IoT-BASED PARALYSIS PATIENT HEALTHCARE MONITORING SYSTEM

A Project Report Submitted

In Partial Fulfillment of the Requirements for the Degree of

BACHELOR OF TECHNOLOGY

In

ELECTRONICS AND COMMUNICATION ENGINEERING

by

Ayush Agarwal (20213)

Vikas Mishra (20257)

Ayush Sharma (20215)

Nidhi Kandoo (20234)

Under the Supervision of

Er. Shambhavi M Shukla

(Assistant Professor)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

INSTITUTE OF ENGINEERING AND TECHNOLOGY

DR. RAMMANOHAR LOHIA AVADH UNIVERSITY

AYODHYA (U.P) – INDIA.

AUGUST, 2024

CERTIFICATE

This is to certify that major project entitled **"IOT-BASED PARALYSIS PATIENT HEALTHCARE MONITORING SYSTEM"** which is submitted by **Ayush Agarwal**, **Vikas Mishra, Ayush Sharma, Nidhi Kandoo** in partial fulfillment of the requirement for the award of the degree of "Bachelor of Technology" in Electronics & Communication Engineering" at "Institute of Engineering & Technology, Dr. Rammanohar Lohia Avadh University, Ayodhya" is a bonafide project work under my supervision and guidance. The assistance and help received during the investigation have been fully acknowledged.

ER. SHAMBHAVI M SHUKLA

(Project Co-ordinator)

Assistant Professor

Dr. ASHUTOSH KUMAR SINGH

(Head of Department)

Electronics & Communication Department

Date:

Place:

Candidate's Declaration

We hereby declare that the major project report titled **"IOT-BASED PARALYSIS PATIENT HEALTHCARE MONITORING SYSTEM"** is presented on behalf of partial fulfillment of the requirements for the award of Bachelor of Technology in Electronics & Communication Engineering, Institute of Engineering and Technology, Ayodhya, under the supervision of **Er. Shambhavi M Shukla**, Assistant Professor Department of Electronics & Communication Engineering.

We have not submitted the matter presented in this report for the award of any other degree from this institution or any other institution.

Ayush Agarwal (20213)

Vikas Mishra (20257)

Ayush Sharma (20215)

Nidhi Kandoo (20234)

<u>Approval Sheet</u>

This Project entitled **"IOT Based Paralysis Patient Healthcare Monitoring System"** by **Ayush Agarwal, Vikas Mishra, Ayush Sharma, Nidhi Kandoo** is approved for degree Bachelor of Technology in electronics and communication engineering.

Examiner:	Supervisor:
----------------------	-----------------------
----------------------	-----------------------

Date: ______________	Head of Department:
Place: ______________	----------------------------

ACKNOWLEDGEMENT

With profound respect and gratitude, we take the opportunity to convey our thanks to our faculty supervisor**, Assistant Prof. Shambhavi M Shukla** for the guidance, information, and knowledge for our major project **"IOT-BASED PARALYSIS PATIENT HEALTHCARE MONITORING SYSTEM"**

We do extend our heartfelt thanks to **Dr. Ashutosh Kumar Singh**, **Head of the Department**, Electronics and Communication engineering for providing us with an opportunity to go through innovation and thinking of out-of-the-box ideas by providing this major project.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication Engineering who were helpful in many ways for the completion of the project.

ABSTRACT

If you have paralysis, you are partly or entirely unable to move the affected parts of the body. Paralysis may be accompanied by a loss of sensation depending on the location of the injury. Strokes and spinal cord injuries cause sudden paralysis. Some medical conditions can cause gradual paralysis.

In India, every day many lives are affected because the patients are not timely and properly operated. Also, real-time parameter values are not efficiently measured in clinics as well as in hospitals. Sometimes it becomes difficult for hospitals to frequently check patients' conditions. Also, continuous monitoring of ICU patients is not possible. To deal with these types of situations, our system is beneficial.

In this project, we propose a Sign Language Glove which will assist those people who are suffering from any kind of speech defect to communicate through gestures i.e. with the help of single-handed sign language the user will make gestures of alphabets. The glove will record all the gestures made by the user and then it will translate these gestures into visual form as well as in audio form.

This project uses an ADRUINO controller to control all the processes and flex sensors along with accelerometer sensors will track the movement of fingers as well as the entire palm. An LCD will be used to display the user's gesture and a speaker to translate the gesture into an audio signal is planned, if possible, for execution.

This project can be further developed to recognize complex foods, water, etc. GSM modem will send SMS to predefined mobile numbers.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviations	Full Form
IoT	Internet of Things
UI	User Interface
WIFI	Wireless Fidelity
BP	Blood Pressure
HR	Heart Rate
ECG	Electrocardiogram
HTTP	Hypertext Transfer Protocol
SQL	Structured Query Language
PHP	Hypertext Preprocessor
LED	Light Emitting Diode
LCD	Liquid Crystal Display
GSM	Global System for Mobile

CHAPTER 1

INTRODUCTION

1.1 Introduction to IOT based paralysis patient monitoring system

An IoT-based paralysis health monitoring system utilizes sensors to track vital signs and movement patterns of individuals with paralysis. These sensors transmit data to a central hub, which then analyzes the information and provides insights to caregivers or healthcare professionals in real-time. By continuously monitoring the health status of patients, this system enables early detection of complications and timely intervention, improving overall quality of care and patient outcome. Hence, we can say that the project IoT-based paralysis health monitoring system revolutionizes healthcare for individuals with paralysis by seamlessly integrating sensor technology with advanced data analytics. By employing a network of sensors placed on the body or within the living environment, this system continuously collects real-time data on vital signs, movement patterns, and environmental conditions. This wealth of information is transmitted wirelessly to a central platform.

This system integrates a network of interconnected sensors and smart devices designed to continuously monitor various health metrics, such as heart rate, blood pressure, body temperature, and respiratory patterns. These sensors can be embedded in wearable devices, such as wristbands or patches, or integrated into the environment, like smart beds equipped with pressure sensors to detect the risk of bedsores. The real-time data collected by these devices are transmitted to a central hub, where it is analysed and monitored by healthcare professionals or caregivers. This setup allows for the immediate detection of any abnormal changes in the patient's health status, enabling timely medical intervention, which is crucial in preventing complications that are common in paralysis patients, such as infections or respiratory distress. Furthermore, the system often includes advanced analytics and machine learning algorithms that can identify patterns and predict potential health issues before they become critical. This predictive capability allows healthcare providers to implement preventative measures, significantly improving the quality of care and reducing the likelihood of hospital readmissions. Additionally, by enabling remote monitoring, IoT-based systems provide patients with greater independence and comfort, allowing them to stay in their own homes while still receiving high-quality care. This not only enhances the patient’s quality of life but also alleviates the burden on healthcare facilities and reduces the overall cost of care. As technology continues to evolve, these systems are expected to become more sophisticated, incorporating even more advanced features such as integration with electronic health records (EHRs) and telemedicine services, further expanding their utility and effectiveness in managing chronic conditions like paralysis.

1.2 Purpose

An IoT-based healthcare system for paralysis patients is designed to significantly improve patient care, enhance safety, and streamline the management of health conditions associated with paralysis. The primary purposes of this system can be summarized as follows:

1. Continuous Monitoring and Real-Time Data Collection:

The system employs various IoT devices, such as wearable sensors and smart home equipment, to continuously monitor vital signs and health metrics, including heart rate, blood pressure, temperature, and respiratory rate. This real-time data is crucial for keeping track of a patient's health status, especially when they cannot communicate effectively due to their condition.

2. Early Detection and Prevention of Complications:

Paralysis patients are prone to complications like pressure ulcers, respiratory issues, and infections due to limited mobility. The IoT system can detect early signs of these complications, enabling timely interventions that can prevent more severe health problems and reduce the need for emergency medical attention.

3. Enhanced Patient Safety:

By continuously monitoring the patient’s environment and health indicators, the system can alert caregivers or medical professionals to potential dangers, such as falls or sudden changes in health metrics. This proactive approach ensures that any emerging issues are addressed swiftly, thereby enhancing patient safety.

4. Increased Independence and Quality of Life:

IoT technology empowers patients with paralysis by providing tools for self-monitoring and alerts. For example, wearable devices can remind patients to take medication or notify them when it’s time to change positions to avoid pressure sores. This level of support helps patients maintain a degree of independence and improves their overall quality of life.

5. Remote Monitoring and Telemedicine Integration:

IoT-based systems facilitate remote patient monitoring, which is particularly beneficial for those who find it challenging to travel to healthcare facilities. This capability is often integrated with telemedicine services, enabling virtual consultations and continuous care, thereby extending the reach of healthcare services to patients in remote or underserved areas.

6. Reduction in Healthcare Costs:

By preventing complications and reducing hospital admissions, IoT-based systems can significantly lower healthcare costs. Early detection and intervention minimize the need for expensive treatments and prolonged hospital stays, making healthcare more cost-effective for both patients and providers.

7. Enhancing Research and Development:

The vast amount of data collected from IoT devices can be invaluable for medical research. Researchers can use this data to gain deeper insights into the health patterns of paralysis patients, potentially leading to the development of new treatments and care methodologies.

CHAPTER 2

LITERATURE REVIEW

2.1 Thesis by previous scientist:

The literature review of IOT based paralysis patient health care system reveals that the Paralytic patient healthcare system is a system designed to help patient convey various messages to other people like doctor, nurse or family member or caretaker, the system makes use of micro controller based circulate to achieve this functionality Physically disabled people many of the times rely on others, even to perform simple action like switching on/off lights, turning on/off fan etc. In order to provide solutions to these activities the system use hand motion-controlled device when even there is motion the rely circuit will be activated which time on/off lights and fan. Patient healthcare system is method in which the doctor or care taker will monitor patient's health from any location any time. There are cases when there is no one hereby him/her to overcome such situation is system continuously monitors health records of patients such as heartbeat and body temperature.

Finally, there are several significant and important papers which focus on paralysis patient healthcare monitoring, or the development of systematic experimental-statistical approaches and other various component which are used develop this project.

Patterson, V.K., et.al. [1] carried out the overview of Telemedicine healthcare system and discusses how telemedicine, enhanced by IoT, can offer continuous monitoring, especially for patients with limited mobility, thus ensuring timely interventions and reducing the need for frequent hospital visits.

Park, H.C., et.al. [2] "Review of Wearable Sensors and Systems with Application in Rehabilitation". And various wearable sensors that are crucial for monitoring patients with physical disabilities, including paralysis. It highlights the potential of these devices in rehabilitation, providing critical data on patient movements and health status, which are essential for tailoring rehabilitation programs and ensuring patient safety.

Chen, M., et.al. [3] state that the use of big data analytics in healthcare, emphasizing the importance of analyzing large datasets from IoT devices. The insights gained from such analysis can lead to better predictive models and early detection systems, which are crucial for managing the health of paralysis patients.

Jha, N.K., et.al. [4] discuss the “Comprehensive Study of Security of Internet-of-Things" and addresses the security concerns related to IoT systems, which are critical when handling sensitive health data. The authors discuss various vulnerabilities and propose security frameworks to protect patient data, which is essential for maintaining trust in IoT healthcare applications.

Sahoo, P.K., et.al. [5] explores the “Integration of big data analytics with telemedicine platforms” and highlighting how IoT devices can support remote monitoring and prediction of health conditions. This approach is particularly beneficial for managing the health of paralysis patients remotely, ensuring continuous care and timely interventions.

John A, Rogers., et.at. [6] explores new materials and technologies for creating “flexible and stretchable electronics”. These advancements are crucial for developing comfortable and effective wearable devices that can continuously monitor health metrics in patients with mobility issues, such as those with paralysis.

Galzarano, S., et.al. [7] describe the “Framework for Collaborative Computing and Multi-Sensor Data Fusion in Body Sensor Networks". And also discussed the framework for integrating data from multiple sensors, which enhances the capability of IoT systems in healthcare. The collaborative computing approach allows for more comprehensive monitoring and data analysis, crucial for understanding and managing complex health conditions like paralysis.

Kumar, R.K., et.al. [8] provides an extensive review of various wearable and smartphone-based technologies used for fall detection. The authors discuss different sensors, including accelerometers and gyroscopes, and the algorithms used to analyse the data for accurate fall detection.

Torrents, C., et.al. [9] discusses the design and implementation of a fall detection system using the ADXL335 accelerometer. The researchers developed an algorithm that utilizes the sensor's output to distinguish between falls and normal activities, emphasizing the accelerometer's high sensitivity and low power consumption.

Lee, Sanghoon., et.al. [10] reviews the design and implementation of wearable ECG monitoring systems. The authors discuss various sensor technologies, including dry and wet electrodes, and their integration into wearable devices like smartwatches and chest straps. The study highlights the advancements in sensor miniaturization and signal processing techniques. The researchers developed an algorithm that utilizes the sensor's output to distinguish between falls and normal activities, emphasizing the accelerometer's high sensitivity and low power consumption.

CHAPTER 3

THEORITICAL BACKGROUND

The development of Internet of Things (IoT)-based healthcare systems specifically designed for paralysis patients involves an interdisciplinary approach, integrating knowledge from healthcare, sensor technology, data analytics, and network communication. These systems aim to improve patient outcomes by providing continuous monitoring and data-driven insights, enhancing the quality of care for individuals with paralysis. Below is a detailed theoretical framework for understanding these systems.

3.1 Fundamental Concepts and Technologies

1. Internet of Things (IoT):

- Definition: IoT refers to a network of interconnected devices capable of collecting, exchanging, and processing data through the internet. In healthcare, IoT facilitates real-time monitoring and management of patient health metrics.
- Components: The IoT ecosystem for healthcare includes sensors, actuators, communication devices, data storage, and processing units. Each component plays a vital role in ensuring the system functions seamlessly.

2. Types of Sensors:

- Physiological Sensors: Measure vital signs such as heart rate (ECG), respiration rate, blood pressure, and body temperature.
- Activity and Movement Sensors: Include accelerometers and gyroscopes, crucial for detecting movement or falls, which are particularly important for monitoring paralysis patients who are at risk of pressure ulcers and other complications due to immobility.
- Environmental Sensors: Monitor room conditions such as temperature, humidity, and air quality, which can affect the patient's health, which also measured the body temperature.

3. Data Communication and Networking:

- Wireless Communication Protocols: Essential for transmitting data from sensors to processing units. Common protocols include Bluetooth, Wi-Fi, Zigbee, and LoRaWAN, chosen based on factors like data rate, range, and power consumption.
- Network Architecture: Typically involves a star topology where sensors connect to a central hub (gateway), which then connects to the internet for data transmission to cloud servers.

3.2 Data Processing and Analysis

1. Edge and Cloud Computing:

- Edge Computing: Involves processing data at or near the source (e.g., on the device or local server), reducing latency and bandwidth use. This is critical for time-sensitive applications such as fall detection or monitoring critical health metrics.
- Cloud Computing: Offers scalable resources for data storage and advanced analytics, allowing for the processing of large volumes of data collected over time. Cloud platforms enable long-term data analysis and storage, supporting comprehensive health monitoring.

2. Data Analytics and Machine Learning:

- Data Preprocessing: Involves cleaning and normalizing data to ensure accuracy. This step is crucial for eliminating noise and errors in sensor readings.
- Machine Learning Algorithms: Applied for predictive analytics, anomaly detection, and classification of health conditions. Algorithms such as decision trees, neural networks, and support vector machines are commonly used.
- Predictive Analytics: Uses historical and real-time data to predict future health events, enabling proactive healthcare interventions.

3.3 Clinical Integration and Decision Support

1. Integration with Healthcare Systems:

- Electronic Health Records (EHRs): IoT systems often integrate with EHRs to provide a holistic view of a patient's health history, improving diagnosis and treatment planning.

- Clinical Decision Support Systems (CDSS): Utilize data from IoT devices to provide healthcare professionals with insights and recommendations, aiding in clinical decision-making.

2. Real-Time Monitoring and Alerts:

- Continuous Monitoring: Allows for the tracking of patient health metrics in real time, which is crucial for managing chronic conditions and preventing acute events.
- Automated Alerts: Systems can be configured to send alerts to healthcare providers or caregivers when certain thresholds are exceeded or abnormal patterns are detected, enabling timely interventions.

3.4 Patient-Centric Care and User Interaction

1. Patient and Caregiver Interfaces:

- User-Friendly Design: Interfaces should be intuitive and accessible, allowing patients and caregivers to easily interact with the system, access health data, and receive notifications.
- Patient Engagement: By providing patients with access to their health data, IoT systems can encourage greater involvement in their own care, improving adherence to treatment plans and lifestyle modifications.

2. Ethical and Privacy Considerations:

- Data Privacy and Security: Protecting sensitive health information is paramount. Systems must comply with regulations such as HIPAA in the U.S. and GDPR in Europe, ensuring data is encrypted and access is controlled.
- Informed Consent: Patients must be fully informed about the data being collected and how it will be used, ensuring their consent is obtained and respected.

3.5. Challenges and Future Directions

1. Technical Challenges:

- Interoperability: The need for standardized protocols to ensure devices from different manufacturers can work together seamlessly.

- Data Accuracy and Reliability: Ensuring sensors are accurate and reliable, as faulty data can lead to incorrect clinical decisions.

2. Adoption and Scalability:

- User Acceptance: Overcoming resistance from patients and healthcare providers, often through education and demonstrating the benefits of the technology.
- Scalability: Developing systems that can be scaled to accommodate large numbers of patients, ensuring that infrastructure and resources can handle increasing data loads.

3. Future Trends:

- Advancements in AI and Machine Learning: Continued development in these fields will enhance the predictive capabilities of IoT healthcare systems, making them more accurate and reliable.
- Integration with Advanced Medical Devices: IoT systems will increasingly integrate with advanced medical devices and wearable technologies, providing more comprehensive health monitoring solutions.

3.6 Embedded Systems:

Embedded systems are electronic devices that incorporate microprocessors with in Their implementations. The main purposes of the microprocessors are to simplify the system design and provide flexibility. Having a microprocessor in the device means that removing the bugs, making modifications, or adding new features are only matters of rewriting the software that controls the device. Or in other words embedded computer systems are electronic systems that include a microcomputer to perform a specific dedicated application. The computer is hidden inside these products. Embedded systems are ubiquitous. Every week millions of tiny computer chips come pouring out of factories finding their way into our everyday products.

Physically, embedded systems range from portable devices such as digital watches and MP3 players, to large stationary installations like traffic lights, factory controllers, or the systems controlling nuclear power plants. In terms of complexity embedded systems can range from very simple with a single microcontroller chip, to very complex with multiple units, peripherals and networks mounted inside a large chassis or enclosure.

Embedded systems are self-contained programs that are embedded within a piece of hardware. Whereas a regular computer has many different applications and software that can be applied to various tasks, embedded systems are usually set to a specific task that cannot be altered without physically manipulating the circuitry. Another way to think of an embedded system is as a computer system that is created with optimal efficiency, thereby allowing it to complete specific functions as quickly as possible.

Embedded systems designers usually have a significant grasp of hardware technologies. They used specific programming languages and software to develop embedded systems and manipulate the equipment. When searching online, companies offer embedded systems development kits and other embedded systems tools for use by engineers and businesses.

CHAPTER 4

PROPOSED SYSTEM

4.1 Proposed Methodology

This project will be available for monitoring at the operational levels. The normal heart beat range of a paralyzed can be nearly 60-100 beats per minute. If the range goes below 60, it may lead to heart block, syncope and when the range goes above 100, it may lead to anxiety and tachycardia. So, when it increases or decreases the status of the patient pulse rate will be intimated. The oxygen level of a paralyzed patient shall be above 93%. If the range reaches below 93%, it leads irrational thinking and health problems. This can be measured to know the fall in oxygen. The normal respiratory rate of a paralysis can be nearly 12-20 breathes per minute. The rate is usually measured when a person is at rest and simply involves counting the number of breaths for one minute by counting how many times the chest rises. If the paralysis increases for the paralyzed patient, then the respiration increases this can be the basic parameter of the paralyzed. If the respiration rate increases or decreases 12-20 breathe per minute, it will be intimated. All the basic parameters are monitored and if there is a dangerous change in paralyzed patient's status then a message will be intimated to the doctor and caretaker about the condition of the paralyzed. This contribution of paralysis monitoring system is for better process management, superior flexibility and increased efficiency within hospitals is further underlining the appeal of wireless networking options for paralysis patient monitoring systems.

Data Acquisition is carried out by the sensors that measure the various physiological data and carries these bioelectrical signals to the microcontroller. Data Transmission Typically the data collected by the microcontroller is transmitted to the internet using IoT module and an SMS can be sent to the caretaker if any critical parameter is recorded. Individual sensor’s data can be accessed via computer or mobile connected to internet. Cloud based Processing Diagnoses and prognosis of several health conditions and diseases can be done using the sensor data. Long term storage of patient’s health information can be done and the health information can be accessed using internet. The IOT just is the web of stuff. Independent technologies can collect and transmit data without human intervention via a wireless network. There are endless personal or business opportunities. In the health care industry, remote monitoring has been possible with IOT-enabled devices that release the possibility of keeping patients safe and healthy and that enable doctors to provide superlative health care. Respond to and enable physical objects to collect information and respond to instructions. To collect, store, handle, manipulate, and manipulate data. The communications infrastructure that includes protocols and technologies that permit two physical objects to exchange at his most important. We used MySQL and PHP as a simple and easy-to-use IOT platform for data storage, data viewing and device control. By using some backend program, we sent an alert on mail, Connect the respective sensors to a Arduino 2560 which gets connected to a database. By using this value, we can monitor the data gathered by the sensors. Based on the patient needs it will give sound and the output will be displayed on the LCD display and if the patients pulse or oxygen become abnormal it will be displayed on the lcd which will be monitored continuously. Also, a mail will be triggered.

4.2 Proposed Solution

The proposed solution of our system is to help a person adapt to life with paralysis by making themes independent as possible. Where we see a problem with these types of devices that are being developed is that they are very large and expensive machines. They seem to be only available in hospitals and not able to be used at the patient’s home or at their convenience.

By using this value, we can monitor the data gathered by the sensors. Based on the patient needs it will give sound and the output will be displayed on the LCD display and if the patients pulse or oxygen become abnormal it will be displayed on the lcd which will be monitored continuously. Also, a mail will be triggered. The normal respiratory rate of a paralysis can be nearly 12-20 breathes per minute. The rate is usually measured when a person is at rest and simply involves counting the number of breaths for one minute by counting how many times the chest rises. If the paralysis increases for the paralyzed patient, then the respiration increases this can be the basic parameter of the paralyzed. If the respiration rate increases or decreases 12-20 breathe per minute, it will be intimated. All the basic parameters are monitored and if there is a dangerous change in paralyzed patient's status then a message will be intimated to the doctor and caretaker about the condition of the paralyzed. This contribution of paralysis monitoring system is for better process management, superior flexibility and increased efficiency within hospitals is further underlining the appeal of wireless networking options for paralysis patient monitoring systems.

Data Transmission Typically the data collected by the microcontroller is transmitted to the internet using IoT module and an SMS can be sent to the caretaker if any critical parameter is recorded. Individual sensor's data can be accessed via computer or mobile connected to internet. Cloud based Processing Diagnoses and prognosis of several health conditions and diseases can be done using the sensor data. Long term storage of patient's health information can be done and the health information can be accessed using internet. The IOT just is the web of stuff. Independent technologies can collect and transmit data without human intervention via a wireless network. There are endless personal or business opportunities.

CHAPTER 5

INTRODUCTION ABOUT IOT

IoT stands for Internet of Things, which means accessing and controlling daily usable equipment and devices using Internet. Our IoT tutorial includes all topics of IoT such as introduction, features, advantage and disadvantage, ecosystem, decision framework, architecture, and domains, biometric, security camera and door unlock system, devices, etc.

5.1 What is an Internet of thing (IOT):

Let us look closely at our mobile device which contains GPS Tracking, Mobile Gyroscope, Adaptive brightness, Voice detection, Face detection etc. These components have their own individual features, but what about if these all communicate with each other to provide a better environment? For example, the phone brightness is adjusted based on my GPS location or my direction. Connecting everyday things embedded with electronics, software, and sensors to internet enabling to collect and exchange data without human interaction called as the Internet of Things (IoT). The term "Things" in the Internet of Things refers to anything and everything in day-to-day life which is accessed or connected through the internet.

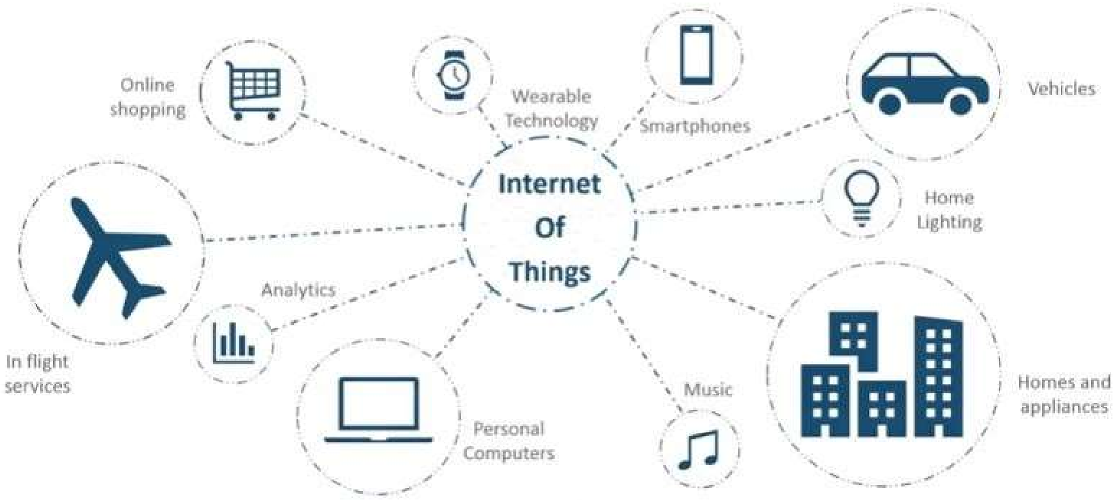

Fig. 5.1: Internet of Things

IoT is an advanced automation and analytics system which deals with artificial intelligence, sensor, networking, electronic, cloud messaging etc. to deliver complete systems for the product or services. The system created by IoT has greater transparency, control and performance.

As we have a platform such as a cloud that contains all the data through which we connect all the things around us. For example, a house, where we can connect our home appliances such as air conditioner, light, etc. through each other and all these things are managed at the same platform. Since we have a platform, we can connect our car, track its fuel meter, speed level, and also track the location of the car.

Fig. 5.2: Example of IoT

If there is a common platform where all these things can connect to each other would be great because based on my preference, I can set the room temperature. For example, if I love the room temperature to to be set at 25 or 26degree Celsius when I reach back home from my office, then according to my car location, my AC would start before 10 minutes I arrive at home. This can be done through the Internet of Things (IoT).

5.2 How does Internet of things work?

The working of IoT is different for different IoT echo system (architecture). However, the key concept of there working is similar. The entire working process of IoT starts with the device themselves, such as smartphones, digital watches, electronic appliances, which securely communicate with the IoT platform. The platforms collect and analyse the data from all multiple devices and platforms and transfer the most valuable data with applications to devices.

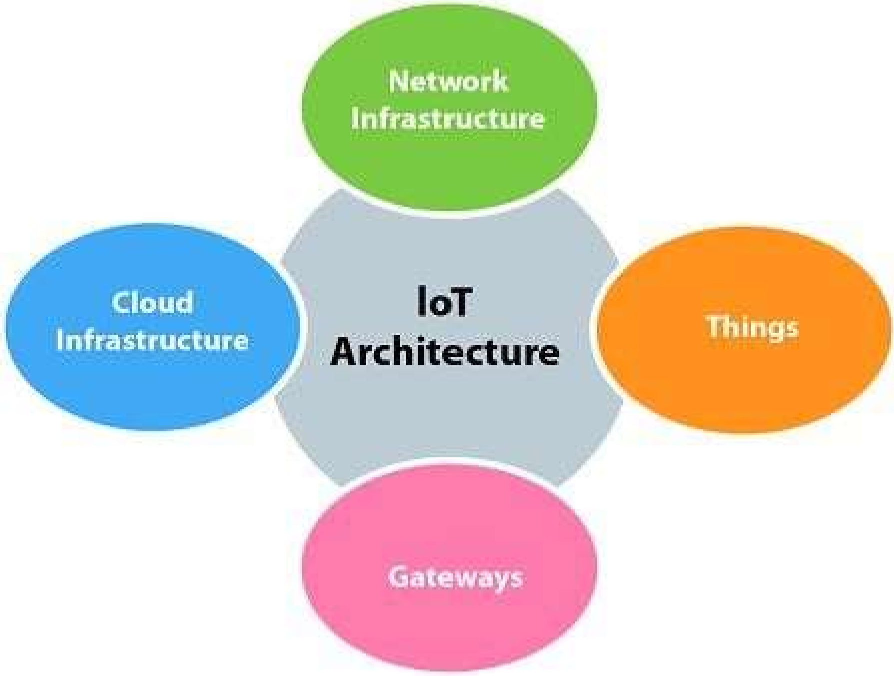

Fig. 5.3: IoT Architecture

5.3 Features of IOT:

The most important features of IoT on which it works are connectivity, analysing, integrating, active engagement, and many more. Some of them are listed below:

Connectivity: Connectivity refers to establish a proper connection between all the things of IoT-to-IoT platform it may be server or cloud. After connecting the IoT devices, it needs a high-speed messaging between the devices and cloud to enable reliable, secure and bi-directional communication.

Analyzing: After connecting all the relevant things, it comes to real-time analyzing the data collected and use them to build effective business intelligence. If we have a good insight into data gathered from all these things, then we call our system has a smart system.

Integrating: IoT integrating the various models to improve the user experience as well.

Artificial Intelligence: IoT makes things smart and enhances life using data. For example, if we have a coffee machine whose beans have going to end, then the coffee machine itself order the coffee beans of your choice from the retailer.

Sensing: The sensor devices used in IoT technologies detect and measure any change in the environment and report on their status. IoT technology brings passive networks to active networks. Without sensors, there could not hold an effective or true IoT environment.

Active Engagement: IoT makes the connected technology, product, or services to active engagement between each other.

Endpoint Management: It is important to be the endpoint management of all the IoT system otherwise, it makes the complete failure of the system. For example, if a coffee machine itself order the coffee beans when it goes to end but what happens when it orders the beans from a retailer and we are not present at home for a few days, it leads to the failure of the IoT system. So, there must be a need for endpoint management.

5.4 Advantages and Disadvantages of IOT:

Any technology available today has not reached to its 100 % capability. It always has a gap to go. So, we can say that **Internet of Things** has a significant technology in a world that can help other technologies to reach its accurate and complete 100 % capability as well. Let us look over the major, advantages, and disadvantages of the internet of things.

5.4.1 Advantages of IOT:

The Internet of Things (IoT) offers numerous advantages across various domains, including personal, industrial, and public sectors. Here are some key benefits:

1. **Improved Efficiency**: IoT devices can automate routine tasks, reducing the need for manual intervention. This leads to faster processes and the optimal use of resources. For example, smart thermostats can adjust temperatures based on occupancy, saving energy.

2. **Cost Savings**: By enabling more efficient operations and resource management, IoT can lead to significant cost savings. Predictive maintenance in industrial settings, for instance, helps prevent equipment failures and reduce downtime.

3. **Enhanced Data Collection and Analysis**: IoT devices generate large amounts of data, which can be analysed to gain insights into various processes. This data can be used to improve decision-making, optimize operations, and create new business models.

4. **Improved Customer Experience**: IoT can enhance the customer experience by providing personalized services and real-time information. For example, wearable devices can track health metrics and provide personalized health recommendations.
5. **Better Safety and Security**: IoT devices can monitor environments for potential safety hazards or security breaches. In smart homes, sensors can detect smoke or gas leaks, while in industrial settings, they can monitor equipment for signs of failure.
6. **Remote Monitoring and Control**: IoT allows for the remote monitoring and control of devices and systems. This is particularly useful in areas like healthcare, where remote monitoring of patients can improve care, and in agriculture, where farmers can monitor crops and livestock remotely.

5.4.2 Disadvantages of IOT:

While the Internet of Things (IoT) offers many advantages, it also presents several challenges and disadvantages. Here are some of the key concerns:

1. **Security Risks**: IoT devices are often vulnerable to cyberattacks, as many lack robust security features. Hackers can exploit these vulnerabilities to gain unauthorized access to personal data, disrupt services, or even control devices.
2. **Privacy Concerns**: The extensive data collection by IoT devices raises significant privacy issues. Personal and sensitive information can be collected, stored, and potentially misused by third parties, leading to concerns about data security and user consent.
3. **Complexity and Interoperability**: The IoT ecosystem consists of a wide variety of devices, platforms, and protocols. Ensuring that these different components can communicate and work together seamlessly is a major challenge. Lack of standardization can lead to compatibility issues and complicate system integration.
4. **High Costs**: The initial setup and maintenance of IoT systems can be expensive. This includes the cost of devices, sensors, infrastructure, and ongoing operational expenses. These costs can be prohibitive for individuals and small businesses.
5. **Data Overload**: IoT devices generate massive amounts of data, which can be overwhelming to manage and analyse. Organizations may struggle to extract valuable insights from this data, and the cost of storing and processing it can be substantial.

6. **Reliability and Dependence**: IoT systems can be prone to failures, whether due to hardware malfunctions, software bugs, or connectivity issues. The dependence on these systems for critical functions, such as healthcare or industrial processes, raises concerns about reliability and the potential consequences of system failures.

5.5 Embedded Devices System In IOT:

It is essential to know about the embedded devices while learning the IoT or building the projects on IoT. The embedded devices are the objects that build the unique computing system. These systems may or may not connect to the Internet.

An embedded device system generally runs as a single application. However, these devices can connect through the internet connection, and able communicate through other network devices.

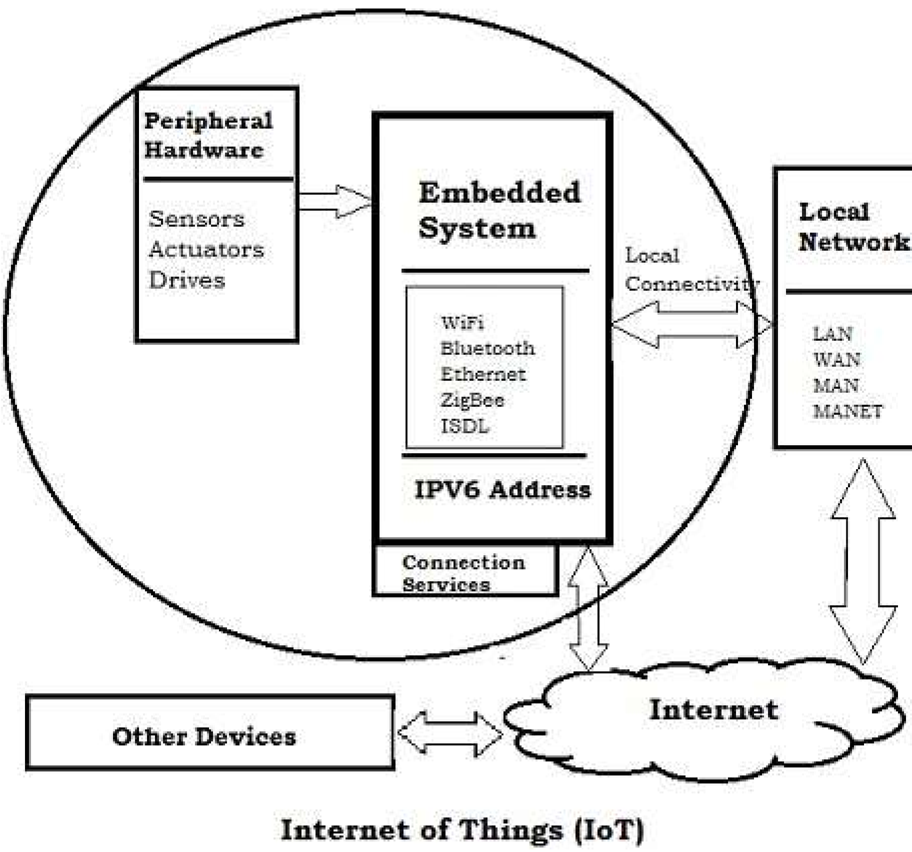

Fig. 5.4: Embedded Devices System in (IoT)

5.6 IOT Ecosystem:

The IoT ecosystem is not easy to define. It is also difficult to capture its proper image due to the vastness and emerging possibility and the rapidity with which it is expanding in the entire sector. However, the IoT ecosystem is a connection of various kind of devices that sense and analyse the data and communicates with each other over the networks. In the IoT ecosystem, the user uses smart devices such as smartphones, tablet, sensors, etc. to send the command or request to devices for information over the networks. The device response and performs the command to send information back to the user through networks after analysed. The typical IoT ecosystem is shown in below image, where the smarter devices send and receive data from the devices themselves in the environment that are integrate over network and Cloud Computing.

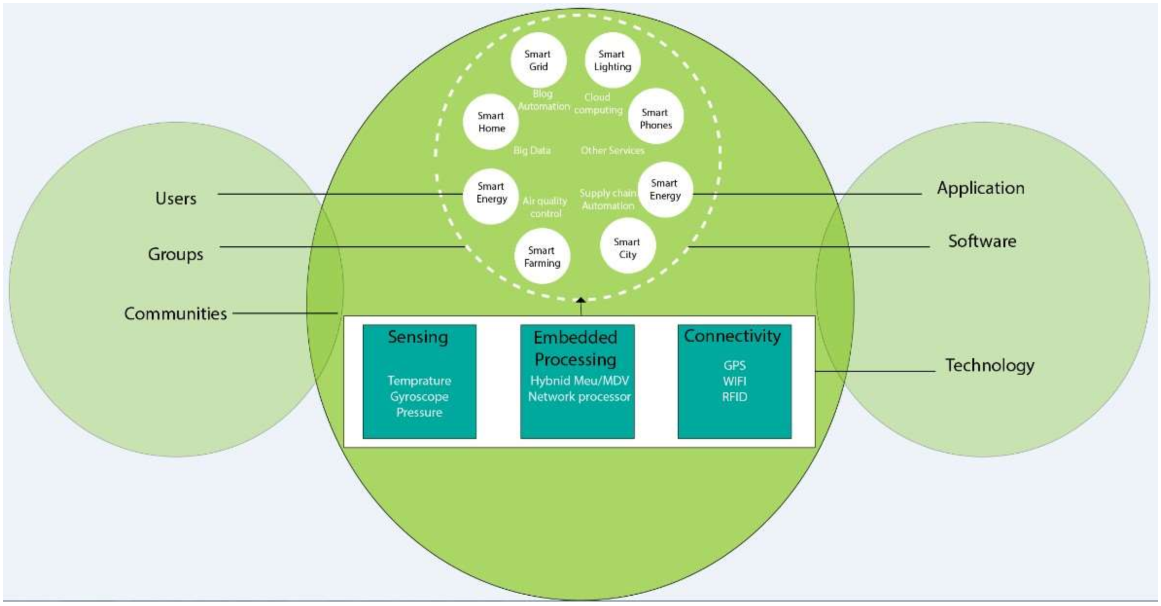

Fig. 5.5: IoT Ecosystem

The IoT is itself an ecosystem of network devices that transfer the data. It is also well interconnected with Big Data and Cloud Computing.

- **Sensing, Embedded processing, Connectivity:** The IoT ecosystem senses its surrounding like temperature, gyroscope, pressure, etc. and make the embedded processing using devices. These devices are connected through any type of devices such as GPS, WIFI, RFID, etc. over the networks.
- **Smart devices and environment, Cloud Computing, Big Data:** The data transfer or receive through smart devices and environments are communicated through Cloud Computing or others Servers and stored as Big Data.
- **Technology, Software, Application:** The IoT ecosystem uses any of different technologies, software, and application to communicate and connect with smart devices and environment.
- **Users or groups of community:** The product or services generated by the IoT ecosystem are consumed by the users or the group of communities to serve the smart life.

5.7 IOT Decision Framework:

The IoT decision framework provides a structured approach to create a powerful IoT product strategy. The IoT decision framework is all about the strategic decision making. The IoT Decision Framework helps us to understand the areas where we need to make decisions and ensures consistency across all of our strategic business decision, technical and more.

The IoT decision framework is much more important as the product or services communicates over networks goes through five different layers of complexity of technology.

1. Device Hardware
2. Device Software
3. Communications
4. Cloud Platform
5. Cloud Application.

Each of these decision areas is evaluated at each of the IoT Technology Stack. The User Experience will be evaluated at Device Hardware, Device Software and so to provide the better user experience. Then at the next step Data Decision Area, we must explore data considerations for all the stages of IoT Technology Stack.

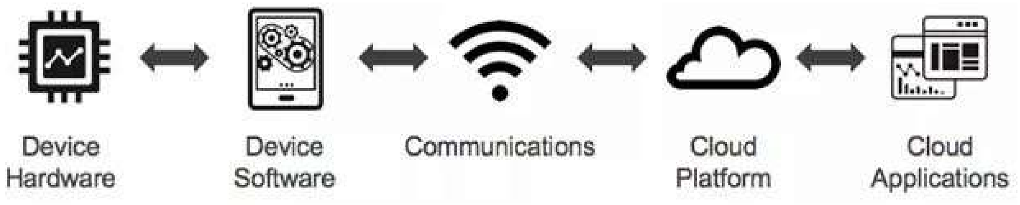

Fig. 5.6: The IoT Technology stack

5.8 IOT Architecture:

The IoT architecture differs from their functional area and their solutions. However, the IoT architecture technology mainly consists of four major components:

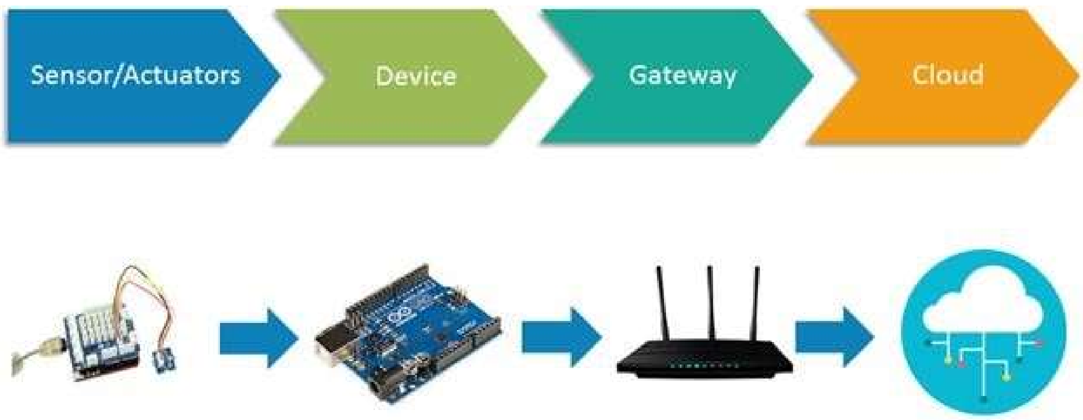

Fig. 5.7: Stages of IoT Solutions Architecture

CHAPTER 6

HARDWARE DESCRIPTION

An IoT-based system for paralysis patients typically includes various hardware components designed to monitor the patient's condition, facilitate communication, and assist with daily activities. Here is an overview of the key hardware components:

6.1 ADXL 335 MEM Body Fall Detection Sensor:

The ADXL335 is a 3-axis accelerometer sensor that is commonly used for detecting body movements, including falls. It operates on the principle of measuring acceleration forces, which can be due to gravity, movement, or vibration. Here is a breakdown of how the ADXL335 works, particularly in the context of body fall detection.

6.1.1 Working Principle:

1. Accelerometer Basics

- The ADXL335 measures acceleration in three dimensions: X, Y, and Z axes. Acceleration is the rate of change of velocity, which can be caused by movement or gravitational forces.
- The sensor provides analog voltage outputs proportional to the acceleration it experiences along each axis. These outputs are typically in the range of 0 to 3.3V or 0 to 5V, depending on the supply voltage.

2. MEMS Technology

- The ADXL335 is a MEMS (Micro-Electro-Mechanical Systems) device, which means it uses microscopic mechanical systems to detect changes in acceleration. These systems include small proof masses attached to springs. When the sensor experiences acceleration, the proof masses deflect, causing a change in capacitance that the sensor can measure.
- The change in capacitance is converted into an analog voltage output. These systems include small proof masses attached to springs. When the sensor experiences acceleration, the proof masses deflect, causing a change in capacitance that the sensor can measure.

3. Detecting Falls

- For fall detection, the ADXL335 monitors the acceleration values along the X, Y, and Z axes. During a fall, the sensor experiences a rapid change in acceleration, especially along the axis corresponding to the direction of the fall.
- By analyzing these changes, the sensor can detect characteristic patterns associated with falls

4. Signal Processing and Alerts

- Once the system detects a fall, it can trigger alerts or alarms. This might involve sending notifications to caregivers, triggering an audible alarm, or communicating with other IoT devices.

6.1.2 Key Characteristics:

- **Range**: The ADXL335 can measure acceleration up to ±3g (where g is the acceleration due to gravity).
- **Low Power Consumption**: It operates with low power, making it suitable for battery-powered applications.

6.1.3 ADXL335 Pin Configuration:

1. **Vcc (Power Supply)**
   - **Pin Function**: Power supply pin for the accelerometer.
   - **Description**: This pin supplies power to the ADXL335. The typical operating voltage range is 1.8V to 3.6V, with 3.3V being common. The voltage supplied here should be stable and regulated.
2. **GND (Ground)**
   - **Pin Function**: Ground pin.
   - **Description**: This is the reference ground for the sensor. It should be connected to the system ground.
3. **X_OUT (X-axis Analog Output)**
   - **Pin Function**: Analog output for the X-axis.

- **Description**: This pin provides an analog voltage proportional to the acceleration along the X-axis. The output voltage varies linearly with acceleration, centered around a midpoint voltage (typically 1.5V at 3.3V supply).

4. **Y_OUT (Y-axis Analog Output)**
   - **Pin Function**: Analog output for the Y-axis.
   - **Description**: Similar to X_OUT, this pin provides an analog voltage for the Y-axis acceleration.
5. **Z_OUT (Z-axis Analog Output)**
   - **Pin Function**: Analog output for the Z-axis.
   - **Description**: Provides the analog output for the Z-axis acceleration.
6. **ST (Self-Test)**
   - **Pin Function**: Self-test pin.
   - **Description**: This pin is used to activate the self-test feature of the accelerometer. Applying a specific voltage (usually Vcc) to this pin enables the self-test mode, which applies an electrostatic force to the internal sensors to verify their operation.

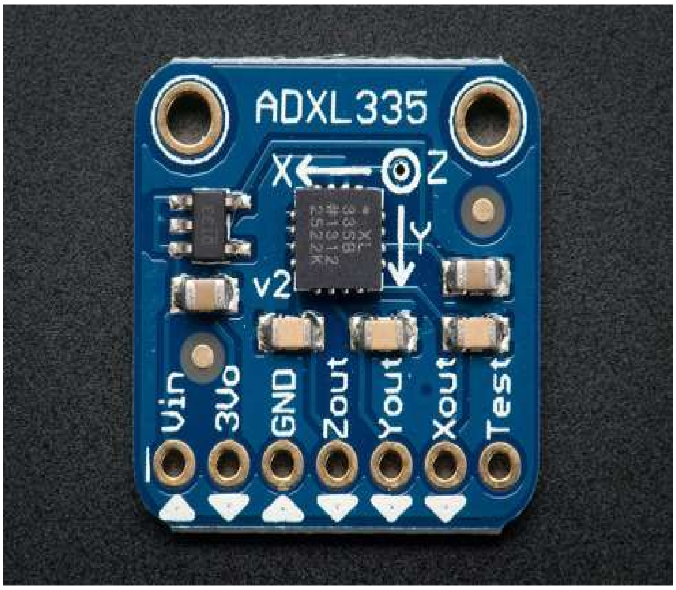

Fig. 6.1: ADXL 335 Body fall detection sensor

6.2 LM 35 Temperature Sensor:

The LM35 is a precision integrated-circuit temperature sensor whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. It's commonly used in various applications, including temperature measurement and control systems. The sensor has several advantages, including its simplicity, accuracy, low cost, and ease of interfacing with microcontrollers or other digital circuits.

6.2.1 Working Principle of LM35

1. Semiconductor-Based Sensing:

- The LM35 is a semiconductor-based temperature sensor. It utilizes the temperature-dependent characteristics of semiconductor junctions. Specifically, the voltage across a diode changes with temperature, which is used to generate a temperature-proportional output.

2. Output Voltage:

- The sensor outputs an analog voltage that is directly proportional to the temperature in degrees Celsius. The output voltage is typically 10 millivolts per degree Celsius (mV/°C). For example, at 25°C, the output voltage would be 250 mV.

3. Linear Relationship:

- The relationship between the temperature and the output voltage is linear, which simplifies the conversion from voltage to temperature. This linear output is one of the key features of the LM35, making it easy to interface with microcontrollers and analog-to-digital converters (ADCs).

4. Calibration and Accuracy:

- The LM35 is factory-calibrated for accuracy. It has a typical accuracy of ±0.5°C at room temperature and ±1°C over a full range of temperatures. This makes it suitable for applications where precise temperature measurements are required.

5. No Need for External Calibration:

- Unlike some temperature sensors, the LM35 does not require external calibration or trimming. It can be used right out of the box with a guaranteed level of accuracy.

6. Operating Range:

- The LM35 can operate over a wide temperature range, typically from -55°C to +150°C, making it versatile for various applications.

7. **Low Self-Heating:**

   - The sensor has low self-heating, which means it does not significantly affect the temperature it measures. This is particularly important in applications requiring high accuracy.

6.2.2 Pin Configuration of LM35 Temperature Sensor:

1. Pin 1 (VOUT) - Output:

- This pin provides the analog voltage output proportional to the temperature. The output voltage is typically 10 mV per degree Celsius (10 mV/°C).

2. Pin 2 (VCC) - Supply Voltage:

- This pin is connected to the positive supply voltage. The operating voltage range for the LM35 is typically from 4V to 30V. This pin powers the sensor.

3. Pin 3 (GND) - Ground:

- This pin is connected to the ground (0V). It serves as the reference point for the output voltage and power supply.

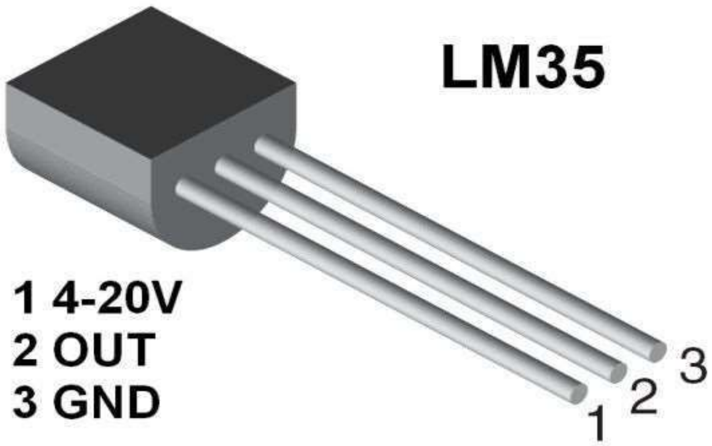

Fig. 6.2: LM35 Temperature Sensor

6.3 Power Supply Board:

A power supply board is a circuit board or module designed to convert input power from a source (such as mains electricity or batteries) into a stable output voltage or current suitable for powering electronic devices or systems. These boards often incorporate components such as transformers, rectifiers, voltage regulators, and filtering capacitors to achieve this conversion while providing protection against voltage spikes, overcurrent, and other potential issues. They can be found in a wide range of applications, from consumer electronics to industrial equipment.

6.3.1 Working Principle of a Power Supply Board:

1. Transformation:

- **Purpose:** The transformation stage changes the input voltage to a desired level suitable for the electronic components. This is often necessary because the mains supply voltage (e.g., 120V or 240V AC) is too high for most electronics.
- **Process:** A transformer is used to step down (or occasionally step up) the voltage. It works on the principle of electromagnetic induction, where an alternating current (AC) in the primary winding induces a magnetic field that generates a voltage in the secondary winding.

2. Rectification:

- **Purpose:** Rectification converts the AC voltage from the transformer into a pulsating direct current (DC) voltage.
- **Process:** Diodes are used in various configurations (e.g., half-wave, full-wave, or bridge rectifier) to allow current to flow in only one direction, effectively blocking the negative half-cycles of the AC input.

3. Filtering:

- **Purpose:** Filtering smooths out the pulsating DC voltage produced by rectification to reduce ripples and produce a more constant DC output.
- **Process:** Capacitors are the primary components used in this stage. They store charge during the peaks of the rectified voltage and release it during the troughs, smoothing the output. Inductors may also be used to filter out high-frequency noise.

4. Regulation:

- **Purpose:** The regulation stage maintains a constant output voltage despite variations in the input voltage or changes in the load (the amount of current drawn by the connected devices).
- **Process:** Voltage regulators (linear regulators like the 78xx series or switching regulators) are used to provide a stable output voltage. Linear regulators dissipate excess power as heat, while switching regulators convert power more efficiently by rapidly switching on and off and using energy storage elements (inductors and capacitors).

5. Protection:

- **Purpose:** Protection mechanisms ensure the safety of the power supply and connected devices.
- **Process:** This can include fuses, circuit breakers, thermal shutdowns, and overvoltage or overcurrent protection circuits. These components disconnect the power or shut down the system in case of faults or dangerous conditions.

6.3.2 Components of power supply board:

1. Bridge rectifier

2. Filter capacitor

3. 7805 Regulator

4. 1 Led

1. Bridge Rectifier: A bridge rectifier is an electronic component used to convert alternating current (AC) into direct current (DC). It consists of four diodes arranged in a bridge configuration. The input AC voltage is applied across the diagonally opposite ends of the bridge, while the output DC voltage is taken from the remaining two ends.

The bridge rectifier operates by allowing current to flow through the diodes in one direction during the positive half-cycle of the input AC voltage and through the other diodes during the negative half-cycle. This arrangement effectively rectifies the AC voltage, producing a pulsating DC output. Additional filtering components such as capacitors are often used downstream to smooth out the pulsations and obtain a more stable DC output.

Bridge rectifiers are commonly used in power supply circuits for various electronic devices and appliances where a steady DC voltage is required from an AC power source.

Fig. 6.3: Bridge Rectifier

2. Filter Capacitor: A filter capacitor, also known as a smoothing capacitor or reservoir capacitor, is an electronic component used in power supply circuits to reduce or eliminate the ripple voltage present in the rectified output of the power supply. When AC voltage is rectified to DC by a bridge rectifier, the resulting waveform has a pulsating nature, which means there are fluctuations in the voltage level. A filter capacitor is connected across the output of the rectifier to smooth out these fluctuations by storing electrical charge during the peaks of the voltage waveform and discharging it during the troughs. This action effectively reduces the ripple voltage, resulting in a more stable DC output voltage.

Filter capacitors are typically large electrolytic capacitors with high capacitance values, capable of storing and discharging significant amounts of charge. They are usually placed in parallel with the load or between the rectifier output and ground in a power supply circuit. In summary, filter capacitors play a crucial role in power supply circuits by improving the quality of the DC output voltage, making it suitable for powering electronic devices and systems without unwanted fluctuations.

A filter capacitor is connected across the output of the rectifier to smooth out these fluctuations by storing electrical charge during the peaks of the voltage waveform and discharging it during the troughs. This action effectively reduces the ripple voltage, resulting in a more stable DC output voltage.

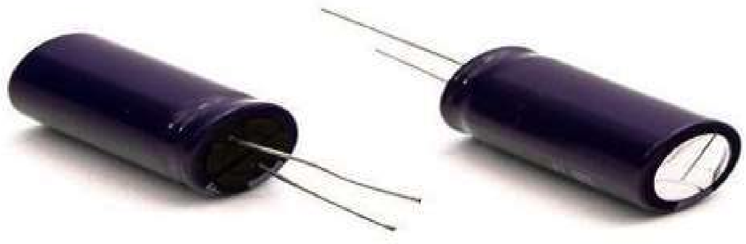

Fig. 6.4: Filter Capacitor

3. 7805 Regulator: The 7805 is a popular linear voltage regulator integrated circuit (IC) commonly used in electronic circuits to provide a stable and fixed output voltage. Here are some key features and details about the 7805-voltage regulator:

Output Voltage: The 7805 regulator produces a fixed output voltage of +5 volts DC.

Input Voltage Range: It typically accepts input voltages ranging from around 7 volts to 25 volts DC. However, it can handle higher input voltages with appropriate heat sinking.

Regulation: The 7805 provides excellent regulation, meaning it maintains a constant output voltage despite variations in the input voltage or load conditions.

Current Capacity: The maximum output current capacity of the 7805 varies depending on the package and manufacturer. Common versions can handle currents up to 1 ampere or more, but this capacity may be reduced if the regulator is not adequately cooled.

Thermal Protection: Some variants of the 7805 include built-in thermal overload protection, which shuts down the regulator if it becomes too hot, preventing damage to the device.

Fixed Output Voltage: Provides a constant 5V output regardless of variations in input voltage or output load, within its specified limits.

Input Voltage Range: Typically requires an input voltage of 7V to 35V. The minimum input voltage must be at least a couple of volts higher than the output voltage to ensure proper regulation (usually 2V above the output, hence a minimum of around 7V).

Thermal Protection: Some variants of the 7805 include built-in thermal overload protection, which shuts down the regulator if it becomes too hot, preventing damage to the device.

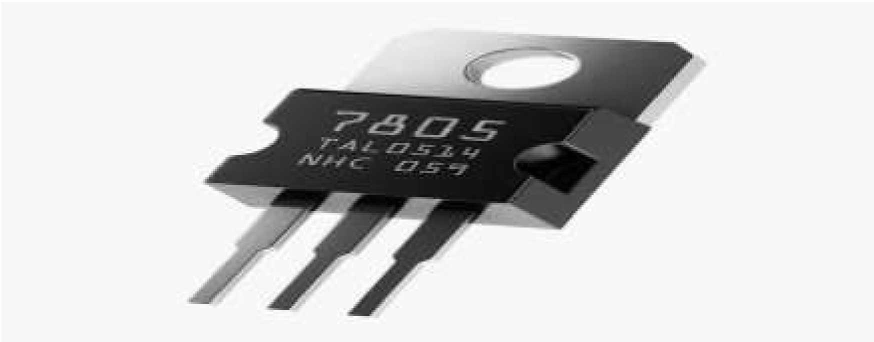

Fig. 6.5: 7805 Regulator

4. LED: A light-emitting diode (LED) is a semiconductor device that emits light when an electric current passes through it. LEDs are commonly used in various applications, including indicator lights, displays, and lighting fixtures, due to their energy efficiency, long lifespan, and compact size.

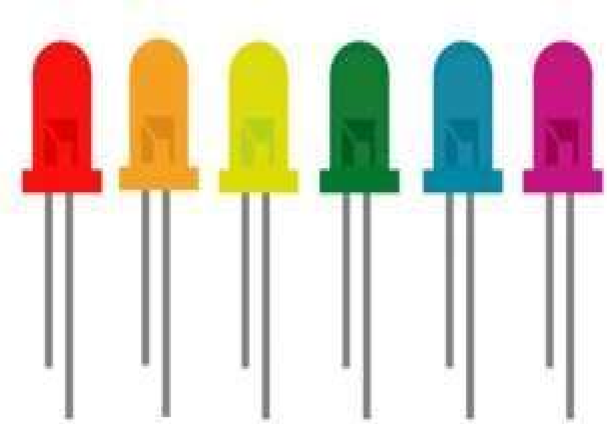

Fig. 6.6: Led

6.4 ECG Sensor:

An ECG (Electrocardiogram) sensor is a device used to monitor the electrical activity of the heart over time. It detects and records the electrical signals generated by the heart as it beats, providing valuable information about the heart's rhythm and function. ECG sensors typically consist of electrodes that are attached to the skin, which pick up the electrical signals and transmit them to a monitoring device or system for analysis.

6.4.1 Working Principle of ECG sensor:

1. Electrode Placement:

- Purpose: Electrodes are placed on specific points on the body to detect the electrical signals generated by the heart.
- Process: The most common placement is on the limbs or chest, where the electrical signals are strongest and can be picked up clearly. The electrodes are typically conductive pads that make good electrical contact with the skin.

2. Signal Acquisition:

- Purpose: The primary function of the ECG sensor is to detect the biopotential signals from the heart.
- Process: The heart generates an electrical signal as it beats. This signal travels through the body and can be detected by the electrodes. The electrical potential differences between pairs of electrodes (leads) are measured to obtain the ECG signal.

3. Amplification:

- Purpose: The electrical signals generated by the heart are very small (microvolts to millivolts). Therefore, amplification is necessary to make these signals usable.
- Process: The ECG sensor includes an amplifier circuit that boosts the signal's strength. The amplification must be done with high precision and low noise to ensure the quality of the recorded signal.

4. Filtering:

- Purpose: To remove noise and interference from the ECG signal, which can come from various sources, including muscle contractions, electromagnetic interference, and baseline wander.

- Process: Filters (high-pass, low-pass, and notch filters) are applied to the amplified signal to eliminate unwanted frequencies, ensuring that the final signal represents only the heart's electrical activity.

5. Analog-to-Digital Conversion (ADC):

- Purpose: Convert the analog ECG signal into a digital format for processing, analysis, and storage.
- Process: The amplified and filtered signal is fed into an ADC, which converts the continuous analog signal into a digital signal that can be interpreted by digital devices such as computers or microcontrollers.

6.4.2 Pin configuration of ECG Sensor:

1. Ground (GND): This pin connects to the ground or reference potential of the circuit.

2. Signal Output: This pin provides the ECG signal output, typically in the form of analog voltage or digital data

3. Positive Power Supply (+V): This pin is connected to the positive power supply voltage, usually 3.3V or 5V, depending on the sensor's requirements.

4. Negative Power Supply (-V or GND): Some ECG sensors may have a separate pin for the negative power supply, while others may connect it directly to the ground pin.

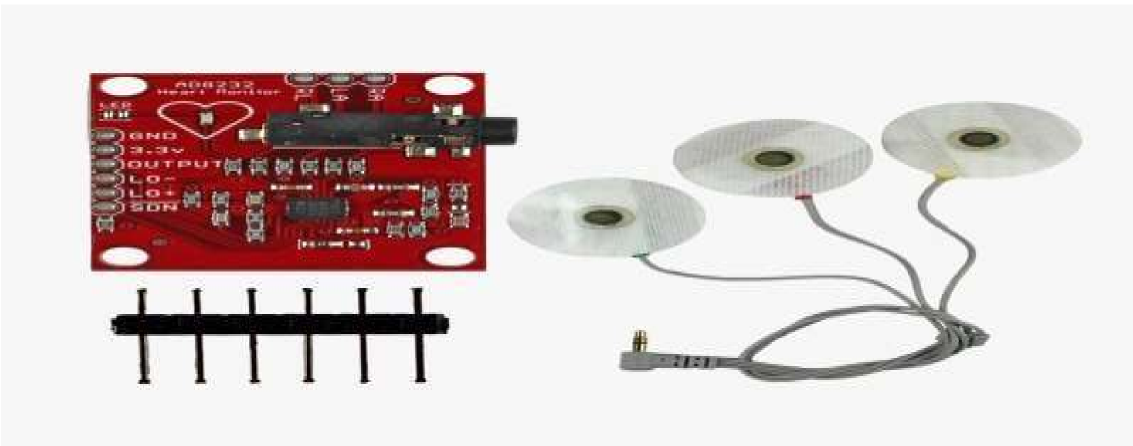

Fig. 6.7: ECG Sensor

6.5 GSM 900A Module:

The GSM 900A module is a GSM (Global System for Mobile Communications) module designed for communication in the 900 MHz frequency band. It is used for a variety of applications, including mobile communication, data transmission, and embedded systems integration. Below is an overview of the GSM 900A module, including its pin configuration, features, and typical applications.

6.5.1 Performance characteristics of GSM 900A:

1. Power Supply and Initialization:

- Power Supply: The GSM module requires a stable power supply (typically 3.4V to 4.5V). It uses this power to operate its internal circuits and communicate with the GSM network.
- Initialization: Upon powering up, the module initializes its internal systems and establishes a connection with the GSM network. This includes registering on the network and acquiring network time.

2. Communication with GSM Network:

- Network Registration: The module communicates with nearby GSM towers to register itself on the network. It sends and receives signals to ensure connectivity and obtain network parameters.
- Frequency Band: The GSM 900A module operates on the 900 MHz frequency band, which is used for GSM communication in many regions around the world.

3. Sending and Receiving Data:

- Voice Calls: To make a voice call, the module sends a request to the network with the phone number. The network connects the call and handles the audio communication, which is transmitted through the module's antenna.
- SMS Messages: For sending an SMS, the module formats the message data and sends it to the network. The network then delivers the message to the recipient's phone.
- Data Transmission: The module can also facilitate GPRS (General Packet Radio Service) for data communication, allowing it to send and receive data packets.

4. UART Communication:

- Serial Communication: The module interfaces with a microcontroller or other devices via UART (Universal Asynchronous Receiver/Transmitter). It uses TX (Transmit) and RX (Receive) pins to send and receive serial data.
- Commands and Responses: The microcontroller sends AT commands (Attention commands) to the GSM module to control its functions (e.g., dialing a number, sending an SMS). The module responds with status information and data.

5. Signal Processing:

- Signal Amplification: The module amplifies the weak signals received from the antenna and processes them to extract data. It also modulates and demodulates signals for transmission and reception.
- Error Correction: The module employs error detection and correction techniques to ensure the integrity of data transmitted over the network.

6.5.2 Pin configuration of GSM 900A:

1. TX (Transmit Data):

- Transmit pin used to send data from the module to the microcontroller or other receiving device. Connect this pin to the RX (Receive) pin of the microcontroller.

2. RX (Receive Data):

- Receive pin used to receive data from the microcontroller or other sending device. Connect this pin to the TX (Transmit) pin of the microcontroller.

3. DTR (Data Terminal Ready):

- Used to indicate the readiness of the module to communicate. It can be used for flow control in serial communication, though it might be optional in some designs.

4. RI (Ring Indicator):

- Signals incoming calls. This pin is used to notify the module of incoming call events. It might be optional or used depending on your application requirements.

5. CTS (Clear to Send):

- Flow control line used in serial communication. The module uses this line to indicate whether it is ready to receive data. It might be optional depending on your setup.

6. RTS (Request to Send):

- Flow control line used in serial communication. It is used by the microcontroller to signal the module to prepare for data reception. This pin might also be optional.

7. NC (Not Connected):

- This pin is not connected to any internal circuitry of the module. It can be left unconnected or used for specific functions in some designs.

8. ANT (Antenna):

- Connects to the external GSM antenna. Proper antenna connection is crucial for reliable GSM communication.

Fig. 6.8: GSM 900A

6.6 16*2 LCD Display:

A 16x2 LCD display is a liquid crystal display (LCD) module with the capability to display 16 characters per line and 2 lines of text. Here is a brief overview of its features and components.

6.6.1 Working Principle of 16*2 lcd display: The working principle of a 16x2 LCD display involves several key processes to display text and control information. Here's a detailed explanation of how a 16x2 LCD operates:

1. Initialization:

- Power Up: Upon powering up, the LCD initializes. This includes configuring internal registers and settings.
- Configuration Mode: Set the mode of operation (e.g., 8-bit or 4-bit) and initialize display parameters (e.g., display on/off, cursor control).

2. Command Mode:

- RS Pin LOW: When the RS pin is LOW, the LCD interprets incoming data as commands rather than character data.
- Sending Commands: Commands control the LCD's behaviour, such as clearing the display, setting cursor position, and defining display mode.
- E Pin Pulsing: To execute a command, a pulse is sent to the E (Enable) pin, which tells the LCD to read the command from the data lines.

3. Data Mode:

- RS Pin HIGH: When the RS pin is HIGH, the LCD interprets incoming data as character data.
- Sending Data: Characters are sent to the display by setting the data lines (D0-D7) and pulsing the E pin to latch the character data into the LCD.
- Displaying Characters: The LCD controller converts the character codes into corresponding patterns on the liquid crystal layer.

4. Contrast and Backlight:

- Contrast Adjustment (V0): Adjusted using a potentiometer to ensure characters are clearly visible.
- Backlight Control (A and K): Provides illumination for the display. Connect the anode (A) to the positive supply and cathode (K) to ground.

5. Data Transfer:

- 8-bit Mode: Data is sent in 8-bit chunks using the D0-D7 pins.
- 4-bit Mode (Optional): Data is sent in two 4-bit nibbles, reducing the number of data lines needed.

6.6.2 Pin Configuration of 16*2 lcd display: The pinout of a typical 16x2 LCD display module with a parallel interface (using the HD44780 controller or compatible) is as follows:

1. **VSS (Ground):** Connect this pin to ground (0V).
2. **VDD (Power Supply):** Connect this pin to a positive power supply voltage (usually +5V).
3. **VO (Contrast Adjustment):** This pin is used to adjust the contrast of the display. Connect it to a potentiometer or a voltage source to adjust the contrast.
4. **RS (Register Select):** This pin is used to select between data and command modes. Connect it to a digital output pin of the microcontroller.
5. **RW (Read/Write):** This pin is used to control the direction of data transfer. Connect it to ground to set the module in write mode.
6. **E (Enable):** This pin is used to enable data transfer. Connect it to a digital output pin of the microcontroller.
7. **D0-D7 (Data Lines):** These are the eight data lines used for transferring data between the microcontroller and the LCD display. Connect them to digital output pins of the microcontroller.
8. **A (Anode, Backlight Positive):** Connect this pin to the positive terminal of the backlight LED (if present).
9. **K (Cathode, Backlight Negative):** Connect this pin to the negative terminal.

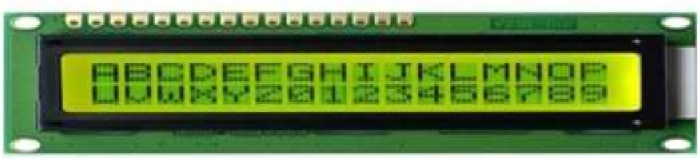

Fig. 6.9: 16*2 LCD Display

6.7 FLEX/BEND Sensor:

A Flex Sensor or sometimes called as Bend Sensor is a device that measures the amount of bend or angular deflection. Usually, a Flex Sensor is made up of a variable resistive surface and the amount of resistance is varied by bending the sensor. The Flex Sensor used in this project is shown in the following image. It is about 0.6cm wide and 8cm long (including the two connectors).

Since a flex sensor is basically a resistor (whose resistance varies depending on its bend), it has two terminals (or leads). Coming to the variable resistance part, an unflexed sensor i.e. a flex sensor as rest, exhibits a normal resistance value. In my case, the normal resistance of the Flex Sensor is around 60KΩ. When I bend the sensor as shown in the following image, the resistance increases with the increase in the bend angle.

6.7.1 Basis Flex Sensor Circuit:

Since the underlying concept of Flex Sensor is a variable resistor, you can easily guess the Basic Flex Sensor Circuit that can be implemented. It is a simple voltage divider circuit. The following image shows the basic flex sensor circuit consisting of a voltage divider (formed by the flex sensor itself and a 10KΩ Resistor) and an Operational Amplifier (Op-Amp).

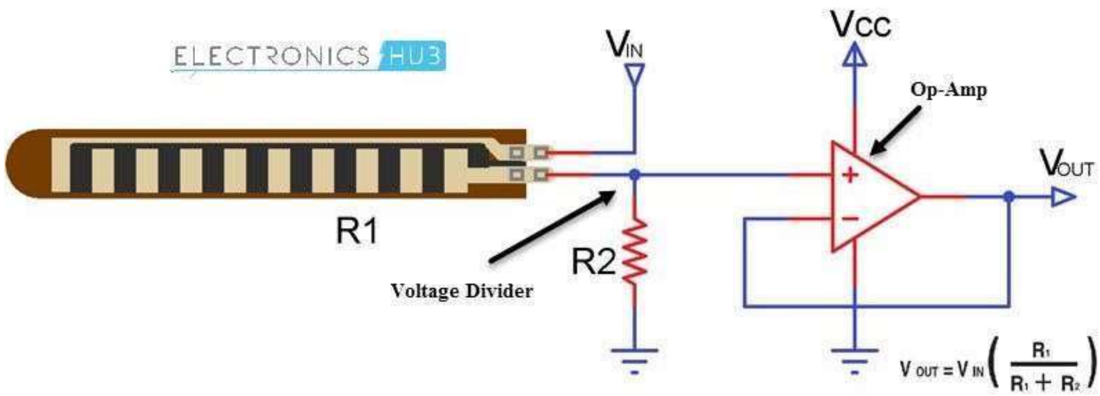

Fig. 6.10: Basis Flex Sensor Circuit

The resistance values mentioned here are specific to the Flex Sensor which I am using and they are not universal.

It is a simple voltage divider circuit. The following image shows the basic flex sensor circuit consisting of a voltage divider (formed by the flex sensor itself and a 10KΩ Resistor) and an Operational Amplifier (Op-Amp).

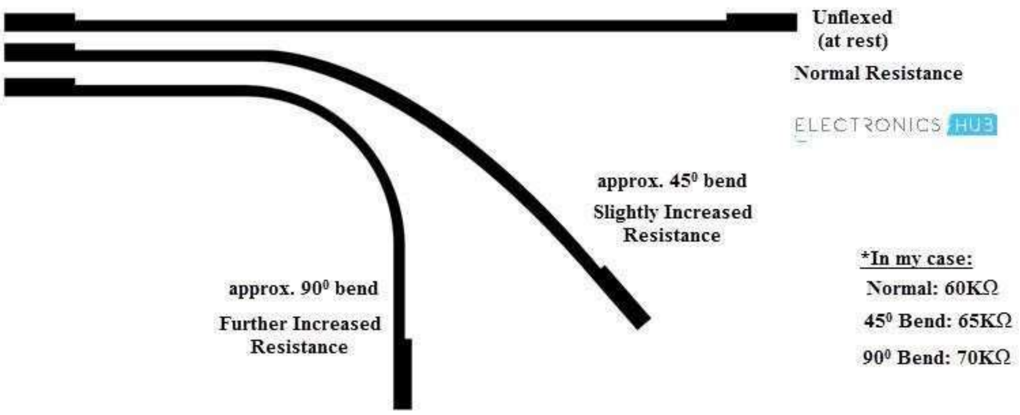

Fig. 6.11: Flex Sensor

6.7.2 Working Principle:

1. Structure of the Flex Sensor: A flex sensor consists of several key components:

- **Flexible Substrate:** A bendable base material that allows the sensor to flex without damage.
- **Conductive Material:** A layer of conductive ink or material is printed or deposited on the substrate. This layer forms a resistive path that changes its resistance when bent.
- **Electrodes:** Metal or conductive pads at the ends of the sensor that connect the conductive layer to external circuits.

2. How Flex Sensors Work

- **Unbent State:**
  - In the unbent or neutral position, the flex sensor has a certain baseline resistance, which is relatively stable.
- **Bending or Flexing:**

- When the sensor is bent, the conductive layer stretches on the outer side of the bend and compresses on the inner side. This stretching and compression alter the resistance of the conductive path.
- **Resistance Change:** As the sensor bends, the resistance increases due to the elongation and thinning of the conductive material. The amount of resistance change is proportional to the degree of bending.

3. Resistance Measurement:

- The sensor is typically connected in a voltage divider circuit or similar setup to measure the change in resistance. A change in resistance results in a change in voltage, which can be read by an analog-to-digital converter (ADC) or microcontroller.

6.7.3 Pin configuration of flex sensor:

1. (Signal Pin): This pin is connected to one end of the flex sensor and is typically connected to an input on a microcontroller or an analog-to-digital converter (ADC) to read the variable voltage.

2. (Ground Pin): This pin is connected to the other end of the flex sensor and is connected to a ground or reference point in the circuit.

Fig. 6.12: Flex sensor output

6.8 ARP 33A3 Voice Module:

The ARP33A3 voice module is a device designed to provide voice playback and recognition capabilities for various applications. It is often used in projects where audio feedback or voice commands are required. Here's an overview of its features, working principle, and typical applications.

6.8.1 Working Principle of ARP 33A3 Voice Module:

1. Power Supply:

- VCC: The module is powered by supplying a voltage to the VCC pin, typically +3.3V or +5V depending on the module's specifications.
- GND: The ground pin (GND) provides the return path for the electrical current.

2. Voice Data Storage:

- Internal Memory: The module contains internal memory where voice data is stored. This memory may be flash memory or another type of non-volatile storage.
- Pre-Programming: Voice data can be pre-recorded and programmed into the module at the factory, or it can be loaded via a serial interface if the module supports updating its memory.

3. Control and Communication:

- Serial Interface:
  - TX (Transmit): Used to send commands and data to the module.
  - RX (Receive): Used to receive data and status from the module.
- Commands: Commands are sent through the serial interface to control the module's functions, such as playing a specific voice message, stopping playback, or configuring settings.
- Control Pins: Additional pins may be used for specific functions, such as PLAY/STOP for controlling playback or RESET for restarting the module.

4. Voice Playback:

- Command Execution: To play a voice message, a command is sent to the module specifying which message to play. The module processes this command and starts playback.

- Audio Output: The voice message is output through the VOICE OUT pin, which is connected to an external speaker or audio system.
- Playback Control: The module can start, pause, or stop playback based on the commands received. Some modules allow for control of playback speed or volume.

5. Voice Recognition (if supported):

- Voice Processing: If the module includes voice recognition features, it processes incoming audio signals to match them with pre-stored voice commands or patterns.
- Command Execution: When a recognized command is detected, the module can trigger specific actions or responses based on the programmed logic**.**

6.8.2 Pin Configuration of ARP 33A3 Voice Module:

1. VCC (Pin 1):

- Function: Power supply input.
- Typical Voltage: +3.3V or +5V (check specific module specifications).

2. GND (Pin 2):

- Function: Ground connection.

3. TX (Pin 3):

- Function: Serial Transmit pin. Used to send commands and data to the module from a microcontroller or other serial device.

4. RX (Pin 4):

- Function: Serial Receive pin. Used to receive data and status from the module.

5. PLAY/STOP (Pin 5):

- Function: Control pin to start or stop voice playback. This pin may be used to trigger playback or halt it based on the module's specific design.

6. VOICE OUT (Pin 6):

- Function: Audio output pin. Connects to a speaker or audio system to output the voice playback.

7. RESET (Pin 7):

- Function: Reset pin. Used to reset the module and reinitialize it if necessary.

8. ADDR (Pin 8):

- Function: Address pin for selecting specific voice files or commands. This pin might be used to address different voice messages or settings.

9. CONTROL (Pin 9):

- Function: Additional control pins for managing various functions of the module. The exact use of these pins can vary based on the module's specific features and firmware.

Fig. 6.13: ARP 33A3 Voice Module

6.9 Arduino AT MEGA 2560:

The Arduino Mega 2560 is a popular microcontroller board based on the ATmega2560 chip. It is designed to provide a high level of flexibility and functionality for a wide range of electronics projects. Here is an overview of the Arduino Mega 2560**.**

The ATmega2560 is a powerful 8-bit microcontroller with a rich set of features, including ample memory, numerous I/O ports, and versatile communication interfaces. It is widely used in various applications, particularly where complex control and extensive I/O are required. Its extensive pin configuration supports a broad range of functions, making it a flexible choice for embedded system designers.

6.9.1 Working Principle:

1. Microcontroller Core:

- The ATmega2560 has an 8-bit AVR core that executes instructions from its Flash memory. The CPU fetches, decodes, and executes instructions to perform various tasks.

2. Memory Management:

- Flash Memory: Stores the program code.
- SRAM: Used for temporary data storage during program execution.
- EEPROM: Stores non-volatile data that should persist between resets.

3. Input/Output Operations:

- Digital I/O: The ATmega2560 can read and write digital signals via its I/O ports. Each port consists of multiple pins that can be configured as input or output.
- Analog I/O: The analog-to-digital converter (ADC) allows the microcontroller to read analog signals from sensors or other sources.

4. Communication Interfaces:

- USART: Allows serial communication with other devices or computers.
- SPI: Provides high-speed synchronous serial communication with peripherals.
- TWI/I2C: Facilitates communication with devices using the I2C protocol.

5. Timers and Counters:

- The microcontroller has several timers and counters for tasks such as generating PWM signals, timing events, or counting pulses.

6. Interrupts:

- The ATmega2560 can respond to external events or internal conditions by triggering interrupts, which temporarily halt the current execution to handle the interrupt service routine.

6.9.2 Pin Configuration of Arduino AT MEGA 2560:

1. Power and Ground Pins:

- VIN (Pin 1): Input voltage to the board when using an external power source (7-12V).
- GND (Pins 2, 22, 39, 61, 79, 99): Ground connections.
- 5V (Pin 3): Regulated 5V output from the board.
- 3.3V (Pin 4): Regulated 3.3V output from the board (limited current).
- AREF (Pin 5): Analog reference pin for the analog-to-digital converter (ADC).
- RESET (Pin 6): Reset pin to restart the board.

2. Digital I/O Pins:

- D0-D13 (Pins 7-20): 14 digital I/O pins (D0-D13), where D0 and D1 are also used for Serial communication (TX/RX).
- D14-D19 (Pins 21-26): Additional digital I/O pins.
- D20-D21 (Pins 27-28): More digital I/O pins.
- D22-D29 (Pins 29-36): Additional digital I/O pins.
- D30-D37 (Pins 37-44): More digital I/O pins.
- D38-D53 (Pins 45-62): Additional digital I/O pins with PWM functionality.

3. Analog Input Pins:

- A0-A15 (Pins 63-78): 16 analog input pins for reading analog signals.

4. Communication Pins:

- Serial Communication (TX/RX):

o TX0/RX0 (Pins 14/15): UART Serial communication (Serial0).

o TX1/RX1 (Pins 16/17): UART Serial communication (Serial1).

o TX2/RX2 (Pins 18/19): UART Serial communication (Serial2).

o TX3/RX3 (Pins 20/21): UART Serial communication (Serial3).

- SPI Communication:

o MISO (Pin 50): Master in Slave Out.

- MOSI (Pin 51): Master Out Slave In.
- SCK (Pin 52): Serial Clock.
- SS (Pin 53): Slave Select.
- SDA (Pin 20): Data Line.
- SCL (Pin 21): Clock Line.

5. Timer and PWM Pins:

- PWM Pins: (Pins capable of PWM output) D2, D3, D5, D6, D7, D8, D9, D10, D11, D12, D13.
- Timer Pins:
  - Timer0: D5, D6.
  - Timer1: D9, D10.
  - Timer2: D3, D11.

6. Special Function Pins:

- XTAL1/XTAL2 (Pins 15 and 16): Connect the external crystal oscillator or clock.
- IOREF (Pin 62): Provides the voltage reference for the I/O pins.

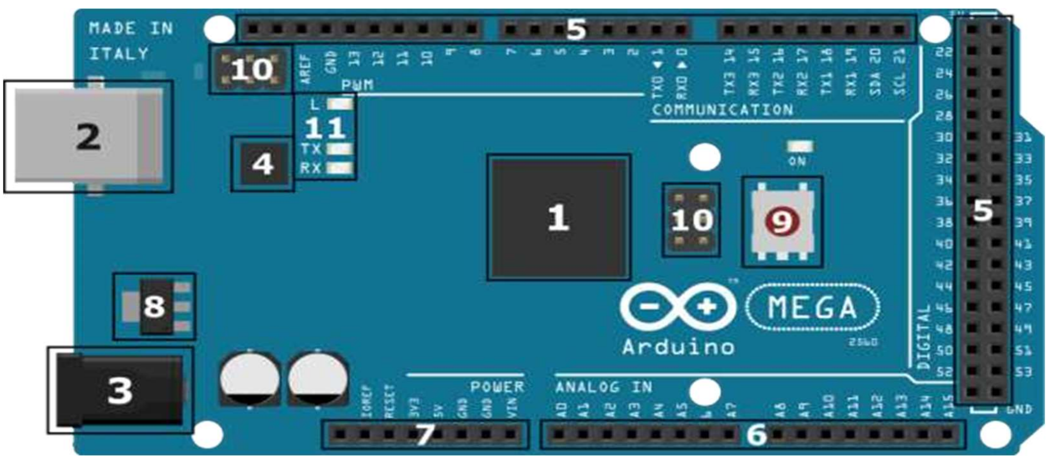

Fig. 6.14: Arduino AT MEGA 2560 Microcontroller

6.10 8 Ohm Speaker:

An 8-ohm speaker is a common type of speaker used in various electronic projects and audio systems. Here is an overview of its characteristics, working principle, and common applications:

6.10.1 Working Principle:

1. Audio Signal:

- The speaker receives an audio signal from an amplifier. This signal is typically an alternating current (AC) that varies in amplitude and frequency.

2. Electromagnetic Interaction:

- Voice Coil: Inside the speaker, there is a voice coil connected to the speaker cone. The voice coil is a coil of wire that generates a magnetic field when an electric current flows through it.
- Magnet: A permanent magnet is fixed to the speaker frame. It creates a static magnetic field.

3. Sound Production:

- When the audio signal passes through the voice coil, it creates a varying magnetic field that interacts with the static field of the permanent magnet.
- This interaction causes the voice coil to move back and forth.
- The movement of the voice coil causes the attached speaker cone to vibrate.
- These vibrations produce sound waves, which travel through the air and are heard as audio.

6.10.2 Pin Configuration of 8-ohm Speaker:

1. Positive Terminal (+): The terminal connected to the positive output of the audio amplifier. It is usually marked with a red dot, "+" symbol, or another indicator.

2. Negative Terminal (−): The terminal connected to the negative output of the audio amplifier. It is typically marked with a black dot, "−" symbol, or another indicator.

6.10.3 Key Characteristics:

1. Impedance: 8 ohms. This is the speaker's resistance to electrical current, which affects how it interacts with the audio amplifier.

2. Power Handling: The maximum power the speaker can handle without distortion or damage. This can vary widely depending on the speaker model, typically ranging from 1 watt to 50 watts or more.

3. Frequency Response: The range of frequencies the speaker can reproduce, usually given as a range (e.g., 20 Hz to 20 kHz).

4. Size and Type: Speakers come in various sizes and types (e.g., full-range, woofer, tweeter). The size impacts the sound quality and frequency range.

Fig. 6.15: 8-ohm Speaker

CHAPTER 7

SOFTWARE DESCRIPTION

Creating an IoT (Internet of Things) webpage typically involves designing a user interface that allows users to interact with and monitor IoT devices through a web browser. This process can include displaying data collected from sensors, controlling devices, and providing real-time updates. Here's an overview of how you might approach creating an IoT webpage:

7.1 Steps for Creating IOT Webpage:

1. Design and Planning: -

a. Define Requirements

- **Purpose:** Determine what functionalities you want on the webpage (e.g., device monitoring, control, data visualization).
- **Devices:** Identify the IoT devices you will be interacting with.
- **Data:** Specify the type of data that will be displayed and how it will be presented.

b. User Interface (UI) Design

- **Layout:** Plan the layout of the page, including sections for device status, controls, and data graphs.
- **Responsive Design:** Ensure the webpage is accessible on various devices (desktops, tablets, smartphones).

2. Technology Stack: -

a. Frontend

- **HTML:** Structure the content of the webpage.
- **CSS:** Style the webpage to make it visually appealing.
- **JavaScript:** Add interactivity to the webpage, such as dynamic data updates and user controls.

b. Backend

- **Server-side Language:** Use languages like **PHP** to handle server-side logic and API interactions.
- **Database:** Store and manage data from IoT devices in this we use **SQL** .

c. IoT Communication

- **API Integration:** Communicate with IoT devices using REST APIs, MQTT, or WebSocket.
- **Data Handling:** Process and display data received from IoT devices.

3. Implementation Steps: -

a. Setup Environment

- **Development Tools:** Set up a development environment using tools like Visual Studio Code, Git, and a local server.
- **Libraries and Frameworks:** Consider using libraries or frameworks such as Bootstrap for styling.

b. Build the Frontend

- **HTML Structure:** Create the basic structure of the webpage.
- **CSS Styling:** Apply styles to make the interface user-friendly and visually appealing.
- **JavaScript:** Implement functionality for data fetching, real-time updates, and user interactions.

c. Develop the Backend

- **Server Setup:** Create a server to handle requests and manage IoT device interactions.
- **API Endpoints:** Develop endpoints for interacting with the IoT devices (e.g., fetching sensor data, sending commands).

d. Integrate IoT Devices

- **Communication Protocol:** Use the appropriate communication protocol to interact with IoT devices.

- **Data Handling:** Ensure data from the devices is correctly processed and displayed on the webpage.

4. Testing: -

- **Functionality Testing:** Verify that all features work as expected, including data display and device control.
- **Responsiveness Testing:** Ensure the webpage works well on different devices and screen sizes.

7.2 Webpage Code:

```
-- phpMyAdmin SQL Dump
-- version 5.2.1
-- https://www.phpmyadmin.net/
--
-- Host: localhost:3306
-- Generation Time: Mar 22, 2024 at 01:25 PM
-- Server version: 8.0.34
-- PHP Version: 8.1.27
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";
/*!40101 SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
--
-- Database: `patient2_thakur`
```

```sql
--
-- --------------------------------------------------------
--
-- Table structure for table `thakur`
--
CREATE TABLE `thakur` (
  `id` bigint NOT NULL,
  `temp` varchar(120) NOT NULL,
  `ecg` varchar(120) NOT NULL,
  `fal` varchar(120) NOT NULL,
  `createdat` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
--
-- Dumping data for table `thakur`
--

INSERT INTO `thakur` (`id`, `temp`, `ecg`, `fal`, `createdat`) VALUES
(1, '30', '120', 'YES', '2024-03-21 08:47:35'),
(2, '59', '0', 'NO', '2024-03-21 11:11:37'),
(3, '37', '0', 'YES', '2024-03-21 11:14:05'),
(4, '53', '0', 'NO', '2024-03-22 06:59:39'),
(5, '45', '0', 'YES', '2024-03-22 07:00:28'),
(6, '59', '0', 'NO', '2024-03-22 07:22:44'),
(7, '48', '0', 'YES', '2024-03-22 07:23:32'),
(8, '40', '332', 'YES', '2024-03-22 07:24:32');
```

```
--
-- Indexes for dumped tables
--
--
-- Indexes for table `thakur`
--
ALTER TABLE `thakur`
  ADD PRIMARY KEY (`id`);
--
-- AUTO_INCREMENT for dumped tables
--
--

-- AUTO_INCREMENT for table `thakur`
--

ALTER TABLE `thakur`
  MODIFY `id` bigint NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=9;
COMMIT;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

7.3 Login Details:

7.3.1 Webpage link:

https://patienthealthcheck.com/iot_patient_Paralysis9/login.php

7.3.2 Login details:

User Id: health

Password: health123

7.4 Webpage Images:

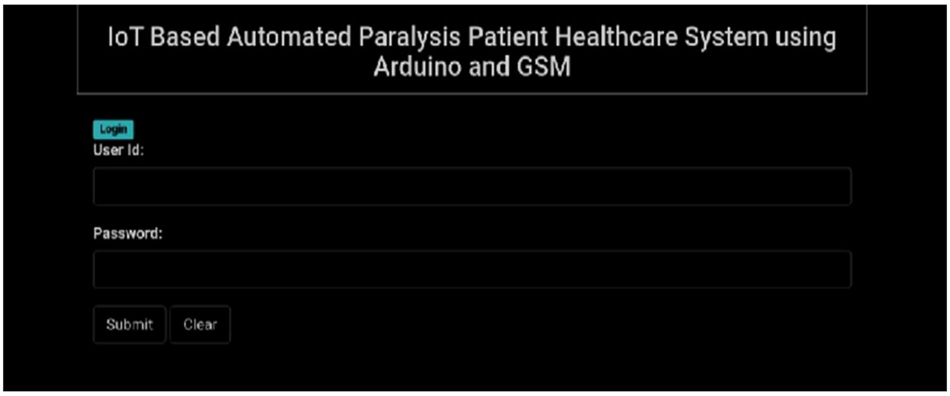

Fig. 7.1: Webpage Image 1

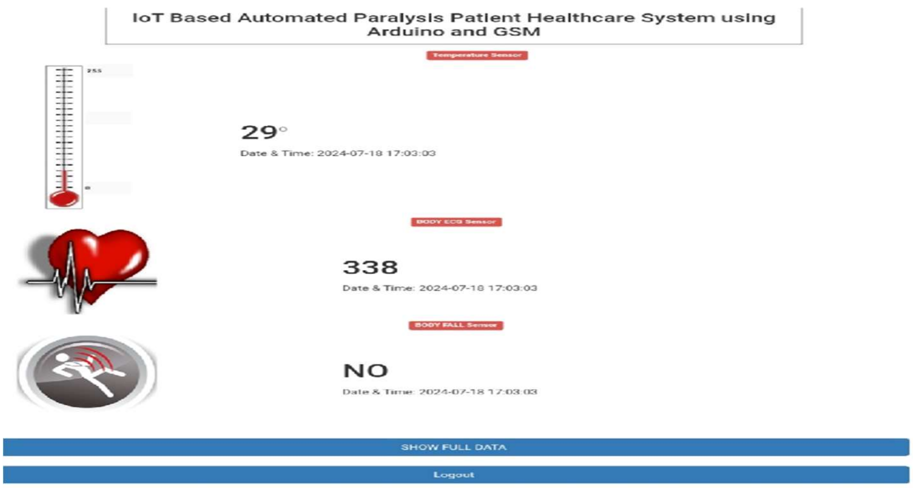

Fig. 7.2: Webpage Image 2

IoT Based Automated Paralysis Patient Healthcare System using Arduino and GSM

MAIN SCREEN

Logout

TEMP	ECG	FALL DETECT	Date / TIME
29	338	NO	2024-07-18 17:03:03
29	338	YES	2024-07-18 17:01:03
29	338	YES	2024-07-18 16:58:32
29	338	YES	2024-07-18 16:57:12
30	164	YES	2024-07-18 16:56:18
43	272	YES	2024-07-18 16:54:59
30	338	YES	2024-07-18 16:53:15
42	0	YES	2024-07-18 16:51:29
41	338	YES	2024-07-18 16:50:08
29	338	YES	2024-07-18 16:48:30
29	338	YES	2024-07-18 16:46:48
29	338	YES	2024-07-18 16:45:19
29	338	NO	2024-07-18 16:44:08
28	338	YES	2024-07-15 16:31:07
36	338	YES	2024-07-15 16:28:42
47	338	NO	2024-07-15 16:27:52
27	338	NO	2024-07-15 16:21:47
49	0	NO	2024-07-15 14:12:31
55	120	NO	2024-07-15 13:48:08
40	120	YES	2024-07-15 13:47:51
40	120	NO	2024-07-15 13:45:47
30	120	YES	2024-07-15 12:33:46

Fig. 7.3: Webpage Image 3

Creating an IoT webpage involves designing a user-friendly interface, integrating with IoT devices, and implementing both frontend and backend functionalities. By using modern web technologies and proper communication protocols, you can build a robust and interactive platform for managing and monitoring IoT devices.

CHAPTER 8

WORKING AND DEMONSTRATION

An IoT-based health care system for paralysis patients is a sophisticated application of technology designed to monitor and manage the health and well-being of individuals with paralysis. Such a system integrates various IoT devices and sensors to collect data, provide real-time monitoring, and facilitate remote health management. Here is an overview of how you might design and implement such a system.

8.1 BLOCK Diagram:

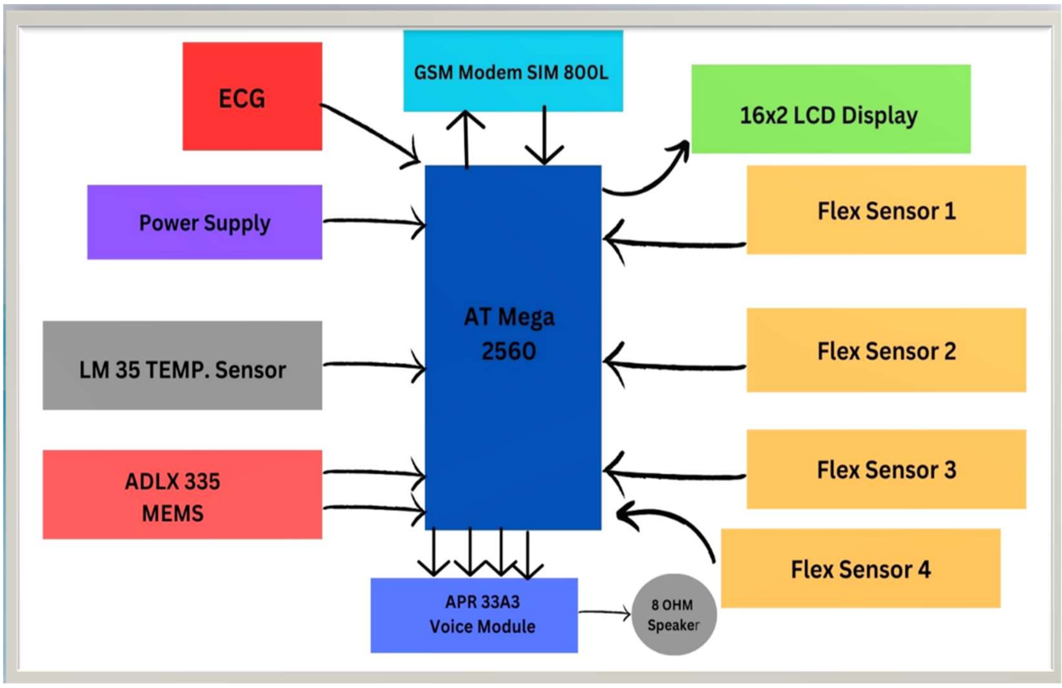

Fig. 8.1: Block Diagram of IOT Based Paralysis patient healthcare system

Here, in this we used several sensors like ecg, gsm module, power supply board, lm 35 temperature sensor, ADXL 335 mem body fall detection sensor, ARP 33a3 voice module, 8-ohm speaker, 4 flex sensor, 16*2 lcd display. All sensors are used to doing appropriate task to fulfil or completed purposefully and without and error.

8.2 RECEIVER SECTION - IOT WEB SITE:

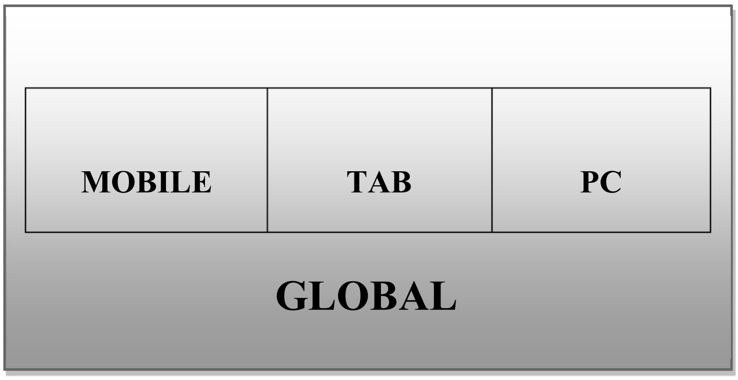

Fig. 8.2: Receiver Section

The receiver section of an IoT (Internet of Things) website is responsible for handling data transmitted from IoT devices, processing it, and presenting it to users. This section typically involves several key components: data reception, processing, storage, and user interface.

8.3 Circuit Diagram:

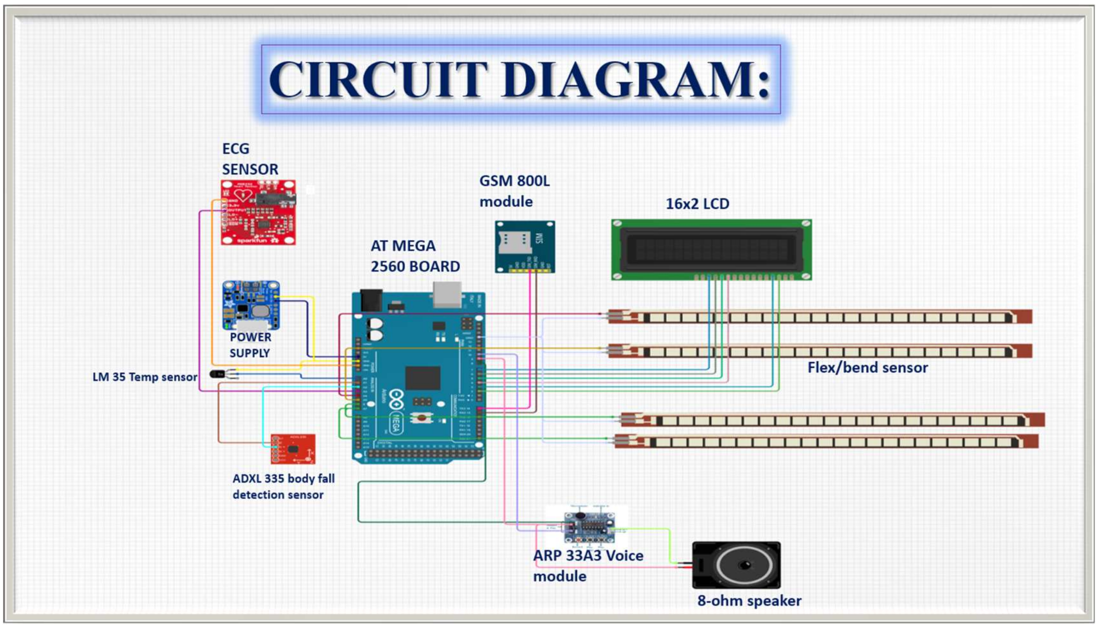

Fig. 8.3: Circuit Diagram (A)

Creating a circuit diagram for an IoT-based system for paralysis patients typically involves several components, including sensors, microcontrollers, communication modules, and possibly actuators or alert systems. Here is a general outline of what such a system might include.

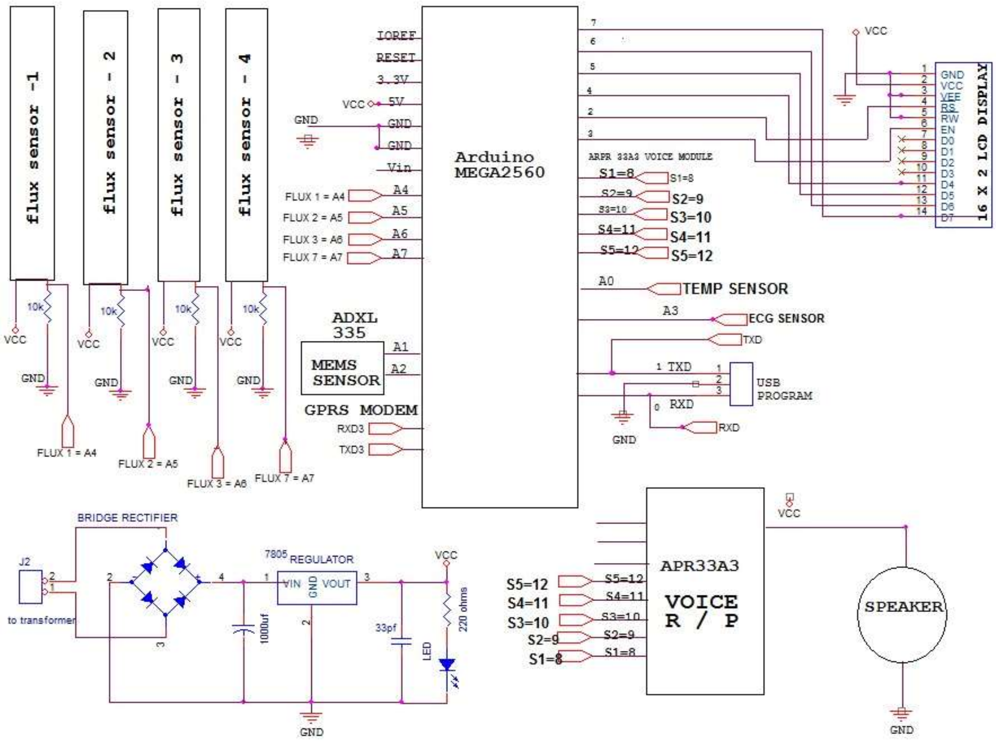

Fig. 8.4: Circuit Diagram (B)

8.4 Working:

An IoT-based healthcare system for paralysis patients leverages modern technology to provide comprehensive monitoring, assistance, and management of patient health. The system integrates wearable sensors, environmental monitors, assistive devices, data processing platforms, and user interfaces. This integration facilitates real-time monitoring, enhances patient safety, and supports caregivers and healthcare professionals in providing better care. In this detailed explanation, we will explore the working principles, components, and functionalities of such a system, emphasizing its importance in improving the quality of life for paralysis patients.

8.4.1 Data Collection and Sensing:

1. Wearable Sensors: Wearable sensors play a crucial role in continuously monitoring vital signs and other health metrics of paralysis patients. These devices, which can be worn on the wrist, chest, or other parts of the body, include:

- **Heart Rate Monitors:** Measure the patient's pulse, providing data on heart rate variability, which can indicate stress levels or potential cardiac issues.
- **Temperature Sensors (LM 35 TEMPERATURE SENSOR):** Monitor body temperature, helping detect fevers or other abnormalities.
- **Electrocardiogram (ECG) Sensors:** Record the electrical activity of the heart, allowing for detailed cardiac monitoring.
- **Accelerometers and Gyroscopes (ADXL 335 SENSOR):** Track movement and orientation, which is essential for monitoring mobility and detecting falls.

2. Environmental Sensors: Environmental sensors are deployed in the patient's living environment to ensure conditions are conducive to their health and comfort:

- **Temperature and Humidity Sensors:** Monitor room temperature and humidity, which are vital for maintaining a comfortable living environment.
- **Light Sensors:** Monitor ambient light levels, which can be adjusted to improve patient comfort and sleep patterns.

3. Assistive Devices: Assistive devices such as smart wheelchairs, robotic arms, and automated beds are equipped with sensors to provide mobility assistance and monitor usage:

- **Smart Wheelchairs:** Include sensors for tracking usage, battery levels, and positioning, allowing for remote monitoring and control.
- **Robotic Limbs:** Assist with daily tasks and rehabilitation, equipped with sensors to monitor movement and exertion.
- **Automated Beds:** Adjust based on patient needs, with sensors to monitor sleep patterns and body movements.

8.4.2 Data Transmission and Communication:

1. Communication Protocols

- **Wireless Communication:** Data from sensors and devices are transmitted wirelessly using protocols like Bluetooth, Wi-Fi.
- **Gateway Devices:** IoT gateways aggregate data from multiple devices and transmit it to the central server or cloud platform.

2. Real-Time Data Transfer

- **Continuous Monitoring:** The system continuously monitors and transmits data in real-time or at regular intervals, ensuring up-to-date information on the patient's condition.
- **Data Security:** Data is encrypted during transmission to ensure privacy and security.

8.4.3 Data Processing and Analysis:

1. Central Server or Cloud Platform

- **Data Storage:** The server or cloud platform stores the collected data in a database for historical analysis and real-time monitoring.
- **Data Processing:** The platform processes incoming data to filter, aggregate, and analyse it, identifying trends and detecting anomalies.

2. Analytics and AI

- **Pattern Recognition:** Machine learning algorithms and AI models analyze the data to recognize patterns and predict potential health issues.

- **Alerts and Notifications:** The system generates alerts for caregivers or healthcare professionals if it detects abnormalities, such as unusual vital signs or environmental conditions.

8.4.4 User Interface and Interaction:

1. Healthcare Professional Dashboard

- **Data Visualization:** A web-based dashboard displays real-time and historical data, visualized through charts, graphs, and indicators.
- **Alerts Management:** Healthcare professionals receive alerts and can take immediate action based on the patient's condition.
- **Patient Records:** Access to detailed patient records and reports for comprehensive care management.

2. Caregiver and Patient App

- **Mobile App:** A mobile application allows caregivers and patients to monitor health metrics, receive alerts, and manage daily activities.
- **Control Features:** The app may include features to control assistive devices, such as adjusting a smart bed or wheelchair.
- **Communication:** In-app messaging or video call features enable direct communication between patients, caregivers, and healthcare providers.

8.4.5 Assistive Technologies:

1. Smart Assistive Devices

- **Smart Wheelchairs:** Equipped with sensors and connectivity features, allowing for monitoring of usage and remote-control capabilities.
- **Robotic Aids:** Assist with movement and rehabilitation exercises, integrated with sensors to track progress and safety.

2. Automated Systems

- **Environment Control:** Automate environmental settings like temperature, lighting, and curtains, based on patient comfort and health data.
- **Emergency Response:** Integrated systems can automatically alert emergency services or caregivers in case of critical health situations or accidents.

CHAPTER 9

SOURCE CODES AND OUTPUT RESULTS

Creating a comprehensive source code for an IoT-based healthcare system for paralysis patients involves multiple components, including data collection from sensors, data transmission, backend processing, real-time data analysis, and user interfaces. Below is a simplified version of the source code to illustrate the core components. This example will focus on three main areas: data collection, backend API, and a basic frontend interface.

9.1 Source Code:

```
#include<LiquidCrystal.h>

LiquidCrystal lcd(2, 3, 4, 5, 6, 7);

int TEMP;

unsigned int MEMSX;

unsigned int MEMSY;

unsigned int ECG;

int a=0;

int aa=0;

int dt=0;///temp

unsigned int F1;

unsigned int F2;

unsigned int F3;

unsigned int F4;

const int S1=8;     //////////    M0

const int S2=9;     //////////    M1

const int S3=10;    //////////    M2

const int S4=11;    //////////    M3

const int S5=12;    //////////    M4
```

```
void setup()
{
lcd.begin(16,2);
Serial3.begin(9600);
pinMode(S1, OUTPUT);
pinMode(S2, OUTPUT);
pinMode(S3, OUTPUT);
pinMode(S4, OUTPUT);
pinMode(S5, OUTPUT);
digitalWrite(S1, HIGH);digitalWrite(S2, HIGH);digitalWrite(S3, HIGH);digitalWrite(S4, HIGH);digitalWrite(S5, HIGH);
lcd.clear();
lcd.setCursor(0,0);lcd.print("  Paralytic  ");
lcd.setCursor(0,1);lcd.print(" Patient    ");
delay(2000);lcd.clear();
lcd.setCursor(0,0);lcd.print("HEALTH CARE ");
lcd.setCursor(0,1);lcd.print("MONITORING  ");
delay(2000);lcd.clear();
lcd.setCursor(0,0);lcd.print("SYSTEM USING ");
lcd.setCursor(0,1);lcd.print(" ARDUINO & GSM");
delay(2000);lcd.clear();
lcd.clear();
lcd.print("GSM Initializing");
delay(500);
gsm_init(); lcd.clear();
lcd.setCursor(0,0);lcd.print(" GPRS MODEM ");
```

```
lcd.setCursor(0,1);lcd.print("ENABLE IOT  ");
delay(2000);lcd.clear();
}
void loop()
{st:
aa=aa+1;
TEMP = analogRead(0);
TEMP=TEMP/2;
TEMP = TEMP/10;
lcd.setCursor(0,0);lcd.print("T:");lcd.setCursor(3,0);lcd.print(TEMP);delay(1000);
if(TEMP>40)
{
dt=dt+1;
if(dt==2)
{
lcd.clear();send_gprs();delay(1000);
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS
1");lcd.setCursor(0,1);lcd.print("TEMP ALERT");
Serial3.println("AT+CMGF=1");delay(1000);
Serial3.println("AT+CMGS=\"9369957859\"");delay(1000);
Serial3.println("Over Temperature\n");delay(1000);
Serial3.print("Temp=");delay(500);Serial3.print(TEMP);delay(1000);Serial3.write(26);
delay(2000);
Serial3.print("AT\r\n");delay(1000);Serial3.print("AT\r\n");delay(1000);
Serial3.println("AT+CMGF=1");delay(1000);lcd.clear();
lcd.clear();lcd.setCursor(0,0);lcd.print("SMS SENT 1");
```

```
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS 2");lcd.setCursor(0,1);lcd.print("TEMP ALERT");
Serial3.println("AT+CMGF=1");delay(1000);
Serial3.println("AT+CMGS=\"8423599949\"");delay(1000);
Serial3.println("Over Temperature\n");delay(1000);
Serial3.print("Temp=");delay(500);Serial3.print(TEMP);delay(1000);Serial3.write(26); delay(2000);
Serial3.print("AT\r\n");delay(1000);Serial3.print("AT\r\n");delay(1000);
Serial3.println("AT+CMGF=1");delay(1000);lcd.clear();
lcd.clear();lcd.setCursor(0,0);lcd.print("SMS SENT 2");
goto st;
}
}
if(TEMP<40){delay(100);dt=0;delay(100);}
ECG = analogRead(3);ECG=ECG/2;
lcd.setCursor(0,1);lcd.print("E: ");lcd.setCursor(3,1);lcd.print(ECG);delay(1000);
lcd.clear() ;
MEMSX = analogRead(8);MEMSX=MEMSX/2;
lcd.setCursor(0,0);lcd.print("X:");lcd.setCursor(3,0);lcd.print(MEMSX);delay(1000);
MEMSY = analogRead(9);MEMSY=MEMSY/2;
lcd.setCursor(0,1);lcd.print("Y: ");lcd.setCursor(3,1);lcd.print(MEMSY);delay(1000);

if(((MEMSX >= 160) & (MEMSX <= 175)) &  ( (MEMSY >= 160) & (MEMSY <= 175 )) )
{a=0;lcd.setCursor(8,0);lcd.print("NORMAL  ");delay(1000);}
```

```
if(((MEMSX >= 130) & (MEMSX <= 150)) &  ( (MEMSY >= 160) & (MEMSY <= 175
)) )
{a=1;
lcd.setCursor(8,0);lcd.print("RIGHT                ");lcd.setCursor(8,1);lcd.print("FALLEN
");delay(500);
lcd.clear();send_gprs();delay(500);
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS 1");delay(500);lcd.clear();
a=1;lcd.setCursor(8,0);lcd.print("RIGHT          ");lcd.setCursor(8,1);lcd.print("FALLEN
");delay(500);
Serial3.println("AT+CMGF=1");delay(500);
Serial3.println("AT+CMGS=\"9369957859\"");delay(500);
Serial3.println("RIGHT FALLEN \n");
delay(1000);Serial3.write(26);delay(2000);
Serial3.print("AT\r\n");delay(1000);Serial3.print("AT\r\n");delay(1000);
Serial3.println("AT+CMGF=1");delay(1000);lcd.clear();
lcd.clear();lcd.setCursor(0,0);lcd.print("SMS SENT 1 ");
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS 2");delay(500);lcd.clear();
a=1;lcd.setCursor(8,0);lcd.print("RIGHT          ");lcd.setCursor(8,1);lcd.print("FALLEN
");delay(500);
Serial3.println("AT+CMGF=1");delay(500);
Serial3.println("AT+CMGS=\"8423599949\"");delay(500);
Serial3.println("RIGHT FALLEN \n");
delay(1000);Serial3.write(26);delay(2000);
Serial3.print("AT\r\n");delay(1000);Serial3.print("AT\r\n");delay(1000);
Serial3.println("AT+CMGF=1");delay(1000);lcd.clear();
lcd.clear();lcd.setCursor(0,0);lcd.print("SMS SENT 2 ");
}
```

```
if(((MEMSX >= 185) & (MEMSX <= 200)) &  ( (MEMSY >= 160) & (MEMSY <= 175
)) )
{a=2;lcd.setCursor(8,0);lcd.print("LEFT          ");lcd.setCursor(8,1);lcd.print("FALLEN
");delay(500);
lcd.clear();send_gprs();delay(500);
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS 1");delay(500);lcd.clear();
a=2;lcd.setCursor(8,0);lcd.print("LEFT          ");lcd.setCursor(8,1);lcd.print("FALLEN
");delay(500);
Serial3.println("AT+CMGF=1");delay(500);
Serial3.println("AT+CMGS=\"9369957859\"");delay(500);
Serial3.println("LEFT FALLEN \n");
delay(1000);Serial3.write(26);delay(2000);
Serial3.print("AT\r\n");delay(1000);Serial3.print("AT\r\n");delay(1000);
Serial3.println("AT+CMGF=1");delay(1000);lcd.clear();
lcd.clear();lcd.setCursor(0,0);lcd.print("SMS SENT 1 ");
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS 2");delay(500);lcd.clear();
a=2;lcd.setCursor(8,0);lcd.print("LEFT          ");lcd.setCursor(8,1);lcd.print("FALLEN
");delay(500);
Serial3.println("AT+CMGF=1");delay(500);
Serial3.println("AT+CMGS=\"8423599949\"");delay(500);
Serial3.println("LEFT FALLEN \n");
delay(1000);Serial3.write(26);delay(2000);
Serial3.print("AT\r\n");delay(1000);Serial3.print("AT\r\n");delay(1000);
Serial3.println("AT+CMGF=1");delay(1000);lcd.clear();
lcd.clear();lcd.setCursor(0,0);lcd.print("SMS SENT 2 ");
}
```

```
if(((MEMSX >= 160) & (MEMSX <= 175)) &  ( (MEMSY >= 130) & (MEMSY <= 150
)) )
{a=3;lcd.setCursor(8,0);lcd.print("FRONT       ");lcd.setCursor(8,1);lcd.print("FALLEN
");delay(500);
lcd.clear();send_gprs();delay(500);
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS 1");delay(500);lcd.clear();
a=3;lcd.setCursor(8,0);lcd.print("FRONT       ");lcd.setCursor(8,1);lcd.print("FALLEN
");delay(500);
Serial3.println("AT+CMGF=1");delay(500);
Serial3.println("AT+CMGS=\"9369957859\"");delay(500);
Serial3.println("FRONT FALLEN \n");delay(500);
delay(1000);Serial3.write(26);delay(2000);
Serial3.print("AT\r\n");delay(1000);Serial3.print("AT\r\n");delay(1000);
Serial3.println("AT+CMGF=1");delay(1000);lcd.clear();
lcd.clear();lcd.setCursor(0,0);lcd.print("SMS SENT 1 ");
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS 2");delay(500);lcd.clear();
a=3;lcd.setCursor(8,0);lcd.print("FRONT       ");lcd.setCursor(8,1);lcd.print("FALLEN
");delay(500);
Serial3.println("AT+CMGF=1");delay(500);
Serial3.println("AT+CMGS=\"8423599949\"");delay(500);
Serial3.println("FRONT FALLEN \n");delay(500);
delay(1000);Serial3.write(26);delay(2000);
Serial3.print("AT\r\n");delay(1000);Serial3.print("AT\r\n");delay(1000);
Serial3.println("AT+CMGF=1");delay(1000);lcd.clear();
lcd.clear();lcd.setCursor(0,0);lcd.print("SMS SENT 2 ");
}
```

```
if(((MEMSX >= 160) & (MEMSX <= 175)) &  ( (MEMSY >= 180) & (MEMSY <= 198
)) )
{a=4;lcd.setCursor(8,0);lcd.print("BACK         ");lcd.setCursor(8,1);lcd.print("FALLEN
");delay(500);
lcd.clear();send_gprs();delay(500);
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS 1");delay(500);lcd.clear();
a=4;lcd.setCursor(8,0);lcd.print("BACK          ");lcd.setCursor(8,1);lcd.print("FALLEN
");delay(500);
Serial3.println("AT+CMGF=1");delay(500);
Serial3.println("AT+CMGS=\"9369957859\"");delay(500);
Serial3.println("BACK FALLEN \n");delay(500);
delay(1000);Serial3.write(26);delay(2000);
Serial3.print("AT\r\n");delay(1000);Serial3.print("AT\r\n");delay(1000);
Serial3.println("AT+CMGF=1");delay(1000);lcd.clear();
lcd.clear();lcd.setCursor(0,0);lcd.print("SMS SENT 1 ");
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS 2");delay(500);lcd.clear();
a=4;lcd.setCursor(8,0);lcd.print("BACK          ");lcd.setCursor(8,1);lcd.print("FALLEN
");delay(500);
Serial3.println("AT+CMGF=1");delay(500);
Serial3.println("AT+CMGS=\"8423599949\"");delay(500);
Serial3.println("BACK FALLEN \n");delay(500);
delay(1000);Serial3.write(26);delay(2000);
Serial3.print("AT\r\n");delay(1000);Serial3.print("AT\r\n");delay(1000);
Serial3.println("AT+CMGF=1");delay(1000);lcd.clear();
lcd.clear();lcd.setCursor(0,0);lcd.print("SMS SENT 2 ");
}
```

```
lcd.clear() ;
F1 = analogRead(4);F1=F1/2;
F2 = analogRead(5);F2=F2/2;
F3 = analogRead(6);F3=F3/2;
F4 = analogRead(7);F4=F4/2;
//lcd.setCursor(0,0);lcd.print(F1);lcd.setCursor(8,0);lcd.print(F2);
//lcd.setCursor(0,1);lcd.print(F3);lcd.setCursor(8,1);lcd.print(F4);
delay (2000);
////////// NO bending finger
if(((F1 >= 110)&(F1 <= 170)) & ((F2 >= 110)&(F2 <= 170)) & ((F3 >= 110) & (F3 <= 170)) & ((F4 >= 110) & (F4 <= 170)) )
{
lcd.setCursor(0,0);lcd.print("NORMAL FINGERS     ");delay(500);
digitalWrite(S1, HIGH);digitalWrite(S2, HIGH);digitalWrite(S3, HIGH);digitalWrite(S4, HIGH);digitalWrite(S5, HIGH);
}
////////// bending index finger
if(((F1 >= 10)&(F1 <= 100)) & ((F2 >= 110)&(F2 <= 170)) & ((F3 >= 110) & (F3 <= 170)) & ((F4 >= 110) & (F4 <= 170)) )
{
lcd.setCursor(0,0);lcd.print("WATER        ");delay(500);
digitalWrite(S1, LOW);digitalWrite(S2, HIGH);digitalWrite(S3, HIGH);digitalWrite(S4, HIGH);digitalWrite(S5, HIGH);delay(1000);
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS 1");delay(500);
lcd.setCursor(0,1);lcd.print("WATER        ");delay(500);
Serial3.println("AT+CMGF=1");delay(500);
Serial3.println("AT+CMGS=\"9369957859\"");delay(500);
```

```
Serial3.println("I NEED WATER \n");delay(500);
delay(1000);Serial3.write(26);delay(2000);
Serial3.print("AT\r\n");delay(1000);Serial3.print("AT\r\n");delay(1000);
Serial3.println("AT+CMGF=1");delay(1000);lcd.clear();
lcd.clear();lcd.setCursor(0,0);lcd.print("SMS SENT 1 ");
digitalWrite(S1, HIGH);digitalWrite(S2, HIGH);digitalWrite(S3, HIGH);digitalWrite(S4, HIGH);digitalWrite(S5, HIGH);
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS 2");delay(500);
lcd.setCursor(0,1);lcd.print("WATER        ");delay(500);
Serial3.println("AT+CMGF=1");delay(500);
Serial3.println("AT+CMGS=\"8423599949\"");delay(500);
Serial3.println("I NEED WATER \n");delay(500);
delay(1000);Serial3.write(26);delay(2000);
Serial3.print("AT\r\n");delay(1000);Serial3.print("AT\r\n");delay(1000);
Serial3.println("AT+CMGF=1");delay(1000);lcd.clear();
lcd.clear();lcd.setCursor(0,0);lcd.print("SMS SENT 2 ");
digitalWrite(S1, HIGH);digitalWrite(S2, HIGH);digitalWrite(S3, HIGH);digitalWrite(S4, HIGH);digitalWrite(S5, HIGH);
}
////////// bending  middle finger
if(((F1 >= 110)&(F1 <= 170)) & ((F2 >= 10)&(F2 <= 100)) & ((F3 >= 110) & (F3 <= 170)) & ((F4 >= 110) & (F4 <= 170)) )
{
lcd.setCursor(0,0);lcd.print("FOOD        ");delay(500);
digitalWrite(S1, HIGH);digitalWrite(S2, LOW);digitalWrite(S3, HIGH);digitalWrite(S4, HIGH);digitalWrite(S5, HIGH);delay(1000);
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS 1");delay(500);
```

```
lcd.setCursor(0,1);lcd.print("FOOD       ");delay(500);
Serial3.println("AT+CMGF=1");delay(500);
Serial3.println("AT+CMGS=\"9369957859\"");delay(500);
Serial3.println("I NEED FOOD \n");
delay(1000);Serial3.write(26);delay(2000);
Serial3.print("AT\r\n");delay(1000);Serial3.print("AT\r\n");delay(1000);
Serial3.println("AT+CMGF=1");delay(1000);lcd.clear();
lcd.clear();lcd.setCursor(0,0);lcd.print("SMS SENT 1 ");
digitalWrite(S1, HIGH);digitalWrite(S2, HIGH);digitalWrite(S3, HIGH);digitalWrite(S4, HIGH);digitalWrite(S5, HIGH);
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS 2");delay(500);
lcd.setCursor(0,1);lcd.print("FOOD       ");delay(500);
Serial3.println("AT+CMGF=1");delay(500);
Serial3.println("AT+CMGS=\"8423599949\"");delay(500);
Serial3.println("I NEED FOOD \n");
delay(1000);Serial3.write(26);delay(2000);
Serial3.print("AT\r\n");delay(1000);Serial3.print("AT\r\n");delay(1000);
Serial3.println("AT+CMGF=1");delay(1000);lcd.clear();
lcd.clear();lcd.setCursor(0,0);lcd.print("SMS SENT 2 ");
digitalWrite(S1, HIGH);digitalWrite(S2, HIGH);digitalWrite(S3, HIGH);digitalWrite(S4, HIGH);digitalWrite(S5, HIGH);
}
////////// bending RING finger
if(((F1 >= 110)&(F1 <= 170)) & ((F2 >= 110)&(F2 <= 170)) & ((F3 >= 10) & (F3 <= 100)) & ((F4 >= 110) & (F4 <= 170)) )
{
lcd.setCursor(0,0);lcd.print("FRESH AIR     ");delay(500);
```

```
digitalWrite(S1, HIGH);digitalWrite(S2, HIGH);digitalWrite(S3, LOW);digitalWrite(S4,
HIGH);digitalWrite(S5, HIGH);delay(1000);
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS 1");delay(500);
lcd.setCursor(0,1);lcd.print("FRESH AIR       ");delay(500);
Serial3.println("AT+CMGF=1");delay(500);
Serial3.println("AT+CMGS=\"9369957859\"");delay(500);
Serial3.println("I WANT FRESH AIR \n");delay(500);
delay(1000);Serial3.write(26);delay(2000);
Serial3.print("AT\r\n");delay(1000);Serial3.print("AT\r\n");delay(1000);
Serial3.println("AT+CMGF=1");delay(1000);lcd.clear();
lcd.clear();lcd.setCursor(0,0);lcd.print("SMS SENT 1 ");
digitalWrite(S1, HIGH);digitalWrite(S2, HIGH);digitalWrite(S3,
HIGH);digitalWrite(S4, HIGH);digitalWrite(S5, HIGH);
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS 2");delay(500);
lcd.setCursor(0,1);lcd.print("FRESH AIR       ");delay(500);
Serial3.println("AT+CMGF=1");delay(500);
Serial3.println("AT+CMGS=\"8423599949\"");delay(500);
Serial3.println("I WANT FRESH AIR \n");delay(500);
delay(1000);Serial3.write(26);delay(2000);
Serial3.print("AT\r\n");delay(1000);Serial3.print("AT\r\n");delay(1000);
Serial3.println("AT+CMGF=1");delay(1000);lcd.clear();
lcd.clear();lcd.setCursor(0,0);lcd.print("SMS SENT 2 ");
digitalWrite(S1, HIGH);digitalWrite(S2, HIGH);digitalWrite(S3,
HIGH);digitalWrite(S4, HIGH);digitalWrite(S5, HIGH);
}
/////////   bending LITTLE finger
```

```
if(((F1 >= 110)&(F1 <= 170)) & ((F2 >= 110)&(F2 <= 170)) & ((F3 >= 110) & (F3 <= 170)) & ((F4 >= 10) & (F4 <= 100)) )
{
lcd.setCursor(0,0);lcd.print("WASHROOM    ");delay(500);
digitalWrite(S1, HIGH);digitalWrite(S2, HIGH);digitalWrite(S3, HIGH);digitalWrite(S4, LOW);digitalWrite(S5, HIGH);delay(1000);
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS 1");delay(500);
lcd.setCursor(0,1);lcd.print("WASHROOM       ");delay(500);
Serial3.println("AT+CMGF=1");delay(500);
Serial3.println("AT+CMGS=\"9369957859\"");delay(500);
Serial3.println("I WANT TO GO WASHROOM \n");delay(500);
delay(1000);Serial3.write(26);delay(2000);
Serial3.print("AT\r\n");delay(1000);Serial3.print("AT\r\n");delay(1000);
Serial3.println("AT+CMGF=1");delay(1000);lcd.clear();
lcd.clear();lcd.setCursor(0,0);lcd.print("SMS SENT 1 ");
digitalWrite(S1, HIGH);digitalWrite(S2, HIGH);digitalWrite(S3, HIGH);digitalWrite(S4, HIGH);digitalWrite(S5, HIGH);
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS 2");delay(500);
lcd.setCursor(0,1);lcd.print("WASHROOM       ");delay(500);
Serial3.println("AT+CMGF=1");delay(500);
Serial3.println("AT+CMGS=\"8423599949\"");delay(500);
Serial3.println("I WANT TO GO WASHROOM \n");delay(500);
delay(1000);Serial3.write(26);delay(2000);
Serial3.print("AT\r\n");delay(1000);Serial3.print("AT\r\n");delay(1000);
Serial3.println("AT+CMGF=1");delay(1000);lcd.clear();
lcd.clear();lcd.setCursor(0,0);lcd.print("SMS SENT 2 ");
```

```
digitalWrite(S1, HIGH);digitalWrite(S2, HIGH);digitalWrite(S3, HIGH);digitalWrite(S4, HIGH);digitalWrite(S5, HIGH);
}
////////// 5th by bending index and middle finger
if(((F1 >= 10)&(F1 <= 100)) & ((F2 >= 10)&(F2 <= 100)) & ((F3 >= 110) & (F3 <= 170)) & ((F4 >= 110) & (F4 <= 170)) )
{
lcd.setCursor(0,0);lcd.print("HELP   ");delay(500);
digitalWrite(S1, HIGH);digitalWrite(S2, HIGH);digitalWrite(S3, HIGH);digitalWrite(S4, HIGH);digitalWrite(S5, LOW);delay(1000);
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS 1");delay(500);
lcd.setCursor(0,1);lcd.print("HELP        ");delay(500);
Serial3.println("AT+CMGF=1");delay(500);
Serial3.println("AT+CMGS=\"9369957859\"");delay(500);
Serial3.println("PLZ HELP ME \n");delay(500);
delay(1000);Serial3.write(26);delay(2000);
Serial3.print("AT\r\n");delay(1000);Serial3.print("AT\r\n");delay(1000);
Serial3.println("AT+CMGF=1");delay(1000);lcd.clear();
lcd.clear();lcd.setCursor(0,0);lcd.print("SMS SENT 2 ");
digitalWrite(S1, HIGH);digitalWrite(S2, HIGH);digitalWrite(S3, HIGH);digitalWrite(S4, HIGH);digitalWrite(S5, HIGH);
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS 1");delay(500);
lcd.setCursor(0,1);lcd.print("HELP        ");delay(500);
Serial3.println("AT+CMGF=1");delay(500);
Serial3.println("AT+CMGS=\"8423599949\"");delay(500);
Serial3.println("PLZ HELP ME \n");delay(500);
delay(1000);Serial3.write(26);delay(2000);
```

```
Serial3.print("AT\r\n");delay(1000);Serial3.print("AT\r\n");delay(1000);
Serial3.println("AT+CMGF=1");delay(1000);lcd.clear();
lcd.clear();lcd.setCursor(0,0);lcd.print("SMS SENT 2 ");
digitalWrite(S1, HIGH);digitalWrite(S2, HIGH);digitalWrite(S3, HIGH);digitalWrite(S4, HIGH);digitalWrite(S5, HIGH);
}
lcd.clear();
digitalWrite(S1, HIGH);digitalWrite(S2, HIGH);digitalWrite(S3, HIGH);digitalWrite(S4, HIGH);digitalWrite(S5, HIGH);
if(aa==10)
{
send_gprs();
aa=0;
goto st;
}

}
void send_gprs()
{
lcd.clear();lcd.print("GPRS SENDING");
boolean test7_flag=1;
while(test7_flag)
{
aa=0;
Serial3.print("AT+HTTPPARA=\"URL\",\"http://patienthealthcheck.com/iot_patient_Paralysis9/put_data.php");
Serial3.print("?temp=");Serial3.print(TEMP);
```

```
Serial3.print("&ecg=");Serial3.print(ECG);
if(a==0){Serial3.print("&fal=");Serial3.print("NO");}
if(a==1){Serial3.print("&fal=");Serial3.print("YES");}
if(a==2){Serial3.print("&fal=");Serial3.print("YES");}
if(a==3){Serial3.print("&fal=");Serial3.print("YES");}
if(a==4){Serial3.print("&fal=");Serial3.print("YES");}
Serial3.print("\"");Serial3.print("\r\n");
while(Serial3.available()>0){if(Serial3.find("OK"))test7_flag=0;}delay(1000);}
lcd.clear();lcd.print("SENT COMPLETED");delay(10000);lcd.clear();
lcd.clear();lcd.print("ACTION");
boolean test8_flag=1;while(test8_flag){Serial3.print("AT+HTTPACTION=0\r\n");
while(Serial3.available()>0){if(Serial3.find("OK"))test8_flag=0;}delay(1000);}
lcd.clear();lcd.print("SEND  OK");delay(5000);
}
void gsm_init()
{
lcd.clear();
lcd.print("GSM TESTING..");
boolean at_flag=1;
while(at_flag)
{Serial3.println("AT");while(Serial3.available()>0){if(Serial3.find("OK"))at_flag=0;}d
elay(1000);}
lcd.clear();lcd.print("GSM CONNECTED");delay(1000);lcd.clear();
//////////////////////////////////////////////////////////////////////////////////////////////
lcd.print("Disabling ECHO");
boolean echo_flag=1;
```

```
while(echo_flag)
{Serial3.println("ATE0");
while(Serial3.available()>0){if(Serial3.find("OK"))echo_flag=0;}delay(1000);}
lcd.clear();  lcd.print("Echo OFF");delay(1000);lcd.clear();
////////////////////////////////////////////////////////////////////////////////////////////////////
lcd.setCursor(0,1);lcd.print("GSM NETWORK");delay(1000);lcd.clear();
////////////////////////////////////////////////////////////////////////////
lcd.clear();lcd.print("TEST MESS");
boolean test_flag=1;while(test_flag){Serial3.println("AT+CMGF=1");
while(Serial3.available()>0){if(Serial3.find("OK"))test_flag=0;}delay(1000);}
lcd.clear();lcd.print("TEST MESSAGE");delay(1000);
/////////////////////////////////////////////////////////////////////////////////////////////////////
lcd.clear();lcd.print("GPRS1");
boolean
test2_flag=1;while(test2_flag){Serial3.print("AT+SAPBR=3,1,\"CONTYPE\",\"GPRS\
"\r\n");
while(Serial3.available()>0){if(Serial3.find("OK"))test2_flag=0;}delay(1000);}
lcd.clear();lcd.print("GPRS START1");delay(1000);
/////////////////////////////////////////////////////////////////////////////////////////////////////
lcd.clear();lcd.print("GPRS2");
boolean
test3_flag=1;while(test3_flag){Serial3.print("AT+SAPBR=3,1,\"APN\",\"internet\"\r\n"
);
while(Serial3.available()>0){if(Serial3.find("OK"))test3_flag=0;}delay(1000);}
lcd.clear();lcd.print("GPRS START2");delay(3000);
/////////////////////////////////////////////////////////////////////////////////////////////////////
lcd.clear();lcd.print("GPRS MAIN");
```

```
boolean test4_flag=1;while(test4_flag){Serial3.print("AT+SAPBR=1,1\r\n");
while(Serial3.available()>0){if(Serial3.find("OK"))test4_flag=0;}delay(1000);}
lcd.clear();lcd.print("GPRS CAME");delay(1000);
////////////////////////////////////////////////////////////////////////////////////////////////////////////
lcd.clear();lcd.print("HTTP1");
boolean test5_flag=1;while(test5_flag){Serial3.print("AT+HTTPINIT\r\n");
while(Serial3.available()>0){if(Serial3.find("OK"))test5_flag=0;}delay(1000);}
lcd.clear();lcd.print("HTTP1");delay(1000);
////////////////////////////////////////////////////////////////////////////////////////////////////////////
lcd.clear();lcd.print("HTTP2");
boolean test6_flag=1;while(test6_flag){Serial3.print("AT+HTTPPARA=\"CID\",1\r\n");
while(Serial3.available()>0){if(Serial3.find("OK"))test6_flag=0;}delay(1000);}
lcd.clear();lcd.print("HTTP2");delay(1000);
////////////////////////////////////////////////////////////////////////////////////////////////////////////
}
```

9.2 Output Results:

The output of the IoT-based healthcare monitoring system for paralysis patients, we can simulate the results for each part of the system: data collection, backend processing, and the frontend interface.

Fig. 9.1: Message Output

Fig. 9.2: Webpage Output

The output of the IoT-based healthcare monitoring system for paralysis patients, we can simulate the results for each part of the system: data collection, backend processing, and the frontend interface.

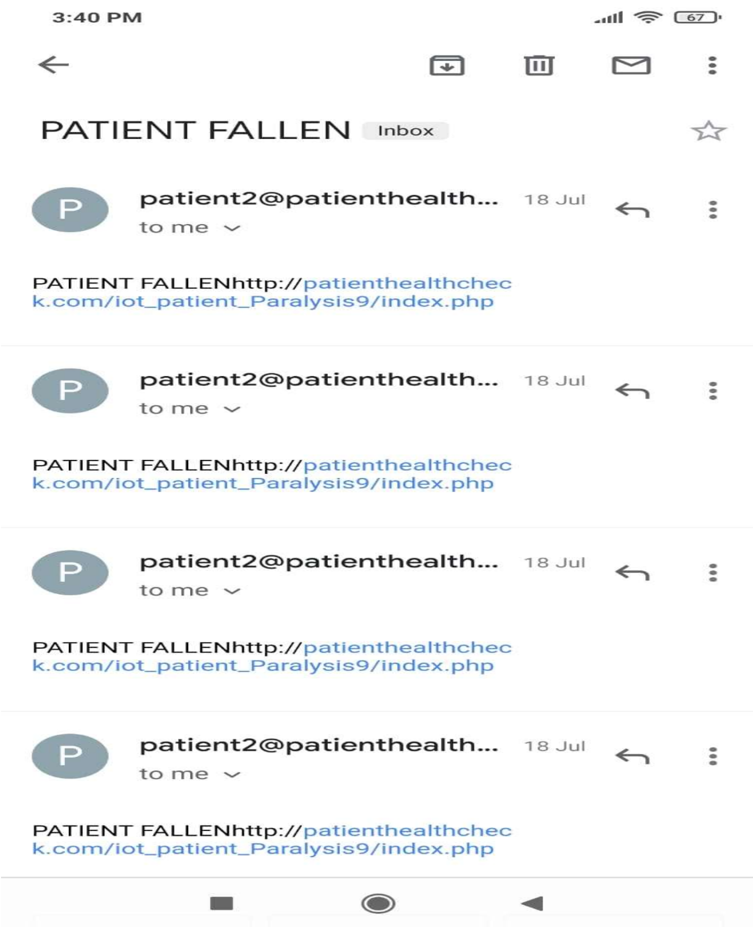

Fig. 9.3: E-mail Output

CHAPTER 10

FEATURES, CHALLENGES & FUTURE SCOPE

An IoT-based paralysis patient monitoring system incorporates a range of advanced features designed to enhance patient care, improve monitoring accuracy, and facilitate effective communication between patients, caregivers, and healthcare professionals. Below are the key features of such a system.

10.1 Features of IOT Based Paralysis Patient Monitoring System:

1. Continuous Health Monitoring:

- **Vital Signs Monitoring:** The system continuously tracks essential health metrics like heart rate, body temperature, blood pressure, and oxygen saturation.
- **Activity and Movement Tracking:** Sensors monitor patient movement, detecting patterns and identifying potential issues like prolonged immobility or falls.

2. Environmental Monitoring:

- **Ambient Condition Sensors:** Monitors environmental factors such as temperature, humidity, and air quality to ensure a comfortable and safe living environment for the patient.
- **Smart Home Integration:** Controls home devices like thermostats, lighting, and ventilation systems to automatically adjust settings for optimal comfort and safety.

3. Real-Time Data Transmission and Alerts:

- **Real-Time Data Transmission:** The system provides real-time data updates to a centralized platform accessible by caregivers and healthcare professionals.
- **Automated Alerts:** Sends immediate alerts for abnormal health readings or environmental conditions, such as a drop in oxygen levels, a significant change in heart rate, or poor air quality.

4. Data Analytics and Reporting:

- **Trend Analysis:** Analyzes historical data to identify trends, such as gradual health changes or the effectiveness of treatment plans.

- **Predictive Analytics:** Uses machine learning models to predict potential health issues, enabling proactive interventions.
- **Customizable Reports:** Generates detailed reports on patient health metrics, which can be customized to meet the needs of healthcare providers.

5. User Interfaces and Accessibility:

- **Web and Mobile Applications:** Provides user-friendly interfaces for viewing real-time data, receiving alerts, and accessing patient records.
- **Customizable Dashboards:** Allows users to customize dashboards to display the most relevant information, such as key health metrics and recent alerts.
- **Voice Command Integration:** Supports voice-activated commands for patients with limited mobility, enabling easy control of the system and home devices.

6. Assistive Devices and Automation:

- **Smart Assistive Devices:** Integrates with devices like smart wheelchairs, robotic limbs, and automated beds to assist with mobility and daily activities.
- **Automation for Daily Routines:** Automates daily tasks such as medication reminders, exercise schedules, and dietary management.

7. Communication and Telemedicine:

- **Integrated Communication Tools:** Provides messaging and video call features for direct communication between patients, caregivers, and healthcare professionals.
- **Telemedicine Support:** Enables remote consultations and follow-ups, reducing the need for physical visits to healthcare facilities.

8. Security and Privacy Features:

- **Data Encryption:** Ensures secure transmission and storage of sensitive health data, protecting it from unauthorized access.
- **Access Control:** Implements strict access controls, allowing only authorized users to view or modify patient data.
- **Compliance with Regulations:** Adheres to healthcare data privacy regulations such as HIPAA, GDPR, and other relevant standards.

10.2 Challenges in IOT Based Paralysis Patient Monitoring System:

1. Data Privacy and Security:

- **Sensitive Data Handling:** The system collects and transmits sensitive health information, which must be protected to prevent unauthorized access and breaches.
- **Encryption and Authentication:** Ensuring secure data transmission and storage through robust encryption methods and authentication mechanisms is crucial.
- **Regulatory Compliance:** Adhering to regulations such as HIPAA (Health Insurance Portability and Accountability Act) in the U.S. or GDPR (General Data Protection Regulation) in Europe is necessary to protect patient privacy.

2. System Reliability and Robustness:

- **Hardware Reliability:** Wearable sensors and other hardware components must be reliable and durable, as malfunctions can lead to incorrect data or system downtime.
- **Connectivity Issues:** The system relies on internet connectivity for real-time data transmission. Connectivity issues can disrupt monitoring and data flow.
- **Redundancy and Backup:** Implementing redundancy and data backup systems is essential to prevent data loss and ensure continuous operation, even during hardware failures or network outages.

3. Scalability and Integration:

- **Scalability Challenges:** The system must be scalable to accommodate a growing number of devices and users, which requires robust infrastructure and efficient data management strategies.
- **Integration with Existing Systems:** Integrating the IoT system with existing healthcare infrastructure, such as electronic health records (EHRs), can be challenging and may require significant customization.

4. Data Management and Analysis:

- **Data Overload:** The system generates large amounts of data, which can overwhelm healthcare providers if not properly managed and filtered.

- **Data Quality and Accuracy:** Ensuring the accuracy and reliability of the data collected by sensors is crucial for making informed healthcare decisions.
- **Real-Time Analysis:** Developing algorithms and systems capable of real-time data analysis is challenging, especially when dealing with large datasets.

5. User Interface and Experience:

- **Usability:** The system must be user-friendly, especially for patients and caregivers who may not be tech-savvy. Complex interfaces can lead to user frustration and errors.
- **Accessibility:** Ensuring the system is accessible to users with disabilities, such as those with visual or motor impairments, is essential for inclusivity.

6. Cost and Economic Viability:

- **High Initial Costs:** The initial setup costs, including purchasing sensors, devices, and backend infrastructure, can be high.
- **Maintenance and Upkeep:** Regular maintenance and updates are necessary to keep the system functional and secure, adding to ongoing costs.
- **Affordability:** Making the system affordable for patients and healthcare providers, especially in resource-limited settings, is a significant challenge.

7. Patient and Caregiver Adoption:

- **Adoption Resistance:** Patients and caregivers may be resistant to adopting new technology due to concerns about privacy, complexity, or reliability.
- **Training and Education:** Providing adequate training and education to patients, caregivers, and healthcare providers is necessary to ensure proper use and maximize the system's benefits.

8. Regulatory and Ethical Issues:

- **Ethical Concerns:** The use of continuous monitoring raises ethical questions about surveillance, autonomy, and the balance between care and privacy.
- **Regulatory Challenges:** Navigating the complex regulatory landscape for medical devices and healthcare technologies can be challenging and time-consuming.

10.3 Future Scope of IOT Based Paralysis Patient Monitoring System:

The future scope of IoT-based paralysis patient monitoring systems is vast, driven by advances in technology, increased awareness of the benefits of remote healthcare, and the growing need for personalized and efficient patient care. Here are several key areas where these systems are expected to evolve and expand.

1. Advanced Sensor Technologies:

- **Improved Wearable Sensors:** Future developments may include more compact, comfortable, and non-invasive sensors capable of continuously monitoring a wider range of physiological parameters with greater accuracy.
- **Integration of Novel Biomarkers:** Sensors that can detect novel biomarkers or molecular signatures, such as biochemical markers for stress, inflammation, or infection, could provide deeper insights into a patient's health status.

2. Artificial Intelligence and Machine Learning:

- **Predictive Analytics:** Leveraging AI and machine learning for predictive analytics can help anticipate health events, such as seizures or respiratory issues, allowing for pre-emptive care.
- **Personalized Care Plans:** AI algorithms can analyze data to create personalized care plans tailored to the specific needs and health trends of individual patients, improving outcomes and quality of life.
- **Natural Language Processing (NLP):** NLP technologies can assist in understanding patient communication, including voice commands or text inputs, improving the interaction with assistive technologies and monitoring systems.

3. Telemedicine and Remote Healthcare:

- **Enhanced Telemedicine Integration:** More advanced telemedicine features, including real-time data sharing during consultations, remote diagnostics, and the ability to control medical devices or administer treatments remotely, can significantly enhance patient care.
- **Remote Rehabilitation and Therapy:** IoT systems can support remote rehabilitation programs, including physical therapy exercises guided by sensors and virtual reality, enabling continuous therapy and monitoring progress.

4. Interoperability and Standards:

- **Universal Standards:** Developing universal standards for data formats, communication protocols, and device interoperability will facilitate seamless integration across different healthcare systems and platforms.
- **Cross-Platform Data Sharing:** Enhanced interoperability will enable more comprehensive data sharing between different healthcare providers, specialists, and caregivers, promoting coordinated and holistic care.

5. Enhanced User Interfaces and Accessibility:

- **Augmented Reality (AR) and Virtual Reality (VR):** AR and VR can provide immersive and intuitive user interfaces, helping patients visualize their health data, interact with virtual caregivers, or engage in therapeutic activities.
- **Voice and Gesture Control:** Improvements in voice and gesture recognition technologies can make it easier for patients with limited mobility or speech difficulties to interact with their monitoring systems and home environments.

6. Security and Privacy Enhancements:

- **Advanced Encryption and Blockchain:** Utilizing advanced encryption techniques and blockchain technology can enhance data security and integrity, providing a secure and transparent way to manage patient data.
- **Privacy-Preserving Technologies:** Techniques such as differential privacy and homomorphic encryption can enable data analysis without compromising individual privacy, crucial for sensitive health data.

7. Scalability and Deployment Models:

- **Edge Computing:** Implementing edge computing can reduce latency, improve data processing speeds, and enhance the system's ability to function even with limited connectivity.
- **Cloud and Hybrid Models:** Scalable cloud-based solutions, possibly combined with local edge computing, will facilitate the deployment and expansion of these systems, accommodating increasing data volumes and user numbers.

CHAPTER 11

CONCLUSION

IoT-based paralysis patient monitoring systems are a promising development in healthcare, offering significant benefits in terms of patient care, safety, and efficiency. While challenges remain, the continued advancement and integration of these technologies hold the potential to revolutionize healthcare for paralysis patients, providing them with better quality of life and more comprehensive medical support.

However, the implementation of such systems also presents challenges, including concerns about data privacy, security, and the need for robust and reliable infrastructure. Addressing these challenges requires a multi-faceted approach, involving collaboration between healthcare providers, technology developers, regulators, and patients.

The development and deployment of IoT-based paralysis patient monitoring systems represent a significant advancement in healthcare technology, offering a transformative approach to managing the health and well-being of individuals with paralysis. These systems provide continuous, real-time monitoring of vital signs and environmental conditions, enabling timely interventions, personalized care, and enhanced patient outcomes. As technology continues to evolve, the capabilities of these systems will expand, integrating advanced sensor technologies, artificial intelligence, and telemedicine, making healthcare more accessible and efficient.

REFERENCE

1. Patterson, V.K., and Craig, J., "Some studies on Telemedicine Healthcare System", *Journal of Institute of Engineering*, Volume 68, Page No.212, November 2019.

2. Park, H.C., and Bonato, P., "A Review of Wearable Sensors and Systems with Application in Rehabilitation", *Journal of Neuro Engineering and Rehabilitation*, Volume 24, Page No.96, May 2020.

3. Chen, M., and Mao, S., "A Review of Big Data: A Data", *Journal of Mobile Networks and Applications*, Volume 21, Page No.66, July 2020.

4. Jha, N.K., and Mosenia, A., "A Comprehensive Study of Security of Internet-of-Things", *Journal of IEEE Transactions on Emerging Topics in Computing*, Volume 13, Page No.64, August 2020.

5. Sahoo, P.K., and Mohapatra, S., "Analyzing the Healthcare Big Data with Prediction of Future Health Condition", *Journal of Access*, Volume 6, Page No.43, March 2021.

6. John A, Rogers., and Ghaffari, Roozbeh., "Study the Stretchable Bioelectronics for Medical Devices and Systems", *Journal of Science and Technology,* Volume 3, Page No.21, October 2021.

7. Galzarano, S., and Gravina, R., "Analyze the Framework for Collaborative Computing and Multi-Sensor Data Fusion in Body Sensor Network", *Journal of Information Fusion,* Volume 2, Page No.17, July 2022.

8. Kumar, R.K., and Bahrami, Majid., "A Comprehensive Review of Wearable and Smartphone Based Technologies for Fall Detection Systems", *Journal of Sensors,* Volume 1, Page No.14, September 2022.

9. Torrents, C., and Garcia, J., "Design and Evaluation of a Fall Detection System Using ADXL 335 Accelerometer", *Journal of IEEE Sensors Conference,* Volume 2, Page No.26, November 2022.

10. Lee, Sanghoon., and Jeong, Hee-Jin., "A Review of Wearable Electrocardiogram Monitoring Systems", *Journal of IEEE Sensors*, Volume 2, Page No.16, December 2022.