VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI-590018, KARNATAKA

SYNOPSIS REPORT ON

"MEDWATCH SYSTEM"

SHREE DEVI INSTITUTE OF TECHNOLOGY

Department of Electronics and Communication Engineering

Affiliated to VTU, Belagavi & approved by AICTE, New Delhi

Airport Road, Kenjar, Mangaluru-574142

Guided by:

Vishwitha Ma'am

Submitted by:

Amrutha K M – 4SH21EC001

Gangotri H S – 4SH21EC003

Suchetha K Naik – 4SH21EC011

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any work would be incomplete without thanking the persons who made it perfect with their constant guidance and encouragement.

We take this opportunity to express my sincere thanks and indebtedness to our Project guide and mentor, **Prof. Vishwitha, Assistant Professor, Department of E&C**, for her support and guidance. Her vision and suggestions throughout the project period have been fundamental in the completion of the project.

We extend my warm gratitude to **Prof. Chitra Prabhu, Head of the Department, Department of E&C**, for her constant support and advice that helped us to complete this project successfully.

We are extremely grateful to our beloved **Principal, Dr. K E Prakash** for his encouragement to come up with new ideas and advice to express them in a systematic manner.

We also like to thank all Teaching & Non-teaching staff of Shree Devi Institute of Technology, Mangalore for their kind co-operation during the course of the work. Finally, we are extremely thankful to our family and friends who helped us in our work and made the project a successful one.

Amrutha K M

Gangotri H S

Suchetha K Naik

Content:

ABSTRACT

The MedWatch System is an integrated health monitoring solution designed to continuously track and display vital signs such as oxygen saturation and heart rate. Utilizing a combination of an Arduino Uno microcontroller, MAX30100 oximeter sensor, LCD display, and Generic ESP8266 Wi-Fi module, the system offers both real-time monitoring and remote data access.

The core of the MedWatch System is the Arduino Uno, which processes data from the MAX30100 oximeter. This sensor measures critical health parameters and sends the data to the Arduino via I2C communication. The Arduino then processes this information and outputs it to the LCD display for immediate visualization by the user.

For remote monitoring and data logging, the Generic ESP8266 is employed. It connects to the Arduino to receive the processed health data and transmits it over Wi-Fi to a remote server or cloud platform. This enables healthcare providers or users to access the data from anywhere, facilitating timely interventions and continuous health monitoring.

The MedWatch System aims to enhance personal health management by providing an easy-to-use, real-time monitoring tool that integrates local display and remote data access, thus improving health oversight and potentially aiding in early detection of anomalies.

CHAPTER 1

INTRODUCTION

An IoT-based patient health monitoring system represents a significant advancement in healthcare technology by seamlessly integrating internet connectivity with a range of medical devices. This system allows for the continuous collection, transmission, and analysis of diverse health metrics such as heartbeat, body temperature, blood pressure, ECG readings, and physical activity levels. These metrics are captured using devices that vary widely in complexity and cost—from consumer-grade smartwatches and fitness trackers to advanced hospital-grade equipment. Each type of device provides different levels of functionality and precision, catering to a broad spectrum of healthcare needs.

The true power of an IoT-based health monitoring system lies in its ability to bridge distances between patients and healthcare providers. For instance, a patient managing a chronic condition or recovering from surgery can remain in the comfort of their home while their health is meticulously monitored remotely. This capability is particularly valuable for patients who require regular health tracking but have limitations in mobility or access to healthcare facilities. The system's real-time data transmission allows healthcare professionals to observe patient conditions as they evolve, leading to timely interventions if necessary. This is especially critical for detecting abrupt changes in health that could indicate potential emergencies, such as sudden fluctuations in blood pressure or irregular heart rhythms.

Furthermore, the continuous flow of data supports ongoing adjustments to treatment plans based on the patient’s evolving health status. This dynamic approach enables healthcare providers to fine-tune therapies and medications, leading to more personalized and effective care. For example, adjustments to medication dosages or lifestyle recommendations can be made promptly in response to real-time data, enhancing overall treatment outcomes and patient satisfaction.

The integration of IoT technology also enhances the analytical capabilities of health monitoring systems. By aggregating and analyzing large volumes of health data, these systems can uncover subtle trends and patterns that might not be visible through traditional methods. This in-depth analysis can provide valuable insights into the progression of chronic conditions, the effectiveness of treatments, and even predictive indicators of potential health issues. Such insights contribute to more accurate diagnoses and the development of tailored treatment strategies that address the specific needs of each patient.

CHAPTER 2

2.1 System overview

The "Medwatch System" project aims to deliver a comprehensive solution for real-time health monitoring by integrating several key components. At its core, the system utilizes the MAX30100 sensor to measure crucial physiological parameters such as SpO2 (blood oxygen saturation) and heart rate. This sensor is renowned for its accuracy and efficiency in capturing these vital statistics, making it an ideal choice for health monitoring applications.

The data collected by the MAX30100 sensor is then processed by an Arduino Uno. The Arduino Uno, a versatile microcontroller, handles the task of interpreting the raw data from the sensor and converting it into meaningful health metrics. This processed data is displayed on an LCD screen, providing immediate, on-site feedback for users or healthcare providers. The visual display allows for easy monitoring and quick assessment of a patient's health status without the need for external devices.

To enhance the system's capabilities, the ESP8266 module is employed to facilitate wireless connectivity. This Wi-Fi-enabled module allows the Medwatch System to send health data to a remote server or cloud-based platform. The ability to transmit data wirelessly expands the system's functionality beyond local monitoring, enabling healthcare providers to access real-time health information from anywhere. This feature is particularly valuable for remote patient monitoring, allowing healthcare professionals to track patient health metrics continuously and intervene promptly if necessary.

In addition to transmitting data to a remote server, the ESP8266 can be configured to send notifications or alerts to healthcare providers in the event of critical health changes. For instance, if the system detects a significant drop in SpO2 levels or an abnormal heart rate, it can automatically trigger an alert, ensuring that medical personnel are promptly informed and can take appropriate action.

The integration of these components—MAX30100 sensor, Arduino Uno, and ESP8266—creates a robust and versatile health monitoring system. The Medwatch System not only provides real-time monitoring and display of health parameters but also leverages wireless technology to facilitate remote oversight and timely intervention. This holistic approach enhances patient care by combining accurate physiological measurement with advanced data transmission capabilities, making it a valuable tool for both personal health management.

2.2 Components Required

Components Required	**Quantity**
Arduino UNO	**1**
Generic ESP8266	**1**
LCD Display	**1**
Bread Board	**1**
Oximeter max30100	**1**
Piezo Buzzer	**1**
Jumper wires	**–**
Potentiometer	**1**
Battery	**1**
Power Supply	**-**

2.2 Description of Components

ESP8266:

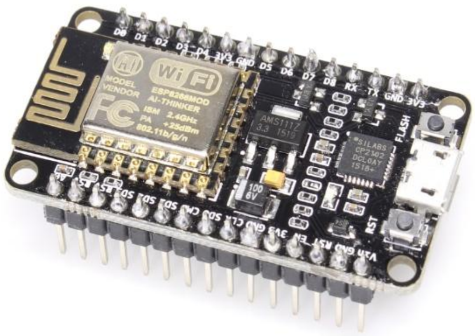


Fig: 2.3.1

The ESP8266 is a microcontroller with built-in Wi-Fi capabilities. In your project, it can be used to connect the Medwatch system to the internet, allowing you to send data (such as oximeter readings and other health parameters) to a remote server or database. It can also be used for local data processing and interfacing with other components.

Oximeter MAX30100:

Fig:2.3.2

The MAX30100 is a pulse oximeter and heart-rate sensor module. It uses infrared light to measure oxygen saturation (SpO2) and heart rate. In your Medwatch project, this sensor provides vital health data which can be displayed on the LCD and transmitted to the ESP8266 for further processing or wireless transmission to a monitoring system.

Arduino uno:

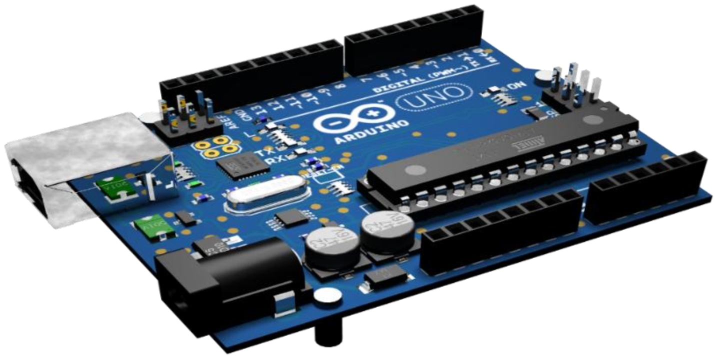


Fig:2.3.3

The Arduino Uno is another microcontroller board. It is commonly used for its ease of use and large community support. In this project, the Arduino Uno can interface with the sensors (like the max30100 oximeter sensor) and handle data processing tasks that the ESP8266 may not directly perform. It can also control the LCD display and manage communication between the sensors and the ESP8266.

LCD I2C:

Fig 2.3.4

An LCD (Liquid Crystal Display) module with I2C (Inter-Integrated Circuit) interface simplifies connections and reduces the number of pins needed to connect to the Arduino Uno. It can be used to display real-time data from the oximeter sensor, such as oxygen saturation levels (SpO2), heart rate, or any warnings/alerts related to the Medwatch system.

Breadboard:

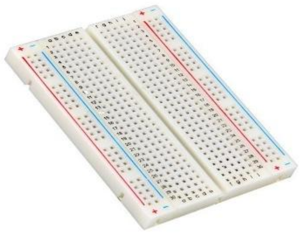


Fig 2.3.5

A breadboard is a fundamental tool for prototyping electronic circuits without soldering. It allows you to quickly connect and rearrange components (such as the ESP8266, Arduino Uno, LCD, and sensors) using jumper wires to test and iterate on your project design.

Battery:

Fig 2.3.6

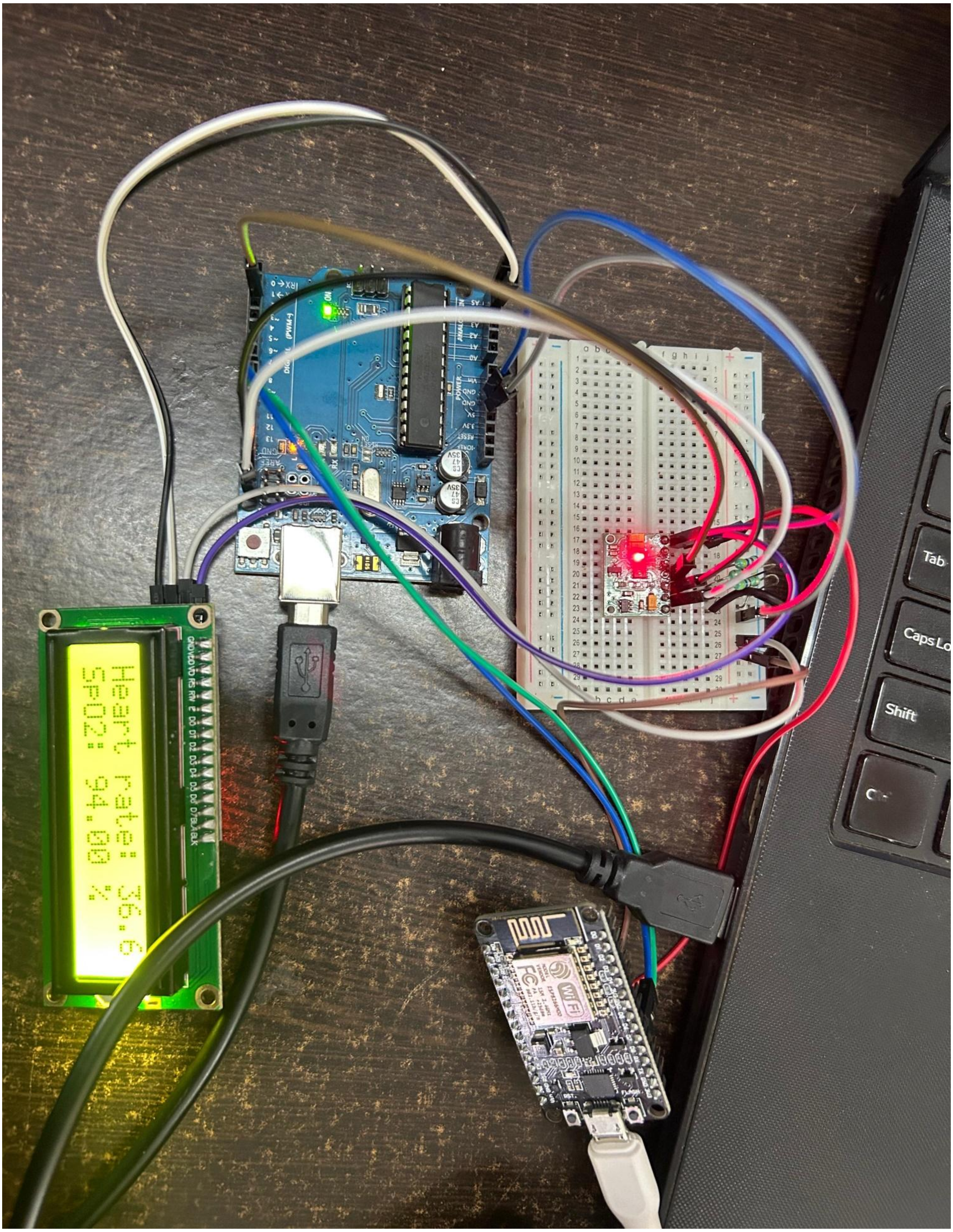
A 9-volt battery, made of six 1.5-volt cells in series, provides higher voltage for devices like smoke detectors, remote controls, and guitar pedals. It's reliable with a long shelf life but may not fit smaller electronics.

Jumper wire:

Fig 2.3.7

Jumper wires are used to make electrical connections between components on the breadboard. They come in various lengths and can be male-to-male, male-to-female, or female-to-female, depending on the type of connection needed between components.

2.4 Circuit Diagram



2.5 Methodology

1. Setup the Hardware:

1.1. Connect the Oximeter to Arduino:

Oximeter Sensor Connections: Most oximeters have four pins (VCC, GND, SDA, SCL). Connect these as follows:

- *VCC* to Arduino 5V.
- *GND* to Arduino GND.
- *SDA* to Arduino A4 (for I2C communication).
- *SCL* to Arduino A5 (for I2C communication).

1.2. Connect the LCD Display to Arduino:

For a 16x2 LCD with an I2C interface:

- *VCC* to Arduino 5V.
- *GND* to Arduino GND.
- *SDA* to Arduino A4.
- *SCL* to Arduino A5.

1.3. Connect the ESP8266 to Arduino:

- *TX* of ESP8266 to *RX* of Arduino (pin 0).
- *RX* of ESP8266 to *TX* of Arduino (pin 1).
- *VCC* and *GND* to Arduino 3.3V and GND respectively.
- Ensure a logic level converter is used if necessary, since ESP8266 operates at 3.3V logic.

2. Develop the Software:

2.1. Write Arduino Code for Sensor Reading:

- Use an appropriate library to interface with the oximeter. Libraries for I2C communication can be used to fetch data from the sensor.
- Implement code to read the heart rate from the sensor.
- Display the heart rate on the LCD display.

2.2. Write Code for ESP8266 Communication:

- On the ESP8266 side, use serial communication to receive data from the Arduino.
- Implement code to send the data to a web server or IoT platform (like Blynk, ThingSpeak, etc.) for monitoring.

3. Testing and Calibration:

- Verify connections and ensure all components are powered correctly.
- Upload the code to the Arduino and ESP8266.
- Check the LCD display to ensure heart rate values are displayed correctly.
- Ensure data is being transmitted to the server or IoT platform by checking the server logs or web interface.

4. Deployment:

- Once tested, you can deploy the system in a real-world environment, such as a health monitoring system or a fitness tracking device.
- This methodology provides a basic framework. Depending on specific requirements, additional features or modifications might be necessary.

CHAPTER 3

3.1 Advantages

1. Arduino Uno

- Ease of Use: User-friendly platform for prototyping and development.
- Versatility: Supports a wide range of sensors and actuators.
- Cost-Effective: Inexpensive compared to other microcontrollers with similar capabilities.

2. Generic ESP8266

- Wireless Communication: Enables data transmission over Wi-Fi for remote monitoring.
- Cost-Efficient: Affordable solution for adding wireless capabilities.
- Community Support: Extensive resources and support available from the community.

3. LCD Display

- Real-Time Visualization: Provides immediate feedback and display of health metrics.
- User Interface: Simple and effective way to interact with the system locally.
- Low Power Consumption: Efficient for continuous use in health monitoring.

4. Oximeter MAX30100

- Accurate Measurements: Provides reliable data on oxygen saturation and heart rate.
- Compact Size: Small and easy to integrate into various systems.
- Low Power Consumption: Suitable for battery-powered applications.

3.2 Limitations

1. Arduino Uno

- Limited Processing Power: May struggle with complex calculations or multiple tasks.
- Memory Constraints: Limited RAM and storage can restrict program size and complexity.

2. Generic ESP8266

- Connectivity Issues: May face challenges with Wi-Fi signal strength and stability.
- Power Consumption: Higher power use compared to some other communication modules.

3. LCD Display

- Limited Data Display: May not be ideal for displaying extensive or complex information.
- Brightness and Visibility: Performance can vary under different lighting conditions.

4. Oximeter MAX30100

- Sensitivity to Movement: Accuracy can be affected by user movement or placement.
- Limited Measurement Range: May not cover all physiological parameters or conditions.

3.3 Applications

1. Arduino Uno

- Prototyping: Ideal for developing and testing new electronic projects.
- Educational Use: Commonly used for teaching electronics and programming.

2. Generic ESP8266

- IoT Projects: Facilitates the creation of Internet-connected devices.
- Remote Monitoring: Enables sending data to remote servers or cloud platforms.

3. LCD Display

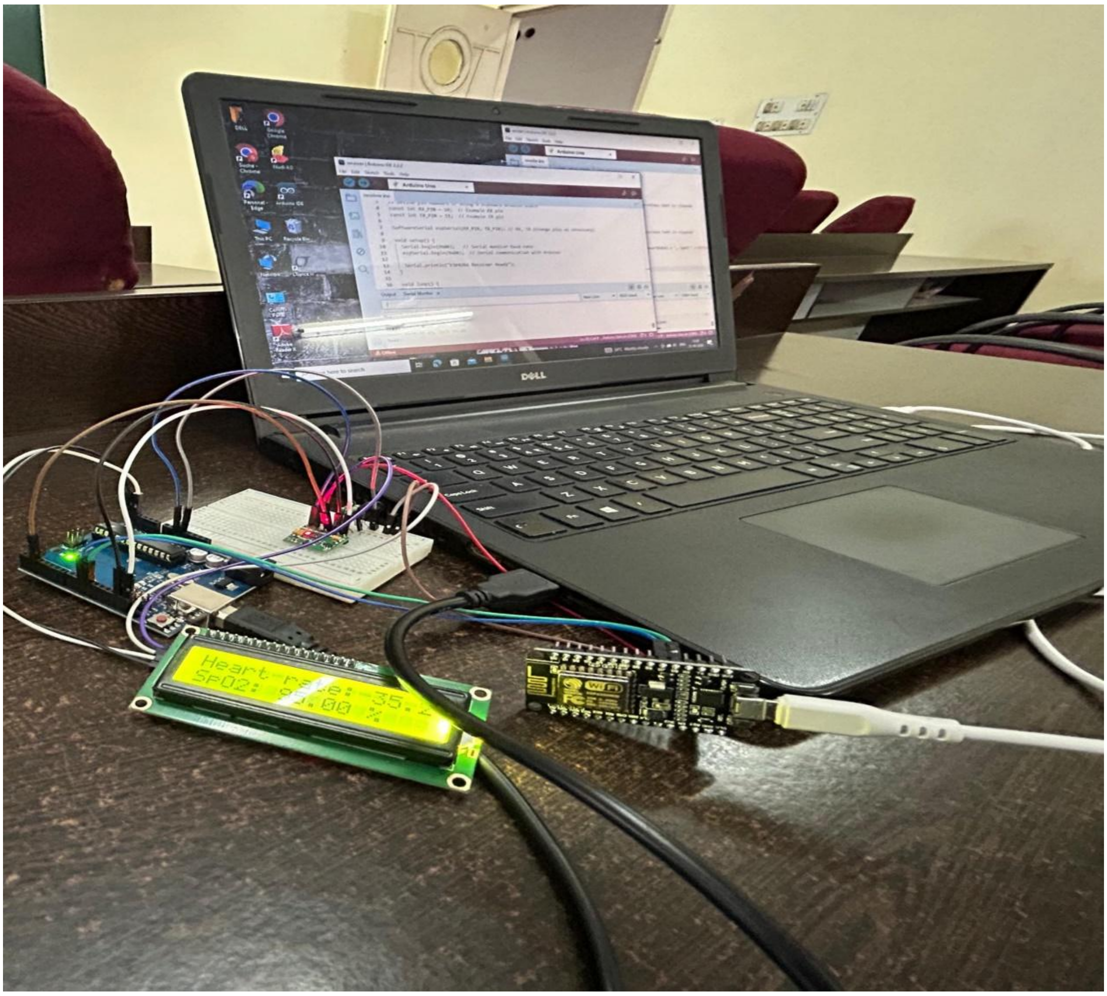
- User Interfaces: Provides a means to interact with and view data from electronic systems.
- Real-Time Data Display: Used in devices where immediate feedback is necessary.

4. Oximeter MAX30100

- Health Monitoring: Used in wearable devices for tracking vital signs.
- Fitness Tracking: Integrated into fitness gadgets to monitor heart rate and oxygen levels during exercise.

CHAPTER 4

4.1 Result



CONCLUSION

The MedWatch System integrates the Arduino Uno, Generic ESP8266, LCD Display, and MAX30100 oximeter to create a robust health monitoring solution. The system efficiently combines real-time data collection with local display and remote access capabilities. The Arduino Uno provides reliable processing power for interpreting data, while the MAX30100 oximeter ensures precise measurement of vital signs like SpO2 and heart rate. The LCD Display offers immediate, user-friendly visualization of health metrics, enhancing on-site monitoring.

The Generic ESP8266 module facilitates seamless wireless communication, allowing the MedWatch System to transmit health data to a remote server or cloud platform. This feature supports continuous remote monitoring by healthcare providers, enabling timely interventions and improving overall health management. Despite some limitations, such as the Arduino's processing constraints and the oximeter's sensitivity to movement, the MedWatch System delivers significant benefits in personal health tracking and early detection of health issues, offering a comprehensive approach to both immediate and long-term health management..

REFRENCES

[1] Remote Patients Monitoring System(Heartbeat and Temperature) using Arduino,Ruaa Shallal Abbas Anooz, Methodology -using IoT, IJSER, Volume 9, ISSN 2229-5518, August 2018.

[2] IoT Based Heart Activity Monitoring Using Inductive Sensors, Adrian Brezulianu, Oana Geman. Methodology-using IoT, Sensors,Volume 19, DOI: 10.3390/s19153284, July 2019.

[3] Heartbeat Monitoring Using IoT, Sayan Banerjee, Souptik Paul, Rohan Sharma, Abhishek Brahma, IEEE Access, DOI:10.1109/IEMCON.2018.8614921, January 2019.

[4] Analysis of Heart Rate and Body Temperature from the wireless monitoring system using Arduino, Mohammad Dabbagh. Methodology–using IoT ,Journal of Physics: ConferenceSeries, DOI: 10.1088/1742-6596/1358/1/012041, Published–November 2019.

[5] Developing IoT Based Smart Health Monitoring Systems: A Review, Ashikur Rahaman, Muhammad Sheikh Sadi, Sheikh Nooruddin. Methodology–using IoT, IIETA access, September 2019.

[6] Development of smart health care monitoring system in IoT environment, Md.Milon Islam, Ashikur rahaman, Md.rashedul Islam. Methodology– using IoT, Springer, Article-185(2020), May2020.

[7] Low cost Heart rate Portable Device for the Risk patients with IoT and warning system, Muhammad Irman, Era Madon. Methodology-using IoT, IEEE Access, INSPEC Accession-18596175, DOI: 10.1109/icaiti.2018.8686761, April 2019.

[8] An IoT-Based Smart Framework for Human Heartbeat Rate Monitoring and Control System, Sani abba, Abubakar Mohammed garba. Methodology-using IoT, DOI: 10.3390/ecsa-606543, November 2019.