

Established – 1961

Subject: IOT

SEVA SADAN'S
R. K. TALREJA COLLEGE
OF
ARTS, SCIENCE & COMMERCE
ULHASNAGAR – 421 003



CERTIFICATE

This is to certify that Mr./Ms. Akash Ashok Balghare
of S.Y. Computer Science (SYCS) Roll No.
2524002 has satisfactorily completed
The Internet Of Thing Mini Project entitled _____
Smart Heart Rate Monitor System

during the academic year 2025 – 2026, as a part of the practical requirement. The project work is found to be satisfactory and is approved for submission.

PROF. INCHARGE

HEAD OF DEPT

INDEX

Sr. No.	Chapters	Page No.
1	Introduction	3- 7
2	Requirement Specification	8-9
3	System Design	10-27
4	Implementation	28-37
5	System Testing and Result	38-43
6	Future Scope and Conclusion	44-45
7	References	46-47
8	Glossary	48-49

INTRODUCTION

IoT Based Heart Rate Monitoring System Introduction:

The **IoT Based Heart Rate Monitoring System** is a simple and practical project that focuses on measuring and monitoring the human heart rate using IoT technology.

The system is designed to calculate the heart rate in terms of **Beats Per Minute (BPM)** and display the result on an OLED screen.

The project is built using an **ESP8266 microcontroller** as the main controller, which interfaces with a **pulse sensor** to detect heartbeats and an **OLED display** to show the output. The system also uses **WiFi connectivity** to upload the heart rate data to the cloud platform for remote monitoring.

This project provides a basic understanding of how IoT can be used in healthcare applications by combining sensors, microcontrollers, and cloud services to perform real-time monitoring.

Motivation :

The motivation behind developing the **IoT Based Heart Rate Monitoring System** is the growing need for simple and affordable health monitoring solutions. Monitoring heart rate regularly can help in understanding a person's health condition and detecting abnormalities at an early stage.

This project provides an educational and practical way for students to learn about **IoT concepts**, **sensor interfacing**, and **wireless communication**. It also demonstrates real-world applications of IoT in healthcare, remote patient monitoring, and fitness tracking systems.

Problem Definition :

The main problem addressed by this project is the need for a **basic and low-cost system** that can measure heart rate and make the data accessible remotely. Traditional heart rate monitoring devices can be expensive and not easily accessible to everyone.

This project aims to solve the problem by designing a system that can **detect heartbeats using a pulse sensor**, **calculate BPM accurately**, **display the result locally**, and **upload the data to the cloud** using IoT technology. The system helps in understanding how real-time health data can be monitored and stored efficiently.

How it Works :

The **IoT Based Heart Rate Monitoring System** works using an ESP8266 microcontroller to interface with a pulse sensor and other components.

The pulse sensor detects changes in blood flow through the finger and generates an analog signal corresponding to heartbeats.

The ESP8266 reads this signal, processes it, and calculates the heart rate in terms of **Beats Per Minute (BPM)**. The calculated BPM is displayed on an OLED screen for local monitoring. Using built-in WiFi connectivity, the system uploads the heart rate data to the ThingSpeak cloud platform, allowing users to monitor the data remotely.

This combination of sensor input, data processing, and cloud connectivity enables real-time heart rate monitoring in a simple and effective manner.

Key Features :

Heart Rate Measurement :

Uses a pulse sensor to accurately detect heartbeats and calculate BPM.

OLED Display Output :

Displays real-time measurement status and final heart rate value on an OLED screen for easy viewing.

IoT Cloud Integration :

Uploads heart rate data to the ThingSpeak cloud platform using WiFi for remote monitoring and data analysis.

Real-Time Monitoring :

Provides continuous and timely heart rate updates, making it useful for basic health tracking.

User-Friendly Operation :

Simple design and easy-to-understand working make the system suitable for beginners and students.

Educational Value :

Helps students learn about IoT, sensor interfacing, data processing, and wireless communication through a practical healthcare application.

Scope of the Project :

The **IoT Based Heart Rate Monitoring System** has wide scope in the field of healthcare and IoT applications. The system provides a simple and cost-effective way to monitor heart rate in real time. It can be used for basic health monitoring, fitness tracking, and educational purposes.

In academic environments, this project helps students understand the integration of sensors, microcontrollers, and cloud platforms. The system also demonstrates real-world applications of IoT in healthcare, such as remote patient monitoring and health data analysis.

The project can be further enhanced by adding more sensors, mobile application support, alert systems, or advanced data analytics, increasing its future potential.

Academic Institutions :

The **IoT Based Heart Rate Monitoring System** can be used as a learning tool in academic institutions. Students can gain hands-on experience with IoT, sensor interfacing, and wireless communication. It helps in understanding practical implementation of healthcare monitoring systems.

Businesses and Healthcare Services :

Healthcare centers, gyms, and wellness programs can use this system for basic heart rate monitoring. The data collected can help in fitness tracking and health analysis. Its low-cost design makes it suitable for small clinics and health awareness programs.

Data Security and Health Monitoring :

The system can be improved to store and manage health data securely on cloud platforms. This helps in maintaining records and ensuring safe access to heart rate information for future reference and analysis.

Online Learning Platforms :

Online learning platforms can use this project to teach IoT and healthcare monitoring concepts. Students can learn remotely by building and testing the system at home, making learning more interactive and practical.

Objective :

The objective of the **IoT Based Heart Rate Monitoring System** project is to design a simple and reliable system capable of measuring and monitoring human heart rate using IoT technology. The project aims to provide an easy-to-use solution for basic health monitoring while helping students understand IoT concepts and sensor integration.

Key goals include :

Promote Learning in IoT and Healthcare :

Encourage students to learn about IoT, microcontrollers, and healthcare monitoring through hands-on practical experience.

Accurate Heart Rate Measurement :

Develop a system that can detect heartbeats and calculate heart rate in terms of BPM using a pulse sensor.

Real-Time Monitoring :

Enable real-time display of heart rate on an OLED screen and upload data to the cloud for remote access.

User-Friendly Design :

Create a simple and easy-to-operate system that can be used by beginners without complex setup.

Cloud Data Storage :

Use IoT cloud platforms to store and visualize heart rate data for future reference and analysis.

Support Future Enhancements :

Provide a foundation for adding advanced features such as alerts, mobile applications, or additional health sensors.

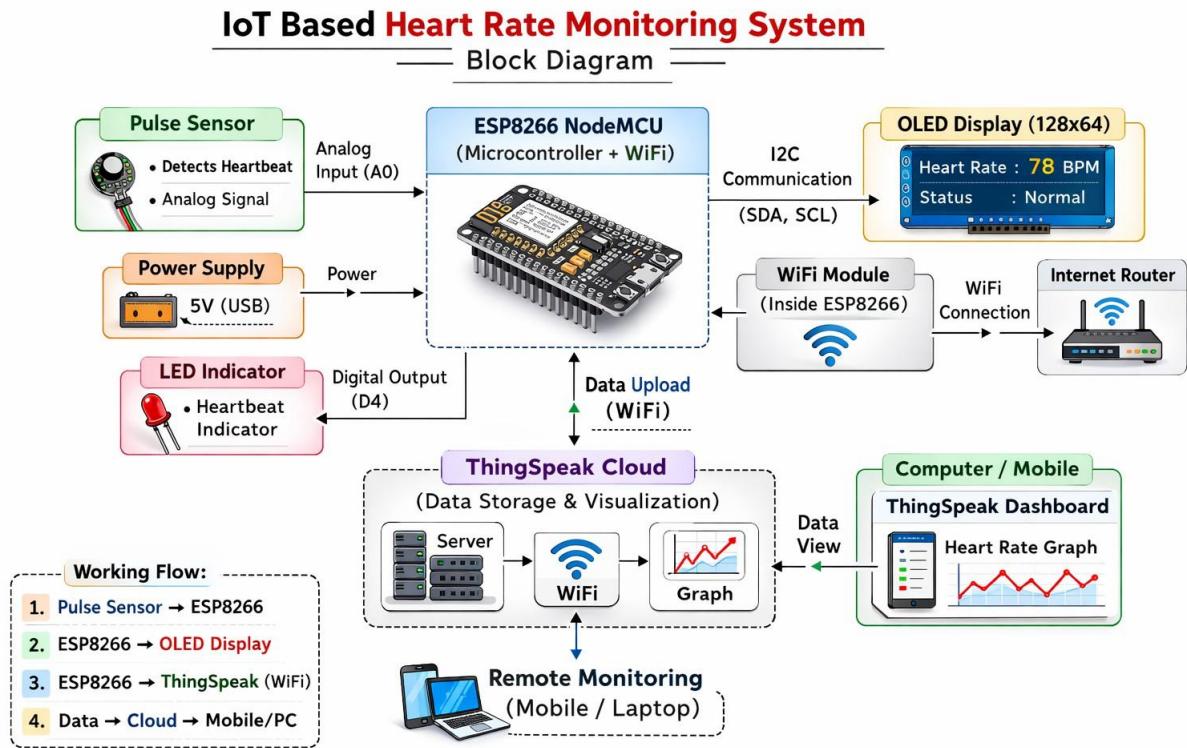
REQUIREMENT SPECIFICATION

Component	Specification
Microcontroller	ESP8266 NodeMCU
Heart Rate Sensor	Pulse Sensor for detecting heartbeats
Display	OLED Display (128×64)
Indicator	LED for heartbeat indication
Connecting Wires	Jumper wires for circuit connections
Breadboard	For mounting and testing components
Power Supply	USB power or 5V external supply

Component	Specification
Arduino IDE	Software for programming the ESP8266
ESP8266 Board Package	Required to program NodeMCU
Display Libraries	Adafruit SSD1306 and Adafruit GFX
IoT Platform	ThingSpeak for cloud data storage
WiFi Network	Internet connection for data upload

SYSTEM DESIGN

Block Diagram :



The block diagram represents the overall structure and interaction between the main components of the IoT Based Heart Rate Monitoring System. It explains how the sensor, microcontroller, display, and cloud platform work together to measure and monitor heart rate.

ESP8266 NodeMCU :

Acts as the main controller of the system. It processes the signal received from the pulse sensor, calculates the heart rate, controls the OLED display, and uploads data to the cloud using WiFi.

Pulse Sensor :

Detects heartbeats by sensing changes in blood flow through the finger and sends analog signals to the ESP8266.

OLED Display (128×64) :

Displays the measuring status, progress, and final BPM value for local monitoring.

WiFi Connectivity :

Allows the ESP8266 to connect to the internet and send heart rate data to the ThingSpeak cloud platform.

ThingSpeak Cloud Platform :

Stores and visualizes the heart rate data in graphical form for remote monitoring and analysis.

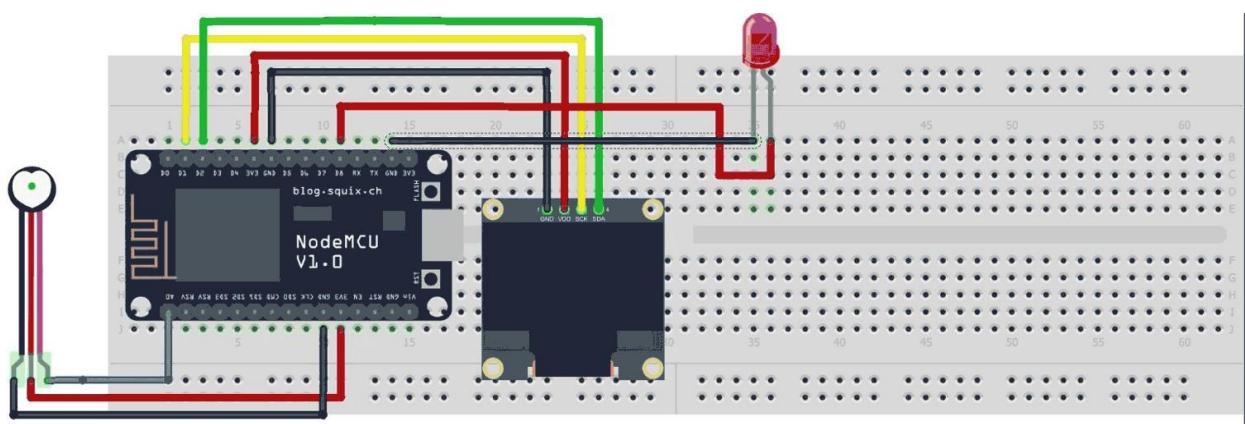
LED Indicator :

Blinks whenever a heartbeat is detected, providing visual confirmation of pulse detection.

Power Supply :

Provides required power to the ESP8266 and connected components through USB or external source.

Circuit Diagram :



Description :

The block diagram demonstrates the interaction between the key components of the **IoT Based Heart Rate Monitoring System**, such as the ESP8266 microcontroller, pulse sensor, OLED display, WiFi module, and cloud platform. The ESP8266 receives analog signals from the pulse sensor to detect heartbeats and calculate the heart rate in BPM.

The calculated heart rate is displayed on the OLED screen for local monitoring, while the WiFi connectivity allows the system to send heart rate data to the ThingSpeak cloud platform for remote access. An LED indicator provides visual feedback for each detected heartbeat. Power is supplied to all components through a USB or external power source.

Key Steps of Circuit Diagram :

ESP8266 Setup :

Connect the ESP8266 NodeMCU as the main controller of the system. All components such as the pulse sensor, OLED display, LED, and WiFi communication are controlled through this board.

Pulse Sensor :

Connect the pulse sensor output pin to the analog pin (A0) of the ESP8266. This sensor detects heartbeats by sensing changes in blood flow through the finger.

OLED Display (128×64) :

Connect the OLED display to the ESP8266 using I2C pins (SDA and SCL). The display is used to show the measuring status and final BPM value.

LED Indicator :

Connect an LED to a digital pin of the ESP8266 with a resistor. The LED blinks whenever a heartbeat is detected, providing visual feedback.

WiFi Connectivity :

Use the built-in WiFi module of the ESP8266 to connect to the internet. This allows heart rate data to be uploaded to the ThingSpeak cloud platform.

Power Supply :

Power the ESP8266 and connected components using a USB cable or a regulated 5V power supply to ensure stable operation.

These steps ensure proper integration of all components for accurate heart rate monitoring.

Benefits of Circuit Diagram :

Clear Visualization :

Provides a clear view of how all components are connected, helping to understand the flow of signals in the system.

Error Detection :

Helps in identifying wiring or connection errors before implementing the actual

hardware.

Documentation :

Acts as an important reference for future modifications, troubleshooting, or project replication.

Ease of Assembly :

Makes the hardware assembly process easier by clearly showing pin connections and component placement.

This is important for projects like the **IoT Based Heart Rate Monitoring System** to ensure safe and correct operation.

When to Use a Circuit Diagram ?

Design Phase :

Used while planning the project to visualize component connections and system structure.

Troubleshooting :

Helps in diagnosing problems by checking connections and signal flow.

Documentation :

Essential for documenting the project for reports, submissions, or sharing with others.

Education :

Helps students understand electronic components, circuit connections, and working principles clearly.

Assembly Guidance :

- The assembly guidance provides a clear roadmap for assembling the **IoT Based Heart Rate Monitoring System**. It helps ensure that all components such as the ESP8266, pulse sensor, OLED display, and LED are connected correctly. Proper pin connections and correct placement of components reduce the chances of errors and ensure smooth functioning of the system.

Circuit Model Testing Methods :

Simulation Testing :

Simulation tools like **Proteus** or **Tinkercad** can be used to understand the circuit design before physical assembly. This helps in identifying possible design issues at an early stage.

Continuity Testing :

After assembling the circuit, a multimeter can be used to check continuity. This ensures that all connections are properly made and there are no short circuits.

Functional Testing :

Power the circuit and verify that each component such as the pulse sensor, OLED display, LED indicator, and WiFi connection works correctly as expected.

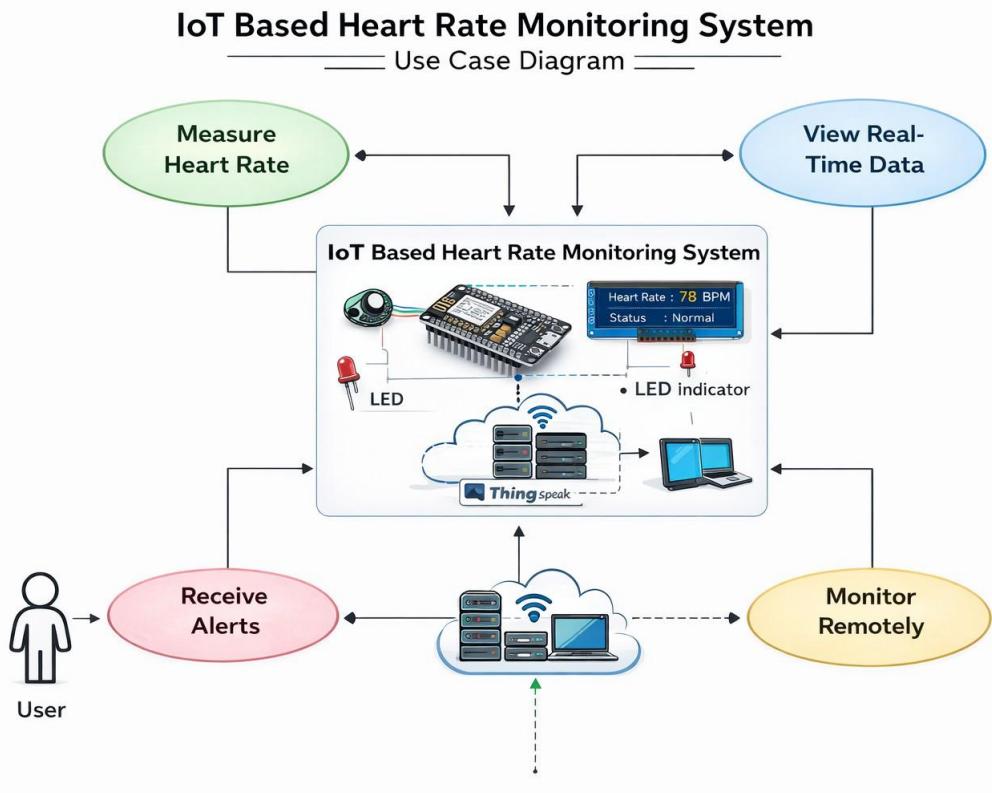
Load Testing :

Test the system under normal operating conditions to ensure that it remains stable while measuring heart rate and uploading data to the cloud.

Debugging :

If any issue or unexpected behavior occurs, debugging is done by checking sensor readings, pin connections, and code logic to identify and fix the problem.

Use Case Diagram :



Description :

1. The **Use Case Diagram for the IoT Based Heart Rate Monitoring System** illustrates the interactions between the user and the system's functionalities. It provides a visual representation of the key actions performed by the user and how the system responds to those actions.
2. The main actors identified in the system are **User**, **Pulse Sensor**, and **Cloud Platform (ThingSpeak)**. Below is a detailed explanation of the use cases included in the diagram.

Measure Heart Rate :

Actor: User, Pulse Sensor

Description:

This use case allows the user to measure their heart rate by placing a finger on the pulse sensor. The pulse sensor detects heartbeats and sends the signal to the ESP8266 microcontroller for processing.

Calculate BPM :

Actor: IoT Based Heart Rate Monitoring System

Description:

The system processes the signals received from the pulse sensor and calculates the heart rate in terms of Beats Per Minute (BPM). This

ensures accurate and meaningful heart rate measurement.

- **Display Heart Rate :**

- **Actor:** IoT Based Heart Rate Monitoring System

Description:

Once the BPM is calculated, the system displays the heart rate on the OLED display. This allows the user to view the result instantly on the device.

- **Upload Data to Cloud :**

- **Actor:** Cloud Platform (ThingSpeak)

Description:

The system uploads the heart rate data to the ThingSpeak cloud platform using WiFi connectivity. This enables remote monitoring and storage of heart rate data for future reference.

-

- **Provide Visual Feedback :**

- **Actor:** IoT Based Heart Rate Monitoring System

Description:

The system provides visual feedback through an LED indicator, which blinks whenever a heartbeat is detected. This confirms that the sensor is working properly.

-

- **Purpose of Use Case Diagrams :**

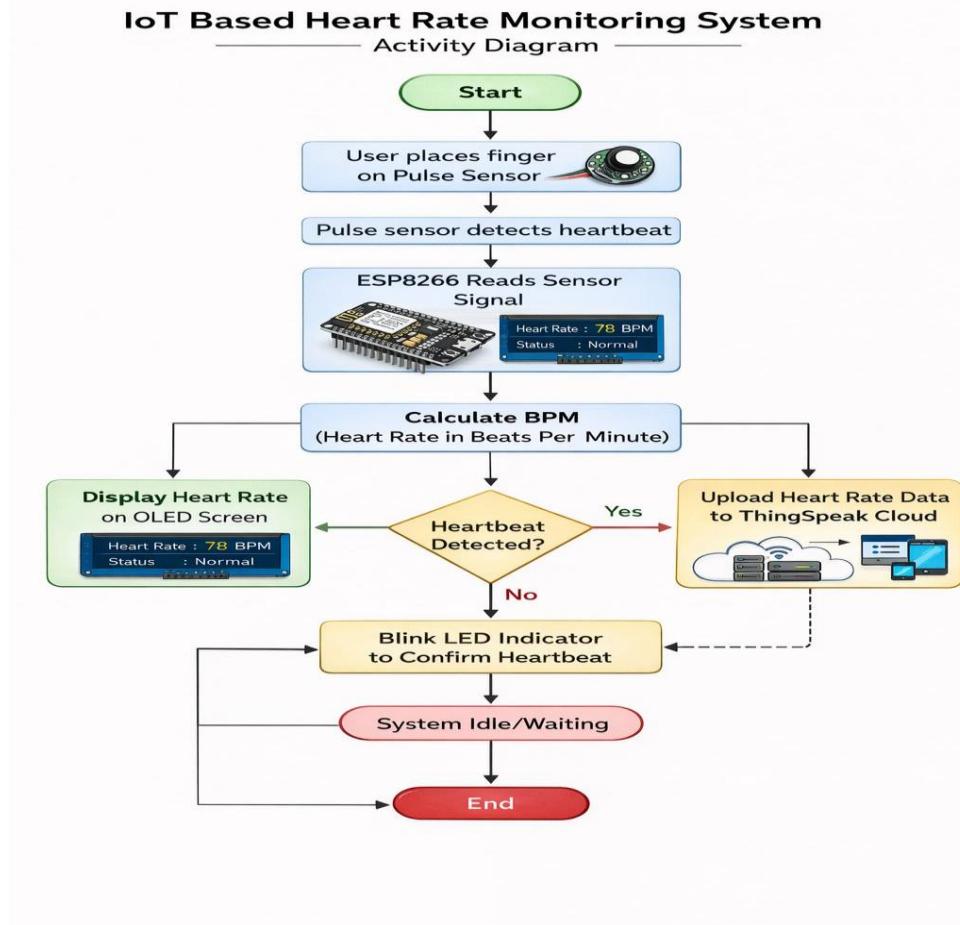
- **Visualizing System Interactions :**

- Use Case Diagrams provide a clear visual representation of how the user interacts with the heart rate monitoring system, making system functionality easy to understand.

- **Defining System Requirements :**

- They help identify and define the functional requirements of the system, ensuring that all features such as measurement, display, and cloud upload are included.
- **Facilitating Communication :**
Use Case Diagrams act as an effective communication tool between students, teachers, and developers by presenting system functionality in a simple visual form.
- **Guiding System Design :**
These diagrams help guide system design by highlighting important functionalities that need to be implemented in the project.
- **Documenting Functional Requirements :**
They serve as documentation for the functional behavior of the system, which is useful for future upgrades or maintenance.
- **Supporting Test Case Development :**
Each use case can be converted into test cases during testing to verify that the system performs correctly.
- **Identifying User Needs :**
By focusing on user interaction, Use Case Diagrams help understand user needs and expectations for effective heart rate monitoring.
- **Simplifying Complex Systems :**
For IoT-based systems, Use Case Diagrams simplify complexity by breaking the system into manageable and understandable interactions.

Activity Diagram :



Description:

Activity diagrams are a type of UML (Unified Modeling Language) diagram that represent the flow of activities within a system. They are useful for showing how different actions are performed step by step. In the **IoT Based Heart Rate Monitoring System**, the activity diagram explains the complete workflow starting from sensor initialization to displaying and uploading the heart rate data. It helps in understanding how the system behaves during real-time operation.

Purpose of Activity Diagrams :

Visualize Workflow :

Activity diagrams provide a clear visual representation of the workflow of the

heart rate monitoring system. They show the sequence of activities such as system start, heartbeat detection, BPM calculation, display output, and data upload to the cloud, making the process easy to understand.

- **Modeling Processes :**

- These diagrams help in modeling both high-level and detailed processes involved in heart rate monitoring. They describe how the system reads sensor data, processes it, and generates meaningful output, which is useful during system design and development.

- **Identify Parallel Activities :**

- Activity diagrams can represent parallel activities such as displaying BPM on the OLED screen while simultaneously sending data to the ThingSpeak cloud platform. This helps in understanding concurrent operations within the system.

- **Clarifying Responsibilities :**

- The diagram clearly shows which component performs which task. For example, the pulse sensor detects heartbeats, the ESP8266 processes the data, and the OLED display shows the results. This clarity helps in proper system understanding and implementation.

- **Supporting Communication :**

- Activity diagrams act as an effective communication tool between students, instructors, and developers. They present the system workflow in a visual format that is easy to explain and understand, even for non-technical users.

- **Facilitating Documentation :**

- These diagrams serve as an important part of project documentation. They provide a reference for future improvements, maintenance, or explanation of the system's working.

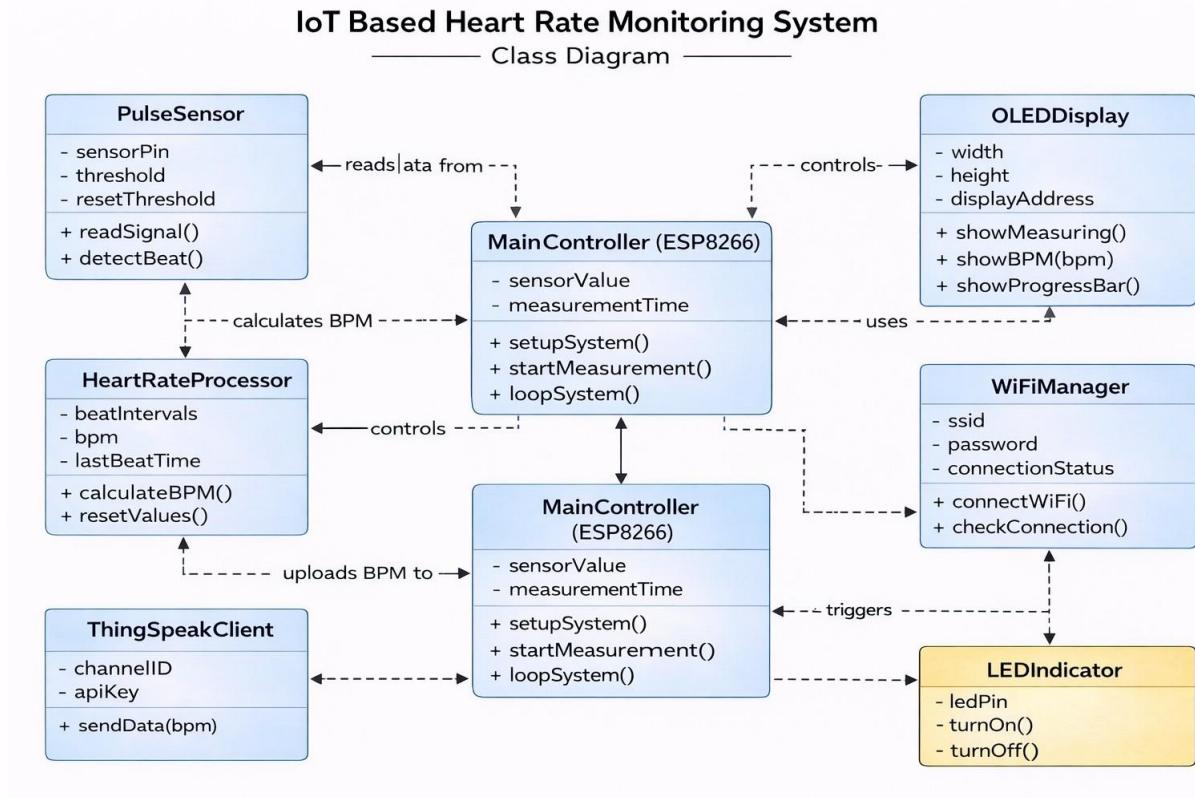
- **Enhancing System Design :**

- During the design phase, activity diagrams help identify possible issues or inefficiencies in the workflow. This allows improvements to be made before actual implementation of the system.

- **Testing and Validation :**

- Activity diagrams assist in testing and validation by clearly showing expected system behavior. Test cases can be created based on each activity to ensure that the system works correctly and reliably.

Class Diagram :



Class Diagram : Description

Class diagrams are a fundamental part of object-oriented design, providing a visual representation of a system's classes and their relationships. They serve as a blueprint for the structure of a software application, illustrating how different classes interact with each other.

In the context of the **IoT Based Heart Rate Monitoring System**, class diagrams play an important role in organizing and defining the system components such as the microcontroller, pulse sensor, display module, and cloud platform. The diagram helps in understanding how data flows from the sensor to the display and cloud service, ensuring clarity among developers and users.

Purpose of Class Diagrams :

Modeling System Structure :

The class diagram visualizes the main components of the heart rate monitoring system, including the ESP8266 microcontroller, pulse sensor, OLED display, and ThingSpeak cloud service. It provides a clear structural overview of the system.

Understanding Relationships :

It shows how different classes such as Sensor, Display, WiFiModule, and CloudService

interact with each other. This helps in understanding dependencies and data communication within the system.

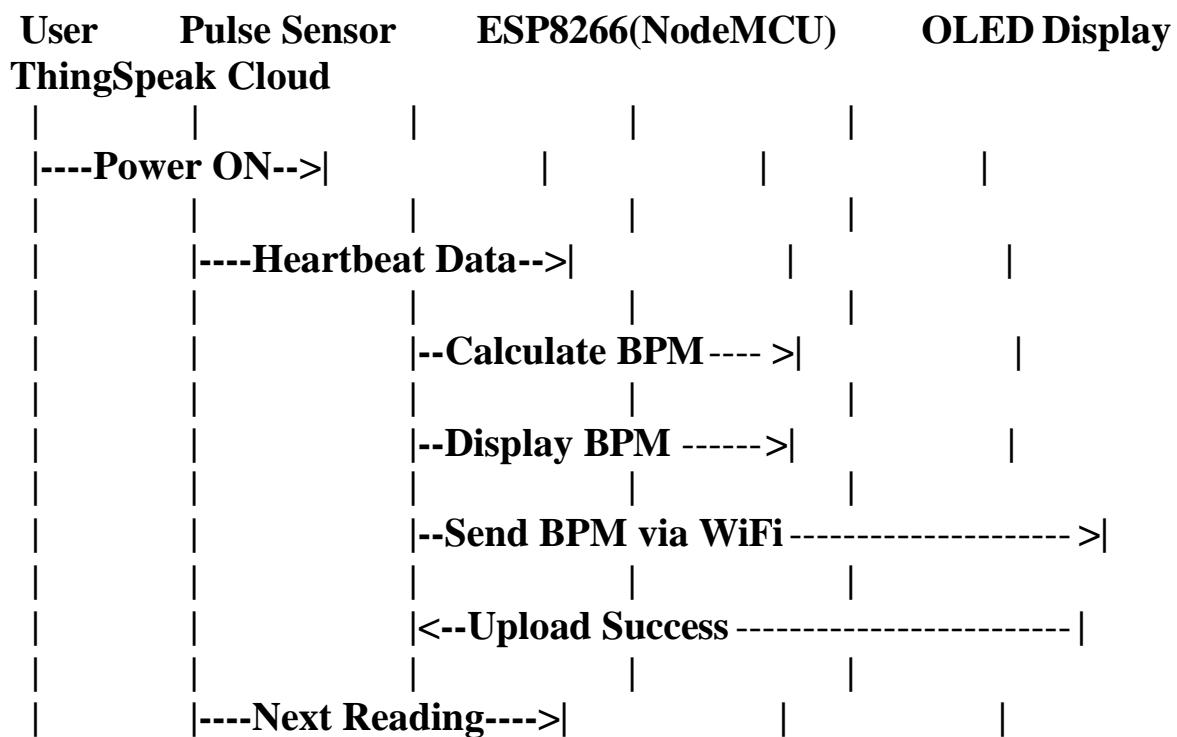
Design Blueprint :

The class diagram acts as a blueprint for implementing the system logic in code. It defines how sensor data is processed, displayed on the OLED screen, and uploaded to the cloud platform.

Documentation :

It serves as an important reference for future maintenance, debugging, and enhancement of the system. Developers can easily understand the system architecture and extend functionality when required.

Sequence Diagram :



Description :

A Sequence Diagram is a type of interaction diagram in Unified Modeling Language (UML) that illustrates how different components of a system interact with each other in a specific sequence over time. It focuses on the order in which messages or data are exchanged between objects to complete a particular process.

In the **IoT Based Heart Rate Monitoring System**, the sequence diagram helps in understanding how heart rate data flows from the pulse sensor to the microcontroller, then to the display and cloud platform, showing the dynamic behavior of the system.

Purpose of Sequence Diagrams :

Visual Representation of Interactions :

Sequence diagrams provide a clear visual representation of interactions between system components such as the pulse sensor, ESP8266 microcontroller, OLED display, and ThingSpeak cloud server in a time-based sequence.

Clarification of Message Flow :

They help clarify how heart rate data is sensed, processed, displayed, and uploaded to the cloud. This makes it easier to understand the flow of control during heart rate measurement and data transmission.

Facilitation of Communication :

The diagram acts as a communication tool among developers and students, helping them discuss and understand the system workflow and data exchange clearly.

Guidance for Development :

It serves as a guide for implementing interactions between hardware and software components. Developers can follow the sequence to program sensor reading, BPM calculation, display output, and cloud upload correctly.

Documentation :

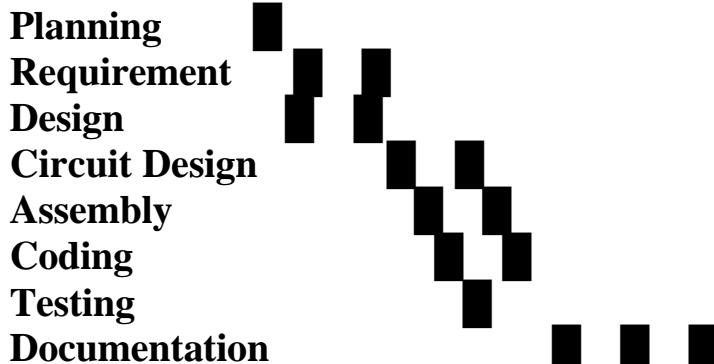
Sequence diagrams provide clear documentation of system operation, making future maintenance, debugging, and enhancements easier.

Identification of Issues :

By visualizing the interaction flow, sequence diagrams help identify delays, incorrect data transfer, or communication issues between components, improving system reliability.

Gantt Chart :

Dates → 13 14 15 16 17 18 19 20



outlines the tasks involved in a project along a timeline, showing when each task starts, how long it will take, and when it should be completed. Gantt charts are widely used in project management to plan, coordinate, and track specific tasks or activities, providing a clear overview of the project's progress and deadlines.

1. The above Gantt chart provides a visual representation of the timeline for the **IoT Based Heart Rate Monitoring System** project. The project tasks are scheduled between **13 February 2026 and 20 February 2026**. The chart breaks the project into key phases such as planning, design, hardware assembly, coding, testing, and documentation, clearly displaying each phase's start and end dates.
2. This representation helps in understanding how the project was completed within a short time frame while ensuring proper planning and timely execution of all activities.
3. _____

Advantages of Using Gantt Charts :

Clear Visualization :

Gantt charts provide a clear and simple visual representation of the project timeline, making it easy to understand task sequences and deadlines.

Effective Planning :

They help in organizing tasks efficiently by assigning time durations to each activity, ensuring systematic project execution.

Tracking Progress :

Gantt charts allow continuous monitoring of project progress, helping identify completed, ongoing, and pending tasks.

Collaboration :

They improve coordination among team members by clearly defining responsibilities and timelines for each phase of the project.

outlines the tasks involved in a project along a timeline, showing when each task starts, how long it will take, and when it should be completed. Gantt charts are widely used in project management to plan, coordinate, and track specific tasks or activities, providing a clear overview of the project's progress and deadlines.

4. The above Gantt chart provides a visual representation of the timeline for the **IoT Based Heart Rate Monitoring System** project. The project tasks are scheduled between **13 February 2026 and 20 February 2026**. The chart breaks the project into key phases such as planning, design, hardware assembly, coding, testing, and documentation, clearly displaying each phase's start and end dates.
5. This representation helps in understanding how the project was completed within a short time frame while ensuring proper planning and timely execution of all activities.

outlines the tasks involved in a project along a timeline, showing when each task starts, how long it will take, and when it should be completed. Gantt charts are widely used in project management to plan, coordinate, and track specific tasks or activities, providing a clear overview of the project's progress and deadlines.
6. The above Gantt chart provides a visual representation of the timeline for the **IoT Based Heart Rate Monitoring System** project. The project tasks are scheduled between **13 February 2026 and 20 February 2026**. The chart breaks the project into key phases such as planning, design, hardware assembly, coding, testing, and documentation, clearly displaying each phase's start and end dates.
7. This representation helps in understanding how the project was completed within a short time frame while ensuring proper planning and timely execution of all activities.

Advantages of Using Gantt Charts :

Clear Visualization :

Gantt charts provide a clear and simple visual representation of the project timeline, making it easy to understand task sequences and deadlines.

Effective Planning :

They help in organizing tasks efficiently by assigning time durations to each activity, ensuring systematic project execution.

Tracking Progress :

Gantt charts allow continuous monitoring of project progress, helping identify completed, ongoing, and pending tasks.

Collaboration :

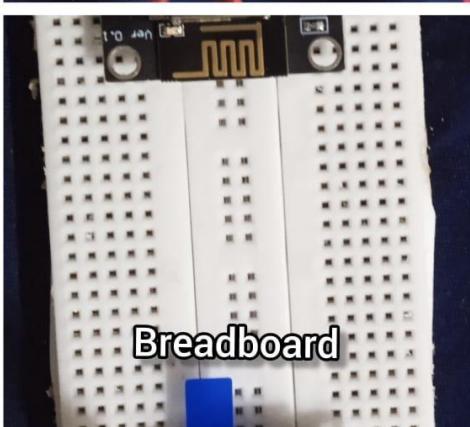
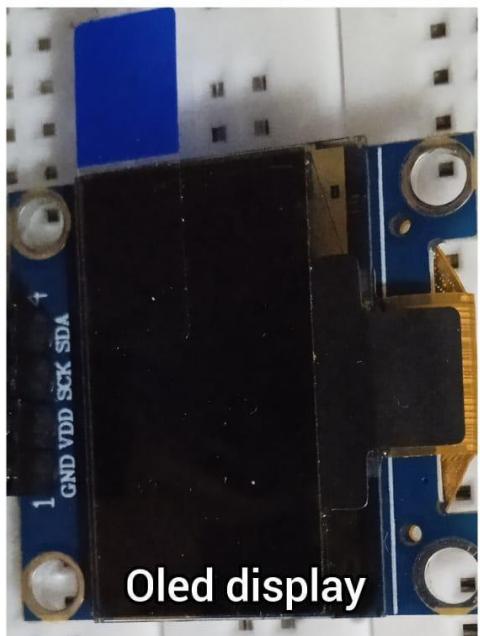
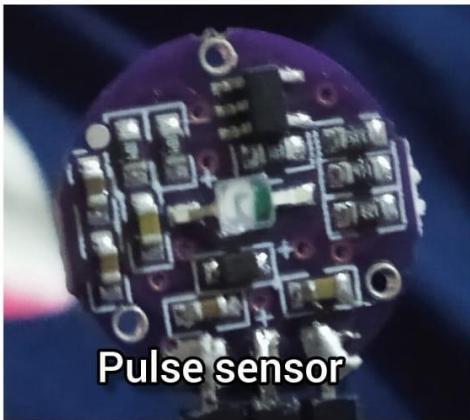
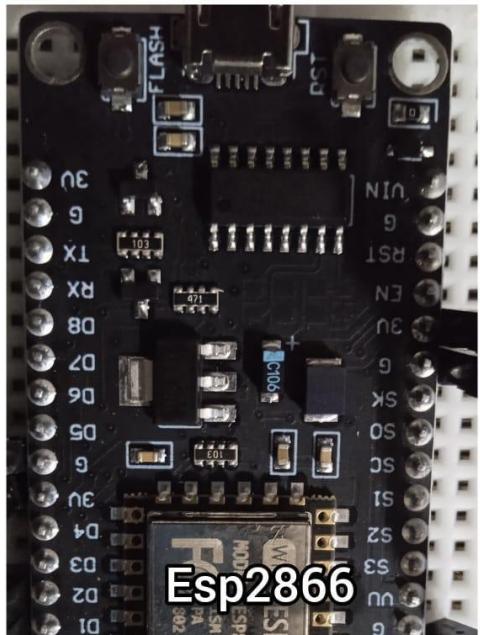
They improve coordination among team members by clearly defining responsibilities and timelines for each phase of the project.

SYSTEM IMPLEMENTATION

Step-by-Step Assembly:

Step 1 :

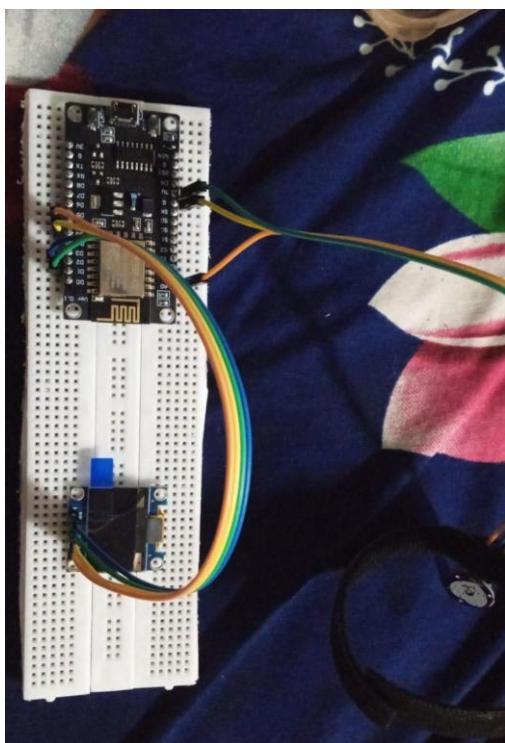
Firstly, identify these components.



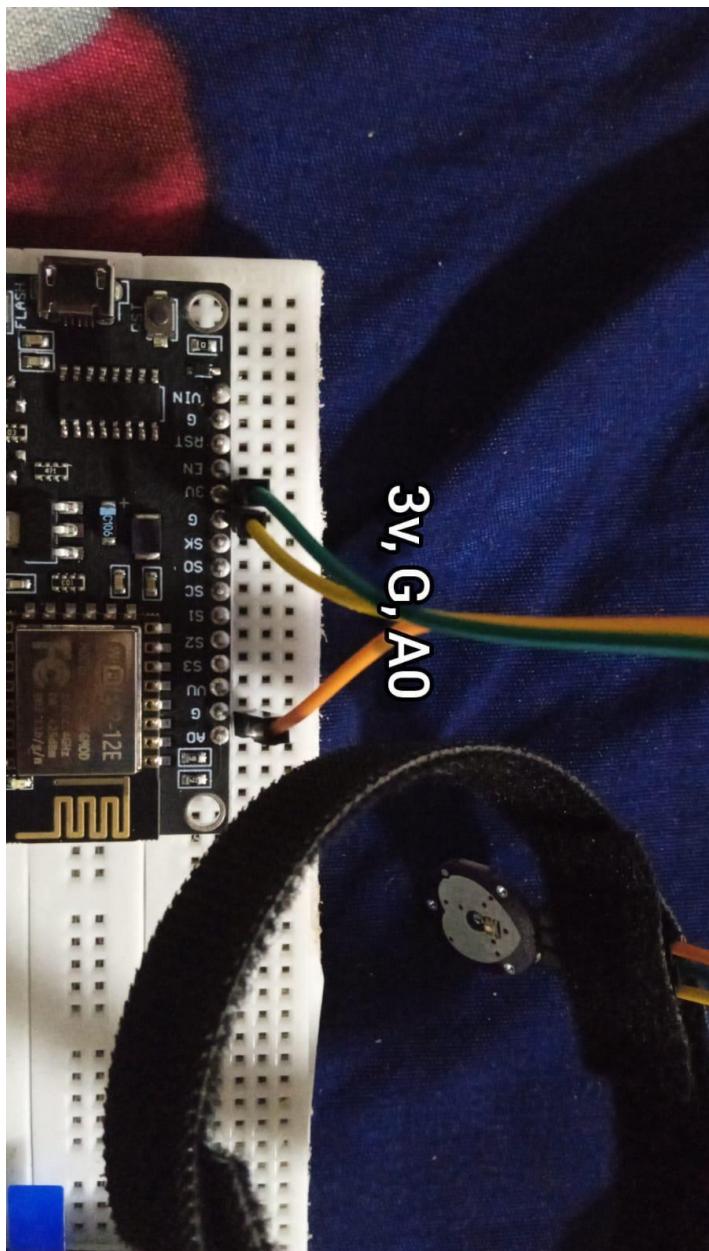


Step 2 :

Place all components on Breadboard :

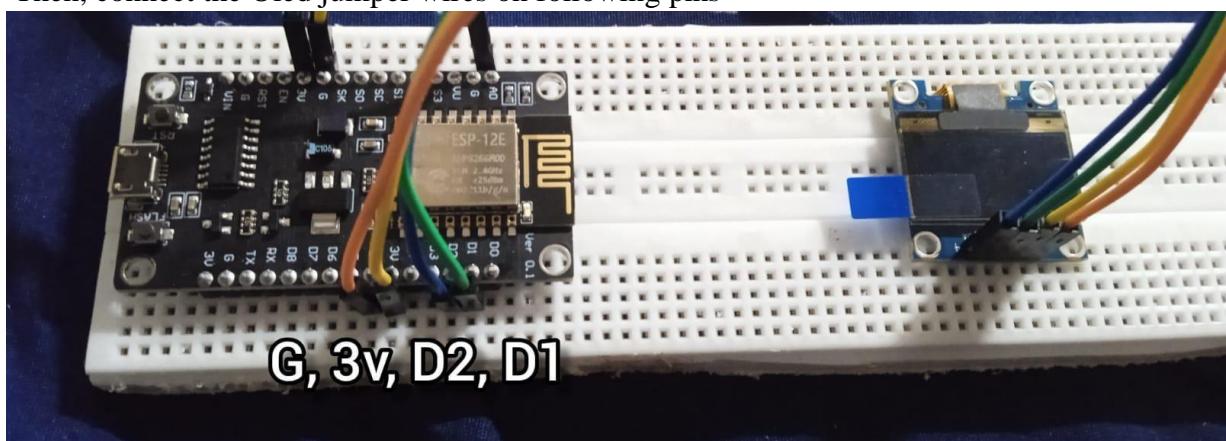


Step 3 :Connect Pulse sensor jumper wires on following pins.



Step 4 :

Then, connect the Oled jumper wires on following pins



Step 5 :

So, let's create the program for this project. This program includes all three functions. We can run these separately. It is as follows.

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <ESP8266WiFi.h>
#include <ThingSpeak.h>

Adafruit_SSD1306 display(128, 64, &Wire);

// ----- WiFi -----
const char* ssid = "ESP_TEST";
const char* password = "123456789";

// ----- ThingSpeak -----
WiFiClient client;
unsigned long channelID = 3263806;
const char* writeAPIKey = "76P4WTT1W9CR9Y9K";

// ----- Sensor -----
const int sensorPin = A0;
const int ledPin = D8;

// ----- Thresholds -----
```

```

int threshold = 580;
int resetThreshold = 550;

// ----- Timing -----
unsigned long lastSampleTime = 0;
unsigned long measureStartTime = 0;
unsigned long lastBeatTime = 0;
unsigned long restartTimer = 0;

const unsigned long sampleInterval = 50;
const unsigned long measureDuration = 20000; // 20 sec
const unsigned long restartDelay = 60000; // 1 minute
const unsigned long minBeatInterval = 300;

// ----- Variables -----
int sensorValue;
bool pulseDetected = false;
bool measurementDone = false;

int beatIntervals[5];
int beatIndex = 0;
int bpm = 0;

void setup() {
    Serial.begin(115200);
}

```

```
pinMode(ledPin, OUTPUT);

display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
display.clearDisplay();
display.setTextColor(WHITE);

WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) delay(300);
```

```
ThingSpeak.begin(client);
```

```
startMeasurement();
```

```
}
```

```
void startMeasurement() {
    measureStartTime = millis();
    lastBeatTime = 0;
    beatIndex = 0;
    bpm = 0;
    measurementDone = false;
```

```
display.clearDisplay();
display.setTextSize(2);
display.setCursor(0, 20);
display.println("Measuring");
```

```

display.display();

}

void loop() {
    unsigned long currentMillis = millis();

    // ----- AUTO RESTART AFTER 1 MIN -----
    if (measurementDone && (currentMillis - restartTimer >= restartDelay)) {
        startMeasurement();
    }

    if (measurementDone) return;

    // ----- SENSOR SAMPLING -----
    if (currentMillis - lastSampleTime >= sampleInterval) {
        lastSampleTime = currentMillis;
        sensorValue = analogRead(sensorPin);

        if (sensorValue > threshold && !pulseDetected &&
            (currentMillis - lastBeatTime > minBeatInterval)) {
            pulseDetected = true;
            digitalWrite(ledPin, HIGH);

            if (lastBeatTime > 0 && beatIndex < 5) {

```

```

beatIntervals[beatIndex++] = currentMillis - lastBeatTime;

}

lastBeatTime = currentMillis;

}

if (sensorValue < resetThreshold) {

    pulseDetected = false;

    digitalWrite(ledPin, LOW);

}

//


// ----- PROGRESS BAR -----


int progress = map(currentMillis - measureStartTime, 0, measureDuration, 0, 128);

display.clearDisplay();

display.setTextSize(1);

display.setCursor(0, 0);

display.println("Measuring...");

display.drawRect(0, 20, 128, 10, WHITE);

display.fillRect(0, 20, progress, 10, WHITE);

display.display();


//


// ----- BPM CALCULATION -----


if (currentMillis - measureStartTime >= measureDuration) {

    long sum = 0;

```

```

for (int i = 0; i < beatIndex; i++) sum += beatIntervals[i];

if (beatIndex > 0) {
    long avgInterval = sum / beatIndex;
    bpm = 60000 / avgInterval;
}

display.clearDisplay();
display.setTextSize(2);
display.setCursor(0, 0);
display.println("Heart Rate");
display.setCursor(0, 30);
display.print("BPM: ");
display.print(bpm);
display.display();

ThingSpeak.writeField(channelID, 1, bpm, writeAPIKey);

Serial.print("Stable BPM: ");
Serial.println(bpm);

measurementDone = true;
restartTimer = currentMillis;

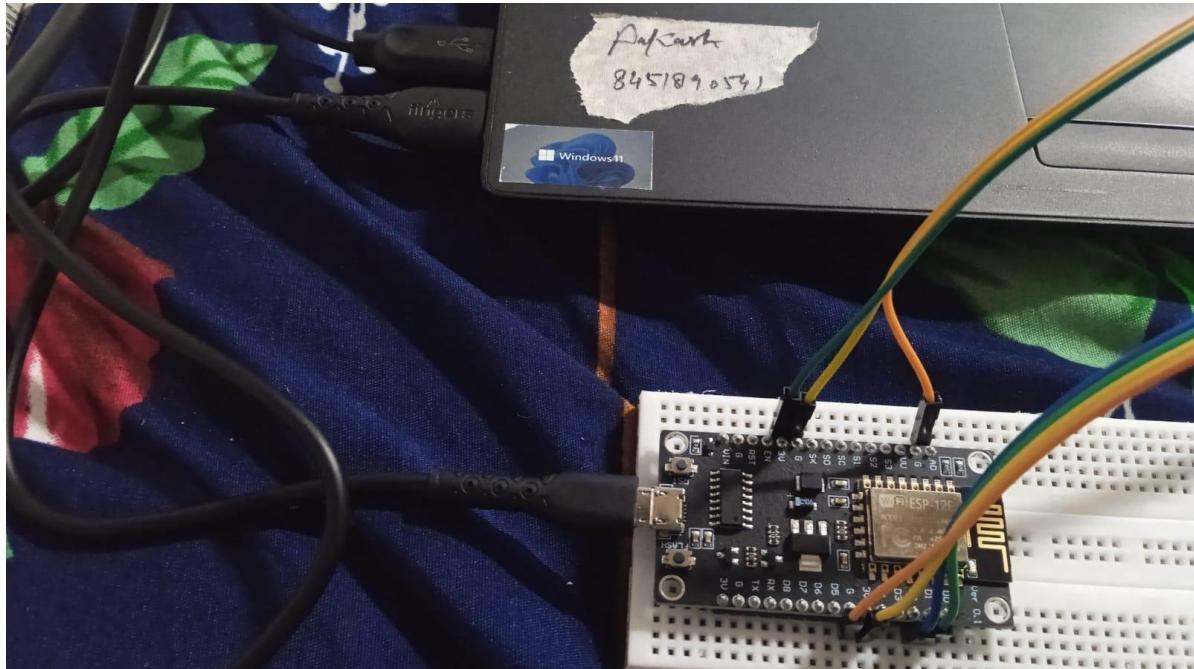
}

```

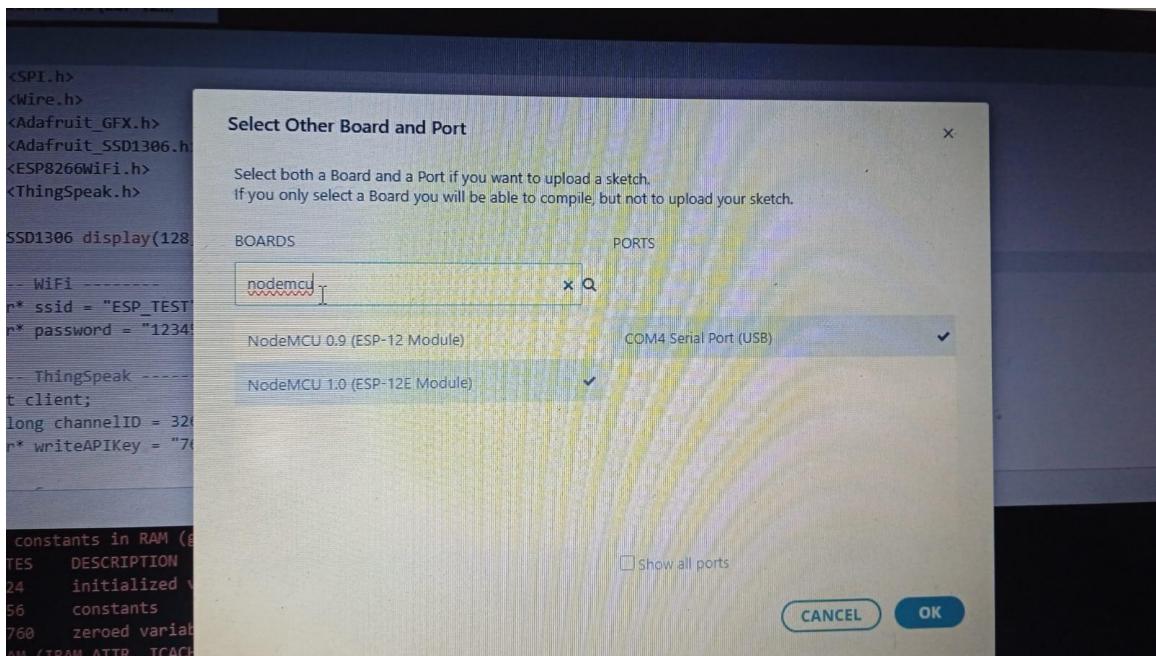
SYSTEM TESTING AND RESULT

Step 6 :

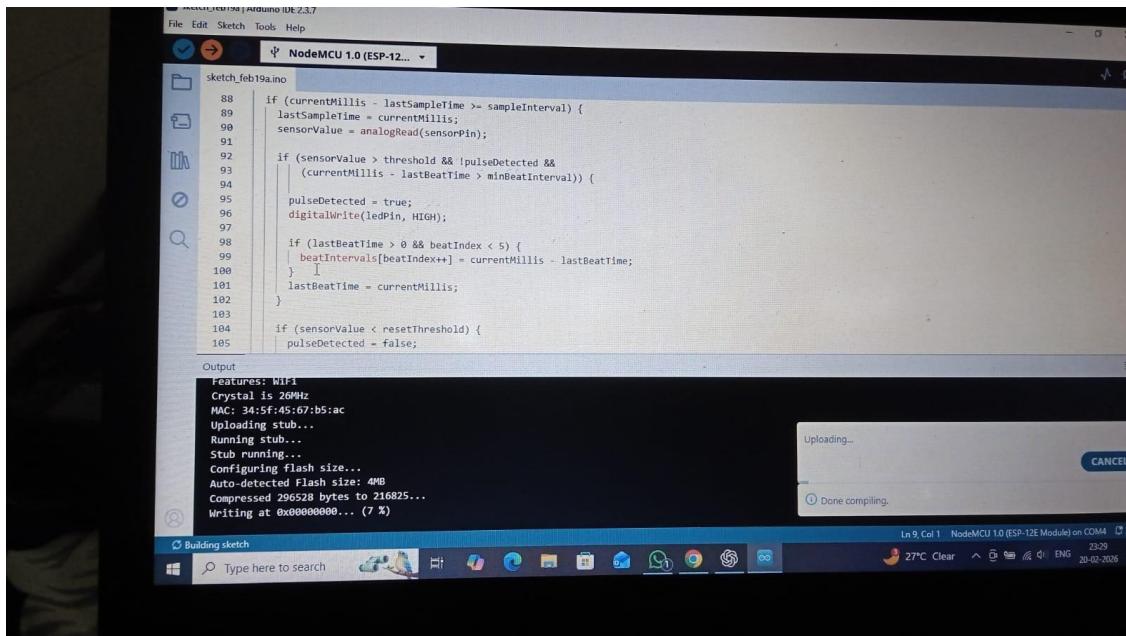
Connect Datacable from Esp8266 to laptop for code uploading.



Step 7:Open Arduino IDE, and select Board and Ports, after that type the code on Arduino IDE

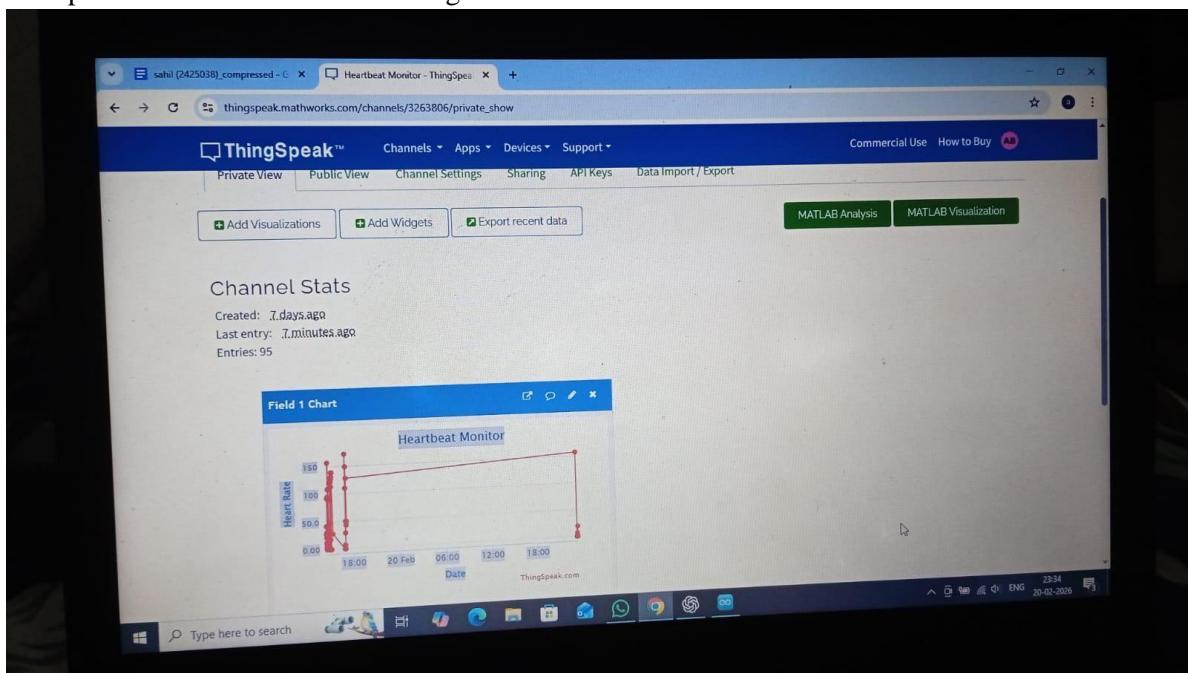


Step 8:Upload the code

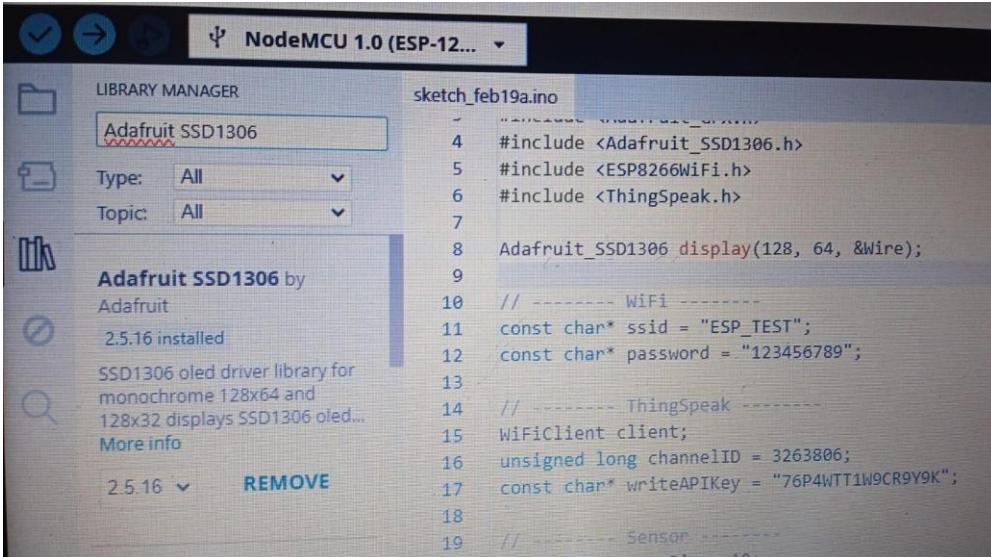


Step 9:

Create an channel on THINKSPEAK .ThingSpeak is an **IoT cloud platform** used to collect, store, and visualize sensor data in real time. In the **IoT Based Heart Rate Monitoring System**, ThingSpeak plays an important role in remote monitoring.



Step 9: Install Libraries in Arduino IDE

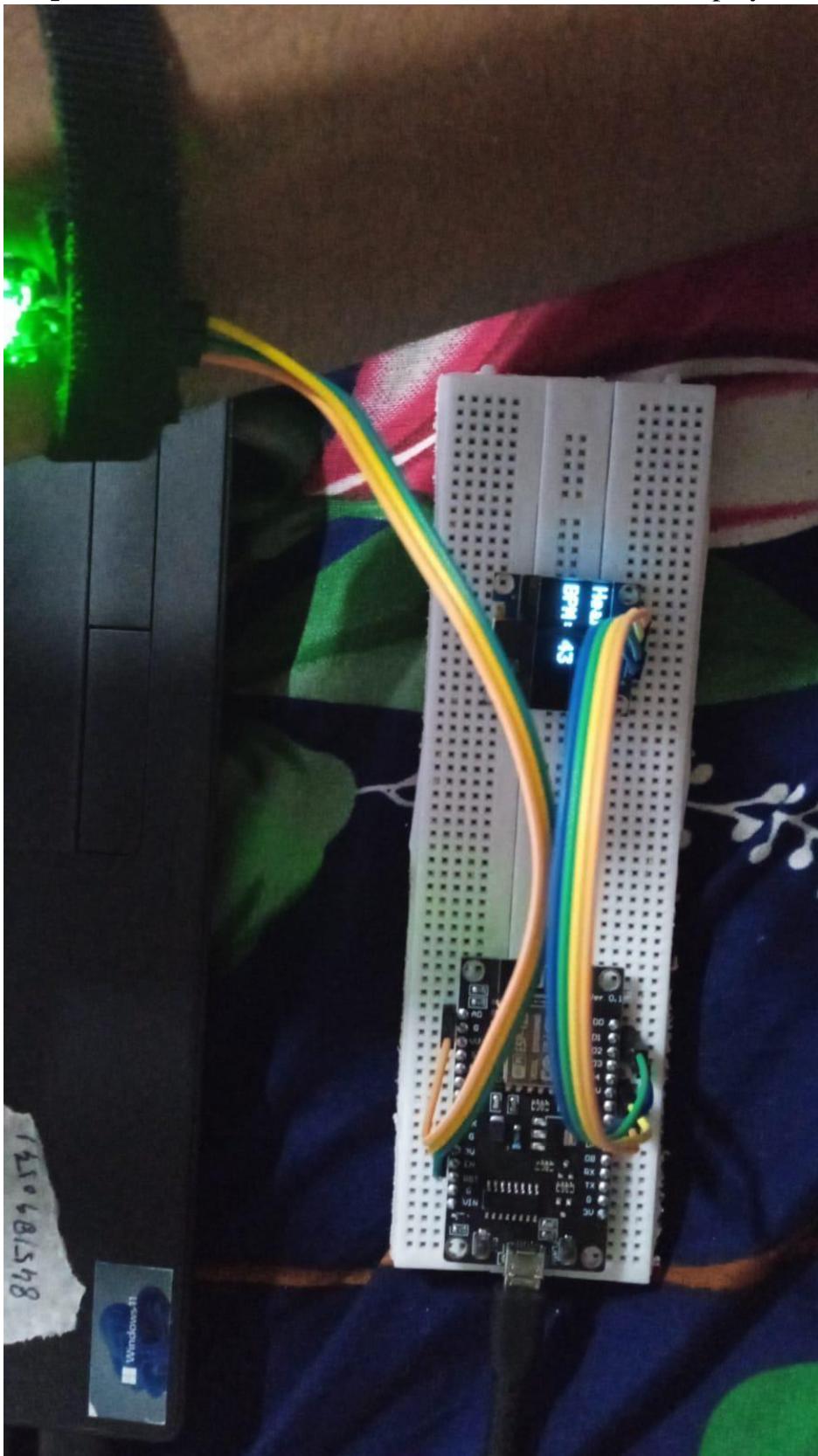


The screenshot shows the Arduino IDE Library Manager. In the search bar, 'Adafruit SSD1306' is typed. The results show the 'Adafruit SSD1306' library by Adafruit, version 2.5.16, which is installed. The library description states it's an SSD1306 oled driver library for monochrome 128x64 and 128x32 displays. Below the search bar, the code editor shows a sketch named 'sketch_feb19a.ino' with the following code:

```

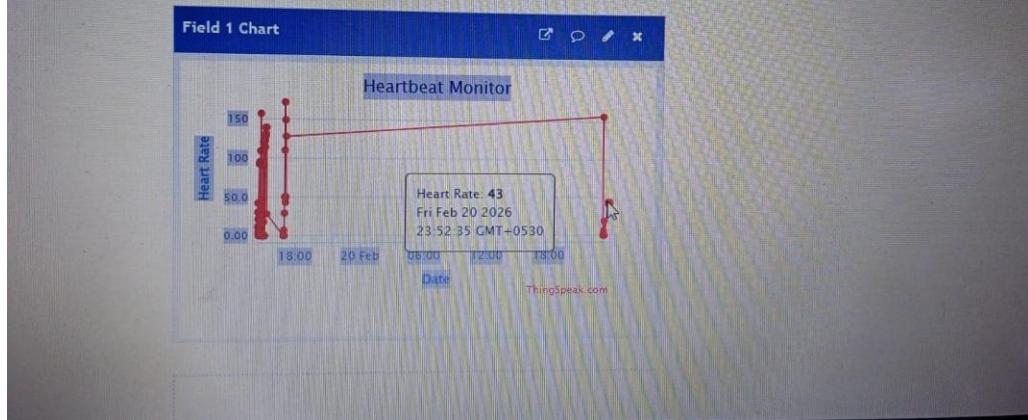
1 //include <Adafruit_GFX.h>
2 #include <Adafruit_SSD1306.h>
3 #include <ESP8266WiFi.h>
4 #include <ThingSpeak.h>
5
6 Adafruit_SSD1306 display(128, 64, &Wire);
7
8 // ----- WiFi -----
9 const char* ssid = "ESP_TEST";
10 const char* password = "123456789";
11
12 // ----- ThingSpeak -----
13 WiFiClient client;
14 unsigned long channelID = 3263806;
15 const char* writeAPIKey = "76P4WTT1W9CR9Y9K";
16
17 // ----- Sensor -----
18 const int sensorPin = A0;
19
20 const int ledPin = D8;
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
287
288
289
289
290
291
292
293
294
295
296
297
297
298
299
299
300
301
302
303
304
305
305
306
307
307
308
309
309
310
311
311
312
312
313
313
314
314
315
315
316
316
317
317
318
318
319
319
320
320
321
321
322
322
323
323
324
324
325
325
326
326
327
327
328
328
329
329
330
330
331
331
332
332
333
333
334
334
335
335
336
336
337
337
338
338
339
339
340
340
341
341
342
342
343
343
344
344
345
345
346
346
347
347
348
348
349
349
350
350
351
351
352
352
353
353
354
354
355
355
356
356
357
357
358
358
359
359
360
360
361
361
362
362
363
363
364
364
365
365
366
366
367
367
368
368
369
369
370
370
371
371
372
372
373
373
374
374
375
375
376
376
377
377
378
378
379
379
380
380
381
381
382
382
383
383
384
384
385
385
386
386
387
387
388
388
389
389
390
390
391
391
392
392
393
393
394
394
395
395
396
396
397
397
398
398
399
399
400
400
401
401
402
402
403
403
404
404
405
405
406
406
407
407
408
408
409
409
410
410
411
411
412
412
413
413
414
414
415
415
416
416
417
417
418
418
419
419
420
420
421
421
422
422
423
423
424
424
425
425
426
426
427
427
428
428
429
429
430
430
431
431
432
432
433
433
434
434
435
435
436
436
437
437
438
438
439
439
440
440
441
441
442
442
443
443
444
444
445
445
446
446
447
447
448
448
449
449
450
450
451
451
452
452
453
453
454
454
455
455
456
456
457
457
458
458
459
459
460
460
461
461
462
462
463
463
464
464
465
465
466
466
467
467
468
468
469
469
470
470
471
471
472
472
473
473
474
474
475
475
476
476
477
477
478
478
479
479
480
480
481
481
482
482
483
483
484
484
485
485
486
486
487
487
488
488
489
489
490
490
491
491
492
492
493
493
494
494
495
495
496
496
497
497
498
498
499
499
500
500
501
501
502
502
503
503
504
504
505
505
506
506
507
507
508
508
509
509
510
510
511
511
512
512
513
513
514
514
515
515
516
516
517
517
518
518
519
519
520
520
521
521
522
522
523
523
524
524
525
525
526
526
527
527
528
528
529
529
530
530
531
531
532
532
533
533
534
534
535
535
536
536
537
537
538
538
539
539
540
540
541
541
542
542
543
543
544
544
545
545
546
546
547
547
548
548
549
549
550
550
551
551
552
552
553
553
554
554
555
555
556
556
557
557
558
558
559
559
560
560
561
561
562
562
563
563
564
564
565
565
566
566
567
567
568
568
569
569
570
570
571
571
572
572
573
573
574
574
575
575
576
576
577
577
578
578
579
579
580
580
581
581
582
582
583
583
584
584
585
585
586
586
587
587
588
588
589
589
590
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
600
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
```

Step 10: Now run the code and see the result on OLED display and THINKSPEAK



Channel Stats

Created: 7.days.ago
Last entry: less.than.a.minute.ago
Entries: 96



FUTURE SCOPE AND CONCLUSION

Future Scope :

The **IoT Based Heart Rate Monitoring System** provides a strong base for future improvements and advanced health monitoring applications. Several enhancements can be implemented to improve accuracy, usability, and functionality.

Future improvements could include :

Advanced Sensors :

More accurate biomedical sensors such as SpO₂ sensors, temperature sensors, or ECG sensors can be added to monitor multiple health parameters along with heart rate.

Mobile Application Integration :

A dedicated mobile application can be developed to display real-time heart rate data, alerts, and historical records for better user experience.

AI and Data Analytics :

Machine learning algorithms can be used to analyze heart rate patterns and detect abnormalities such as irregular heartbeat or stress conditions.

Real-Time Alerts :

The system can be enhanced to send instant alerts via SMS, email, or app notifications in case of abnormal BPM values.

Wearable Design :

The system can be converted into a wearable device such as a smart band or wristwatch for continuous health monitoring.

Healthcare Integration :

The collected data can be shared directly with hospitals or doctors, enabling remote patient monitoring and telemedicine applications.

Conclusion :

The **IoT Based Heart Rate Monitoring System** successfully achieves its objective of measuring and monitoring heart rate in real time. The system detects heartbeats using a pulse sensor, displays BPM values on an OLED screen, and uploads the data to the ThingSpeak cloud platform for remote monitoring.

This project demonstrates the effective use of IoT technology in healthcare-related applications and provides valuable hands-on experience in embedded systems, sensors, and cloud-based data monitoring. It is especially useful for **students** to understand the working of biomedical sensors, IoT communication, and real-time data visualization.

It is important to note that this project is developed for educational, testing, and demonstration purposes only. The system is not intended for professional medical diagnosis or real-life clinical use, as it does not meet medical-grade accuracy and certification standards.

Overall, due to its simplicity, low cost, and expandability, this project serves as an excellent learning platform for students and beginners. With further improvements and medical-grade components, it can be extended into a more advanced health monitoring system.

REFERENCES

References :

1. **YouTube Reference –**
Project reference and implementation guidance taken from a YouTube tutorial. URL:
<https://youtu.be/u-gq7wABn9Q?si=jdlq0EduSs7s5vBf>
2. **Arduino IDE –**
Software used for writing, compiling, and uploading the program to the ESP8266 board. URL:
<https://www.arduino.cc/en/software>
3. **Pulse Sensor and Circuit Connections –**
Information related to the pulse sensor and hardware connections was referred from online resources and Google search.
4. **Mentor Guidance –**
Guidance and support were provided by **Sahil Sir** during the design, development, and testing phases of the project.
5. **Components Purchased –**
ESP8266 (NodeMCU), pulse sensor, OLED display, LEDs, and connecting wires were purchased from local electronics stores.

GLOSSARY

Glossary :

ESP8266 (NodeMCU):

A low-cost WiFi-enabled microcontroller used as the main controller in this project. It reads sensor data, processes heart rate values, and uploads data to the cloud.

Pulse Sensor:

A biomedical sensor used to detect heartbeats by measuring changes in blood flow. It provides analog signals that are used to calculate heart rate in beats per minute (BPM).

OLED Display (SSD1306):

A small organic light-emitting diode display used to show real-time information such as measuring status and heart rate values.

ThingSpeak:

An IoT cloud platform used to store, visualize, and analyze heart rate data uploaded through the internet.

WiFi:

A wireless networking technology that allows the ESP8266 to connect to the internet and send data to the cloud platform.

LED Indicator:

A light-emitting diode used to indicate heartbeat detection by blinking whenever a pulse is detected.

Arduino IDE:

An Integrated Development Environment used to write, compile, and upload programs to the ESP8266 microcontroller.

Analog Pin (A0):

A pin on the ESP8266 used to read analog signals from the pulse sensor.

BPM (Beats Per Minute):

A unit used to measure heart rate, representing the number of heartbeats in one minute.

IoT (Internet of Things):

A technology that enables devices to connect to the internet and exchange data without human intervention.

Threshold Value:

A predefined sensor value used to detect valid heartbeats and avoid false readings.

Cloud Platform:

An online service used to store and access sensor data remotely, allowing real-time monitoring and analysis.