Team name: Besart Gashi, Konsta Hovivuori, Lauri Karhu

VitalTrack

Project Report

First-year Hardware Project

School of ICT

Metropolia University of Applied Sciences

12 May 2024

Abstract

This project involved the development of "VitalTrack," a portable, user-friendly device for monitoring heart rate (HR) and heart rate variability (HRV). The project was completed by Besart Gashi, Konsta Hovivuori, and Lauri Karhu as part of the first-year hardware project for the School of ICT at Metropolia University of Applied Sciences.

The primary objective of VitalTrack was to integrate heart health monitoring into daily life without the complexity of traditional medical devices, aiming for real-time, accessible health insights. Using a Raspberry Pi Pico, an OLED display, and a Crowtail pulse sensor v2.0, the device offers functionalities such as continuous HR monitoring, HRV analysis, and data transmission over MQTT for further analysis. VitalTrack enables the display of heart rate data every 5 seconds and performs HRV analysis to provide insights into the user's stress levels and overall cardiovascular health.

The development process included designing a user interface on the OLED display with menu options for HR and HRV metrics, testing in various conditions to ensure accuracy and reliability, and implementing features to connect to Wi-Fi and transmit data for advanced analysis via MQTT and Kubios Cloud.

Overall, VitalTrack successfully met the project criteria for grade 1-2, achieving a blend of technical reliability, ease of use, and accessibility. Future work could expand its features and improve data management and analysis capabilities. This project underscores the potential of portable health monitoring devices in enhancing individual health management.

Version history

Ver	Description	Date	Author(s)
1.0	Introduction, Mid Summary	29.4.2024	Konsta, Besart, Lauri
1.1	Theoretical Background, Methods and Material	1.5.2024	Konsta, Besart, Lauri
1.2	Implementation and adding the abbreviations and acronyms list	3.5.2024	Konsta, Besart, Lauri
1.3	Group Work Summary, some cleaning up and editing to previous versions.	5.5.2024	Konsta, Besart, Lauri
1.4	Conclusions, Abstract	6.5.2024	Konsta, Besart, Lauri
1.5	Adding more text to theoretical background and pictures to Methods and Material	10.5.2024	Konsta, Besart, Lauri
1.6	Editing text and adding references	11.5.2024	Konsta, Besart, Lauri
1.7	Final editing for the project report and preparing for the return	12.5.2024	Konsta, Besart, Lauri

Contents

Abbreviations and acronyms

HRV	Heart Rate Variability is the variation in the time intervals between heartbeats
HR	Heart Rate refers to the number of heartbeats per minute
OLED	Organic Light Emitting Diode is a display technology that uses organic compounds to emit light when electrified
MQTT	Message Queuing Telemetry Transport is a messaging protocol designed for minimal bandwidth and device resource usage
PPG	Photoplethysmography uses light-based technology to detect blood volume changes in the microvascular bed of tissue
I2C	Inter-Integrated Circuit
PPI	Peak to Peak Interval is the time measured between the peaks of a waveform
SDNN	Standard Deviation of NN Intervals represents overall variability in heart rate over a specified period
RMSSD	Root Mean Square of Successive Differences is used to quantify the variability between successive heartbeat intervals, reflecting the parasympathetic activity of the heart
FIFO	First In, First Out is a type of data structure or queue where the first element added to the queue is the first one to be removed
ADC	Analog-to-Digital Converter is a device or a circuit that converts analog signals into digital data that computers can process

1 Introduction

1.1 Topic and Purpose

The project for the Hardware 2 course involves developing a heart rate and heart rate variability (HRV) monitoring device. This project's purpose is to bring heart health monitoring into everyday life without the complexity of traditional medical devices. We are focused on making a device that is small, effective, and easy to use, so you can get real-time updates on your heart’s health whenever you need them.

1.2 Motivation

Our motivation for this project comes from an increasing interest in maintaining health in our daily lives. With this device, you can consistently monitor your heart rate and its variability. Reliable tracking of these metrics is key, not just for athletes, but for anyone interested in their cardiovascular health. The device can offer valuable insights, helping you spot potential early signs of stress or cardiovascular issues

1.3 Goals

1.3.1 Functionality

Develop a device that accurately measures and displays the heart rate and HRV data on an OLED screen. The device will be able to update heart rate data every 5 seconds and provide a HRV analysis.

1.3.2 User Interface

Implement a user-friendly interface on the OLED display, which includes multiple menu options for displaying heart rate data and HRV analysis, as well as historical data storage and visualization capabilities.

1.3.3 Connectivity and Quality

Ensure the device can connect to a Wi-Fi network and communicate with other devices via MQTT, enabling data sharing and analysis. Our goal is to make sure that the heart rate measurements are both accurate and reliable, matching up to the standards you would expect from a traditional heart rate monitoring device.

2 Theoretical Background

2.1 Heart Rate and Heart Rate Variability

Heart rate (HR), measured in beats per minute (bpm), is the frequency at which the heart beats in a given time frame, usually a minute. Heart rate variability (HRV) measures the time variation between heartbeats, in milliseconds (ms). HRV is an indicator of the nervous system's influence on the heart, providing insights into stress, relaxation, and cardiac coherence.

2.2 Physiological Basis

Heart rate and HRV are influenced by various different factors such as age, fitness, stress, and underlying health conditions. A typical resting heart rate for adults ranges from 60 to 100 bpm. Athletes may have lower resting heart rates, sometimes below 60 bpm. HRV can vary significantly between individuals and is not confined to a specific "normal" range, but higher HRV generally suggests better health and fitness.

2.3 Sensor Technology

In this project, we use a photoplethysmography (PPG) sensor, technology that detects blood volume changes in the microvascular bed of tissue. This sensor emits a light that penetrates the skin and is absorbed by the blood. By measuring the changes in light absorption due to blood flow, the PPG sensor provides a signal that can be analyzed to determine heart rate.

The readings are read with the use of an Analog-to-Digital Converter (ADC) and stored in a First-In-First-Out (FIFO) buffer for processing. We used functions to calculate the average peak-to-peak interval (PPI), from which the heart rate and HRV are derived.

3 Methods and Material

3.1 Hardware

In this project we used a Crowtail Pulse Sensor v2.0. This sensor operates by photoplethysmography (PPG), which measures the blood volume changes within the microvascular bed of tissue via light absorption. [1]

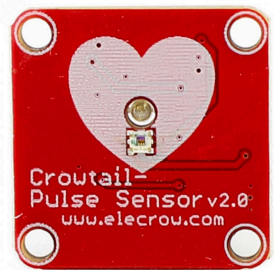

Figure 1: Crowtail Pulse Sensor v2.0 [1]

We utilized an OLED display that is 128x64 wide and connects through I2C communication. This display shows high-contrast images while using little power. [2]

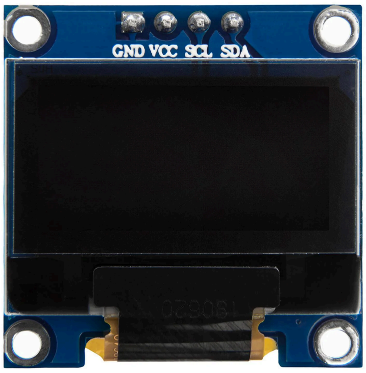

Figure 2: SSD1306 OLED display [2]

The Raspberry Pi Pico serves as the brains of our operation. This microcontroller supports USB connectivity. It is able to handle all our data processing and communication needs. The board also includes a rotator which is used to communicate with the code. [3]

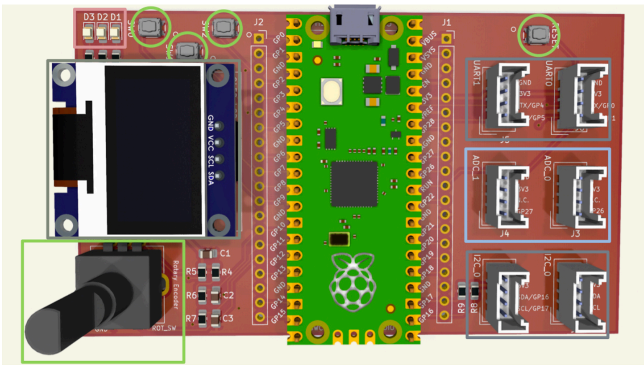

Figure 3: Raspberry Pi Pico model [3]

3.2 Software

For the programming, we used Thonny IDE as the platform and MicroPython as the language. Thonny is designed for embedded systems, facilitating the download of MicroPython firmware onto the Raspberry Pi Pico.

MicroPython, while having fewer libraries than standard Python, offers more options for embedded systems. For instance, the ssd1306 library, built on MicroPython's FrameBuffer library, manages the OLED display. [4]

Additionally, two files, fifo and piotimer are essential to this project. The fifo script ensures data storage without interruptions, while piotimer includes a class designed to handle hardware timing. [4]

```
from piotimer import Piotimer as Timer
from ssd1306 import SSD1306_I2C
from machine import Pin, ADC, I2C, PWM
from fifo import Fifo
import utime
import array
import time
```

Figure 4: Libraries used [4]

3.3 System Testing

The testing for the crowtail pulse sensor were carried out in different environments, including noisy, well-lit, dark, and crowded settings. We also examined how the sensor performs when placed in different parts of the body.

The OLED display was tested for its ability to clearly visualize data in real-time, again in different settings.

The microcontroller was tested on different computers to ensure it was recognized and that the results are accurate

4 Implementation

The VitalTrack device is a comprehensive system designed to measure and analyze heart rate (HR) and heart rate variability (HRV) using the Raspberry Pi Pico as the central processing unit. The implementation of VitalTrack involved both hardware setup and software development to ensure seamless integration and functionality.

4.1 Hardware configuration

The core of our hardware setup includes the Raspberry Pi Pico microcontroller, which orchestrates all operations of the device. The heart rate data is collected via a Crowtail Pulse Sensor v2.0, which utilizes photoplethysmography (PPG) technology. This sensor is sensitive enough to detect blood volume changes in the microvascular bed of the tissue at the fingertip. Data display is managed through an OLED display connected via I2C communication protocol, allowing real-time monitoring of HR and HRV metrics visually.

For user interaction, a simple interface consisting of a rotary encoder allows users to navigate through menu options on the OLED screen, which includes heart rate display, HRV analysis, and historical data review.

4.2 Software Development

The device's software was developed using MicroPython, an efficient choice for running on embedded systems like the Raspberry Pi Pico. The main program initializes the system components and manages the device's operational modes based on user input received through the rotary encoder.

4.3 Algorithm

The heart rate measurement algorithm involves analyzing the PPG signal obtained from the pulse sensor to detect beat-to-beat intervals. These intervals are then converted to beats per minute (BPM) for display. For HRV analysis, we implemented time-domain methods to calculate metrics such as RMSSD (Root Mean Square of the Successive Differences) and SDNN (Standard Deviation of NN intervals), which are critical for assessing autonomic nervous system activity.

The HRV analysis algorithm captures at least 30 seconds of heartbeat data to ensure accuracy in variability assessment. This data is processed to extract meaningful HRV statistics that provide insights into the user's stress and health status.

4.4 System testing and Validation

Thorough testing was conducted to validate the accuracy of the HR and HRV measurements. The device was tested under various environmental conditions and compared against a standard pulse oximeter to ensure the heart rate readings fell within the acceptable error margin of ±10%. Additionally, the system's software stability and response times were tested extensively to guarantee that the device could operate reliably in real-time use.

5 Group Work Summary

5.1 Midway Summary

Konsta Hovivuori: Created the project repository in gitlab and programming.

Lauri Karhu: Background research and programming.

Besart Gashi: Writing the project report and programming.

Plans for next week: More programming, researching and learning how to "refine" the code to function better.

Problems: at first, displaying info issues on the OLED screen.

5.2 Final Summary

Throughout the project, each team member played an important role to reach the necessary.

Konsta Hovivuori focused on developing the algorithm and trying different methods to enhance its functionality. He also contributed to the project report and presentation.

Besart Gashi did the programming of our functions, and helped with the development of the algorithm. He also contributed to the project report and presentation.

Lauri Karhu did background research to make sure our project had the correct libraries and how to use them. He also played a role in cleaning up and finalizing the code and contributed to the project report and presentation.

5.3 Evaluation

Our team work was efficient, maintaining open communication and ideas. We faced challenges, particularly with programming and managing our time effectively. Despite the late start we managed to develop a functional heart rate and HRV monitoring device.

However, there were areas where we could have improved. The Wi-Fi connectivity and MQTT messaging were not implemented, which were in our project goals. This shortfall was due to our limited experience with these technologies.

6 Conclusions

Overall the project could have gone better. The project goals were modestly met. We succeeded in developing a functional device capable of monitoring heart rate and HRV and displaying this data on an OLED screen. The device's ability to connect via Wi-Fi and send MQTT messages was not implemented due to lack of skill and time.

Our project faced some challenges due to late start and difficulties to find time to do the project as a team. Programming was difficult and we started it from scratch multiple times. At the class we got some base programming done but in the end every team member completed different parts of the code at home.

Our prototype's functionality was limited to heart rate measurement and HRV analysis. In the future we could continue the project by adding features like MQTT messaging and Kubios cloud data processing.

References

1. Crowtail pulse sensor v2.0. Accessed 10.5.2024

https://www.partco.fi/22872-thickbox_default/crowtailpulsesensor20.jpg

2. SSD1306 OLED display. Accessed 10.5.2024

https://cdn.shopify.com/s/files/1/1509/1638/products/096-zoll-oled-ssd1306-display-i2c-128-x-64-pixel-kompatibel-mit-arduino-und-raspberry-pi-562312.jpg?v=1679397959

3. Raspberry Pi Pico. Accessed 10.5.2024

https://oma.metropolia.fi/delegate/download_workspace_attachment/9627536/Pico_Devboard_Pinout.pdf