.


Smart Phone utilizes 3G/4G connection to send data on cloud
Inform Hospital about accident
Android
Wifi Hotspot/ 3G/4G
Cloud Server
Dispatch Ambulance
Accident Spot

CHAPTER-3

BLOCK DIAGRAM AND DISCRIPTION

3.1 BLOCK DIAGRAM

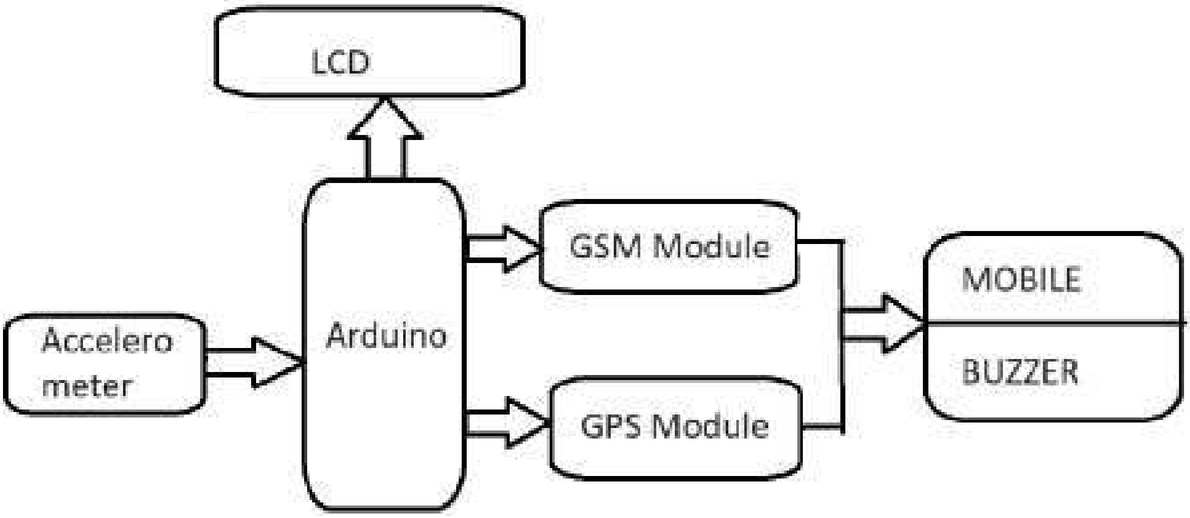


Fig3.1 Block diagram

3.2HARDWARE DISCRIPTION

3.2.1 INTRODUCTION TO AURDINO UNO

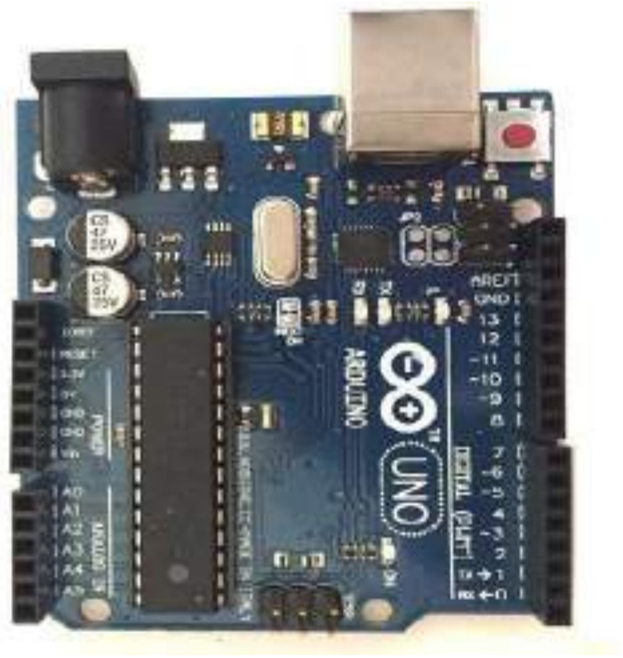


Fig3.2ARDUINOUNO

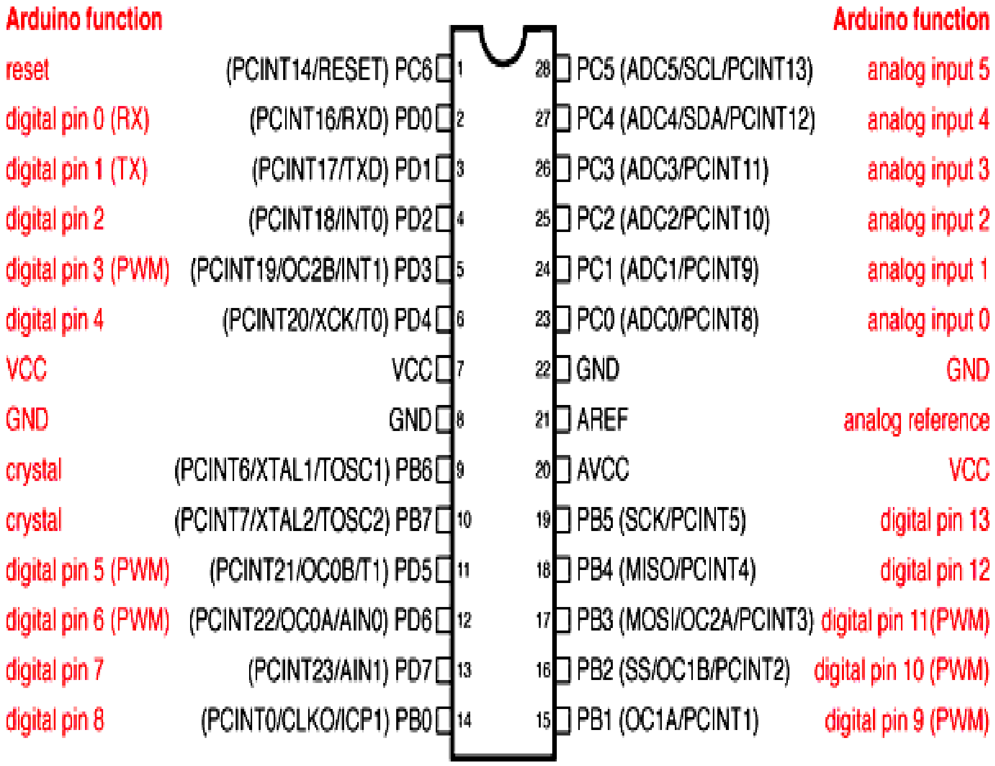
Arduino Uno is a microcontroller board based on 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller.

Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs),6 analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

Arduino **is** a great tool for developing interactive objects, taking inputs from a variety of switches or sensors and controlling a variety of lights, motors and other outputs**.** Arduin**o** project scan be stand-alone or they can be connected to a computer using USB

The Arduino UNO has only 32K bytes of Flash memory and 2K bytes of SRAM. That is more than100,000 times LESS physical memory than a low-end PC! And that's not even counting the disk drive! Working in this minimalist environment, you must use your resources wisely.

ARDUINO UNO PIN DIAGRAM


Arduino function
Arduino function
reset (PCINT14/RESET) PC6 1 28 PC5 (ADC5/SCL/PCINT13) analog input 5
digital pin 0 (RX) (PCINT16/RXD) PD0 2 27 PC4 (ADC4/SDA/PCINT12) analog input 4
digital pin 1 (TX) (PCINT17/TXD) PD1 3 26 PC3 (ADC3/PCINT11) analog input 3
digital pin 2 (PCINT18/INT0) PD2 4 25 PC2 (ADC2/PCINT10) analog input 2
digital pin 3 (PWM) (PCINT19/OC2B/INT1) PD3 5 24 PC1 (ADC1/PCINT9) analog input 1
digital pin 4 (PCINT20/XCK/T0) PD4 6 23 PC0 (ADC0/PCINT8) analog input 0
VCC VCC 7 22 GND GND
GND GND 8 21 AREF analog reference
crystal (PCINT6/XTAL1/TOSC1) PB6 9 20 AVCC VCC
crystal (PCINT7/XTAL2/TOSC2) PB7 10 19 PB5 (SCK/PCINT5) digital pin 13
digital pin 5 (PWM) (PCINT21/OC0B/T1) PD5 11 18 PB4 (MISO/PCINT4) digital pin 12
digital pin 6 (PWM) (PCINT22/OC0A/AIN0) PD6 12 17 PB3 (MOSI/OC2A/PCINT3) digital pin 11(PWM)
digital pin 7 (PCINT23/AIN1) PD7 13 16 PB2 (SS/OC1B/PCINT2) digital pin 10 (PWM)
digital pin 8 (PCINT0/CLKO/ICP1) PB0 14 15 PB1 (OC1A/PCINT1) digital pin 9 (PWM)

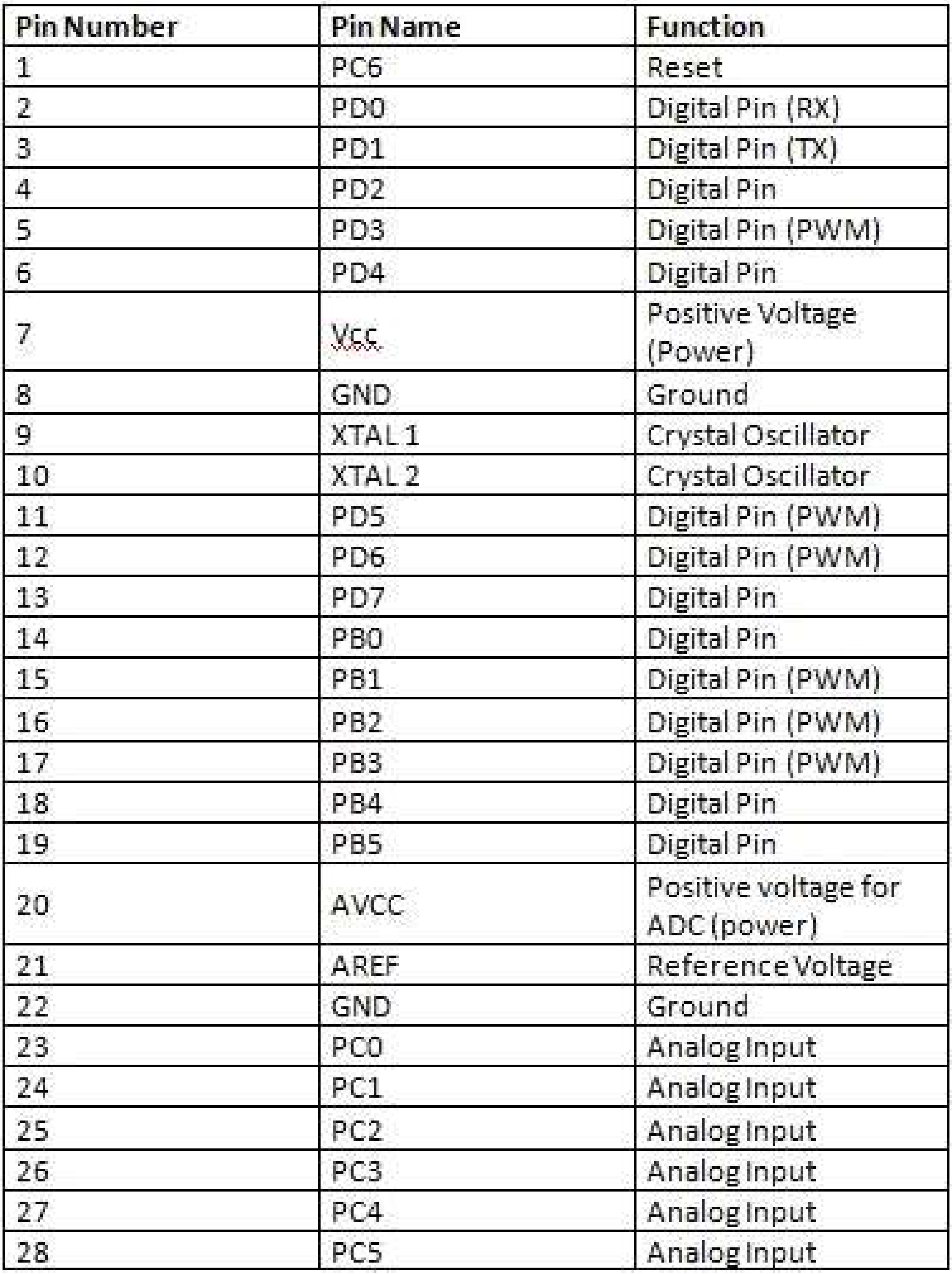
Digital Pins 11,12 & 13 are used by the ICSP header for MOSI, MISO, SCK connections (Atmega168 pins 17,18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

Fig3.3ARDUINO UNOPINDIAGRAM

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs-light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online.

Arduino can input and output analog signals as well as digital signals. An analog signal is one that can take on any number of values, unlike a digital signal which has only two values: HIGH and LOW. ... The function used to output a PWM signal is analog Write (pin, value).pin is the pin number used for the PWM output.

ARDUINO UNO PIN SPECFICITIONS



Pin Number	Pin Name	Function
1	PC6	Reset
2	PD0	Digital Pin (RX)
3	PD1	Digital Pin (TX)
4	PD2	Digital Pin
5	PD3	Digital Pin (PWM)
6	PD4	Digital Pin
7	Vcc	Positive Voltage (Power)
8	GND	Ground
9	XTAL 1	Crystal Oscillator
10	XTAL 2	Crystal Oscillator
11	PD5	Digital Pin (PWM)
12	PD6	Digital Pin (PWM)
13	PD7	Digital Pin
14	PB0	Digital Pin
15	PB1	Digital Pin (PWM)
16	PB2	Digital Pin (PWM)
17	PB3	Digital Pin (PWM)
18	PB4	Digital Pin
19	PB5	Digital Pin
20	AVCC	Positive voltage for ADC (power)
21	AREF	Reference Voltage
22	GND	Ground
23	PC0	Analog Input
24	PC1	Analog Input
25	PC2	Analog Input
26	PC3	Analog Input
27	PC4	Analog Input
28	PC5	Analog Input

table3.1ARDUINO UNO PIN SPECIFICATIONS

ARDUINO UNO TECHNICAL SPECIFICATION

Microcontroller	ATmega328P– 8 bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6(A0 –A5)
Digital I/O Pins	14(Out of which 6 provide PWM output)
DC Current on I/O Pins	40Ma
DC Current on 3.3Vpin	50Ma
Flash Memory	32KB (0.5KB is used for Boot loader)
SRAM	2KB
EEPROM	1KB
Frequency(Clock Speed)	16MHz

Table3.2 .ARDUINO UNO TECHNICAL SPECIFICATIONS

ACCELEROMETER MODULE (ADXL-345)

The ADXL345 is a high-resolution, low-power, 3-axis accelerometer that plays a crucial role in accident detection within the proposed alarming system. It measures acceleration forces resulting from motion, impact, or vibration, providing precise data necessary for identifying accidents. The sensor offers high resolution (13-bit) measurements up to ±16 g, making it sensitive enough to detect even minor changes in vehicle motion.

In the accident alarming system, the ADXL345 continuously monitors the vehicle's acceleration. When a sudden change, such as a collision or rollover, is detected, it sends a signal to the Arduino microcontroller. The Arduino then processes this data to determine the occurrence of an accident and triggers the subsequent alert mechanisms.

The ADXL345 is not only accurate but also cost-effective, contributing to the overall affordability of the system. Its compact size and low power consumption make it an ideal choice for continuous monitoring in vehicular applications, ensuring reliable and timely accident detection.

GPS MODULE (NEO-6M)

The NEO-6M GPS module is a highly efficient and reliable component used for accurate location tracking in the proposed accident alarming system. This module provides real-time geographic positioning by receiving signals from GPS satellites, ensuring precise and continuous location data with an accuracy of up to 2.5 meters.

In the accident alarming system, the NEO-6M module interfaces with the Arduino microcontroller. Upon detection of an accident by the accelerometer, the Arduino queries the NEO-6M for the vehicle's current coordinates. This real-time location data is then included in the automated alert message sent via the GSM module to pre-configured emergency contacts.

The NEO-6M's high sensitivity and fast time-to-first-fix (TTFF) make it ideal for vehicular applications, ensuring quick acquisition of satellite signals and reliable performance even in challenging environments. Its low power consumption and compact size further contribute to the system's overall efficiency and ease of integration, making it an essential component for timely and accurate emergency response.

BUZZER

The buzzer is a critical component in the accident alarming system, serving as an immediate audible alert mechanism. Upon detecting an accident through the accelerometer and verifying the location via the GPS module, the Arduino microcontroller activates the buzzer. This instant alert sound helps in quickly drawing attention to the accident, which can be crucial for nearby responders and for ensuring that the vehicle occupants are aware of the system's activation.

The buzzer is chosen for its simplicity, reliability, and low power consumption. It provides a clear and loud sound that can be heard even in noisy environments, enhancing the system's effectiveness. Additionally, its small size and ease of integration with the Arduino make it an ideal choice for the system. The buzzer's activation serves as a local alert, complementing the GSM module's remote notification function, ensuring a comprehensive response strategy to vehicular accidents.

GSM SIM800L

The SIM800L GSM module is a versatile component used for communication in the accident alarming system. It enables the system to send SMS alerts and make voice calls to emergency contacts or services upon detecting an accident. The module operates on GSM (Global System for Mobile Communications) networks, providing reliable connectivity for transmitting critical information.

In the accident alarming system, the Arduino microcontroller interfaces with the SIM800L module to send automated SMS alerts containing the vehicle's GPS coordinates and accident details to pre-configured contacts. This ensures prompt notification of emergency services or designated individuals, facilitating rapid response to the accident scene.

The SIM800L module supports quad-band GSM frequencies, making it suitable for global deployment. It features low power consumption, which is crucial for maintaining the system's operation during emergencies. Its compact size and compatibility with Arduino make it easy to integrate into the system, enhancing its overall efficiency and reliability in emergency communication scenarios. The SIM800L module plays a pivotal role in enhancing road safety by enabling timely and effective emergency response through mobile network connectivity.

LM2596 Step converter:

The LM2596 is a step-down (buck) voltage regulator module used in the accident alarming system to efficiently manage power supply. It converts higher DC voltages (up to 40V) from the vehicle's power source to a stable lower voltage (adjustable between 1.25V to 35V) required for powering components such as Arduino, GPS module, GSM module, and sensors.

In the accident alarming system, the LM2596 ensures consistent and reliable operation by providing a regulated voltage output, even when the vehicle's battery voltage fluctuates. This stable power supply prevents voltage spikes that could potentially damage sensitive electronic components and ensures continuous monitoring and communication capabilities during critical situations.

The module's adjustable output voltage and high efficiency (up to 92%) minimize heat dissipation, making it suitable for automotive applications where space and thermal management are crucial. Its compact size, ease of integration, and affordability contribute to the overall cost-effectiveness and reliability of the accident alarming system, ensuring uninterrupted functionality and timely response to emergencies on the road.

CHAPTER 4

SOFTWARE DESCRIPTION

4.1 ArduinoIDE

The Arduino Integrated Development Environment - or Arduino Software (IDE) –contains a texted it or for writing code. It connects to the Arduino and Genuine hardware to upload programs and communicate with them.

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension.

The editor has features for cutting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors.

The console displays text output by the Arduino Software (IDE), including complete error messages and other information.

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data.

#include statements will insert one or more at the top of the sketch and compiles the library with your sketch.

Because libraries are uploaded to the board with your sketch, they increase the amount of space It takes up. If ask no longer needs a library, simply delete its #include statements from the top of your code.

If you want to program your Arduino Uno while offline you need to install the Arduino Desktop (IDE)

The Uno is programmed using the Arduino Software (IDE), our Integrated Development Environment common to all our boards.

Before you can move on, you must have installed the Arduino Software (IDE) on your PC.

The serial monitor displays serial sent from the Arduino or Genuine board over USB or serial connector. To send data to the board, enter text and click on the "send" button or press enter.

Note that on Windows, Macor Linux the board will reset(it will rerun your sketch) when you connect with the serial monitor.

Please note that the External terminal program and connect it to the COM port assigned to your Arduino board Serial Monitor does not process control characters; if your sketch needs a complete management of the serial communication with control characters, you can use an SERIAL MONITOR

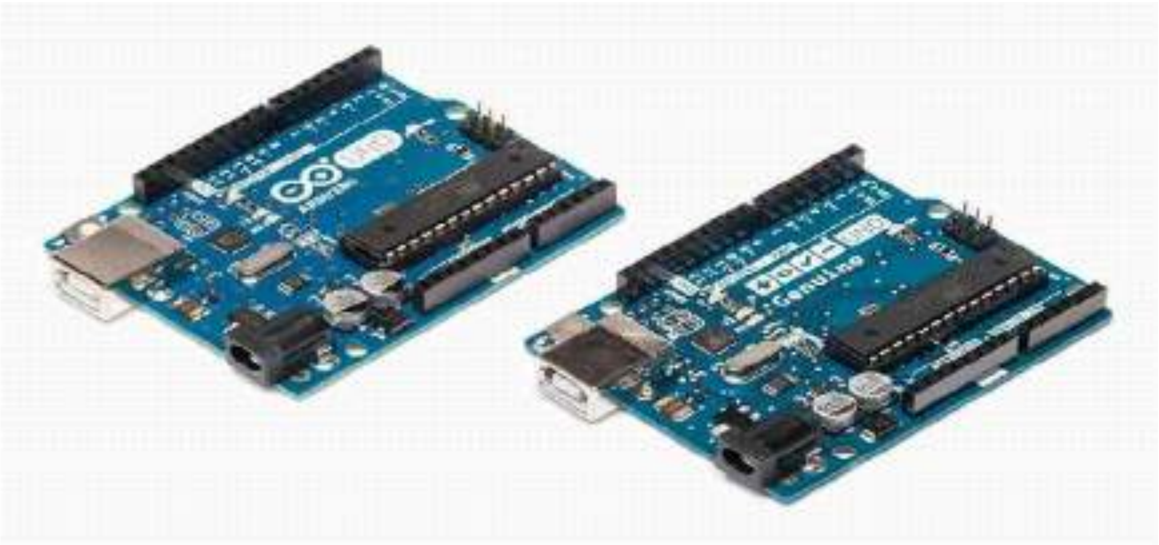


Fig4.1ARDUINOUNO

Arduino was born at the Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IOT applications, wearable, 3D printing, and embedded environments.

All Arduino boards are completely open-source, empowering users to build the min dependently and eventually adapt them to their particular needs.

4.2 ARDUINOSETUPANDINSTIALLISATION

The Uno is programmed using the Arduino Software (IDE).Connect your Uno board with an USB cable.

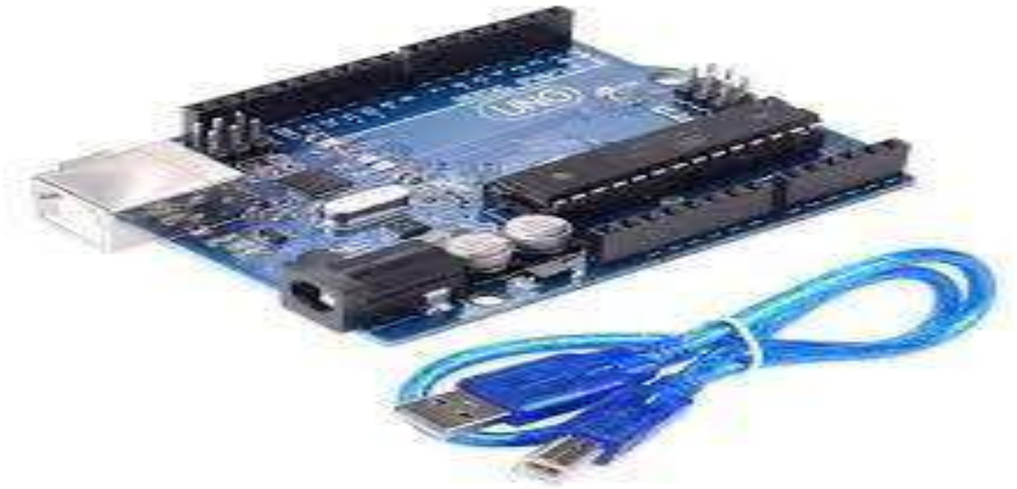


Fig4.2arduino uno and usb cable

If you want to program your Arduino Uno while offline you need to install the Arduino Desktop IDE. The Uno is programmed using the Arduino Software (IDE), our Integrated Development Environment common to all our boards.

Before you can move on, you m**ust** have installed the Arduino Software (IDE) on your PC, as explained in our introduction. Connect your Uno board with an A B USB cable.

The USB connection with the PC is necessary to program the board and not just to power it up. The Uno automatically draw power from either the USB or an external power supply. Connect the board to your computer using the USB cable.

If you used the Installer, Windows from XP upto 10 will install drivers automatically as soon as you connect your board.

If you downloaded you need to follow the procedure step by step

- Click on the Start Menu, and open up the Control Panel.
- While in the Control Panel, navigate to System and Security. Next, click on System. Once the System window is up, open the Device Manager.

- Look under Ports (COM & LPT). You should see an open port named "Arduino UNO (COMxx)". If there is no COM & LPT section, look under "Other Devices "for "Unknown Device".
- Right click on the "Arduino UNO (COMxx)" port and choose the "Update Driver Software" option.
- Next, choose the "Browse my computer for driver software" option.
- Finally, navigate to and select the driver file named "arduino.inf", located in the"Drivers" folder of the Arduino Software download not the "FTDI USB Drivers"sub-directory. If you are using an old version of the IDE older one, choose the Uno driver file named"ArduinoUNO.inf"
- Windows will finish up the driver installation from there.

Open your first sketch

Open the LED blink examplesketch:**File>Examples>01.Basics>Blink**.

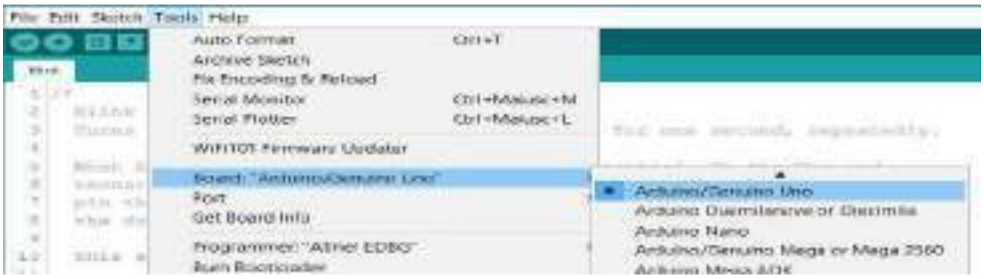


Fig4.3UNOblink

Select your board type and port

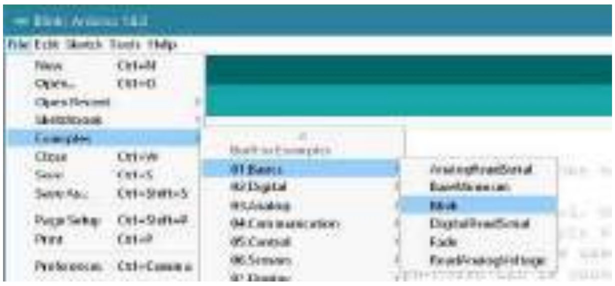


Fig4.4Uno board type port

You'llneedtoselecttheentryinthetools>boardmenuthatcorrespondstoyourarduinoboard.

Select the serial device of the board from the Tools | Serial Port menu. This is likely to be **COM3** or higher (**COM1** and **COM2** are usually reserved for hardware serial ports). To find out, you can disconnect your board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port. Upload the program

A few seconds after the upload finishes, you should see the pin 13 (L) LED on the board start to blink (in orange). If it does, congratulations. You've gotten Arduino up-and-running

If you have problems on uploading programs to the Arduino. Here are some specific suggestions for troubleshooting each of the pieces. They include: the drivers for the board, the board and serial port selections in the Arduino software, access to the serial port, the physical connection to the board, the firmware on the Uno and Mega 2560, the boot loader on the main microcontroller on the board.

Some specific suggestions for troubleshooting each of the pieces are as follows:

Arduino software, drivers, Access to the serial port, physical connection, auto-reset,
boot loader are some of the troubleshooters

Arduino Software

Make sure you have the right item selected in the Tools>Board menu. If you have an Arduino Uno, you'll need to choose it. Also, newer Arduino boards come with an ATmega328, while older ones have an ATmega168. To check, read the text on the microcontroller (the larger chip)on your Arduino board

Then, check that the proper port is selected in the Tools>Serial Port menu (if your port doesn't appear, try restarting the IDE with the board connected to the computer) On Windows, it will be a COM port but you'll need to check in the Device Manager(under Ports) to see which one. If you don't seem to have a serial port for your Arduino board, see the following information about drivers.

Drivers

Drivers provide away for software on your computer (i.e. the Arduino software) to talk to hardware you connect to your computer (the Arduino board).

The Arduino on Windows, if the software is slow to start or crashes on launch, or the Tools menu is slow to open, you may need to disable Bluetooth serial ports or other networked COM ports in the Device Manager. The Arduino software scans all the serial (COM) ports on your computer when it starts and when you open the Tools menu, and these networked port scan sometimes cause large delays or crashes.

Physical Connection

First make sure your board is on (the green LED is on) and connected to the computer.

The Arduino Uno and Mega2560 may have trouble connecting to a Mac through a USB hub. If nothing appears in your "Tools > Serial Port" menu, try plugging the board directly to your computer and restarting the Arduino IDE.

Disconnect digital pins 0 and 1 while uploading as they are shared with serial communication with the computer Try uploading with nothing connected to the board

Make sure the board isn't touching anything metallic or conductive.

Auto-Reset

If you have a board that doesn't support auto-reset, be sure that you are resetting the board a couple of seconds before uploading

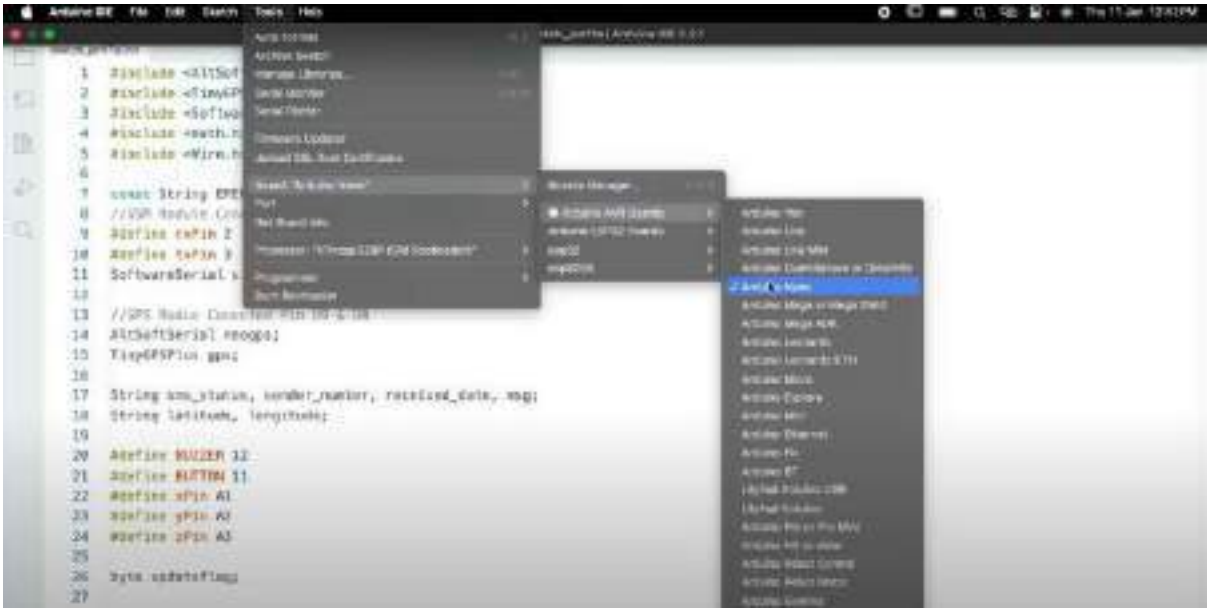
However, on some computers, you may need to press the reset button on the board after you hit the upload button in the Arduino environment. Try different intervals of time be tweent-two, upto 10seconds or more.

INSTALLING ARDUINO LIBRARY AND EXECUTION PROCEDURE IN ARDUINO IDE SOFTWARE

Installing Arduino libraries is straightforward using the Arduino IDE software. First, download the library as a ZIP file from the library's source or through the Arduino Library Manager. In the Arduino IDE, go to Sketch > Include Library > Add .ZIP Library, then navigate to the downloaded ZIP file and select it. The IDE will install the library automatically.

To use the installed library in your sketch, include it at the beginning of your code with `#include <LibraryName.h>`. Libraries extend the functionality of Arduino by providing pre-written code for specific tasks, such as interfacing with sensors or modules. After including the library, write your code to utilize its functions and classes as per the library's documentation

Compile your sketch by clicking the Verify button. If there are no errors, upload it to your Arduino board using the Upload button. The IDE compiles your code into machine language and uploads it via USB or other interfaces supported by your Arduino board.



```
#include <AltSoft
#include <TinyGP
#include <Softwa
#include <math.h
#include <Wire.h

const String EME
//GSM Module Con
#define rxPin 2
#define txPin 3
SoftwareSerial s

//GPS Modue Conected Pin D9 & D8
AltSoftSerial neogps;
TinyGPSPlus gps;

String sms_status, sender_number, received_date, msg;
String latitude, longitude;

#define BUZZER 12
#define BUTTON 11
#define xPin A1
#define yPin A2
#define zPin A3

byte updateflag;
```

4.3 DEVELOPING THE CODE

#include <AltSoftSerial.h>

#include <TinyGPS++.h>

#include <SoftwareSerial.h>

#include <math.h>

#include <Wire.h>

const String EMERGENCY_PHONE = "+9188305848xx";

//GSM Module Conected Pin D2 & D3

#define rxPin 2

#define txPin 3

SoftwareSerial sim800(rxPin, txPin);

//GPS Modue Conected Pin D9 & D8

```
AltSoftSerial neogps;
TinyGPSPlus gps;
String sms_status, sender_number, received_date, msg;
String latitude, longitude;
#define BUZZER 12
#define BUTTON 11
#define xPin A1
#define yPin A2
#define zPin A3
byte updateflag;
int xaxis = 0, yaxis = 0, zaxis = 0;
int deltx = 0, delty = 0, deltz = 0;
int vibration = 2, devibrate = 75;
int magnitude = 0;
int sensitivity = 20;
double angle;
boolean impact_detected = false;
unsigned long time1;
unsigned long impact_time;
unsigned long alert_delay = 30000;
void setup() {
  Serial.begin(9600);
  Serial.println("SIM800L serial initialize");
  sim800.begin(9600);
  Serial.println("NEO6M serial initialize");
  neogps.begin(9600);
  pinMode(BUZZER, OUTPUT);
  pinMode(BUTTON, INPUT_PULLUP);
  sms_status = "";
  sender_number = "";
  received_date = "";
  msg = "";
  sim800.println("AT");
  delay(1000);
  sim800.println("ATE1");
  delay(1000);
  sim800.println("AT+CPIN?");
  delay(1000);
  sim800.println("AT+CMGF=1");
  delay(1000);
  sim800.println("AT+CNMI=1,1,0,0,0");
  delay(1000);
  time1 = micros();
  xaxis = analogRead(xPin);
  yaxis = analogRead(yPin);
  zaxis = analogRead(zPin);
}
void loop() {
```

```
if (micros() - time1 > 1999) Impact();
if (updateflag > 0) {
  updateflag = 0;
  Serial.println("Impact detected!!");
  Serial.print("Magnitude:");
  Serial.println(magnitude);
  getGps();
  digitalWrite(BUZZER, HIGH);
  impact_detected = true;
  impact_time = millis();
}
if (impact_detected == true) {
  if (millis() - impact_time >= alert_delay) {
    digitalWrite(BUZZER, LOW);
    makeCall();
    delay(1000);
    sendAlert();
    impact_detected = false;
    impact_time = 0;
  }
}
if (digitalRead(BUTTON) == LOW) {
  delay(200);
  digitalWrite(BUZZER, LOW);
```

```
    impact_detected = false;
    impact_time = 0;
  }
  while (sim800.available()) {
    parseData(sim800.readString());
  }
  while (Serial.available()) {
    sim800.println(Serial.readString());
  }
}
void Impact() {
  time1 = micros();
  int oldx = xaxis;
  int oldy = yaxis;
  int oldz = zaxis;
  xaxis = analogRead(xPin);
  yaxis = analogRead(yPin);
  zaxis = analogRead(zPin);
  vibration--;
  if (vibration < 0) vibration = 0;
  if (vibration > 0) return;
  deltx = xaxis - oldx;
  delty = yaxis - oldy;
  deltz = zaxis - oldz;
  magnitude = sqrt(sq(deltx) + sq(delty) + sq(deltz));
  if (magnitude >= sensitivity) {
    updateflag = 1;
    vibration = devibrate;
  } else {
    if (magnitude > 15)
      Serial.println(magnitude);
    magnitude = 0;
  }
}
void parseData(String buff) {
  Serial.println(buff);
  unsigned int len, index;
  index = buff.indexOf("\r");
  buff.remove(0, index + 2);
  buff.trim();
  if (buff != "OK") {
    index = buff.indexOf(":");
    String cmd = buff.substring(0, index);
    cmd.trim();
    buff.remove(0, index + 2);
    if (cmd == "+CMTI") {
      index = buff.indexOf(",");
      String temp = buff.substring(index + 1, buff.length());
      temp = "AT+CMGR=" + temp + "\r";
      sim800.println(temp);
    }
    else if (cmd == "+CMGR") {
      if (buff.indexOf(EMERGENCY_PHONE) > 1) {
        buff.toLowerCase();
        if (buff.indexOf("get gps") > 1) {
          getGps();
          String sms_data;
          sms_data = "GPS Location Data\r";
          sms_data += "http://maps.google.com/maps?q=loc:";
          sms_data += latitude + "," + longitude;
          sendSms(sms_data);
        }
      }
    }
  }
}
void getGps() {
  boolean newData = false;
  for (unsigned long start = millis(); millis() - start < 2000;) {
    while (neogps.available()) {
      if (gps.encode(neogps.read())) {
        newData = true;
        break;
      }
    }
  }
  if (newData) {
    latitude = String(gps.location.lat(), 6);
    longitude = String(gps.location.lng(), 6);
    newData = false;
  } else {
    Serial.println("No GPS data is available");
    latitude = "";
    longitude = "";
  }
  Serial.print("Latitude= ");
  Serial.println(latitude);
  Serial.print("Lngitude= ");
  Serial.println(longitude);
}
void sendAlert() {
  String sms_data;
  sms_data = "Accident Alert!!\r";
  sms_data += "http://maps.google.com/maps?q=loc:";
  sms_data += latitude + "," + longitude;
  sendSms(sms_data);
}
void makeCall() {
  Serial.println("calling....");
  sim800.println("ATD" + EMERGENCY_PHONE + ";");
  delay(20000);
  sim800.println("ATH");
  delay(1000);
}
void sendSms(String text) {
  //return;
  sim800.print("AT+CMGF=1\r");
  delay(1000);
  sim800.print("AT+CMGS=\"" + EMERGENCY_PHONE + "\"\r");
  delay(1000);
  sim800.print(text);
  delay(100);
  sim800.write(0x1A);
  delay(1000);
  Serial.println("SMS Sent Successfully.");
}
boolean SendAT(String at_command, String expected_answer, unsigned int timeout) {
  uint8_t x = 0;
  boolean answer = 0;
  String response;
  unsigned long previous;
  //Clean the input buffer
  while (sim800.available() > 0) sim800.read();
  sim800.println(at_command);
  x = 0;
  previous = millis();
  do {
    if (sim800.available() != 0) {
      response += sim800.read();
      x++;
      if (response.indexOf(expected_answer) > 0) {
        answer = 1;
        break;
      }
    }
  } while ((answer == 0) && ((millis() - previous) < timeout));
  Serial.println(response);
  return answer;
}
```

CHAPTER 5

WORKING OF THE PROJECT

Components and their roles:

1. Accelerometer:

- Role: Detects changes in vehicle motion, such as sudden deceleration or impact.

- Working Principle: Measures acceleration forces in multiple axes (x, y, z) using internal sensors. Outputs analog or digital signals to indicate motion changes to the Arduino.

2. GPS Module (e.g., NEO-6M):

- Role: Provides real-time geographical coordinates of the vehicle's location.

- Working Principle: Receives signals from GPS satellites, calculates position using trilateration, and sends coordinate data (latitude, longitude) to the Arduino via serial communication.

3. GSM Module (e.g., SIM800L):

- Role: Enables communication over GSM networks for sending alerts.

- Working Principle: Connects to a mobile network using a SIM card. Sends SMS alerts containing GPS coordinates and accident details to emergency contacts or a central server upon command from the Arduino.

4. Buzzer:

- Role: Provides audible alert signals to notify nearby individuals of an accident.

- Working Principle: Activates upon receiving a signal from the Arduino, emitting a loud sound to attract attention and inform bystanders of the emergency.

Working Together:

- The accelerometer continuously monitors vehicle motion. Upon detecting an abrupt change (e.g., collision), it signals the Arduino.

- The Arduino receives signals from the accelerometer and processes them to confirm an accident based on predefined thresholds for acceleration changes.

- Once an accident is confirmed, the Arduino queries the GPS module for the vehicle's current coordinates to verify the accident location.

- Using the GSM module, the Arduino sends SMS alerts to emergency contacts or a central server with precise GPS coordinates and accident details.

- Simultaneously, the Arduino activates the buzzer to emit audible alerts locally, notifying nearby individuals of the accident.

This integrated system ensures rapid and reliable communication of critical accident information, facilitating prompt emergency response. By leveraging these components' capabilities, the system enhances road safety by reducing response times and improving situational awareness during emergencies.

CIRCUIT DIAGRAM

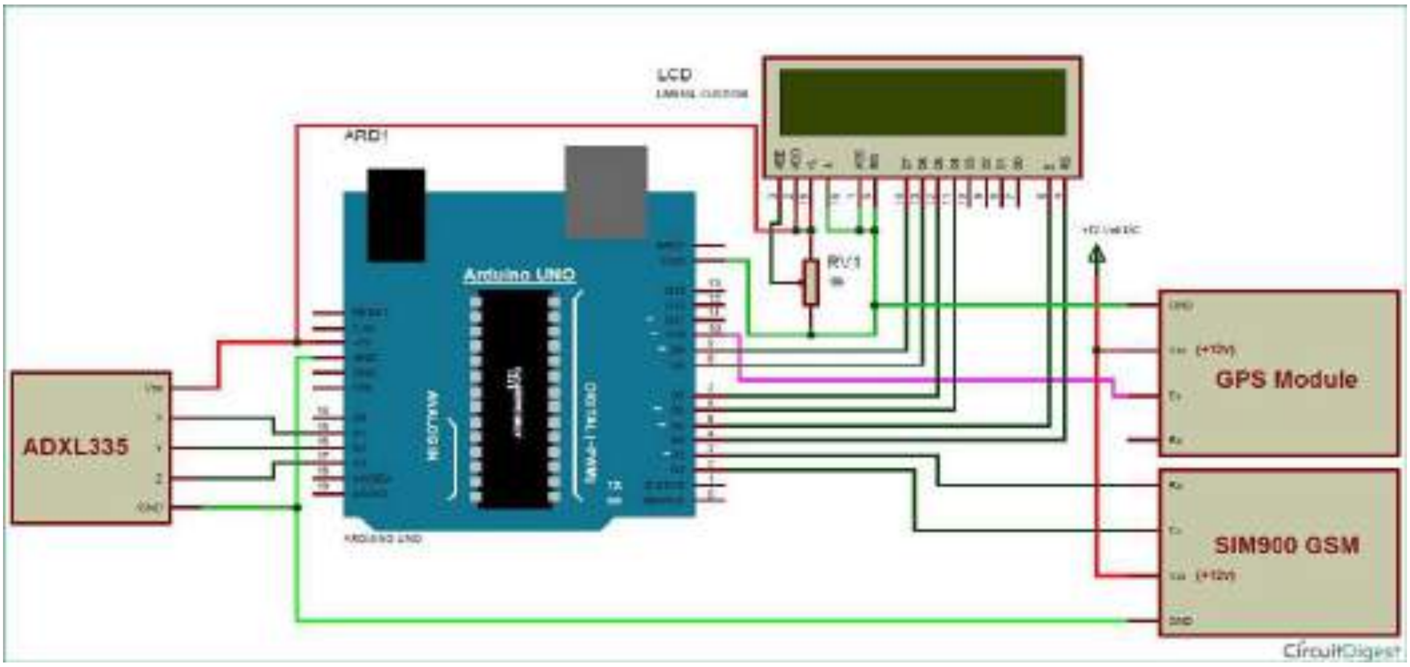
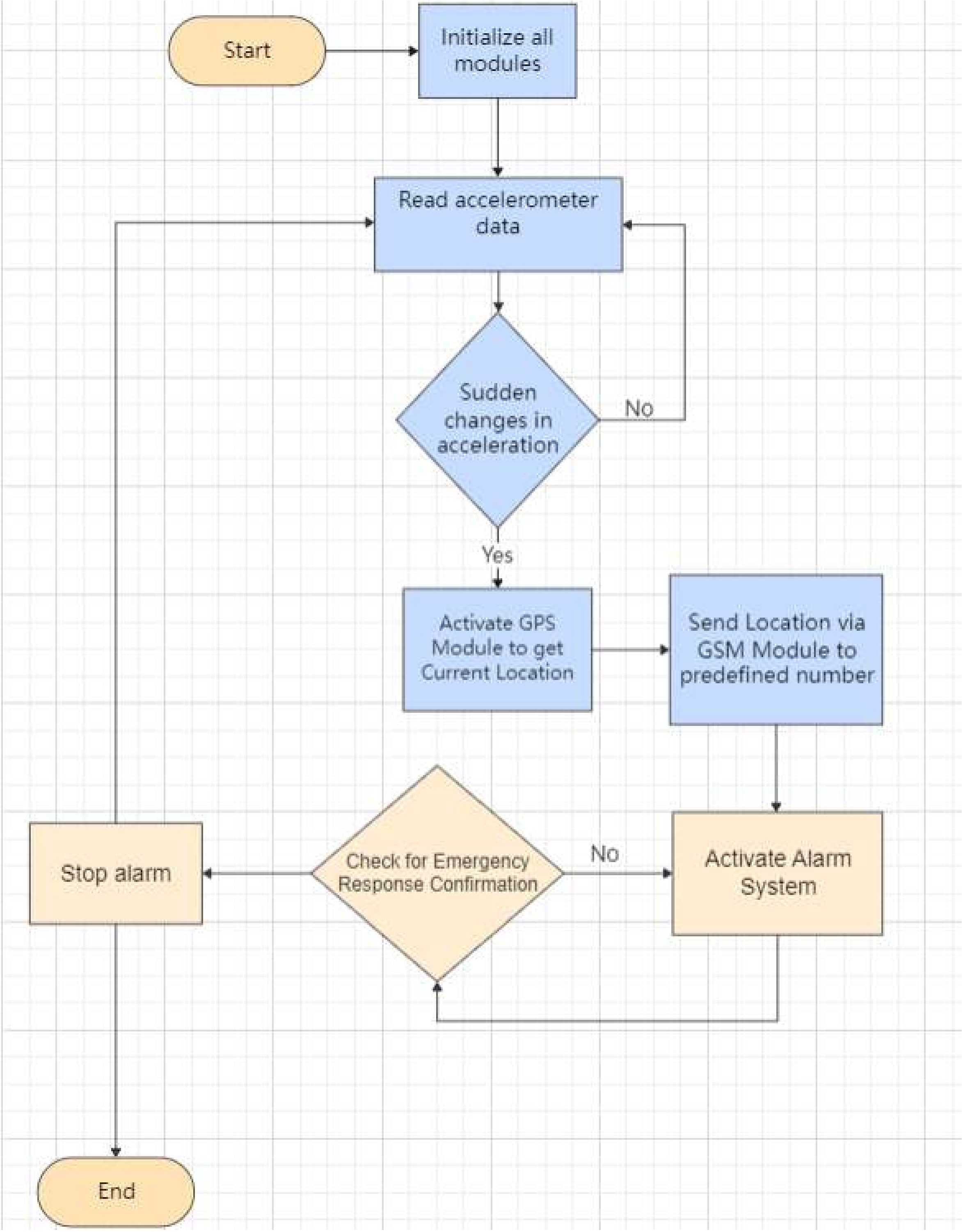


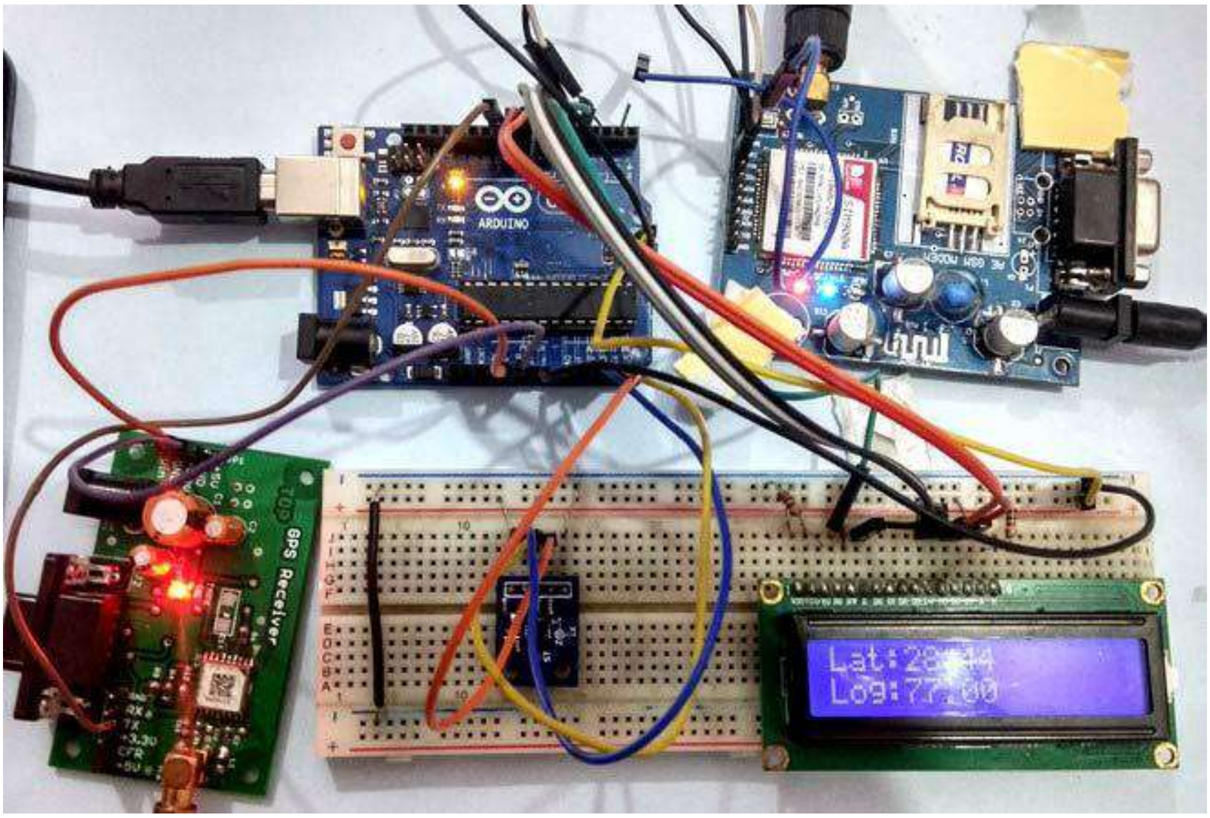
Fig5.2.2circuit diagram of the system

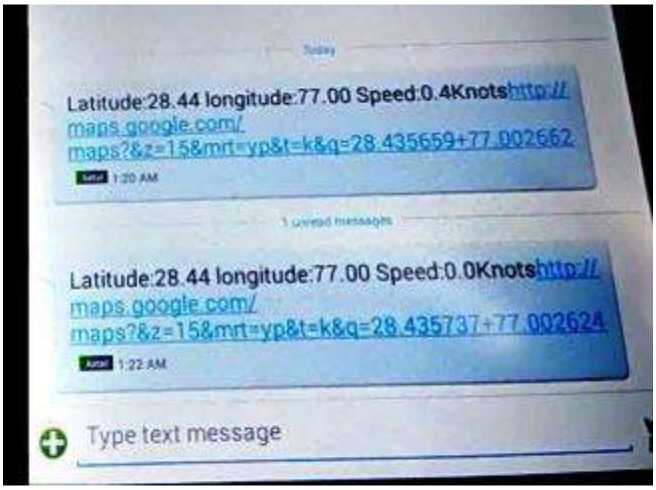
FLOWCHART


Start
Initialize all modules
Read accelerometer data
Sudden changes in acceleration
No
Yes
Activate GPS Module to get Current Location
Send Location via GSM Module to predefined number
Activate Alarm System
Check for Emergency Response Confirmation
No
Stop alarm
End

CHAPTER - 6

RESULTS




Today

Latitude:28.44 longitude:77.00 Speed:0.4Knotshttp://maps.google.com/maps?&z=15&mrt=yp&t=k&q=28.435659+77.002662

1:20 AM

1 unread messages

Latitude:28.44 longitude:77.00 Speed:0.0Knotshttp://maps.google.com/maps?&z=15&mrt=yp&t=k&q=28.435737+77.002624

1:22 AM

Type text message

CHAPTER-7

APPLICATIONS AND ADVANTAGES

APPLICATIONS

Vehicle Safety Systems:

Implementing the system in automobiles for real-time accident detection and immediate alerting of emergency services, ensuring prompt assistance to accident victims.

Fleet Management:

Integrating the system into fleet vehicles to monitor driver safety and respond swiftly to accidents, thereby reducing downtime and improving operational efficiency.

Public Transportation:

Installing the system in buses, trains, or taxis to enhance passenger safety by detecting accidents and notifying relevant authorities or emergency contacts.

Personal Safety Devices:

Adapting the system into wearable devices or personal safety equipment for individuals engaged in high-risk activities, providing immediate assistance in case of accidents or emergencies.

Smart City Infrastructure:

Deploying the system as part of smart city initiatives to improve urban safety and emergency response capabilities, integrating with existing traffic management systems.

IoT and Connected Vehicles:

Incorporating the system into IoT frameworks and connected vehicle technologies to enable data-driven insights into accident patterns and optimize emergency response strategies.

Remote and Rural Areas:

Utilizing the system in remote or rural regions where access to immediate emergency services is limited, enhancing response times and improving survival rates in accidents.

Research and Development:

Supporting research efforts in transportation safety and accident prevention through data collection and analysis facilitated by the system's capabilities.

ADVANTAGES

Early Accident Detection:

Detects accidents promptly through accelerometer sensors, enabling immediate response and potentially reducing injury severity.

Real-Time Location Tracking:

Utilizes GPS modules to provide accurate vehicle location data, facilitating quick emergency response and rescue operations.

Immediate Alerting:

Sends SMS alerts via GSM modules to emergency contacts with precise accident details, ensuring timely assistance.

Versatility:

Can be integrated into various vehicles and applications, enhancing road safety across different transportation sectors.

Cost-Effective:

Uses affordable Arduino components and modules, making it accessible for widespread adoption and deployment.

Enhanced Safety Measures:

Provides audible alerts with buzzers to notify bystanders and nearby vehicles, improving situational awareness.

Scalability:

Can be expanded with additional sensors or communication modules, adapting to evolving safety requirements and technological advancements.

Integration with IoT:

Supports integration with IoT platforms for advanced data analytics and smart city applications, contributing to improved traffic management and accident prevention strategies.

CHATPER-8

CONCLUSION

In conclusion, the Arduino-based accident alarming system utilizing components like accelerometer, GPS module, GSM module, and buzzer represents a significant advancement in enhancing road safety and emergency response capabilities. By integrating these technologies, the system enables early detection of accidents, precise location tracking, and immediate alerting of emergency services or designated contacts. This proactive approach not only reduces response times but also improves the chances of mitigating injury and damage in critical situations.

The system's versatility allows for deployment across various vehicles and applications, from personal safety devices to fleet management systems and smart city infrastructure. Its cost-effectiveness and scalability make it accessible for widespread adoption, contributing to safer roads and enhanced public safety overall. Furthermore, the integration of IoT capabilities opens doors for continuous improvement through data-driven insights and optimized emergency response strategies.

In essence, the Arduino-based accident alarming system represents a vital tool in modernizing transportation safety, addressing critical needs in accident prevention, rapid response, and effective communication during emergencies on the road.

FUTURE SCOPE

As technology continues to evolve, the future scope of Arduino-based accident alarming systems holds promising advancements and applications across various domains:

1.Integration with Autonomous Vehicles

The rise of autonomous vehicles presents an opportunity to integrate accident alarming systems with advanced driver assistance systems (ADAS). Arduino-based platforms can play a crucial role in enhancing safety by providing real-time accident detection and response capabilities in autonomous vehicles. These systems could potentially communicate with centralized control centers or other vehicles to coordinate emergency responses more effectively.

2.Enhanced Sensor Technologies

Future developments in sensor technologies, such as more advanced accelerometers and GPS modules with higher accuracy and faster response times, will further improve the capabilities of accident alarming systems. Integration of sensors capable of detecting biometric data or environmental conditions could provide additional insights into accident severity and enhance emergency response strategies.

3. IoT and Cloud Integration

The integration of IoT frameworks and cloud computing will enable Arduino-based systems to leverage extensive data analytics and predictive modeling. By collecting and analyzing data from multiple sources, including traffic patterns, weather conditions, and vehicle telemetry, these systems can anticipate potential accidents and preemptively alert drivers or emergency services.

4. Smart City Applications

In smart city initiatives, Arduino-based accident alarming systems can contribute to comprehensive traffic management and urban planning strategies. By integrating with smart infrastructure such as traffic lights, road signs, and surveillance cameras, these systems can optimize traffic flow, reduce congestion, and improve emergency response times in urban environments.

REFERENCES

[1] DR.C.K.Gomathy , V.Geetha , S.Madhumitha , S.Sangeetha , R.Vishnupriya Article: A Secure With Efficient Data Transaction In Cloud Service, Published by International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 5 Issue 4, March 2016, ISSN: 2278 – 1323.

[2] Dr.C.K.Gomathy,C K Hemalatha, Article: A Study On Employee Safety And Health Management International Research Journal Of Engineering And Technology (Irjet)- Volume: 08 Issue: 04 | Apr 2021

[3] Dr.C K Gomathy, Article: A Study on the Effect of Digital Literacy and information Management, IAETSD Journal For Advanced Research In Applied Sciences, Volume 7 Issue 3, P.No-51-57, ISSN NO: 2279-543X,Mar/2018