A Project Report on

"IOT BASED AUTOMATED PARALYSIS PATIENT HEALTHCARE MONITORING SYSTEM USING ARDUINO"

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS & COMMUNICATION ENGINEERING

Submitted By

P.PRABHAVATHI	20MH1A04H1
M.MANIKANTA	20MH1A04G2
V.LAKESH MOULI	20MH1A04F7
T.SAI VISWANADH	20MH1A04H9

Under the Esteemed Guidance of

Mr.K.CHANDRA SEKHAR, M.Tech, (Ph.D)

Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

Aditya College of Engineering(A)

(Approved by AICTE, New Delhi & Permanently Affiliated to JNTUK, Kakinada,accredited by NBA & NAAC-A++)

Recognized by UGC under the sections 2(f) and 12(B) of the UGC Act, 1956

Aditya Nagar, ADB Road, Surampalem , East Godavari District,A.P - 533 437

2020-2024

ADITYA COLLEGE OF ENGINEERING(A)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that the Project Report entitled

IOT Based Automated Paralysis Patient Healthcare Monitoring System Using Arduino

being submitted by

P.PRABHAVATHI	**20MH1A04H1**
M.MANIKANTA	**20MH1A04G2**
V.LAKESH MOULI	**20MH1A04F7**
T.SAI VISWANADH	**20MH1A04H9**

In partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Electronics and Communication Engineering, at Aditya College of Engineering is a bonafide work carried out by them under my guidance and supervision. The result embodied in this report has not been submitted to any other University or Institution for the award of any degreeor diploma.

Project Guide	**Head of the Department**
Mr.K.CHANDRA SEKHAR, M.Tech, (Ph.D)	**Mrs Ch.JANAKI DEVI, M.Tech, (Ph.D)**
Assistant Professor	Associate Professor & HOD
Department of ECE	Department of ECE
Aditya College of Engineering(A)	Aditya College of Engineering(A)

External Examiner

VISION & MISSION OF THE INSTITUTE

VISION

To induce higher planes of learning by imparting technical education with

- International standards
- Applied research
- Creative Ability
- Value based instruction and to emerge as a premiere institute.

MISSION

Achieving academic excellence by providing globally acceptable technical education by forecasting technology through

- Innovative Research And development
- Industry Institute Interaction
- Empowered Manpower

VISION & MISSION OF THE DEPARTMENT

VISION

"To be a center of excellence and renowned for Electronics &Communication Engineering education and research"

MISSION

M1:Enlighten the graduates in the basic concepts underlying the principles of analog and digital electronics, communication systems and advanced technologies.

M2:Provide state of the art infrastructure and research facilities

M3:Organizing industrial programs and social activities in collaboration with industries, NSS to disseminate knowledge

DECLARATION

We hereby declare that the entire project work embodied in this dissertation entitled "**IOT Based Automated Paralysis Patient Healthcare Monitoring System Using Arduino**" has been independently carried out by us. As per our knowledge, no part of this work has been submitted for any degree in any institution, university, and organization previously. We hereby boldly state that to the best of our knowledge, our work is free from plagiarism.

Yours sincerely

P.PRABHAVATHI	**20MH1A04H1**
M.MANIKANTA	**20MH1A04G2**
V.LAKESHMOULI	**20MH1A04F7**
T.SAIVISWANADH	**20MH1A04H9**

ACKNOWLEDGEMENT

We express our sincere gratitude and heartful thanks to the under stated person for the successful completion of our final project on **“IOT Based Automated Paralysis Patient Healthcare Monitoring System Using Arduino”.**

First and foremost we wish to thank our beloved guide **Mr.K.CHANDRA SEKHAR,** forhis kind guidance, valuable advices and utmost care at every stage of our final project.

We would like to thank **Mrs Ch.JANAKI DEVI,** head of the department of E.C.E who have provided vital information, which was necessary for success of the project.

The credit in successful completion of our final project goes to highly esteemed **Dr.A.RAMESH,** Principal of **Aditya College of Engineering(A)** for his kind guidance, commendable and noteworthy advices and even memorable encouragement to the same.

Yours sincerely

P.PRABHAVATHI	**20MH1A04H1**
M.MANIKANTA	**20MH1A04G2**
V.LAKESHMOULI	**20MH1A04F7**
T.SAIVISWANADH	**20MH1A04H9**

ABSTRACT

The IOT-based automated paralysis patient healthcare monitoring system represents anovel approach to address the challenges of continuous monitoring and management of patients with paralysis. This system integrates advanced IOT technologies with healthcare monitoring to provide real-time tracking of vital signs and movement patterns, enabling proactive intervention and personalized care for patients. The project encompasses meticulous planning and design, including the identification of essential sensors, actuators,and communication devices, as well as the development of a robust system architecture andsoftware applications. Through the implementation phase, hardware components are assembled, software solutions are developed, and rigorous testing is conducted to ensure thesystem's reliability and accuracy.

The testing and validation phase evaluates the system's performance under various conditions, validating its effectiveness in meeting clinical requirements and user expectations. Overall, the IOT based automated paralysis patient healthcare monitoring system offers promising potential to enhance patient outcomes, improve caregiver efficiency, and revolutionize healthcare delivery for individuals with paralysis

CONTENTS

LIST OF FIGURES **PAGE NO**

LIST OF TABLE PAGE NO

CHAPTER – 1
EMBEDDED SYSTEM

1.1 INTRODUCTION

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware.

A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious. All embedded systems are including computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer. The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes.

In some cases a microprocessor may be designed in such a way that application software fora particular purpose can be added to the basic software in a second process, after which it is notpossible to make further changes The applications software on such processors is sometimes referred to as firmware. The simplest devices consist of a single microprocessor (often called a "chip”), which may itself be packaged with other chips in a hybrid system or Application Specific Integrated Circuit (ASIC). Its input comes from a detector or sensor and its output goes to a switch or activator which (for example) may start or stop the operation of a machine or, by operating a valve, may control the flow of fuel to an engine.

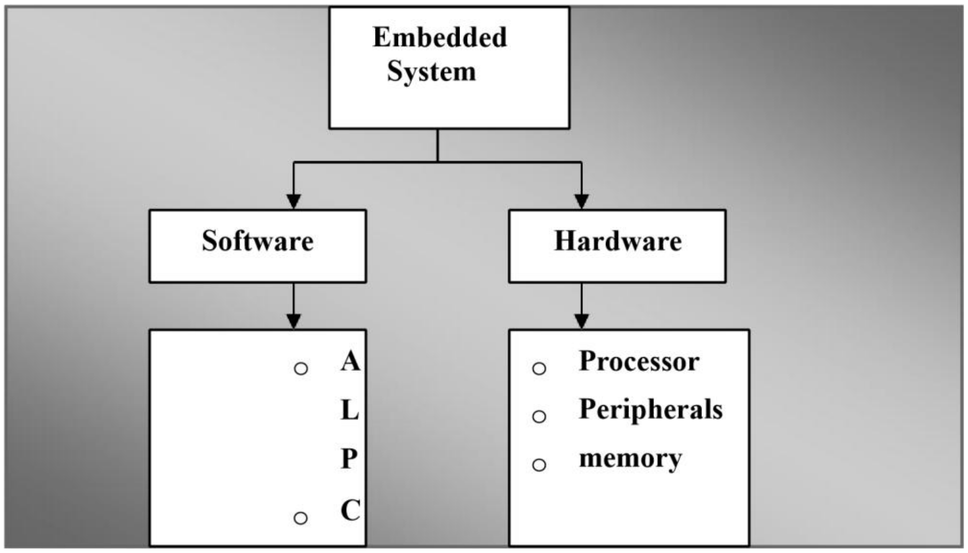


Figure 1.1: Block diagram of Embedded System

As the embedded system is the combination of both software and hardware Software deals withthe languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory.

Memory: It is used to store data or address.

Peripherals: These are the external devices connected.

Processor: It is an IC which is used to perform some task.

1.2 LITERATURE SURVEY

In First survey learn to IOT Based Patient Health Monitoring System, Author name is Amitabha Chakrabarty (25MAR.2018), Conference/Journal is international Research Journal ofEngineering and Technology. (IRJET), Advantages is telemedicine is the new technology used to improve patients health.

In second survey learn the paper title is Arduino Based Heart Rate Monitoring And Heart Attack Detection System, the author name is Bandana Mallick (Jan. 2016) Conference/Journal is international Journal of Science, Engineering and Technology Research.(IJSETR) the advantage is Quick response time

In third survey learn the paper title is A Smart patient Health Monitoring System using IOT, Author name is Narasimha Rao Jasti Madhu (MAY.2018) Conference/Journal is International Journal of Pure and Applied Mathematics. (IJPAM) Advantage is Reduces the human activity.

In fourth survey learn the paper title is Sensor Based Wearable System to Assist Paralytic Patient with Continuous Health Monitoring, author name is Kumara K R (MAR. 2017) Conference/Journal is International Journal on Future Revolution In Computer Science & Communication Engineering.(IJFRCSCE) Advantage is Real-time application.

CHAPTER 2
INTRODUCTION TO IOT

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

2.1 What is an Internet of Things (IOT)

Let's 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 locationor 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 Internetof 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.

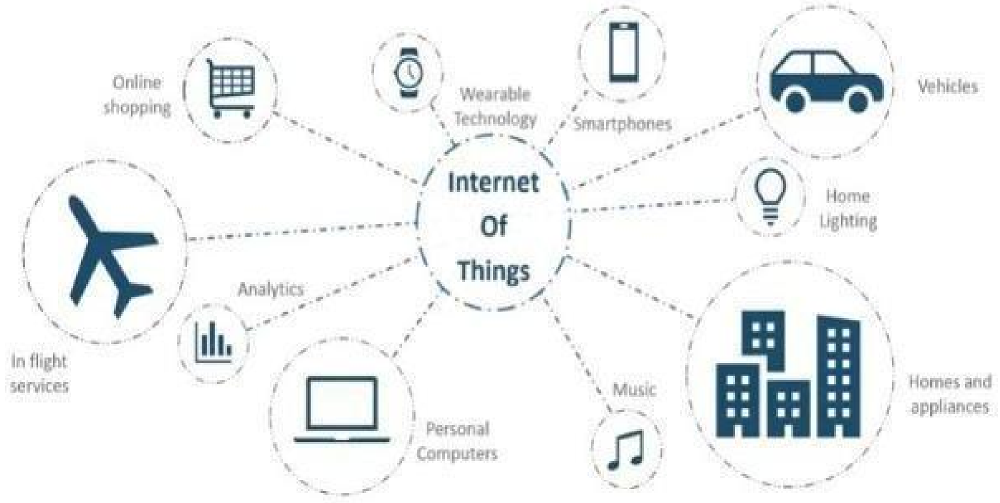


Figure 2.1: Internet of things

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 connectall the things around us. For example, a house, where we can connect our home appliances suchas 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, andalso track the location of the car.

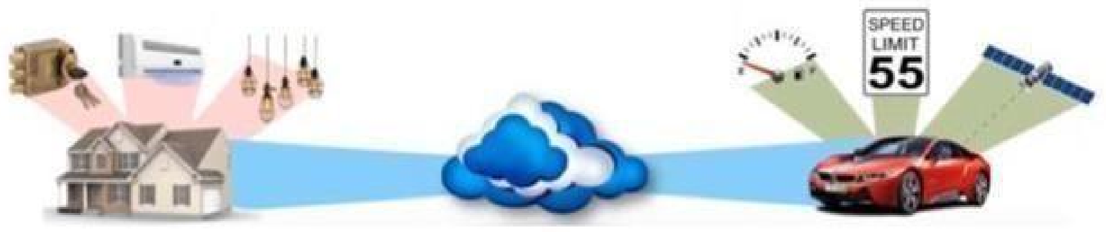


Figure 2.2: Example of IOT

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 theroom temperature to to be set at 25 or 26- 19 degree 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).

2.2 How does IOT work

The working of IOT is different for different IOT echo system (architecture). However, thekey concept of there working are 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 analyze the data from all multipledevices.

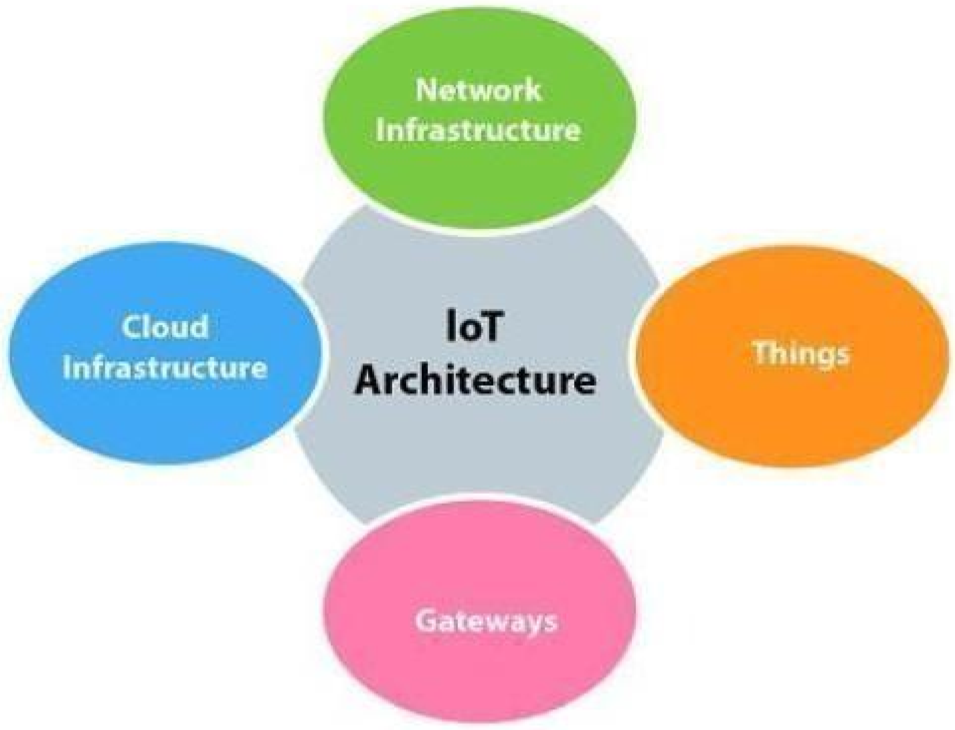


Figure 2.3: IOT Architecture

2.3 Features of IOT

The most important features of IOT on which it works are connectivity, analyzing, 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 bidirectional 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 through the use of data.

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

2.4 Advantages of IOT

Internet of things facilitates the several advantages in day-to-day life in the business sector.Some of its benefits are given below:

Efficient resource utilization: If we know the functionality and the way that how each device work we definitely increase the efficient resource utilization as well as monitor natural resources.

Minimize human effort: As the devices of IOT interact and communicate with each other and do lot of task for us, then they minimize the human effort.

Save time: As it reduces the human effort then it definitely saves out time. Time is the primaryfactor which can save through IOT platform.

Enhance Data Collection: o Improve security: Now, if we have a system that all these things are interconnected then we can make the system more secure and efficient.

2.5 Disadvantages

Interoperability: As IOT is still an evolving field, there is a lack of standardization across different platforms and devices. This lack of standards can lead to issues with interoperability, where devices from different manufacturers do not work well together, complicating thedeployment and operation of IOT solutions.

Increased Dependence on Technology: IOT encourages greater reliance on technology in everyday activities. This dependence can lead to a disruption of basic tasks in the event of Internet downtime or if the IOT systems fail for any reason.

Data Overload and Management: IOT devices generate vast amounts of data that need to be stored, processed, and analyzed. Managing this data effectively poses significant challenges, requiring sophisticated algorithms and storage solutions, which can be costly and resource- intensive.

Environmental Impact: The widespread adoption of IOT devices also raises environmental concerns. The production, operation, and disposal of millions of electronic devices contribute to e-waste, energy consumption, and environmental degradation.

Legal and Regulatory Issues: The rapid development and deployment of IOT technologies often outpace the corresponding legal and regulatory frameworks. This can lead to uncertainties regarding liability and compliance issues, especially when data breaches or device malfunctionsoccur.

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

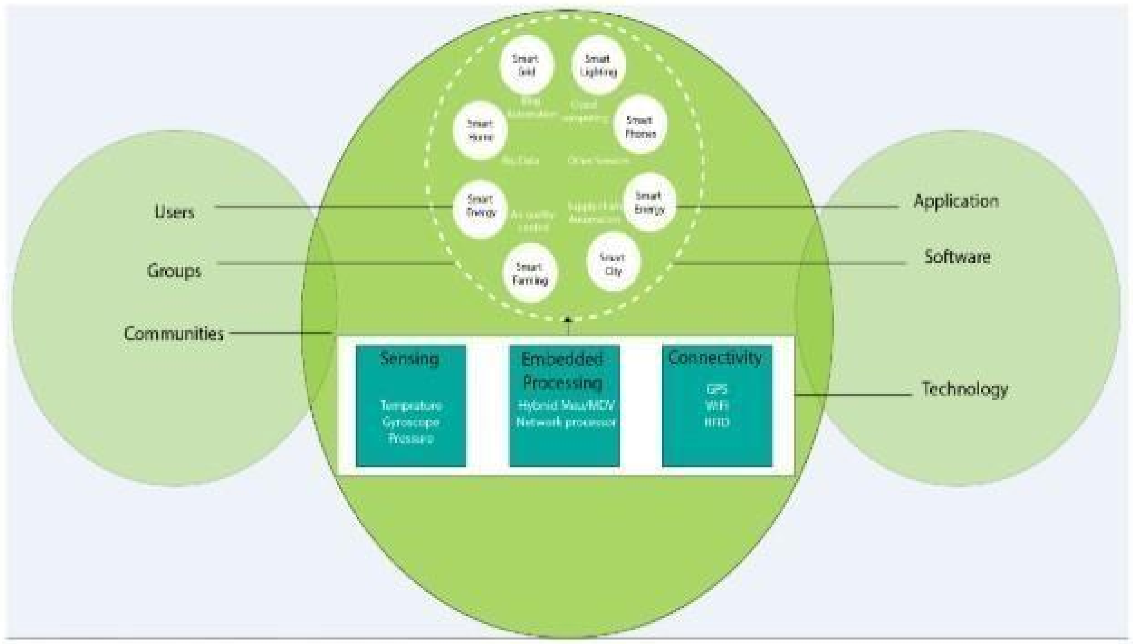


Figure 2.4: IOT Ecosystem

Sensing, Embedded processing, Connectivity: The IOT ecosystem senses its surrounding like temperature, gyroscope, pressure, etc. and make the embedded processing using devices. Thesedevices 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 25 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.

2.7 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

The IOT decision framework pays attention to six key decision areas in any IOT product.These decision areas are:

A.User Experience (UX)

B.Data

C.Business

D.Technology

E.Security

F.Standards & Regulations

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 betteruser experience. Then at the next step Data Decision Area, we have to explore data considerations for all the stages of IOT Technology Stack.

Let's see each of the Decision Area of IOT Decision Framework in detail:

1. User Experience Decision Area: This is the area where we concentrate about who are the users,what are their requirements and how to provide a great experience at each step of IOT stack without worrying about the technical details.

2. Data Decision Area: In this area, we make the overall data strategy such as the data flow overthe entire IOT stack to fulfill the user's requirements.

3. Business Decision Area: Based on the previous decisions area, we make the decision how product or services will became financial potential. At each of the IOT Stack level are monetizedabout the costs of providing services.

4. Technology Decision Area: In this area, we work with the technology for each layer to facilitatethe final solution.

5. Security Decision Area: After going through the implementation of technology it is importantto decide and provide the security at each stage of the IOT Stack.

6. Standards & Regulations Decision Area: At the last stage of IOT Decision Area, we identify the standards and regulations of product or services that will affect your product at each layer ofthe IOT Stack.

2.8 IOT DEVICES

Internet of Things Devices is non-standard devices that connect wirelessly to a network with each other and able to transfer the data. IOT devices are enlarging the internet connectivitybeyond standard devices such as smartphones, laptops, tablets, and desktops. Embedding these devices with technology enable us to communicate and interact over the networks and they can be remotely monitored and controlled There are large varieties of IOT devices available based on IEEE 802.15.4 standard. These devices range from wireless motes, attachable sensor boards to interface-board which are useful for researchers and developers.

IOT devices include computer devices, software, wireless sensors, and actuators. TheseIOT devices are connected over the internet and enabling the data transfer among objects or people automatically without human intervention.

PROPERTIES OF IOT DEVICES

Some of the essential properties of IOT devices are mention below:

1. Sense: The devices that sense its surrounding environment in the form of temperature, movement, and appearance of things, etc.

2.Send and receive data: IOT devices are able to send and receive the data over the networkconnection.

3.Analyze: The devices can able to analyze the data that received from the other device over theinternet networks.

4.Controlled: IOT devices may control from some endpoint also. Otherwise, the IOT devices are themselves communicate with each other endlessly leads to the system failure.

2.9 IOT – PLATFORM

As in IOT, all the IOT devices are connected to other IOT devices and application to transmit and receive information using protocols. There is a gap between the IOT device and IOT application. An IOT Platform fills the gap between the devices (sensors) and application(network). Thus we can say that an IOT platform is an integrated service that fulfills the gap between the IOT device and application and offers you to bring physical object online.

An IOT platform is a central software solution that enables seamless integration, management, and data exchange between IOT devices and cloud-based applications or services. It acts as a bridge between the physical world of connected devices and the digital world of data processing and analytics. By providing these capabilities, an IOT platform simplifies the development, deployment, and management of IOT solutions, enabling businesses to focus on their core applications and services while leveraging the power of connected devices and data- driven insights

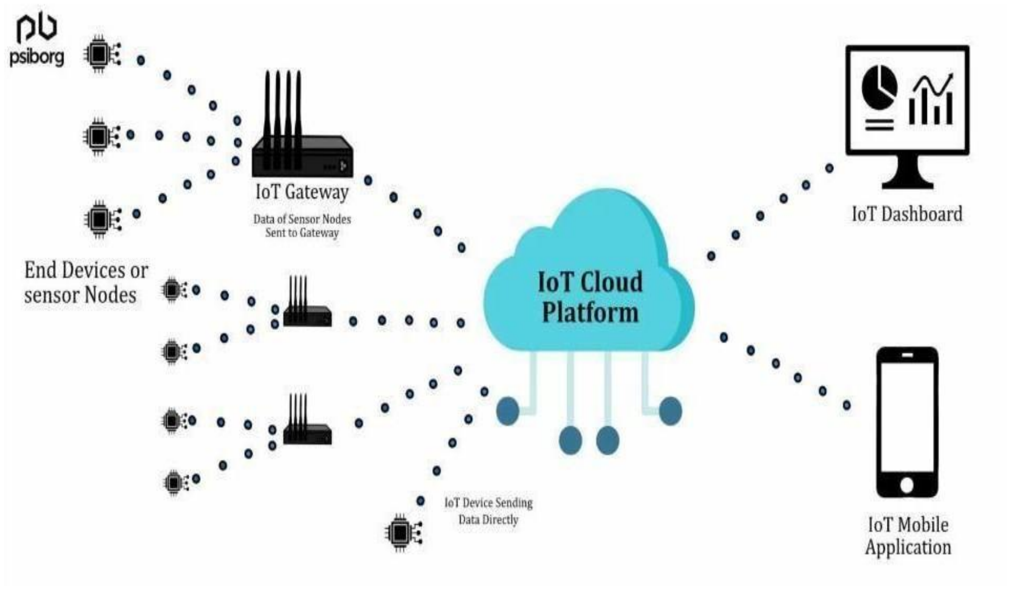

psiborg
IoT Gateway
Data of Sensor Nodes Sent to Gateway
End Devices or sensor Nodes
IoT Cloud Platform
IoT Dashboard
IoT Device Sending Data Directly
IoT Mobile Application

Figure 2.5: IOT Cloud Platform

The Internet of Things (IOT) strives to connect devices remotely for seamless functioning and ease of operations. An IOT platform bridges the gap between device sensors and data networks. It provides an insight into the data used in the backend application. An IOT platform is a set of components that allows developers to spread out the applications, remotely collect data, secure connectivity, and execute sensor management.

An IOT platform manages the connectivity of the devices and allows developers to build new mobile software applications. It facilitates the collection of data from devices and enables business transformation. It connects different components, ensuring an uninterrupted flow of communication between the devices.

The IOT platform helps in understanding the customers' needs better and facilitate the creation of products that fulfill their requirements. It provides organizations with greater intelligence and visibility into operations, which enables better decision-making.

CHAPTER – 3
ARDUINO UNO

3.1 INTRODUCTION

In this chapter 3.1, we will learn about the different components on the Arduino board. Wewill study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

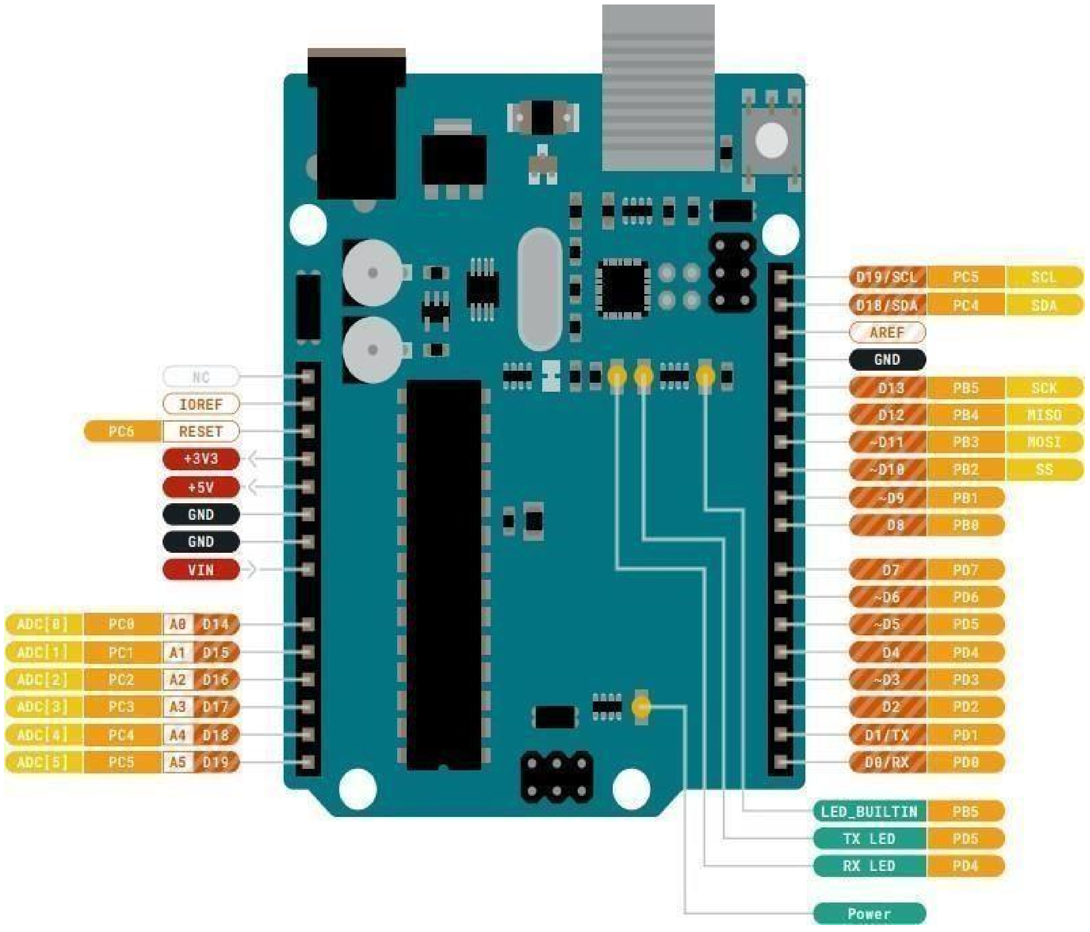


Figure 3.1: Arduino Pin Diagram

POWER USB

Arduino board can be powered by using the USB cable from your computer.All you need todo is connect the USB cable to the USB connection (1).

Power (Barrel Jack)

Arduino boards can be powered directly from the AC mains power supply by connectingit to the Barrel Jack (2).

Voltage Regulator

The function of the voltage regulator is to control the voltage given to the Arduino board andstabilize the DC voltages used by the processor and other elements.

Crystal Oscillator

The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.

5, 17. Arduino Reset

You can reset your Arduino board, i.e., starts your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).

6,7,8,9 Pins (3.3, 5, GND, Vin)

i.3.3V (6) − Supply 3.3 output volt

ii.5V (7) − Supply 5 output volt

iii. Most of the components used with Arduino board works fine with 3.3 volt and 5 volt.

iv.GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit

Analog pins

The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it intoa digital value that can be read by the microprocessor.

Main microcontroller

Each Arduino board has its own microcontroller (11). You can assume it as the brain of yourboard. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your boardhas before loading up a new program from the Arduino IDE. This information is available on thetop of the IC. For more details about the IC construction and functions, you can refer to the datasheet.

ICSP pin

Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slavingthe output device to the master of the SPI bus.

Power LED indicator

This LED should light up when you plug your Arduino into a power source to indicate thatyour board is powered up correctly. If this light does not turn on, then there is something wrongwith the connection.

TX and RX LEDs

On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate usedby the board. RX flashes during the receiving process

Digital I/O

The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse WidthModulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labelled “~” can be used to generate PWM.

AREF

AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage(between 0 and 5 Volts) as the upper limit for the analog input pins. After learning about the mainparts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once welearn this, we will be ready to upload our program on the Arduino board

ATMEGA328 CONTROLLER

ATMEGA328P is high performance, low power controller from Microchip. ATMEGA328P is an 8-bit microcontroller based on AVR RISC architecture. It is the most popular of all AVR controllers as it is used in ARDUINO boards

Figure 3.2: Atmega328 Controller

3.2 ARDUINO IDE SETUP

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computerand prepare the board to receive the program via USB cable.

Step 1:

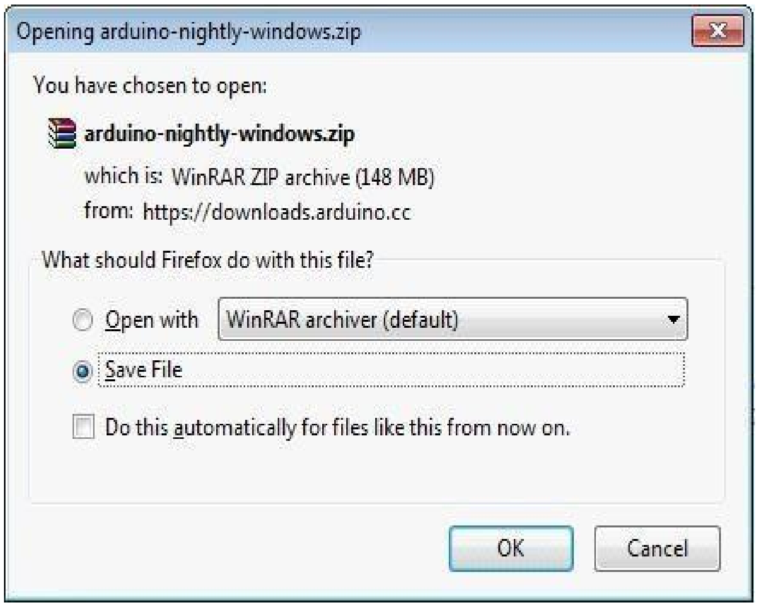
First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connectto a USB printer as shown in the following image.

Figure 3.3: Arduino Cable

Step 2:

Download Arduino IDE Software. You can get different versions of Arduino IDE fromthe Download page on the Arduino Official website

You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.



Opening arduino-nightly-windows.zip

You have chosen to open:

arduino-nightly-windows.zip

which is: WinRAR ZIP archive (148 MB)

from: https://downloads.arduino.cc

What should Firefox do with this file?

( ) Open with WinRAR archiver (default)

(●) Save File

[ ] Do this automatically for files like this from now on.

OK Cancel

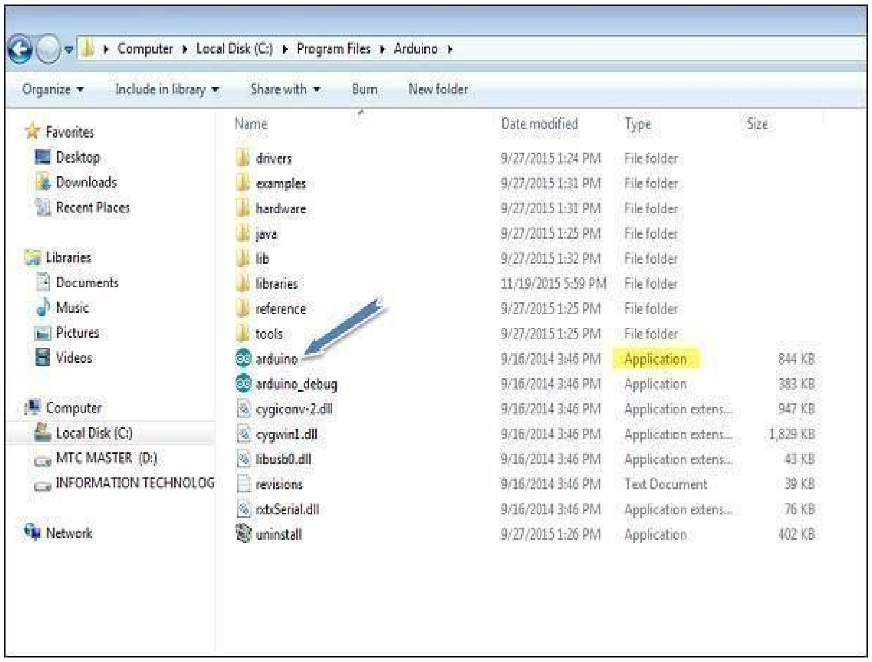
Figure 3.4: Arduino IDE

Step 3:

Power up your board. The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external powersupply. If you are using an Arduino Diecimila, you have to make sure that the board is configuredto draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check thatit is on the two pins closest to the USB port. Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4:

Launch Arduino IDE. Inside the folder, you can find the application icon with an infinity label (application.exe). Double - click the icon to start the IDE.



Name	Date modified	Type	Size
drivers	9/27/2015 1:24 PM	File folder	
examples	9/27/2015 1:31 PM	File folder	
hardware	9/27/2015 1:31 PM	File folder	
java	9/27/2015 1:25 PM	File folder	
lib	9/27/2015 1:32 PM	File folder	
libraries	11/19/2015 5:59 PM	File folder	
reference	9/27/2015 1:25 PM	File folder	
tools	9/27/2015 1:25 PM	File folder	
arduino	9/16/2014 3:46 PM	Application	844 KB
arduino_debug	9/16/2014 3:46 PM	Application	383 KB
cygiconv-2.dll	9/16/2014 3:46 PM	Application extens...	947 KB
cygwin1.dll	9/16/2014 3:46 PM	Application extens...	1,829 KB
libusb0.dll	9/16/2014 3:46 PM	Application extens...	43 KB
revisions	9/16/2014 3:46 PM	Text Document	39 KB
rxtxSerial.dll	9/16/2014 3:46 PM	Application extens...	76 KB
uninstall	9/27/2015 1:26 PM	Application	402 KB

Figure 3.5: Arduino IDE Launcher

Step 5:

Open your first project. Once the software starts, you have two options –

- Create a new project.
- Open an existing project example. To create a new project, select File → New.

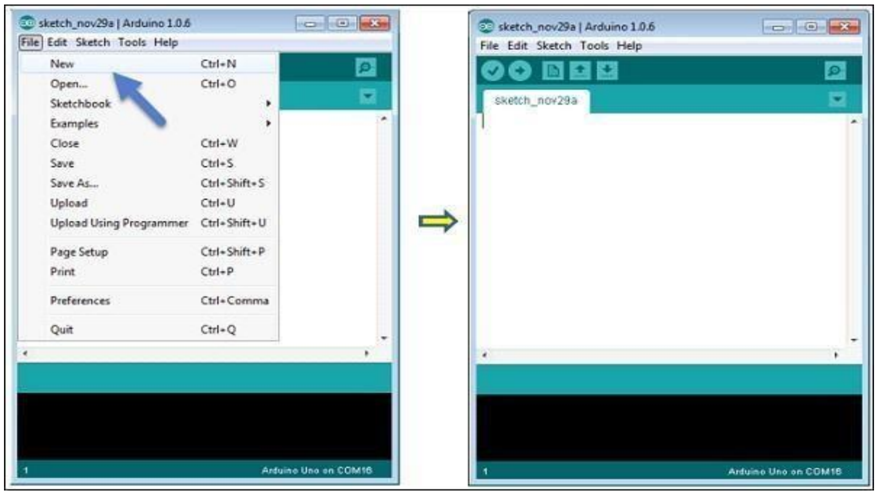
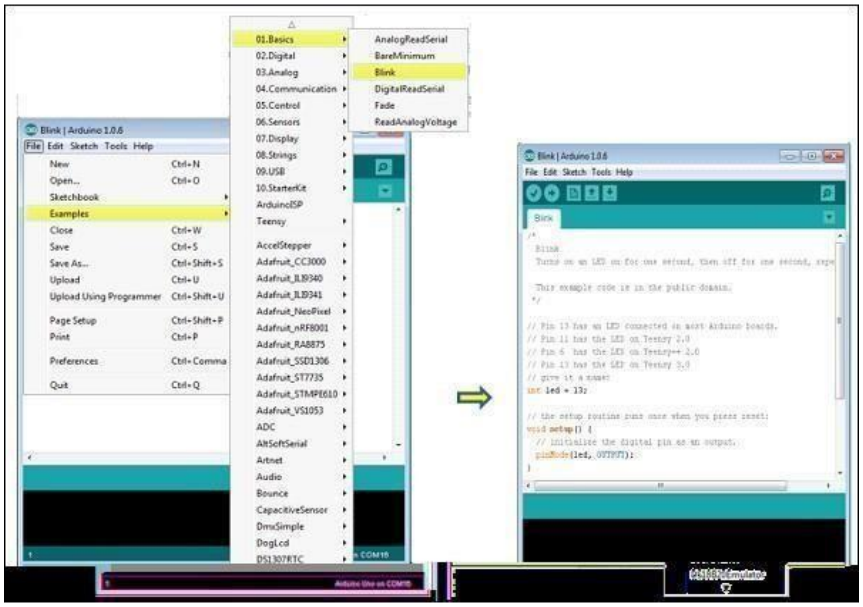


Figure 3.6: Create new Project

To open an existing project example, select File → Example → Basics → Blink.



```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repe

  This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// Pin 11 has the LED on Teensy 2.0
// Pin 6  has the LED on Teensy++ 2.0
// Pin 13 has the LED on Teensy 3.0
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}
```

Figure 3.7: First Project

Here, we are selecting just one of the examples with the name Blink. It turns the LED onand off with some time delay. You can select any other example from the list.

Step 6:

Select your Arduino board. To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board

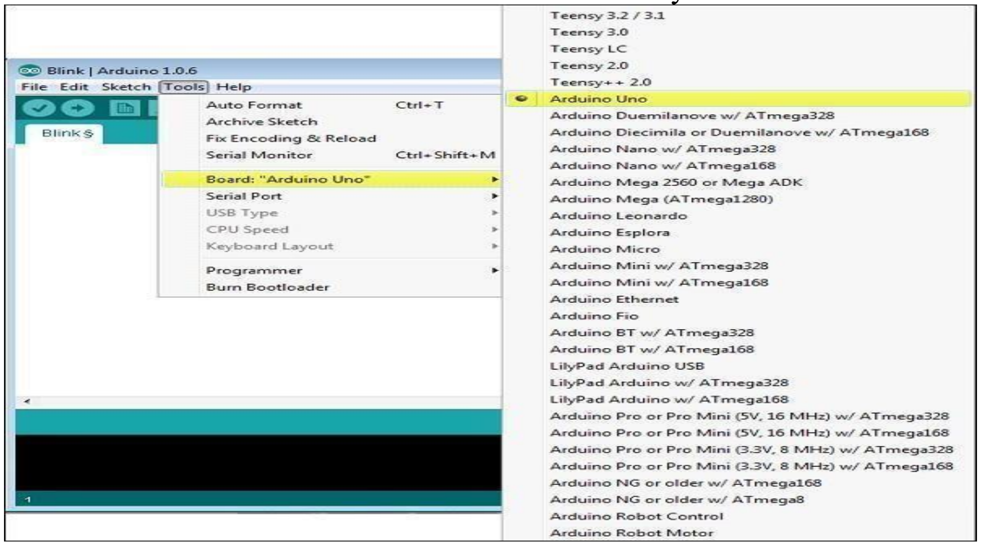

Blink | Arduino 1.0.6
File Edit Sketch Tools Help
Blink §
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board: "Arduino Uno"
Serial Port
USB Type
CPU Speed
Keyboard Layout
Programmer
Burn Bootloader
Teensy 3.2 / 3.1
Teensy 3.0
Teensy LC
Teensy 2.0
Teensy++ 2.0
Arduino Uno
Arduino Duemilanove w/ ATmega328
Arduino Diecimila or Duemilanove w/ ATmega168
Arduino Nano w/ ATmega328
Arduino Nano w/ ATmega168
Arduino Mega 2560 or Mega ADK
Arduino Mega (ATmega1280)
Arduino Leonardo
Arduino Esplora
Arduino Micro
Arduino Mini w/ ATmega328
Arduino Mini w/ ATmega168
Arduino Ethernet
Arduino Fio
Arduino BT w/ ATmega328
Arduino BT w/ ATmega168
LilyPad Arduino USB
LilyPad Arduino w/ ATmega328
LilyPad Arduino w/ ATmega168
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega168
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168
Arduino NG or older w/ ATmega168
Arduino NG or older w/ ATmega8
Arduino Robot Control
Arduino Robot Motor

Figure 3.8: Tool Box

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7:

Select your serial port. Select the serial device of the Arduino board. Go to **Tools → Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your

Arduino board and re-open the menu, the entry that disappears should be of the Arduino board.Reconnect the board and select that serial port.

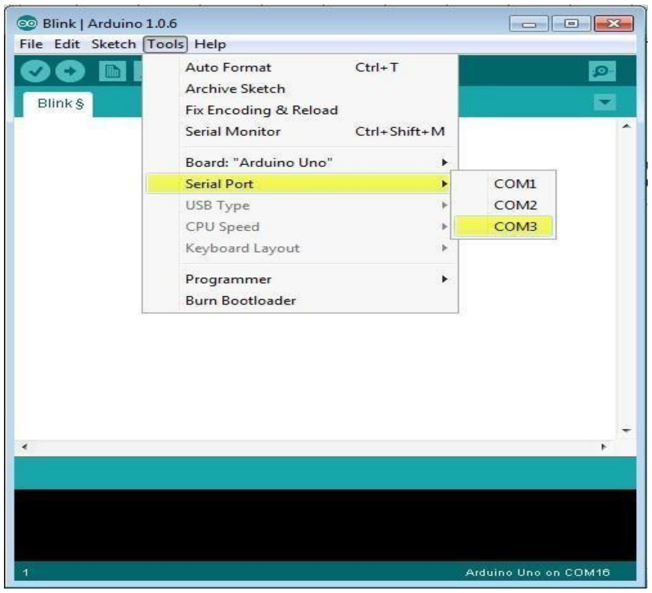


Figure 3.9: Selecting a Serial Port

Step 8:

Upload the program to your board. Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDEtoolbar.

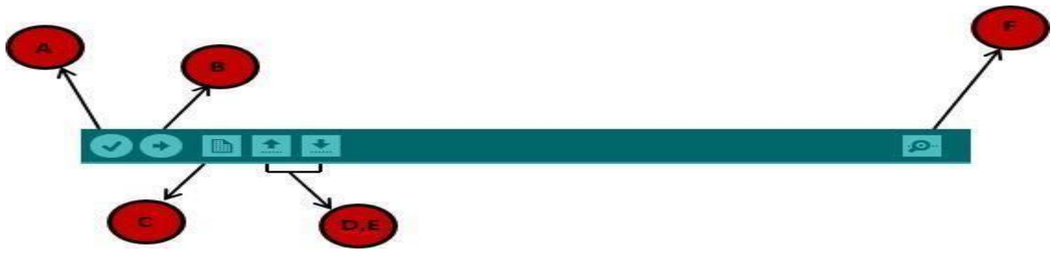


Figure 3.10: Menu Bar

A− Used to check if there is any compilation error.

B− Used to upload a program to the Arduino board.

C− Shortcut used to create a new sketch.

D− Used to directly open one of the example sketch.

E− Used to save your sketch.

F− Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see theRX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note:

If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

In this chapter, we will study in depth, the Arduino program structure and we will learn more new terminologies used in the Arduino world. The Arduino software is open-source. The sourcecode for the Java environment is released under the GPL and the C/C++ microcontroller librariesare under the LGPL.

Sketch:

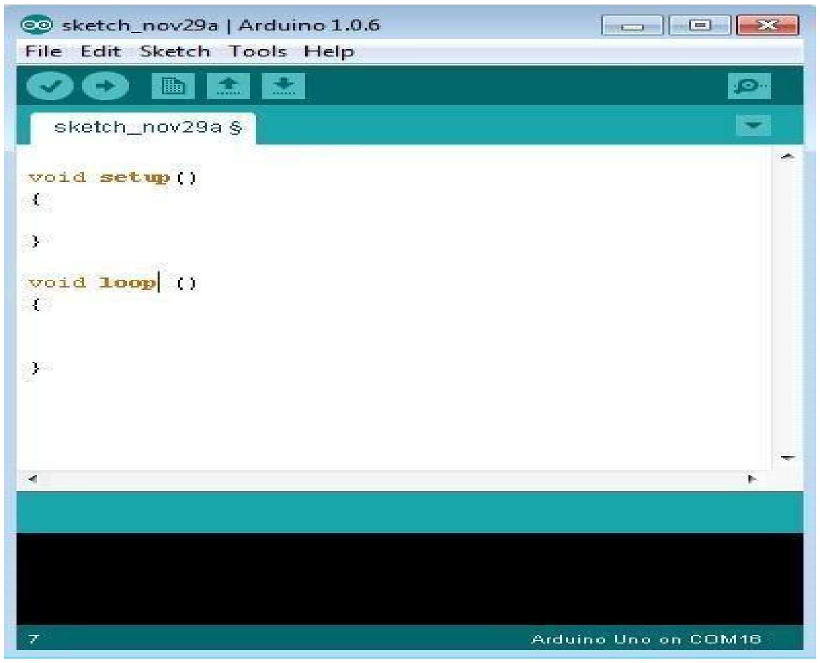
The first new terminology is the Arduino program called "**sketch**".

3.3 ARDUINO STRUCTURE

Arduino programs can be divided in three main parts: Structure, Values and Functions. In this tutorial, we will learn about the Arduino software program, step by step, and how we can write the program without any syntax or compilation error. Let us start with the Structure. Software structure consist of two main functions –

i.Setup( ) function

ii. Loop( ) function



```
void setup()
{

}

void loop ()
{


}
```

Figure 3.11: Arduino Code Structure

CHAPTER – 4
NodeMCU

4.1 DEFINITION OF NodeMCU:

Node Micro Controller Unit is an open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espressif Systems, contains the crucial elements of a computer: CPU, RAM, networking (WiFi), and even a modern operating system and SDK. Thatmakes it an excellent choice for Internet of Things (IOT) projects of all kinds.

However, as a chip, the ESP8266 is also hard to access and use. You must solder wires, with the appropriate analog voltage, to its pins for the simplest tasks such as powering it on or sending a keystroke to the "computer" on the chip. You also have to program it in low-level machine instructions that can be interpreted by the chip hardware. This level of integration is nota problem using the ESP8266 as an embedded controller chip in mass-produced electronics. It isa huge burden for hobbyists, hackers, or students who want to experiment with it in their own IOT projects.

4.2 ESP8266 ARDUINO CORE

As Arduino.cc began developing new MCU boards based on nonAVR processors like the ARM/SAM MCU and used in the Arduino Due, they needed to modify the Arduino IDE so thatit would be relatively easy to change the IDE to support alternate toolchains to allow Arduino C/C++ to be compiled for these new processors. They did this with the introduction of the Board Manager and the SAM Core. A "core" is the collection of software components required by theBoard Manager and the Arduino IDE to compile an Arduino C/C++ source file for the target MCU's machine language. Some ESP8266 enthusiasts developed an Arduino core for the ESP8266 WiFi SoC, popularly called the "ESP8266 Core for the Arduino IDE".[16] This has become a leading software development platform for the various ESP8266-based modules and development boards, including NodeMCUs.

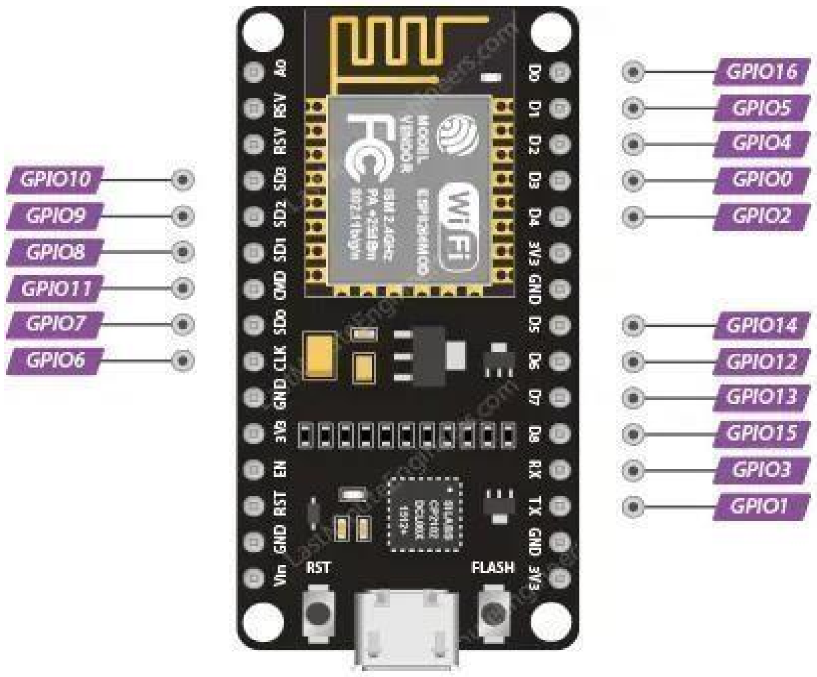


Figure 4.1: ESP8266

4.3 NODEMCU DEVELOPMENT

NodeMCU is open source platform, their hardware design is open for edit/modify/build.NodeMCU Dev Kit/board consist of ESP8266 wifi enabled chip. The ESP8266 is a low-cost Wi-Fi chip developed by Espressif Systems with TCP/IP protocol. For more information about ESP8266, you can refer ESP8266 WiFi Module. There is Version2 (V2) available for NodeMCU Dev Kit i.e. NodeMCU Development Board v1.0 (Version2), which usually comes in black colored PCB

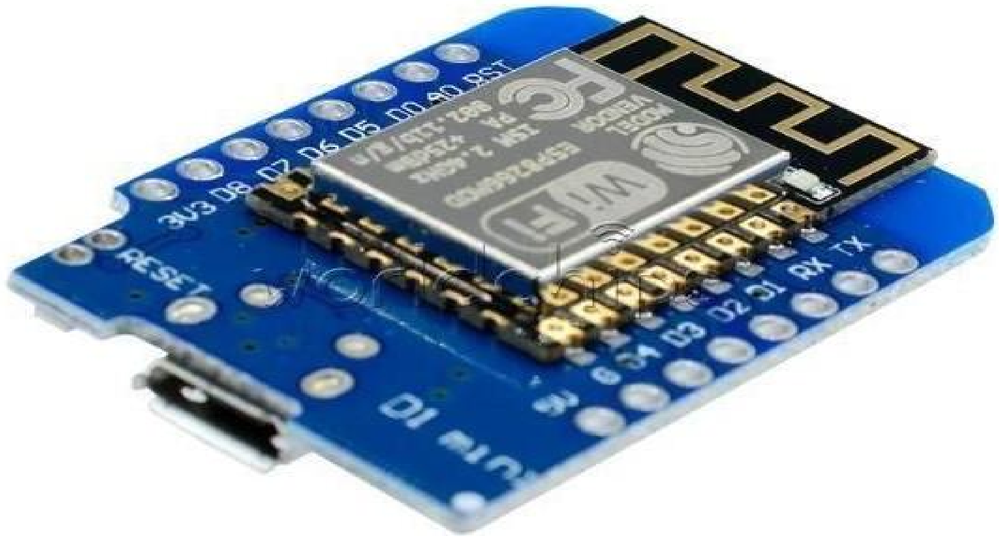


Figure 4.2: Nodemcu development

For more information about NodeMCU Boards available in market refer NodeMCU Development Boards. NodeMCU Dev Kit has Arduino like Analog (i.e. A0) and Digital (D0- D8) pins on its board. It supports serial communication protocols i.e. UART, SPI, I2C etc. Using such serial protocols we can connect it with serial devices like I2C enabled LCD display, Magnetometer HMC5883, MPU-6050 Gyro meter + Accelerometer, RTC chips, GPS modules,touch screen displays, SD cards etc.

NodeMCU Development board is featured with wifi capability, analog pin, digital pins and serial communication protocols. 45 To get start with using NodeMCU for IOT applications first we need to know about how to write/download NodeMCU firmware in NodeMCU Development Boards. And before that where this NodeMCU firmware will get as per our requirement.

4.4 Codes For NodeMCU

The ESP8266 NodeMCU can be programmed using different Integrated Development Environments (IDEs) and programming languages. You mentioned two popular options:

ESPlorer IDE with Lua Scripts:

i.ESPlorer is a dedicated IDE for programming the NodeMCU using the Lua scripting language. **ii**.Lua is a lightweight, embeddable scripting language built on top of C, making it a good fit forthe resource-constrained ESP8266.

iii. Using Lua scripts with the ESPlorer IDE allows you to write code specifically tailored for theNodeMCU and leverage its capabilities directly.

iv. The Lua approach provides a more low-level and closely integrated programming experiencewith the NodeMCU hardware.

Arduino IDE with Arduino Sketches:

i. The Arduino IDE is a widely popular development environment, primarily designed forprogramming Arduino boards.

ii. However, thanks to the Arduino core for the ESP8266, you can also develop applications for the NodeMCU using the familiar Arduino IDE and the Arduino programming language (whichis based on C/C++).

iii. This approach allows Arduino developers to leverage their existing knowledge and easily transition to programming the NodeMCU without having to learn a new language (Lua) or IDE(ESPlorer).

iv. The Arduino core for ESP8266 provides a higher-level abstraction layer, simplifying the programming experience but potentially sacrificing some low-level control compared to using Lua scripts directly.

Both options are viable and widely used for developing applications on the ESP8266 NodeMCU. The decision ultimately comes down to your personal preferences, project requirements, and existing knowledge or willingness to learn a new language and IDE.

4.5 Difference in using ESPlorer and Arduino IDE

Well, there is a programming language difference we can say while developing application for NodeMCU using ESPlorer IDE and Arduino IDE. We need to code in C\C++ programming language if we are using Arduino IDE for developing NodeMCU applications andLua language if we are using ESPlorer IDE. Basically, NodeMCU is Lua Interpreter, so it can understand Lua script easily. When we write Lua scripts for NodeMCU and send/upload it to NodeMCU, then they will get executes sequentially. It will not build binary firmware file of code for NodeMCU to write. It will send Lua script as it is to NodeMCU to get execute. In Arduino IDE when we write and compile code, ESP8266 toolchain in background creates binary firmware file of code we wrote. And when we upload it to NodeMCU then it will flash all NodeMCU firmware with newly generated binary firmware code. In fact, it writes the complete firmware. That's the reason why NodeMCU not accept further Lua scripts/code afterit is getting flashed by Arduino IDE. After getting flashed by Arduino sketch/code it will be nomore Lua interpreter and we got error if we try to upload Lua scripts.

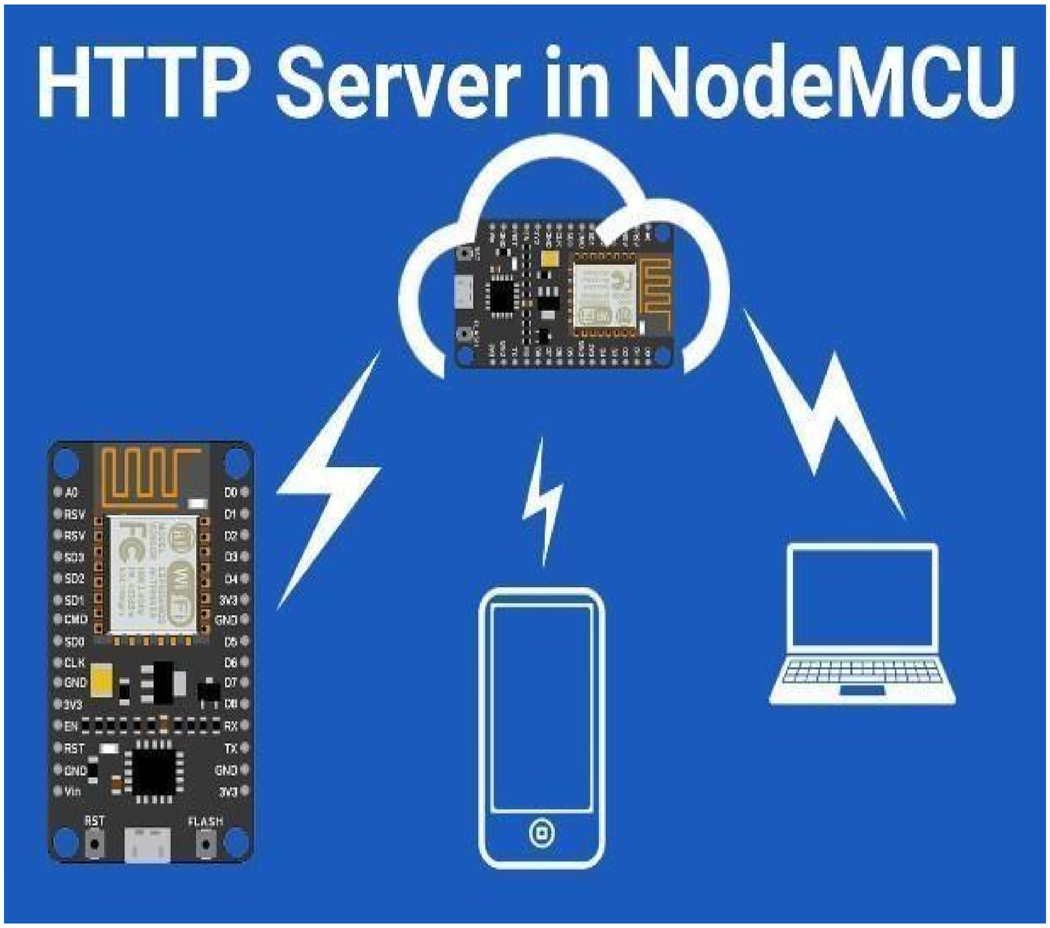


Figure 4.3: Http Server On Nodemcu With Arduino IDE

4.6 NodeMCU as HTTP Server using Wi-Fi AP mode

NodeMCU wi-fi has Access Point (AP) mode through which it can create Wireless LAN to which any wi-fi enabled device can connect as shown in the below figure.

As we are making an HTTP server for LED on/off functionality, we are going to make a simple HTML page that will be visible at the client-side and able to take user input for LED on/off. It is a user-friendly representation of button input that takes input from the user on click.

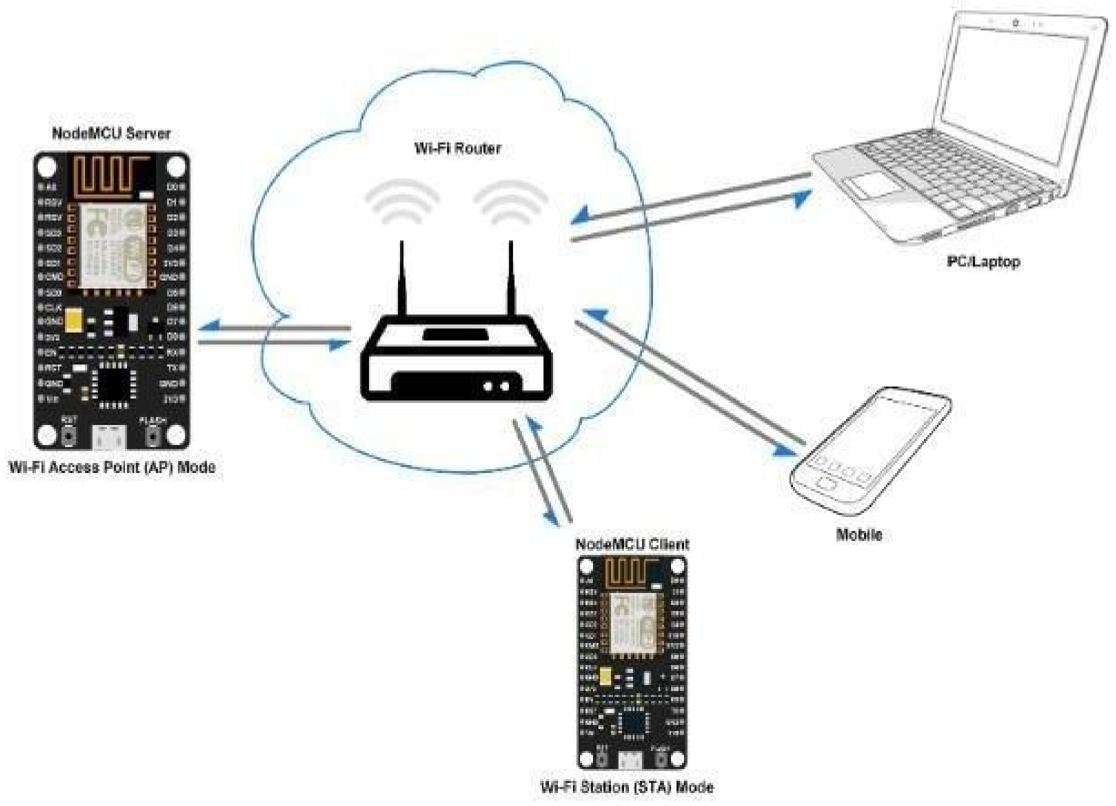


Figure 4.4: NodeMCU as HTTP Server using Wi-Fi STA mode

We need to write two HTML pages for LED ON and LED OFF state i.e. when a client clicks the LED ON button, then in the next action, we need to provide options for LED OFF. Below are the two HTML code snippets for LED ON and LED OFF state presentation.NodeMCU gets IP from the Wi-Fi router to which it is connected. With this IP address, it can actas an HTTP server to which any wi-fi device can connect.

CHAPTER – 5
LIQUID CRYSTAL DISPLAY

5.1 INTRODUCTION

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

Liquid crystal display is very important device in embedded system. It offers highflexibility to user as he can display the required data on it. But due to lack of proper approach toLCD interfacing many of them fail. Many people consider LCD interfacing a complex job but according to me LCD interfacing is very easy task, you just need to have a logical approach. Thispage is to help the enthusiast who wants to interface LCD with through understanding. Copy and Paste technique may not work when an embedded system engineer wants to apply LCDinterfacing in real world projects.

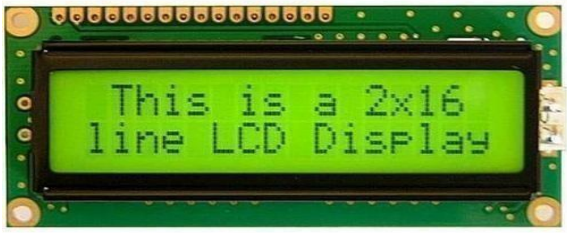


Figure 5.1: 2x16 LCD

5.2 PIN DESCRIPTION

Most LCDs with 1 controller has 14 Pins and LCDs with 2 controller has 16 Pins (two pinsare extra in both for back-light LED connections).

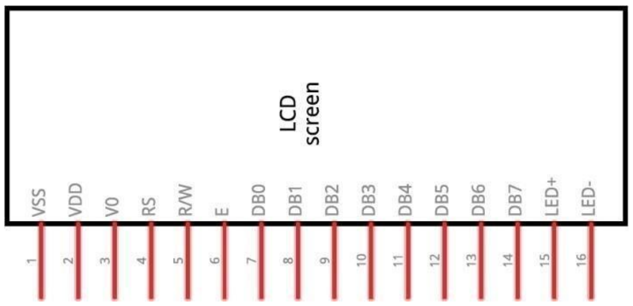


Figure 5.2: Pin Diagram of 2x16 lines LCD

PIN	SYMBOL	FUNCTION
1	Vss	Power Supply(GND)
2	Vdd	Power Supply(+5V)
3	Vo	Contrast Adjust
4	RS	Instruction/Data Register Select
5	R/W	Data Bus Line
6	E	Enable Signal
7-14	DB0-DB7	Data Bus Line
15	A	Power Supply for LED B/L(+)
16	K	Power Supply for LED B/L(-)

Table 5.2 : Pin Description

5.3 CONTROL

LinesE.N:

Line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD, your program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

RS:

Line is the "Register Select" line. When RS is low (0), the data is to be treated as a commandor special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter"T" on the screen you would set RS high.

RW:

Line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. RW will almost always be low. Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

Logic status on control lines:

i. E - 0 Access to LCD disabled
- 1 Access to LCD enabled

ii. R/W - 0 Writing data to LCD
- 1 Reading data from LCD

iii. RS - 0 Instructions
-1 character

Writing data to the LCD:

1. Set RS bit to logic 0 or 1 (instruction or character)

2. Set R/W bit to low

3. Set data to data lines (if it is writing)

4. Set E line to high

5. Set E line to low

Read data from data lines (if it is reading) on LCD:

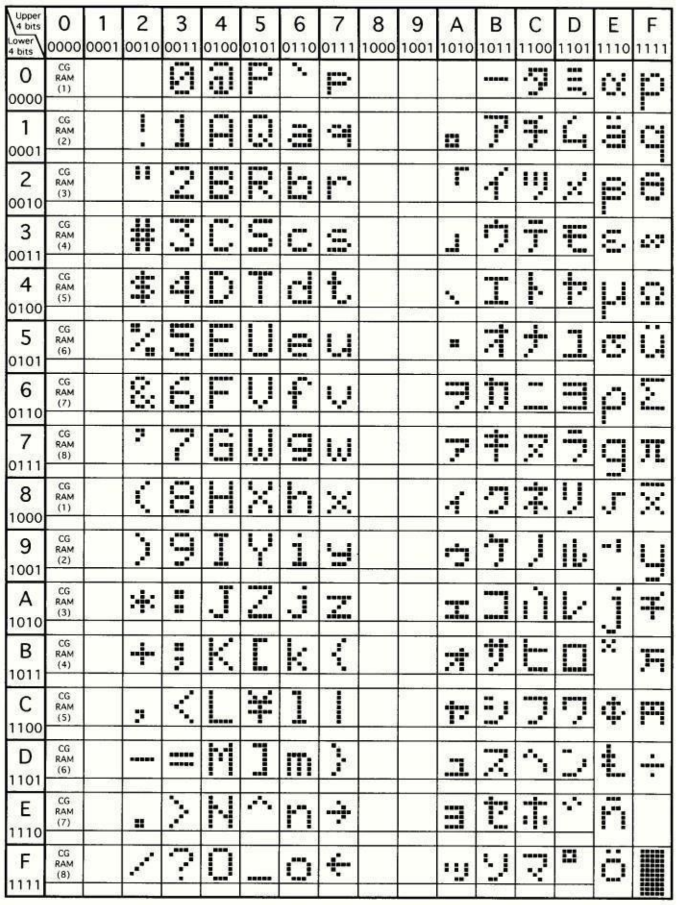
1. Set RS bit to logic 0 or 1 (instruction or character)

2. Set R/W bit to low

3. Set data to data lines (if it is writing)

4. Set E line to high

5. Set E line to low



Upper 4 bits / Lower 4 bits	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111
0 0000	CG RAM (1)			0	@	P	`	p				ー	タ	ミ	α	p
1 0001	CG RAM (2)		!	1	A	Q	a	q			。	ア	チ	ム	ä	q
2 0010	CG RAM (3)		"	2	B	R	b	r			「	イ	ツ	メ	β	θ
3 0011	CG RAM (4)		#	3	C	S	c	s			」	ウ	テ	モ	ε	∞
4 0100	CG RAM (5)		$	4	D	T	d	t			、	エ	ト	ヤ	μ	Ω
5 0101	CG RAM (6)		%	5	E	U	e	u			・	オ	ナ	ユ	σ	ü
6 0110	CG RAM (7)		&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
7 0111	CG RAM (8)		'	7	G	W	g	w			ァ	キ	ヌ	ラ	g	π
8 1000	CG RAM (1)		(	8	H	X	h	x			ィ	ク	ネ	リ	√	x̄
9 1001	CG RAM (2)		)	9	I	Y	i	y			ゥ	ケ	ノ	ル	⁻¹	y
A 1010	CG RAM (3)		*	:	J	Z	j	z			ェ	コ	ハ	レ	j	千
B 1011	CG RAM (4)		+	;	K	[	k	{			ォ	サ	ヒ	ロ	ˣ	万
C 1100	CG RAM (5)		,	<	L	¥	l	|			ャ	シ	フ	ワ	¢	円
D 1101	CG RAM (6)		-	=	M	]	m	}			ュ	ス	ヘ	ン	£	÷
E 1110	CG RAM (7)		.	>	N	^	n	→			ョ	セ	ホ	゛	ñ	
F 1111	CG RAM (8)		/	?	O	_	o	←			ッ	ソ	マ	゜	ö	█

Figure 5.3: Character Details in LCD

CHAPTER - 6
SENSORS

6.1 TEMPERATURE SENSOR:

6.1.1 Introduction

Temperature is a fundamental parameter in countless processes and systems, influencingeverything from chemical reactions to climate control in our daily lives. To quantify and monitor temperature accurately, temperature sensors have become indispensable tools across a broad spectrum of industries and applications. In essence, temperature sensors serve as the sensory organs of modern technology, providing vital feedback for systems to maintain optimal performance, ensure safety, and enhance efficiency. By converting temperature variations into electrical signals, these sensors enable precise monitoring, control, and analysis of thermal conditions in diverse environments.

6.1.2 Working

- The LM35 consists of a temperature-sensitive integrated circuit housed within a smallpackage.
- Inside the package, the heart of the sensor is a temperature-sensitive diode connected in adifferential configuration with a precision amplifier.
- The integrated circuit is designed to exhibit a linear change in output voltage with temperature.
- The temperature-sensitive diode within the LM35 has a known characteristic where itsforward voltage changes linearly with temperature.
- This temperature-dependent voltage is then amplified by the internal amplifier stage toprovide a proportional output voltage.
- The output voltage varies at a rate of 10 mV per degree Celsius change in temperature.
- Many LM35 sensors are factory calibrated to ensure accurate temperature readings acrosstheir specified operating range.
- Calibration involves adjusting the internal circuitry to provide a linear and accurate responseto temperature changes.

- This linear relationship simplifies the interpretation of temperature readings, as no complexconversion algorithms are required.
- The LM35 exhibits low self-heating characteristics, minimizing the impact of internal temperature changes on the accuracy of temperature measurements.

6.1.3 Specifications

Temperature Range: Typically -55°C to +150°C. This range may vary slightly between different manufacturers or specific versions of the LM35.

Output Voltage: 10mV per degree Celsius. For instance, at 25°C, the output voltage would be250mV.

Supply Voltage: Can operate within a wide range of supply voltages, typically between 4 to 30volts. This feature makes it compatible with various power sources commonly found in electronic circuits.

Accuracy: The LM35 offers high accuracy with low self-heating and low impedance output. It's typically calibrated to provide accurate readings without requiring additional adjustments. Temperature Sensing Element: The heart of the LM35 is its semiconductor temperature sensor, which ensures reliable and precise temperature measurements over its operational range.

Package: The LM35 is available in different packages, including TO-92, metal can, and surface amount variants, providing flexibility in design and integration into different projects. Calibration: Some variants of the LM35 come pre-calibrated by the manufacturer, reducing the need for additional calibration steps in the end-user application.

6.1.4 Advantages

High Accuracy: The LM35 offers high accuracy and stability over a wide temperature range,making it suitable for precision temperature measurement applications.

Ease of Use: With its linear output voltage proportional to temperature, the LM35 is easy to interface with microcontrollers and other electronic components, requiring minimal additionalcircuitry for temperature measurement.

Low Power Consumption: The LM35 operates with low power consumption, making it suitable for battery-powered applications where energy efficiency is essential.

Robustness: LM35 sensors are robust and reliable, with low self-heating and low impedance output, ensuring accurate temperature measurements even in harsh operating environments.

Precision: They offer accurate measurements, crucial for applications like climate control, industrial processes, and medical devices.

Versatility: Temperature sensors come in various types, such as thermocouples, RTDs, thermistors, and infrared sensors, allowing flexibility in usage across different environments and temperature ranges.

Efficiency: They enable automated systems to maintain optimal conditions, reducing energy consumption and operational costs.

Reliability: Modern temperature sensors are robust and durable, ensuring consistent performance even in harsh conditions.

Safety: By monitoring temperature, sensors can help prevent overheating and mitigate risks offire or equipment failure.

Real-time monitoring: They provide real-time data, enabling timely interventions and adjustments to prevent damage or optimize processes.

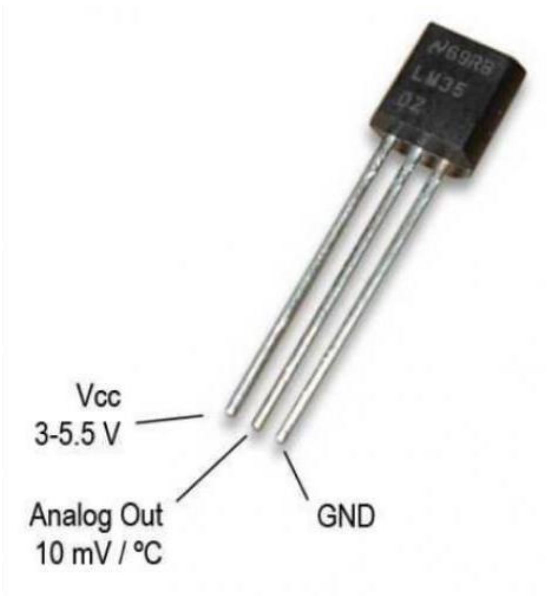


Figure 6.1: Temperature Sensor

6.2 FLEX SENSORS:

6.2.1 Introduction

The flex sensor is a versatile and responsive component that translates mechanical bending or flexing into electrical signals. With its slender, flexible construction, it can be easily integrated into wearable devices, robotics, and various human-machine interface applications. The sensor's operation relies on changes in resistance as it bends, with resistance decreasing as the sensor is flexed and increasing as it returns to its original position. This resistance variation is directly proportional to the degree of bending, allowing precise measurement of flexion angles. The flex sensor's simplicity, reliability, and accuracy make it invaluable in applications ranging from motion tracking in virtual reality systems to prosthetic limb control. This introductory overview provides insights into the working principle, applications, and advantages of the flex sensor, highlighting its pivotal role in enabling intuitive and responsiveinteractions between humans and technology.

6.2.2 Working

- Flex sensors operate based on the principle of resistive sensing.
- When the sensor is in a neutral or unbent position, its resistance remains relatively constant. However, when the sensor is bent or flexed, the conductive material within the sensor stretchesor compresses, causing its resistance to change
- This change in resistance is proportional to the degree of bending or flexing, allowing the sensor to provide a measurable output.

6.2.3 Specifications

Flexibility Range: Flex sensors typically have a defined range of flexibility, indicating the maximum angle of bending they can endure without damage. This range can vary depending on the specific model and construction.

Resistance Variation: Flex sensors exhibit a change in resistance proportional to the degree of bending. The sensitivity of this resistance variation is an important specificationto consider.

Range: Flex sensors have a baseline resistance value in their neutral, unbent state, and this resistance varies as the sensor is flexed. The resistance range describes the minimum and maximum resistance values achievable through bending.

Operating Voltage: Flex sensors operate within a specified voltage range, typically compatible with common power sources used in electronic circuits. It's important to ensure that the operating voltage matches the requirements of the application.

Temperature Range: The temperature range within which the flex sensor operates effectivelyis an essential specification, ensuring reliable performance across different environmental conditions.

Response Time: The response time of the flex sensor refers to how quickly it reacts to changes in bending. This specification is crucial for applications requiring real-time feedbackor rapid adjustments.

Durability: Flex sensors should be durable and resilient to repeated bending and flexing, ensuring longevity and reliability in demanding applications.

Size and Form Factor: The physical dimensions and form factor of the flex sensor are important considerations, especially for applications where space is limited or where the sensorneeds to conform to specific shapes or contours.

Compatibility: Flex sensors may come with different interface options, such as analog or digital outputs, to suit the requirements of different electronic systems and microcontrollers.

6.2.4 Advantages

Flexibility: Flex sensors are highly flexible and conformable, allowing them to bend and flex with minimal resistance, making them suitable for applications requiring precise motion tracking.

Customization: Flex sensors can be customized in terms of size, shape, and sensitivity tomeet the specific requirements of different applications and designs.

Easy Integration: Flex sensors are easy to integrate into electronic circuits and systems, requiring minimal additional components for signal conditioning.

Reliability: High-quality flex sensors offer reliable and consistent performance over time, even after multiple bending and flexing cycles, ensuring accurate and repeatable measurements.

Magnetic Field Detection: Flux sensors can accurately measure the strength and direction of magnetic fields. This capability is essential in various applications where magnetic fieldsneed to be monitored or controlled.

Navigation and Positioning: They are used in navigation systems such as compasses and inertial navigation systems to determine direction and orientation relative to the Earth's magnetic field. This makes them valuable in applications like GPS navigation in smartphones, autonomous vehicles, and drones.

Magnetic Object Detection: Flux sensors can detect the presence and movement of magnetic objects. This feature is useful in industrial applications for detecting metallic objects in proximity sensors, security systems, and metal detectors.

In summary, flex sensors are versatile and adaptable devices that find applications in awide range of fields, from wearable technology to robotics, providing precise and reliable measurement of bending and flexing movements. Their flexibility, ease of integration, and durability make them indispensable components in many innovative applications.

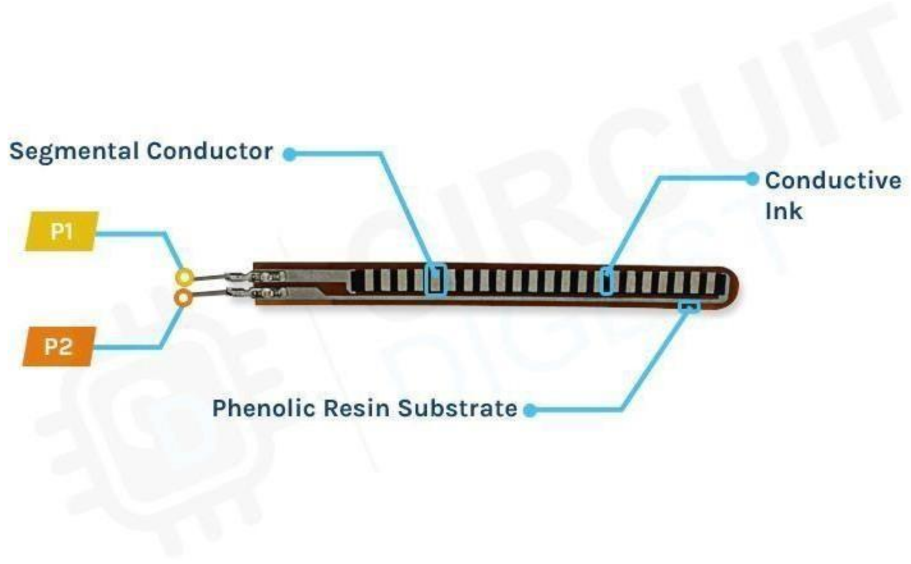


Figure 6.2: Flux Sensor

6.3 ACCELEROMETER SENSOR

6.3.1 Introduction

The MPU-6050 accelerometer sensor module, a compact and versatile MEMS device,combines a three-axis accelerometer and a three-axis gyroscope in a single chip. Developed by InvenSense (now TDK Corporation), this module offers precise motion sensing capabilities for a wide range of applications. Its integrated design, programmable features, and compatibility with common communication interfaces like I2C and SPI make it popular amonghobbyists, engineers, and manufacturers alike. The accelerometer measures linear acceleration, while the gyroscope detects angular velocity along three axes, providing comprehensive motion data. By combining information from both sensors, the MPU-6050 accurately tracks orientation changes, tilt angles, and motion dynamics in real-time. This sensor is commonly employed in electronic stability control systems, robotics, wearable devices, and motion-controlled gadgets, owing to its high accuracy, compact form factor, andease of integration.

6.3.2 Working

Accelerometer

- The accelerometer in the MPU-6050 measures acceleration along three orthogonal axes:X, Y, and Z.
- It operates based on the principle of capacitance sensing or piezoelectric effect.
- Inside the accelerometer, tiny microstructures deflect in response to acceleration, causing changes in capacitance or generating piezoelectric signals proportional to the applied acceleration.
- These changes are converted into digital signals by onboard analog-to-digital converters (ADCs) for further processing.

Gyroscope

- The gyroscope in the MPU-6050 measures angular velocity or rotation rate along the samethree axes: X, Y, and Z.
- It typically utilizes the Coriolis effect, where a vibrating mass experiences a force

perpendicular to its motion when subjected to angular velocity.

- As the MPU-6050 rotates, the vibrating mass inside the gyroscope experiences this force,generating a signal proportional to the angular velocity.

Data Fusion

- The MPU-6050 combines data from the accelerometer and gyroscope to provide accuratemotion tracking and orientation estimation.
- By employing sensor fusion algorithms, such as Kalman filtering or complementary filtering, the MPU-6050 compensates for the strengths and weaknesses of each sensor, providing robust and reliable motion sensing capabilities.

6.3.3 Specifications

Acceleration Range: The MPU-6050 accelerometer typically has a programmable full-scale range, commonly selectable between ±2g, ±4g, ±8g, or ±16g, depending on the application requirements.

Gyroscope Range: The gyroscope on the MPU-6050 usually has a programmable full-scale range as well, commonly selectable between ±250, ±500, ±1000, or ±2000 degrees per second(dps).

Data Output Rate (DOR): The MPU-6050 can provide sensor data at various output rates, typically ranging from several Hz to several hundred Hz, depending on the selected configuration.

Communication Interface: The MPU-6050 communicates with external microcontrollers or other devices via an I2C (Inter-Integrated Circuit) or SPI (Serial Peripheral Interface) interface, making it compatible with a wide range of embedded systems.

Integrated Features: In addition to the accelerometer and gyroscope, the MPU-6050 may include integrated temperature sensor, digital motion processor (DMP), and other features to enhance performance and functionality.

6.3.4 Advantages

Compact and Integrated: The MPU-6050 combines both accelerometer and gyroscope functions into a single chip, reducing the need for multiple sensors and simplifying system design.

High Accuracy: The MPU-6050 offers high accuracy and sensitivity, enabling precise measurement of motion and orientation in various conditions and environments.

Versatility: With its programmable full-scale ranges and output rates, the MPU-6050 can be configured to suit a wide range of applications, from low-power wearable devices to highperformance motion control systems.

Ease of Integration: The MPU-6050 communicates over standard I2C or SPI interfaces, making it easy to integrate into existing electronic circuits and microcontroller-based projects.

Motion Sensing: Accelerometers can detect motion, acceleration, and tilt in various directions. This capability is crucial in applications such as mobile devices, gaming consoles, and wearable fitness trackers.

Orientation Detection: They can determine the orientation of a device relative to the Earth'sgravity, enabling features like auto-rotation in smartphones and tablets.

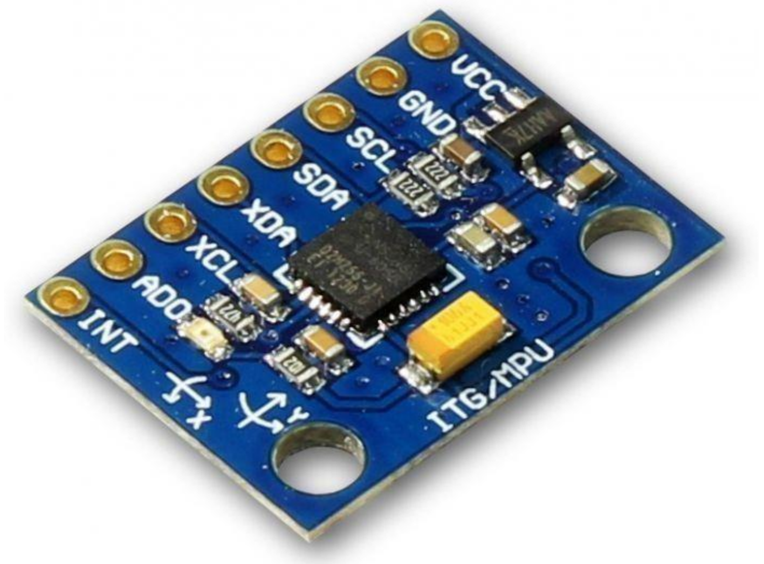


Figure 6.3: Mems Sensor

6.4 HEARTBEAT SENSOR

6.4.1 Introduction

The heartbeat sensor, a pivotal component in modern health and fitness technology, offers a noninvasive means of measuring heart rate and pulse rate. Utilizing advanced techniques like photoplethysmography (PPG), it detects subtle changes in blood volume duringeach heartbeat, translating them into electrical signals for analysis.

With its compact design and real-time monitoring capabilities, the sensor finds widespread application in fitness trackers, smartwatches, and medical devices, empowering users to track their cardiovascular health and fitness levels seamlessly. Beyond consumer electronics, heartbeat sensors play a vital role in healthcare, biometric authentication, and stress management, contributing to a holistic approach to wellness. This introductory overview delves into the working principle, specifications, applications, and advantages of this transformative technology.

6.4.2 Working

- The heartbeat sensor operates on the principle of photoplethysmography (PPG), whichrelies on the interaction between light and biological tissues.
- It typically consists of an emitter, which shines light onto the skin, and a photodetector,which measures the amount of light transmitted or reflected back.
- When light is emitted onto the skin, it penetrates the tissue and is partially absorbed byblood vessels.
- During each heartbeat, blood flow in the vessels changes, causing variations in the amountof light absorbed.
- These fluctuations in light intensity are captured by the photodetector and converted intoelectrical signals.
- Signal conditioning and processing techniques are then employed to isolate the pulsatile component corresponding to the heart rate from the ambient light and other noise sources.
- By analyzing the waveform generated by these electrical signals, the heartbeat sensor canaccurately determine the heart rate or pulse rate of the individual.

6.4.3 Specifications

Detection Method: Heartbeat sensors can use various methods to detect heart rate, includingoptical, electrical, or acoustic methods. Optical sensors are commonly used in wearable devices and employ LEDs and photodiodes to measure the changes in blood volume under theskin caused by the heartbeat.

Accuracy: The accuracy of the sensor is a crucial factor, especially in medical and fitness applications.Higher accuracy ensures reliable heart rate measurements.

Sampling Rate: This refers to how frequently the sensor samples heart rate data. A higher sampling rate allows for more precise monitoring of heart rate variations. Common sampling rates range from a few Hz to several hundred Hz.

Power Consumption: Heart rate sensors should have low power consumption, particularly forwearable devices, to prolong battery life. Efficient power management is essential for continuous monitoring without frequent recharging.

Compatibility: Heart rate sensors may come with different interfaces for connecting to various devices, such as Bluetooth, ANT+, or proprietary protocols. Compatibility with a wide range of devices and platforms enhances their usability.

Size and Form Factor: For wearable devices, the size and form factor of the sensor are critical. Smaller sensors with a low profile are preferable for integration into wearable gadgets like smartwatches, fitness trackers, or chest straps.

Water Resistance: Depending on the intended application, the sensor may need to be water- resistant or even waterproof. This feature is particularly important for fitness trackers used during swimming or other water-related activities.

Signal Processing: Advanced signal processing algorithms may be incorporated into the sensor to filter out noise and motion artifacts, improving the accuracy of heart rate measurements, especially during physical activity.

Data Output: Heart rate sensors typically provide heart rate data in digital format, which can be directly processed by microcontrollers or transmitted wirelessly to other devices. The outputformat and communication protocol should be compatible with the intended application.

6.4.4 Advantages

Non-Invasive: Heartbeat sensors are non-invasive and can measure heart rate without the need for physical contact with the skin, making them comfortable and convenient to use.

Real-Time Monitoring: Heartbeat sensors provide real-time feedback on heart rate, allowing individuals to monitor their cardiovascular health and adjust their activities accordingly.

Compact and Portable: Many heartbeat sensors are compact and portable, making them suitable for wearable devices and mobile applications for on-the-go monitoring.

Cost-Effective: Heartbeat sensors are often cost-effective solutions for heart rate monitoring compared to traditional medical devices, making them accessible to a wide range of users.

In summary, heartbeat sensors are valuable tools for monitoring heart rate and assessing cardiovascular health in various applications, from fitness tracking to medical diagnostics. Their noninvasive nature, real-time monitoring capabilities, and compact design make them indispensable in today's digital health and wellness landscape.

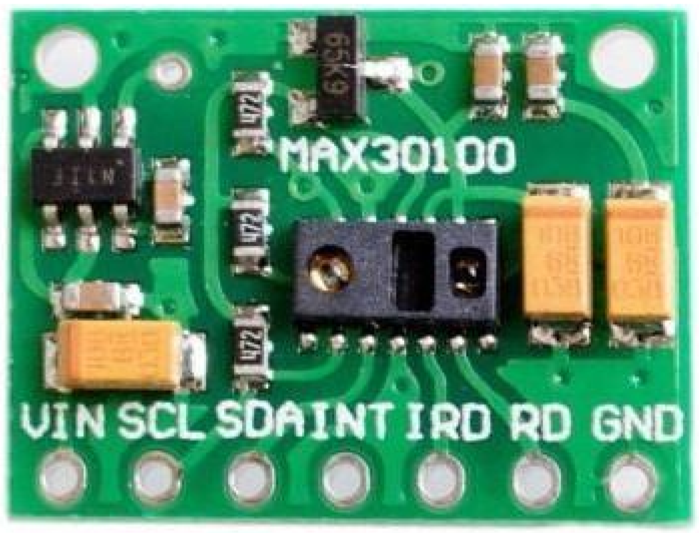


Figure 6.4: Heartbeat Sensor

CHAPTER – 7
WORKING METHODOLOGY

7.1 Sensor Data Acquisition

The Arduino UNO serves as the central hub for data acquisition, continuously gathering information from a diverse array of sensors strategically placed to monitor various aspects of the patient's health and well-being. Alongside the LM35 temperature sensor, heartbeat sensor, MPU 6050 accelerometer sensor, and flex sensors, additional sensors may beintegrated based on specific healthcare needs.

7.2 Data Processing

Upon receiving sensor data, the Arduino UNO employs sophisticated algorithms tailored to each sensor's output characteristics. For instance, temperature data undergoes meticulous analysis to detect subtle deviations from the norm, potentially indicating early signs of illness or discomfort. Similarly, heartbeat data is scrutinized for irregularities, with algorithms capable of identifying arrhythmias or anomalies requiring immediate attention.

7.3 Alert Generation

The system is equipped with robust alert mechanisms designed to swiftly notify caregivers or healthcare providers of critical developments. When abnormal temperature readings or irregular heartbeats are detected, the system triggers SMS and email alerts, promptly notifying designated individuals for timely intervention. When abnormal temperature readings or irregular heartbeats are detected, the system triggers SMS and email alerts, promptly notifying designated individuals for timely intervention

7.4 Communication

Seamless communication is facilitated through the integration of the GSM SIM800L module, enabling the system to transmit SMS alerts directly to caregivers' mobile devices.

The inclusion of a GPS module further enhances response capabilities, enabling precise location tracking for swiftintervention in emergencies.

7.5 User Interface

A user-friendly interface, comprising an LCD display and intuitive controls, provides caregivers with real-time access to vital patient data and system status updates. The display prominently features key metrics such as temperature, heartbeat, and alerts, empowering caregivers to make informed decisions and take prompt action when necessary. Voice alerts, facilitated by the ISD 1805 IC, offer an additional layer of communication, providing auditoryfeedback or instructions as needed.

7.6 Power Management

Efficient power management is paramount to ensuring uninterrupted operation and prolonged battery life, particularly in scenarios requiring continuous monitoring over extendedperiods. A dedicated power supply module regulates power delivery to all system components, optimizing energy consumption and minimizing wastage. Advanced power-saving techniques further enhance efficiency, prolonging battery life and reducing maintenance requirements. By integrating these comprehensive features and functionalities, the system delivers unparalleled monitoring capabilities, empowering caregivers to provide proactive and responsive care tailored to each patient's unique needs.

APPENDIX CODE

Code For Arduino Uno 1

```
#include <SoftwareSerial.h>
#include <Wire.h>
#include "MAX30100_PulseOximeter.h"
#include "MAX30100.h"
MAX30100 sensor;
#define REPORTING_PERIOD_MS     1000
PulseOximeter pox;
uint32_t tsLastReport = 0;
const byte LED=13;
const byte rxPin = 10;
const byte txPin = 11;
const int LM35P = A0;
int temp_val1=0;
int temp_adc_val=0;
int temp_val=0;
int temp=0;
SoftwareSerial mySerial (rxPin, txPin);
String food=" ";
String water=" ";
String medicine=" ";
String washroom=" ";
String PULSE=" ";
String OXYGEN=" ";
String TEMPARATURE=" ";
String falled=" ";
bool newValue = false;
void onBeatDetected()
{
//Serial.println("Beat!");
```

```
}

void setup()
{
   Serial.begin(9600);
   mySerial.begin(9600);
   pinMode(LED,OUTPUT);
   digitalWrite(LED,LOW);
   if (!pox.begin())
   {

      for(;;);
   }
   else
   {

      digitalWrite(LED,HIGH);
   }
   pox.setOnBeatDetectedCallback(onBeatDetected);


}
int hr=0;
void loop()
{
  digitalWrite(LED,LOW);
   int temp_adc_val;
   float temp_val;
   float temp_val1;
   int temp;
   temp_val1=0;
   for(int i=0;i<=300;i++)
   {
```

```
temp_adc_val = analogRead(A0);  /* Read Temperature */
temp_val1=temp_val1+temp_adc_val;
}
temp_val1=temp_val1/300;
temp_val = (temp_val1 * (5.0 / 1024.0));  /* Convert adc value to equivalent voltage */
temp = (temp_val*10)+2;   /* LM35 gives output of 10mv/°C */
TEMPARATURE=String(temp);
pox.update();
if (millis() - tsLastReport > REPORTING_PERIOD_MS)
{
 digitalWrite(LED,HIGH);
 hr=0;int rhr=75;int diff=0;int sp0=0;int rsp=98;int diffsp=0;
 hr=pox.getHeartRate()-10;
 sp0=pox.getSpO2();
 if(hr>=0)
 {
  if(hr<=50)
  {
    diff=rhr-hr;
    hr=hr+diff;
  }
  if(hr>=120)
  {
    //diff=rhr-hr;
    hr=89;
  }
  if(sp0<90)
  {
   sp0=95;
  }

  PULSE=String(hr);
  OXYGEN=String(sp0);
```

```
        tsLastReport = millis();

        mySerial.println(PULSE+","+OXYGEN+","+TEMPARATURE+",");

      }

    }
    Serial.println(PULSE+","+OXYGEN+","+TEMPARATURE+",");
}
```

Code For Arduino Uno 2

```
#include <SoftwareSerial.h>
#include <Wire.h>
#include "MAX30100_PulseOximeter.h"
#include "MAX30100.h"

#define REPORTING_PERIOD_MS     1000
PulseOximeter pox;
MAX30100 sensor;
uint32_t tsLastReport = 0;

const byte rxPin = 10;
const byte txPin = 11;
const int LM35P = A3;
int temp_val1=0;
int temp_adc_val=0;
int temp_val=0;
int temp=0;
SoftwareSerial mySerial (rxPin, txPin);
String food=" ";
String water=" ";
String medicine=" ";
String washroom=" ";
String PULSE=" ";
String OXYGEN=" ";
bool newValue = false;



void onBeatDetected()
{
Serial.println("Beat!");
}



void setup()
{
 Serial.begin(9600);
 mySerial.begin(9600);
 if (!pox.begin())
 {
        Serial.println("FAILED");
        for(;;);
 }
 else
 {
        Serial.println("SUCCESS");
 }
 // Configure sensor to use 7.6mA for LED drive
        pox.setOnBeatDetectedCallback(onBeatDetected);
}

void loop()
{
temp_val1=0;
temp_adc_val=0;
temp_val=0;
```

```
temp=0;
   for(int i=0;i<=1000;i++)
   {
   temp_adc_val = analogRead(LM35P);      /* Read Temperature */
   temp_val1=temp_val1+temp_adc_val;
   }
   temp_val1=temp_val1/1000;
   temp_val = (temp_val1 * (5.0 / 1024.0));  /* Convert adc value to equivalent voltage */
   temp = (temp_val*10)+2;   /* LM35 gives output of 10mv/°C */

   if(analogRead(LM35P)>=500)
   {
   food="YES";
   }
   else
   {
   food="NO";
   }
   if(analogRead(LM35P)>=500)
   {
   water="YES";
   }
   else
   {
   water="NO";
   }
   if(analogRead(LM35P)>=500)
   {
   medicine="YES";
   }
   else
   {
    medicine="NO";
```

```
  }
  pox.update();

  // Asynchronously dump heart rate and oxidation levels to the serial
  // For both, a value of 0 means "invalid"
  if (millis() - tsLastReport > REPORTING_PERIOD_MS) {
    Serial.print("Heart rate:");
    Serial.print(pox.getHeartRate());
    Serial.print("bpm / SpO2:");
    Serial.print(pox.getSpO2());
    Serial.println("%");

    tsLastReport = millis();
  }


  PULSE=String(pox.getHeartRate());
  OXYGEN=String(pox.getSpO2());




  Serial.print("Temperature = ");
  Serial.print(temp);
  Serial.print(" Degree Celsius\n");
  delay(1000);
  Serial.println(temp_val1);
  Serial.println(temp_val);
  Serial.println(temp);
}
```

Code For Node MCU or ESP

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClientSecureBearSSL.h>
#include <Wire.h>

const char* ssid     = "IOTBPMS";
const char* password = "12345678";



const char* serverName = "https://tech-guide.online/iot-1-iotbpms.php";

String apiKeyValue = "IOTBPMS";

String sensorName = "BME280";
String sensorLocation = "Office";
int NETWLED=D0;
int BULB=D4;
int BULB1=D2;
int FAN=D5;
int AC=D6;
int PUMP=D7;

#define SEALEVELPRESSURE_HPA (1013.25)
int i=0;
String  datastr = "20,OFF,32,OFF,80,ON,WET,OFF";

void dataposting(void)
{
digitalWrite(NETWLED,LOW);
delay(500);
digitalWrite(NETWLED,HIGH);
```

```
delay(50);
digitalWrite(NETWLED,LOW);
}
void setup()
{
 Serial.begin(9600);
 pinMode(NETWLED,OUTPUT);
 digitalWrite(NETWLED,HIGH);
 WiFi.begin(ssid, password);
 Serial.println("IOT BASED PATIENT MONITORING SYSTEM");
 Serial.println("Connecting");
 while(WiFi.status() != WL_CONNECTED) {
   delay(500);
   Serial.print(".");
 }
 Serial.println("");
 Serial.print("Connected to WiFi network with IP Address: ");
 Serial.println(WiFi.localIP());
 digitalWrite(NETWLED,LOW);
}

void loop()
{
  yield();

          if(Serial.available()>0)
          {
           yield();
              String Readdata=Serial.readString();
              String datastr[8];
              int StringCount = 0;
              Serial.println(Readdata);
                while (Readdata.length() > 0)
```

```
{
  int index = Readdata.indexOf(',');
  if (index == -1) // No space found
    {
      datastr[StringCount++] = Readdata;
      break;
    }
  else
    {
    datastr[StringCount++] = Readdata.substring(0, index);
    Readdata = Readdata.substring(index+1);
    }
}
for (int i = 0; i < StringCount; i++)
{
  Serial.print(i);
  Serial.print(": \"");
  Serial.print(datastr[i]);
  Serial.println("\"");
}


if(WiFi.status()== WL_CONNECTED)
{
      std::unique_ptr<BearSSL::WiFiClientSecure>client(new
BearSSL::WiFiClientSecure);
      client->setInsecure();
      HTTPClient https;
      https.begin(*client, serverName);
      https.addHeader("Content-Type", "application/x-www-form-
urlencoded");
      String httpRequestData = "api_key=" + apiKeyValue + "&sensor="
+ sensorName
```

```
                              + "&location=" + sensorLocation + "&value1=" +
datastr[0]
                              + "&value2=" +datastr[1] + "&value3=" + datastr[2]
+ "&value4=" + datastr[3] +  "&value5=" + datastr[4] +  "&value6=" + datastr[5] +
"&value7=" + datastr[6] + "&value8=" + datastr[7] + "&value9=" + String(i) +
"&value10=" + String(i) +"";
                  Serial.print("httpRequestData: ");
                  Serial.println(httpRequestData);
                  i=i+1;
                  int httpResponseCode = https.POST(httpRequestData);
                           if (httpResponseCode>0)
                           {
                            Serial.print("HTTP Response code: ");
                            Serial.println(httpResponseCode);
                            dataposting();
                           }
                           else
                           {
                            Serial.print("Error code: ");
                            Serial.println(httpResponseCode);

                           }
                           https.end();
                }
                else
                {
                 Serial.println("WiFi Disconnected");
                 digitalWrite(NETWLED,HIGH);
                }
           }

}
```

Code For Playback Module

```
import board
import audiomp3
import audiopwmio
import time
import analogio
import audiocore
import audiomixer
audio = audiopwmio.PWMAudioOut(board.GP0)

mp3files = ["FOOD.mp3", "MEDICINE.mp3","WATER.mp3","WASHROOM.mp3"]
FOOD = analogio.AnalogIn(board.A1)
WATER = analogio.AnalogIn(board.A2)
WASHROOM = analogio.AnalogIn(board.A3)

mp3 = open(mp3files[0], "rb")
decoder = audiomp3.MP3Decoder(mp3)

while True:


  print(FOOD.value)
  print(WATER.value)
  print(WASHROOM.value)
for filename in mp3files:
  decoder.file = open(filename, "rb")
  audio.play(decoder)
  print("Playing:", filename)
```

```
    while audio.playing:
        pass

print("Done playing!")
```

CHAPTER-8

RESULT

An IoT-based automated paralysis patient healthcare monitoring system using the specified components. This system will help paralyzed patients convey their needs and monitor their health. Here's how we can integrate the components:

Arduino Uno and NodeMCU ESP8266:

Use the Arduino Uno as the main microcontroller and the NodeMCU ESP8266 for wireless communication.The NodeMCU will handle data transmission to the cloud or a local server.

8.1 Sensors

MEMS Sensor (Accelerometer and Gyroscope):

The MPU6050 sensor combines both accelerometer and gyroscope functionalities. It detects motion and orientation changes.

LM35 Temperature Sensor: Measures the patient's body temperature.

MAX30100 Pulse Oximeter Sensor: Monitors heart rate and oxygen saturation levels.

Flux Sensors:

These could be various environmental sensors (e.g., light, humidity, gas) to provide additional context.

Components for Interaction:

LCD Display: We'll use an LCD screen to display messages or vital signs.

Potentiometer: For adjusting display contrast or brightness.

Loudspeaker/Buzzer: To provide audio alerts or notifications.

System Functionality:

Motion-Based Message Conveyance:

When the patient moves any part of their body (e.g., tilting their head), the MPU6050 detects the motion.The Arduino processes this motion and displays a predefined message on the LCD screen.

Health Monitoring:

The LM35 measures body temperature, and the MAX30100 monitors heart rate and oxygen saturation.Data from these sensors is transmitted to the NodeMCU.The NodeMCU can send this data to a cloud server or a caregiver's device.

Emergency Alerts:

If the patient needs urgent assistance (e.g., heart rate irregularities), the system triggers an alert.The loudspeaker/buzzer can sound an alarm, and an alert will be sent to website.

Communication:

The NodeMCU communicates with the Arduino Uno via serial communication (UART).

It also connects to the internet via Wi-Fi (using the ESP8266 module).

Power Supply:

Ensure a stable power supply for all components. Consider using rechargeable batteries or an external power source. Remember that this is a high-level overview, and detailed implementation would require further exploration and development. You may need to write code to handle sensor data, communication protocols, and alert mechanisms.

8.2 IMAGE OF THE PROTOTYPE

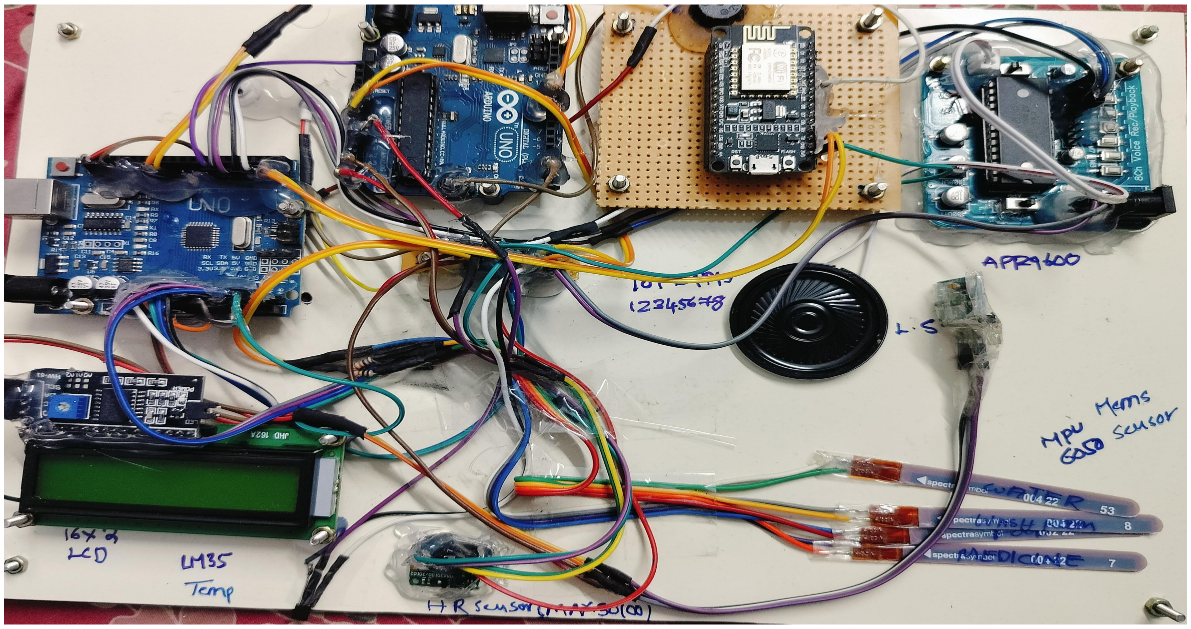
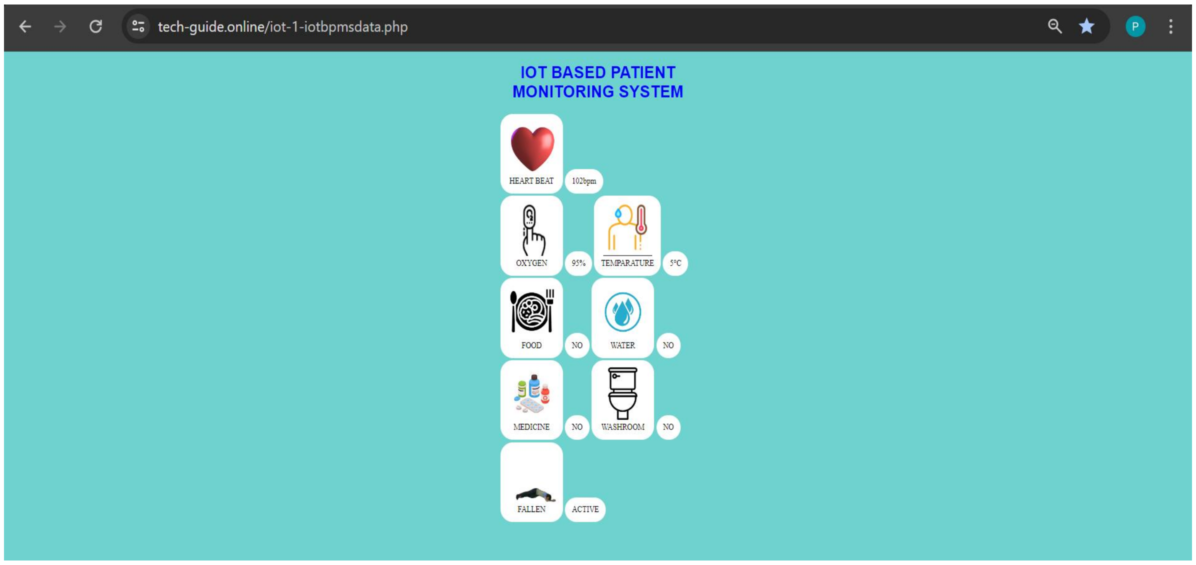


Figure 8.2:Image of the prototype

8.3 IOT Based Patient Monitoring System

Website: https://tech-guide.online/iot-1-iotbpmsdata.php



IOT BASED PATIENT MONITORING SYSTEM

HEART BEAT 102bpm

OXYGEN 95%

TEMPARATURE 5°C

FOOD NO

WATER NO

MEDICINE NO

WASHROOM NO

FALLEN ACTIVE

Figure 8.3: IOT Based Patient Monitoring System

CONCLUSION

The IOT-based automated paralysis patient healthcare monitoring system represents a significant advancement in medical technology, particularly in the management and care of individuals with paralysis.By leveraging a sophisticated array of sensors and modern communication technologies, this system provides continuous, real-time monitoring of vital parameters such as body temperature, heart rate, and physical movement, ensuring comprehensive oversight of the patient’s well-being.

The seamless integration of the Arduino UNO with sensors like the LM35 temperature sensor, heartbeat sensor, MPU 6050 accelerometer, and flex sensors allows for precise data acquisition. This data is meticulously processed to detect any abnormalities that could indicatehealth risks or emergencies.

Such capabilities are crucial for paralysis patients, who may not be able to communicate their needs or discomforts effectively. Automated alerts via SMS and email ensure that caregivers and healthcare providers are informed promptly about any concerning changes, allowing for quick responses that can be literally lifesaving. Moreover, the communication suite equipped with the GSM SIM800L module and GPS tracking further enhances the system's effectiveness by facilitating swift communication and precise patient location tracking.

This is particularly important in emergency scenarios where immediate intervention is required. The system's user interface, highlighted by an LCD display and voice alert capabilities, offers an intuitive platform for caregivers to monitor and interact with the system.This not only simplifies the user experience but also enhances the engagement and efficiencyof the caregiving process.

In conclusion, this IOT-based system is more than just a technical solution; it embodies a comprehensive approach to patient care that integrates technology, healthcare, and human oversight. By addressing the specific needs of paralysis patients and providing tools to manage their health proactively, the system significantly contributes to improving their quality of life.

FUTURE SCOPE

The future scope of IOT-based healthcare monitoring systems for paralysis patients is both vast and promising, as advancements in technology continue to enhance the capabilities and effectiveness of these systems. As the field of IOT expands, we can anticipate more sophisticated sensors and devices being integrated into the monitoring systems, offering moredetailed and comprehensive data collection. This could include sensors that monitor neurological activity or muscle responses, providing deeper insights into the patient's condition and potentially facilitating breakthroughs in treatment methods. Artificial intelligence (AI) and machine learning (ML) are set to play a crucial role in the evolution of these systems. By implementing AI algorithms, the system can learn from vast amounts of data to predict potential health issues before they become critical, thereby offering preventive care solutions that are tailored to individual patients. Additionally, the integration of augmented reality (AR) and virtual reality (VR) technologies could revolutionize patient interaction and rehabilitation processes. These technologies can be used to create simulation-based therapies and exercises that are engaging and tailored to the specific rehabilitation needs of paralysis patients. This approach not only enhances the recovery process but also adds a layer of interaction that can improve the mental and emotional well-being of patients. Enhanced connectivity will also facilitate remote monitoring and telehealth services, making healthcare more accessible, especially for patients living in rural or underserved areas. In the long term, the adoption of such technologies in healthcare could lead to a shift towards more personalized and precision- based medical approaches. This could significantly alter how healthcare is delivered, moving away from a one-size-fits-all model to one that is adaptable to the unique conditions and needsof each patient.

As a result, paralysis patients could expect not only improved life expectancy but also enhanced quality of life, with greater independence and participation in daily activities.Overall, the future of IOT-based healthcare monitoring systems is intrinsically linked to technological advancements and their integration into healthcare practice.

REFERENCES

[1] Abhijeet Botre et al., “Assistance system for paralyzed”, International Journal Of Innovative Research In Electrical, Electronics, Instrumentation And Control Engineering ,Vol 4, Issue5, 2016.

[2] Alexis Bell, Paul Rogers, Chris Farnell, Brett Sparkman, ScottC Smith, "Wireless Patient Monitoring System”, HIC 2014 IEEE .

[3] Alyson Matheus de Carvalho Souza et al., “Hand copter Game”, Virtual and Augmented Reality (SVR), 2012.

[4] Gomez-Vilda et al., “Vocal-fold paralysis patients treated with stemcellgrafting”, International Conference on Pattern Recognition Systems (ICPRS-16), 2016.

[5] Hemakshi Pawar et al., “Assistive Interactive Device using ElectroOculography”, International Journal of Advanced Research in Computer Engineering and Technology ,Vol4, Issue 1, 2015.

[6] Herta Flor et al., “The Thought Translation Device (TTD) for Completely Paralyzed Patients”, IEEE Trans actions on Rehabilitation Engineering, Vol 8, No 2, 2000.

[7] Massage Coll et al., “Treatment of Peripheral Facial Paralysis (PFP)” , Photonics and Optoelectronics (SOPO), 2012. [8] Rolga Roy, “Methodologies to assist paralysed patients”, International Journal Of Advanced Research In Electrical, Electronics And Instrumentation Engineering, Vol 5, Issue 3, 2016.