

ECE 1000 Final Report: Heartbeat Sensor

Jack Massengille, Anthony Pfaff
Department of Electrical and Computer Engineering
Cookeville, TN, USA
jtmassengi42@tntech.edu, ajpfaff42@tntech.edu

Abstract- We chose the heartbeat sensor because it seemed like a cool and interesting challenge, and we wanted to see how well we could design and build it. The sensor works by detecting raw pulse data when a user places their finger on the reader and then displays the corresponding BPM on an OLED screen. While the sensor was a little finicky and required precise placement for accurate readings, the process of coding, building the circuit, and troubleshooting was an enjoyable and rewarding experience.

I. Introduction

Jack Massengille Electrical Engineering major and Anthony Pfaff Computer Engineering and the project that we have been working on is a heartbeat sensor. Where you put your finger on the beat reader and it reads out its raw data then prints the BPMs on the OLED display. We thought this project had importance because heartbeat sensor are needed everywhere in a hospital although ours is not as accurate it was very fun to see how it actually works.

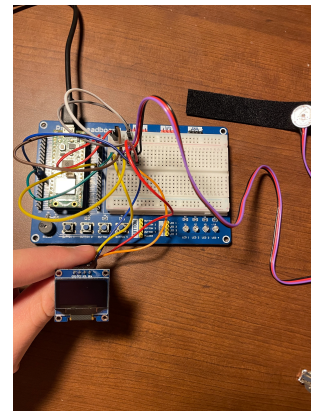
II. Background

While designing our project we had a couple of sources help us along the way. One source was a website called peppe80.com. This website had a project just like ours and also had circuit and codes to help test your sensor with a excel sheet that showed graphs when pressed vs. empty. We also used Chat GPT to help form our main code and to help with further smoothing and averaging to help our finicky sensor get better results.

III. Project Description and Formulation

This project works by having the user place their finger over the light sensor which then reads a line of raw data every ms. The circuit for this project was very simple we just connected the sensor and OLED screen up to the raspberry pi

pico using the breadboard and jumper wires. The code is where things got complicated the first code we did was a sensor test to see if our sensor would give consistent results when pressed vs. empty. It did not. However we did some further research and realized we had to code the moving average and get a baseline and threshold before every run of the sensor. We did a lot of troubleshooting testing different offset values and amount of readings to make the BPMs smoother. Although it is definitely not the smoothest reading, we were happy with the results.



VI. Discussion and Results

The result of a project was a heartbeat sensor that read the raw data and then print a correspondent BPM however our sensor was very finicky and was giving us very inconsistent raw data when pressed which made the information less accurate it will give a right reading if you have pressure and placement perfect. If we were to continue working on this project, we would have smoothed out the averaging and smoothing of the code and ordered a different sensor as well. Thing we enjoyed most about the project is just seeing it all come together at the end. Jack completed the building of the circuit and the lab report and

V. Conclusion

This was a very fun project and we thoroughly enjoyed making it and developing the code. It was very rewarding to see the final product although we wished it was a little bit more accurate we were still happy with how it turned out.

Jack and Anthony did the coding and created the GitHub.

References

[1] Inspiration for github layouts <https://github.com/JackBender42/ECE-1000-Final-Project-Spring-2024>

<https://github.com/JCWilliams1003/ECE-1000-Spring-2024-Final-Project-Insert-Project-Name>

[2] <https://peppe80.com/pulse-sensor-with-raspberry-pi-pico-hearth-beat-check-with-micropython/>

[3] Chat GPT [ChatGPT \(openai.com\)](https://openai.com)