Indian Institute OF Technology, Ropar

GE – 109

TINKERING LAB PROJECT

(Write – Up)

Arduino - Based Heart Rate Monitoring System

Submitted By -: G02_Monday

Group Members -:

Name -:	*Entry Number-:*
1) Honey Garg	***2023EEB1207***
2) Chirag Tayal	***2023EEB1195***
3) Jaidev Gatla	***2023EEB1200***
4) Harshit Gupta	***2023EEB1205***
5) Arnav Bhandari	***2023EEB1061***

Objective-:

Arduino - Based Heart Rate Monitoring System

To measure and display heart rate using an Arduino and a low-cost pulse sensor. To help understand. bio-signal acquisition, real-time data processing, and visualization using Arduino.

Components Used -:

1. *Microcontroller – ESP32*
2. *USB Cable (USB to B-type and data transferrable)*
3. *Power Source – 9V batteries * 3, with Battery Snap Connector*
4. *Buck Module DC-DC 9V to 5 & 3.3V * 2*
5. *OLED Display (Driver IC: SSD1306, Resolution: 128 x 64)*
6. *MAX30100 Pulse Oximeter Sensor*
7. *Pulse Sensor Amped*
8. *Bread-Board (400 Tie Points)*
9. *Header Pins*
10. *Soldering Iron and Solder Wire*
11. *Jumper Wires (Male-Male, Male-Female, Female-Female)*
12. *Resistors - 220 ohms *3*
13. *RGB led (2)*
14. *Buzzer (2)*
15. *Galvanic Skin Response (GSR) (or any other similar product - need not be the same brach and product code)*

Brand: **SeeedStudio**

Product Code: 13133)

Functionality Of the Project -:

Step 1:
Connect the Arduino to a power source via USB. Attach the heart rate sensor to the Arduino using the 5V and GND pins.

Step 2:
Place a finger on the heart rate sensor. The LED on the sensor emits light, and the photodiode (optional) detects changes in light absorption caused by blood flow variations.

Step 3:
The sensor amplifies and filters the weak pulse signal to reduce noise. The processed signal is then sent to the analog input pin (A0) of the Arduino.

Step 4:
The Arduino reads this analog voltage and converts it into a digital signal using the ADC (Analog-to-Digital Converter).

Step 5:
The Arduino detects peak values in the signal, where each peak represents a heartbeat. It calculates the time difference between two consecutive peaks to determine the Beats Per Minute (BPM).

Step 6:
Finally, the BPM is displayed on an LCD screen.

Step 7:

Stress Detection with Galvanic Skin Response (GSR) – Combine heart rate and GSR sensor readings to detect stress levels.

Possible Future Enhancements -:

1.) Wearable Design: Create a smart wearable device (like a smartwatch) using an ESP32, Li-Po battery, and flexible PCB.
2.) Wi-Fi (ESP8266 or ESP32): Sends real-time heart rate data to cloud servers for online monitoring the heart beat of a person and sending the alert to Known ones and location of the person
3.) Implement Machine Learning (ML) to detect heart rate patterns and predict potential heart conditions.

Conclusion -:

The Arduino-based Heart Monitoring System provides an efficient way to track heart rate in real time. It is a cost-effective solution for personal and medical applications. With further enhancements, this system can become a vital part of wearable health technology