Group 3: Ade Aiho, Topias Aho, Jonne Roponen

Cardio Pulse PRO

Project Report

First-year Hardware Project

School of ICT

Metropolia University of Applied Sciences

12 May 2024

In this document ChatGPT was used for phrasing and language refinement.

Abstract

This report details the development and implementation of the project device Cardio Pulse PRO. The heart rate detection and analysis system was designed by first-year students of group 3 as the hardware 2 project work at Metropolia University of Applied Sciences. The project aimed to create a system capable of measuring heart rate variability (HRV) using photoplethysmography (PPG) and then provide an understanding into stress and recovery levels. The methods used involved the integration of a Raspberry Pi Pico W microcontroller with a Crowtail Pulse Sensor v2.0, employing algorithms for accurate heart rate and HRV measurement. The results of the implementation showed the capability of the system to reliably detect and analyze heart rate data, which was confirmed against a patient monitor. The conclusions of the project confirm the potential of the system as a device for health-conscious users, while also emphasizing its educational value in introducing first-year engineering students to practical experience with health technology and embedded systems.

Version history

Ver	Description	Date	Author(s)
1.0	First iteration	12.5.2024	Ade Aiho, Topias Aho, Jonne Roponen
1.1			
1.2			
1.3			
2.0			

Contents

1 Introduction

The Hardware 2 project at Metropolia University of Applied Sciences involves the development and production of a heart rate detection and analysis system. The system is designed to measure heart rate variability (HRV) using photoplethysmography (PPG) and give an understanding into stress and recovery levels [1]. The device will be suitable for health-conscious general users at home.

This project introduces the group of first year ICT students to health technology and hardware engineering, making practical use of the theory learned in the Hardware 1 and Hardware 2 courses. The group will learn how to operate and program an embedded system, detect and calculate HRV, and establish a connection with a cloud service.

The goals for the project include developing a heart rate detection algorithm that operates locally on the device, calculating heart rate variability for analysis, establishing connection with Kubios Cloud HRV for more complete HRV analysis, and displaying the estimated stress and recovery status indicators on the device.

The project report is organized into sections as follows:

1. **Theoretical Background** provides an introduction to the pulse sensor and discusses key concepts such as heart rate and heart rate variability, what influences these measurements, and their normal values.
2. **Methods and Materials** covers how the project was carried out, detailing the components used, and how the system was tested and verified
3. **Implementation** describes the system architecture and the algorithms developed for data collection and processing.
4. **Group Work Summary** outlines the contributions of each group member.
5. **Conclusions** summarizes the results and potential implications of the project.

Glossary

ANS	Autonomous nervous system
APMHR	Age-predicted maximal heart rate
AWS	Amazon Web Services
BPM	Beats per minute
IBI	Inter-beat-interval, measured from PPG signal, unit in milliseconds (ms)
HR	Heart rate, generally unit in beats per minute (BPM)
HRV	Heart rate variability, measure of variability in consecutive heartbeats, indicated by many different parameters
Hz	Hertz, The SI unit of frequency, equal to one cycle per second.
LAN	Local area network
LED	Light-emitting diode
ms	millisecond
NN interval	Time interval between successive peaks in ECG or PPG signals, referred to as PPI or RRI, measured in milliseconds (ms)
NN50 count	Count of adjacent NN intervals that differ by more than 50 ms
OLED	Organic light-emitting diode
pNN50	Ratio of NN50 count to the total number of NN intervals
Poincaré plot	A type of recurrence plot used to quantify self-similarity in processes, usually periodic functions
PPI	Peak-to-peak interval: the time between two consecutive peaks, measured in milliseconds (ms)
PPG	Photoplethysmography, optical detection of heart signals from peripheral blood circulation
PNS	Parasympathetic nervous system, part of the autonomic nervous system
RMSSD	The square root of the sum of the squares of successive differences between NN intervals, measured in milliseconds (ms)

RRI	RR-interval, the time difference between two consecutive R-peaks in an ECG signal, measured in milliseconds (ms)
SD1	Poincaré plot index, the first ellipse shape parameter estimated from the Poincaré plot
SD2	Poincaré plot index, the second ellipse shape parameter estimated from the Poincaré plot
SDANN	Standard deviation of the mean NN intervals, measured in milliseconds (ms)
SDNN	Standard deviation of all NN intervals, measured in milliseconds (ms)
SI	Baevsky’s stress index
SNS	Sympathetic nervous system, part of autonomic nervous system
USB	Universal serial port
WiFi	Wireless fidelity

2 Theoretical Background

This section introduces the theoretical and technical background that the project is based on. It provides a basic understanding of cardiac functions, and technical specifications of the sensor utilized in the project. The first subsection discusses heart rate and heart rate variability, the second subsection explains the role of the autonomic nervous system and related HRV analysis methods, the third subsection details the sensor and its functional basis.

2.1 Heart Rate and Heart Rate Variability

The main tasks of the heart are to circulate blood throughout the body and to support other bodily functions. Heart rate variability (HRV) plays an important role in this. Multiple factors can change the rhythm of the heart, causing differences between consecutive heartbeats. The brain and the nervous system regulate the output of the heart through neural and hormonal pathways [2].

The way the heart operates is affected by age, physical activity, metabolic disorders, nutrition, pathological conditions, and changes in physiology [3]. Heart rate changes depending on physical, mental, and emotional conditions. This means that heart rate variability (HRV) can help measure how well the body reacts to different internal and external changes to stay balanced [4].

Heart rate (HR) is measured as the number of heartbeats per minute (BPM), while heart rate variability (HRV) is the time difference between consecutive heartbeats, measured in milliseconds (ms) [2]. Figure 1 shows the peak-to-peak intervals in milliseconds, HRV is the difference between successive beats.

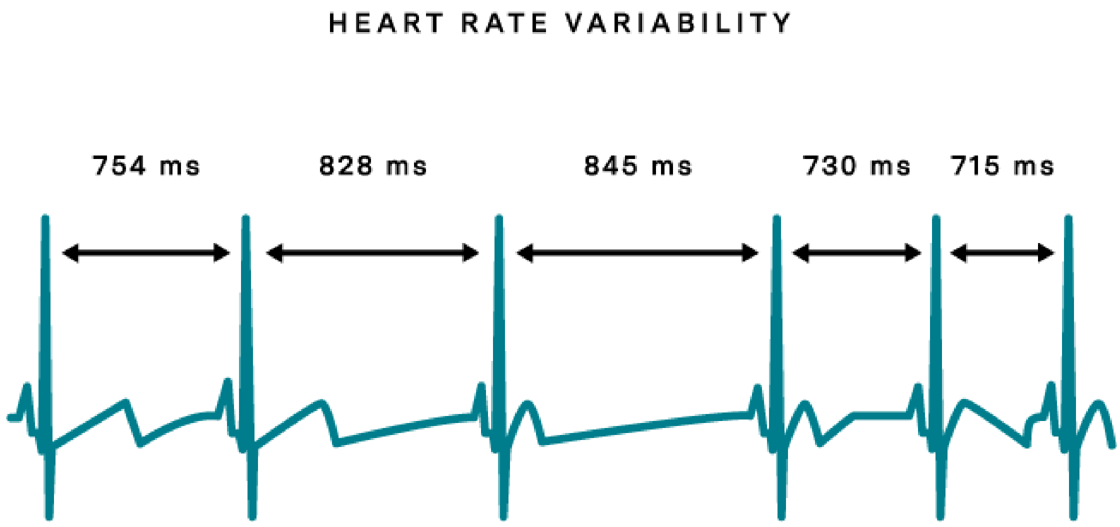

Figure 1: HRV is the time interval between consecutive heartbeats [5].

Research shows that the average HRV in healthy adults is 42 milliseconds, typically ranging from 19-75 milliseconds and goes as high as 120 milliseconds in athletes [2]. The average heart rate according to multiple studies as well as the expert consensus suggest that normal resting HR for adults ranges from 60 to 100 BPM [6]. Typical resting heart rates by age and sex are shown in figure 2. The maximum heart rate is determined by using the age-predicted maximal heart rate (APMHR), with the equation HR_{max} = 220-Age [7]. Figure 2 shows the average resting heart rate based on age and sex.

AGE	ATHLETE	VERY GOOD	ABOVE AVERAGE	AVERAGE	BELOW AVERAGE	POOR
20-39	47-54	55-60	61-68	69-75	76-83	84-94
40-59	46-54	55-60	61-67	68-76	77-84	85-94
60-79	45-53	54-59	60-66	67-74	75-83	84-97

AGE	ATHLETE	VERY GOOD	ABOVE AVERAGE	AVERAGE	BELOW AVERAGE	POOR
20-39	52-59	60-65	66-73	74-81	82-88	89-98
40-59	51-58	59-63	64-70	71-78	79-85	86-96
60-79	52-58	59-63	64-69	70-77	78-85	86-95

Figure 2: Resting heart rate by age and sex. Upper chart represents male and the lower female resting heart rate [8].

2.2 Autonomic Control of Heart Rate

The autonomic nervous system (ANS) helps control the heart rate through its two branches, the sympathetic nervous system (SNS) and the parasympathetic nervous system (PNS). They have the opposite effects on heart functions. SNS activation boosts heart rate, especially during stress and physical activity to increase blood flow. PNS activation reduces heart rate, helping the body conserve energy and recover during calm states [9]. Recovery and stress indexes that show the parasympathetic and sympathetic activity of the heart have been developed using common heart rate variability measures.

HRV analysis methods are used to calculate the HRV. The different time-domain HRV analysis methods include Mean RR, RMSSD, and SDNN. The methods used for analyzing heart rate variability (HRV) in the time-domain are based on the successive differences between heartbeats, known as RR intervals [10].

The mean RR interval can be calculated with the equation in figure 3. The mean RR interval is the average duration between consecutive heartbeats. N is the total number of RR intervals that is being measured [10].

$$\overline{RR} = \frac{1}{N}\sum_{i=1}^{N} RR_i$$

Figure 3: Formula for the mean RR interval [10].

The mean heart rate is calculated with the equation shown in figure 4. Since heart rate is the number of beats per minute, and the RR interval is the time between successive heartbeats, the mean HR is calculated by dividing the number of seconds in a minute with the mean RR [10].

$$\text{Mean HR} = \frac{60}{\overline{RR}}$$

Figure 4: Formula for the mean HR.

The variability within the RR time intervals in the time-domain can be measured using SDNN with the equation in figure 5. The summation inside the square root calculates the squared differences between each interval and the mean interval [10].

$$SDNN = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(RR_i - \overline{RR})^2}$$

Figure 5: Formula for the SDNN, standard deviation of the RR intervals [10].

Root mean square of successive differences (RMSSD), is used to measure the variability in the time interval between heartbeats. The portion $(RR_{i+1} - RR_i)^2$ calculates the squared differences between successive RR intervals [10].

$$RMSSD = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N-1}(RR_{i+1} - RR_i)^2}$$

Figure 6: Formula for RMSSD, root mean square of successive differences [10].

The PNS index is calculated using the mean RR interval, the RMSSD (Root mean square of successive RR interval differences), and the normalized SD1 from the Poincaré plot. Higher values indicate stronger parasympathetic activation. This index is compared to average values to see how it deviates from the normal population [10][11]. Figure 3 shows the PNS index that is formed using the previously mentioned calculations.

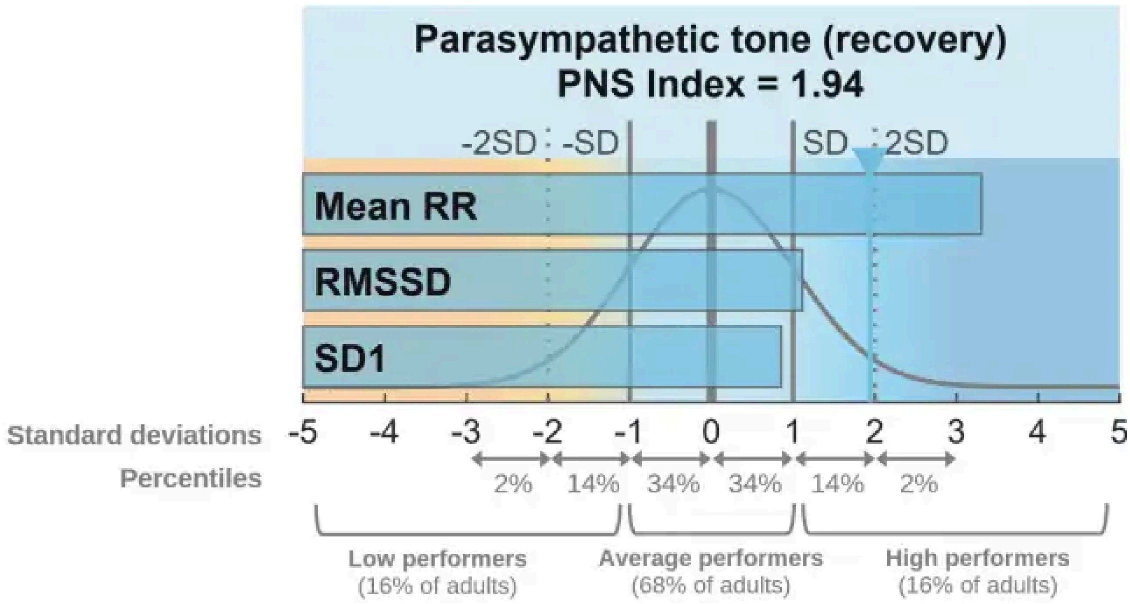

Figure 3: Parasympathetic nervous system index [10].

The SNS index assesses the activity of the sympathetic nervous system. It uses the average HR interval, Baevsky's stress index (SI), and the normalized SD2 from the Poincaré plot to measure sympathetic influence. Higher values indicate stronger sympathetic activation. Similar to the PNS index it is compared to the average values to see how it deviates from the normal population [10][11]. Figure 4 shows the SNS index that is formed using the previously mentioned calculation, measurement, and index.

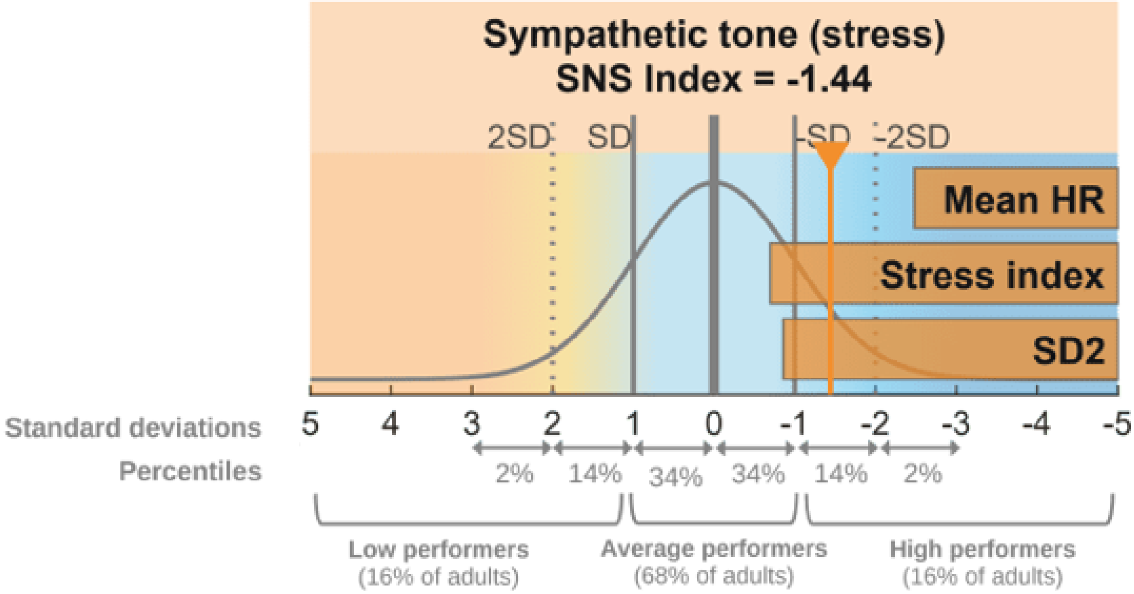

Figure 4: Sympathetic nervous system index [10].

These indexes help understand heart health by showing the difference between relaxation effects from the parasympathetic system and stress responses from the sympathetic system.

2.3 PPG Sensor Technology and Heart Rate Detection

The pulse sensor used in the project produces a photoplethysmography (PPG) signal. It is an optically obtained plethysmogram, which is generated by detecting changes in the blood volume under the skin using an LED and a photodetector [12]. It is commonly used in wearable devices for monitoring cardiovascular parameters such as heart rate (HR) and heart rate variability (HRV)[13]. Figure 5 shows how the sensor is used and the principle of PPG.

The PPG sensor typically uses infrared, red or green LEDs as light sources along with a photodetector that measures the light reflected by the blood flow under the skin [14][13]. The raw PPG signal is often noisy when obtained from a moving subject and therefore needs to be filtered. After conditioning the quality of the signal is reliable for peak detection. Peak detection is done by finding the local maximum, which is the point where signal gradient changes from positive to negative and this indicates a peak. An algorithm is applied for this peak detection to be possible [14].

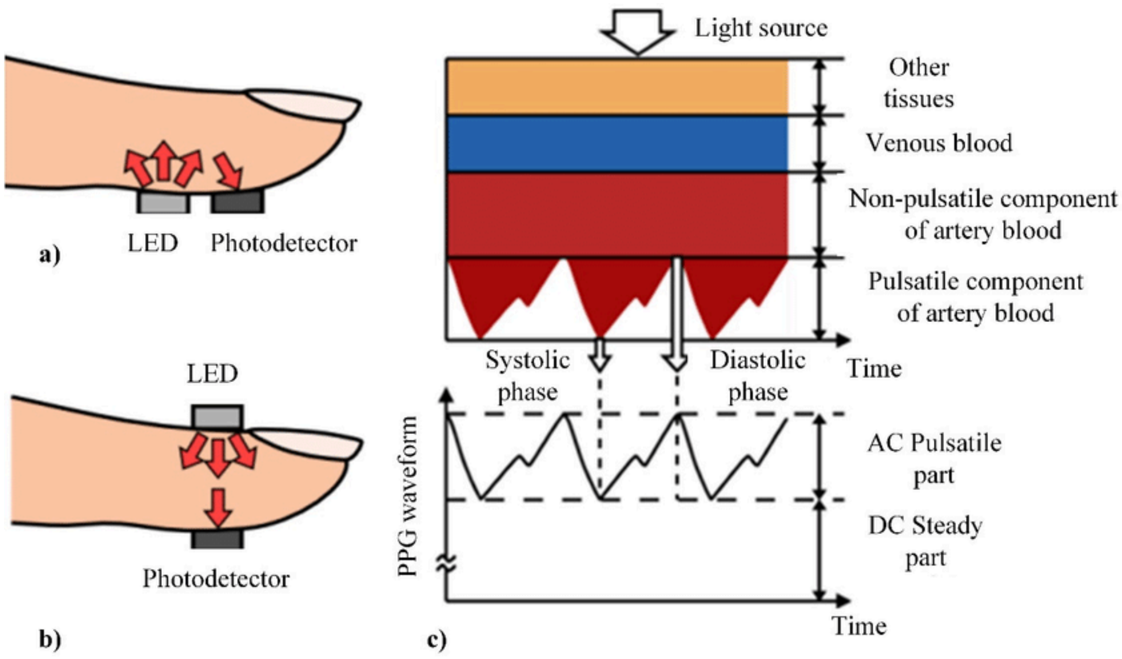

Figure 5: Principle of the photoplethysmography, a) reflective mode, b) transmitting mode, c) example of the PPG signal [15].

Heart rate is determined by calculating the interval between successive peaks in the PPG signal, typically expressed in beats per minute (BPM). This measurement involves recording the duration between detected peaks, known as the peak-to-peak interval (PPI). To calculate BPM, divide the number of seconds in a minute by the average PPI duration in seconds [16].

The pulse sensor utilized in this project is the Crowtail Pulse Sensor v2.0. It uses the previously discussed PPG as the heart rate extraction method. The technical properties and use of the sensor will be discussed later in sections “3. Methods and Materials” and “4. Implementation” respectively.

3 Methods and Material

This section discusses the methods and materials utilized in the project. The first subsection details the technical specifications and roles of the components. The second subsection describes the software and development tools utilized in the project. The third subsection outlines how the system was tested and verified in order to produce valid readings for the measured parameters.

3.1 Components

This section describes the hardware and their technical properties in separate subsections.

3.1.1 Pulse Sensor

The Project makes use of Crowtail-Pulse Sensor model 2.0 CT010712P, it operates within a voltage range of 3-5V, which gives flexibility to users depending on different power setups that can be used, even for mobile devices. It comes with a 609 nm wavelength LED, suitable for detecting changes in blood flow through the skin. The pulse sensor is composed of an optical heart rate sensor, LED, amplification component, noise cancellation component, and a connection with the Raspberry Pi Pico is formed with Grove-connectors for transmitting the analog signal [17]. Figures 6 and 7 show the component side and the front of the pulse sensor respectively.

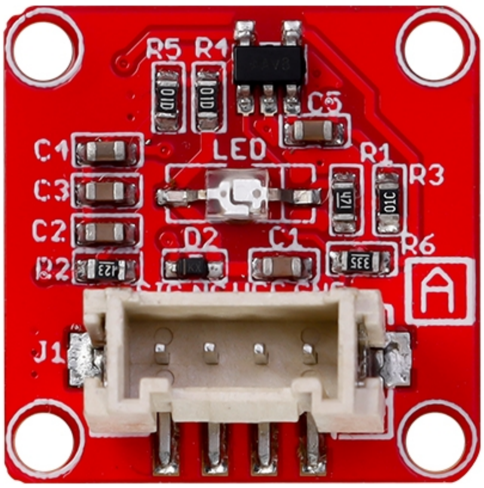

Figure 6: Crowtail Pulse Sensor v2.0 shown from the back with components [17].

Figure 7: Crowtail Pulse Sensor v2.0 shown from the front with the sensor [17].

Grove connectors that are compatible with Crowtail connectors and Grove cables are used to connect the sensor to the development board. Figure 8 and 9 show the Grove connector and Grove cables respectively.

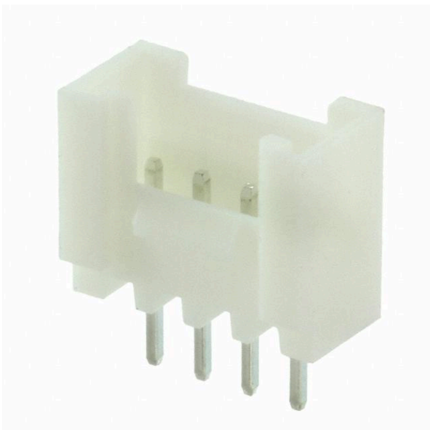

Figure 8: A DIP Grove female header [18].

Figure 9: A 4-pin Grove connector cable [19].

3.1.2 Display

The display used in this project is an organic light emitting diode (OLED). Light is emitted when a current is applied. The OLED display is a basic dot matrix graphic display with 128x64 pixels. A graphical image can be displayed on the screen by turning on or off the LEDs of these pixels. The SSD1306 compatible display uses the I2C interface for communication [20]. Figure 10 shows the display used in the project.

Figure 10: SSD1306 compatible monochrome OLED display [20]

3.1.3 Microcontroller Board

Raspberry Pi Pico W is the microcontroller board utilized in this project. The Pico W includes built-in Wi-Fi and Bluetooth capabilities, which will be applied when transmitting data wirelessly to the Kubios Cloud HRV analysis service for more advanced HRV analysis. With 26 GPIO pins, multiple UART, SPI, and I2C interfaces, the Pico W can connect and manage multiple different devices, including sensors, and display [21][22]. Figure 9 displays the Raspberry Pi Pico W Rev3 board from the top and bottom sides respectively.

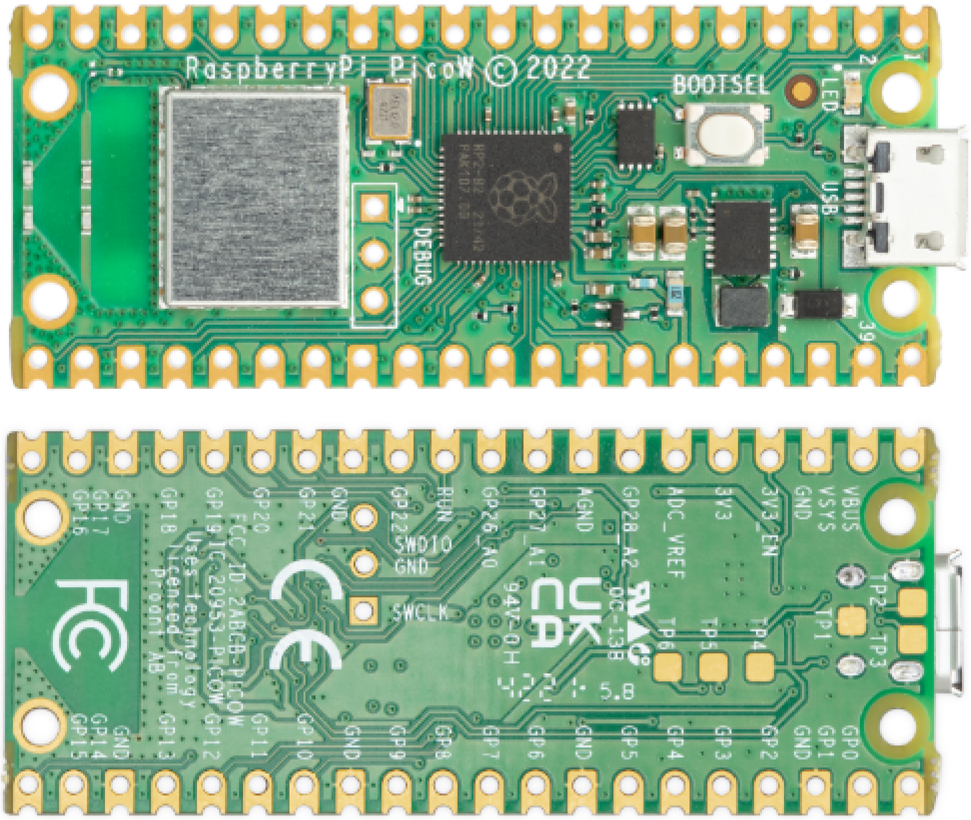

Figure 11: The Raspberry Pi Pico W Rev3 board, component side on top and backside on bottom [22].

The upper image on figure 9 shows the layout of the components on the Raspberry Pi Pico W Rev3 board. The RP2040 dual-core Arm Cortex-M0+ processor sits in the middle, and is responsible for preprocessing data from the optical heart rate sensor. The wireless 2.4 GHz interface Infineon CYW43439 is on the left of the processor closer to the short edge of the board. On the right, shorter edge of the board is the micro-USB port, this connection is used for power and the ability to program the device. On the upper left is an LED and right next to it is the BOOTSEL button. The BOOTSEL button is used to reset the board and enter bootloader mode. GPIO pins are along the longer edges of the board, suitable for connecting with various devices [21][22].

3.1.4 Rotary Encoder

For the user interaction the project uses a rotary encoder designed by Joseph Hotchkiss, project engineer of Metropolia UAS. It is used to navigate through the menu by rotating the encoder and pushing it to select options on the OLED screen. Figure 10 shows the rotary encoder utilized in the project without the 3D printed cap for the knob [1].

Figure 12: Rotary encoder with a push button switch. [1].

The device comes with three different outputs ROT A, ROT B, and Rot switch. The Rot switch functions just like a standard push button. Once the knob is pressed down, the electrical signal changes from high to low or low to high, depending on the wiring of the encoder. By using one of the outputs as a clock signal, the state of the other pin at the time of the clock pulse indicates the direction [1].

3.2 Software and Development Tools

This subsection discusses the software, libraries, and the integrated development environment (IDE) utilized in the project.

3.2.1 Thonny IDE and MicroPython

The software for this project is developed using the Thonny IDE, which is customized for programming the Raspberry Pi Pico, and uses MicroPython as the coding language [23]. Thonny also allows you to download MicroPython firmware to embedded boards, like the one used in the project, Raspberry Pi Pico [1]. Micropython is a streamlined version of the latest Python releases. It offers fewer libraries and features compared to the standard Python. It comes with some special libraries such as machine, utime, urequests, ujson, and ssd1306 [1].

In the project, a pico library provided by the Metropolia UAS is downloaded locally on the device. This file contains the following libraries: fifo.py, filefifo.py, led.py, and piotimer.py. Fifo (first-in-first-out) is a data storage where items added first are also read first. It enables a way to store integers in an interrupt safe way. Filefifo uses the same principle as Fifo to store items and it is utilized during the course assignments to read values from a file. Led contains methods to control the LED in several ways, on/off, brightness and a combination of these. Piotimer class is designed as a hardware timer that offers an interface identical to the timer class used in MicroPython. The ssd1306 is used to manipulate the OLED display and relies on the MicroPython library called Frame-Buffer for graphical operations [24][25][26].

3.2.2 Kubios

Kubios Cloud is an Amazon Web Services (AWS) based cloud service that uses REST API interface for comprehensive HRV analysis. The project will utilize this service for detailed HRV analysis on the Raspberry Pi Pico by sending inter-beat-interval (IBI) data to Kubios Cloud and receiving analysis results [1].

3.3 System Testing

The system testing began with the development of the signal detection and processing algorithm. Once the algorithm produced values that were found rational, these results were benchmarked against a patient monitor. The test subject had one finger on the project device sensor and another in the finger clip of the pulse oximeter. This way the results were compared in real time. After this, more refinement of the algorithm ensued to get closer to the values measured from the patient monitor. The group decided to test different pulse sensors from the development case of each member to see if there was any variation between measurements. In order to have more consistent results the device was tested on multiple different subjects and body parts with good peripheral blood flow.

Once a satisfactory level of accuracy was achieved, the team tested the functions developed for HRV calculations to ensure they produced data within an acceptable range and reflected reliable results to a significant degree. A 30 second measurement period was used to calculate the mean values of PPI, HR, SDNN, RMSSD. The results of these calculations were compared to research data, to ensure the calculators yielded plausible results.

There were slight variations in results from time to time, which can be attributed to several factors such as motion artifacts, cold hands, and ambient lighting. Several more revisions later, the team was satisfied with the precision of the program.

4 Implementation

This section gives a complete description of the final end-to-end system developed by group 3 for the hardware 2 project. It details interconnected devices and algorithms created for calculating and measuring heart rate and heart rate variability.

4.1 Final system: Hardware

The hardware configuration of the final system consists of the Raspberry Pi Pico W microcontroller board, the Crowtail pulse sensor, a rotary encoder, and an OLED display. These components communicate with the Pico via a protoboard. The pulse sensor is connected to the protoboard with a grove cable and attached to the ADC_0 pin, while the OLED screen is connected to its designated I2C_1 pin.

To power on the system, it needs to be connected to an external power source via a USB cable. Interaction with the system is done by rotating or pressing the rotary encoder.

The pulse sensor captures a PPG signal, which is converted to digital using the AD-converters of the Raspberry Pi Pico [1]. The data is transferred to the Pico, and processed according to the algorithms developed.

In order to send data to Kubios Cloud for further processing, the Raspberry Pi Pico establishes a Wi-Fi connection with a local router. Once the data is processed, the results are transferred back to the Pico, and displayed on the OLED screen. Additionally, the Wi-Fi connection is used to publish messages to other devices using the MQTT protocol.

Figure 13: Depiction of the complete system architecture.

4.2 Final system: Software

The system is represented in the flow chart in appendix 1. Once the system is connected to a power source, the program starts with an introduction screen.

Following the introduction screen, a four-option menu is displayed. The first option leads to a basic heart rate measurement, the second option executes the local analysis, the third option integrates the Kubios Cloud results, and the fourth option opens the history. The menu can be navigated by rotating the rotary encoder, and each option can be selected by pressing the rotary encoder. When a measurement option is chosen, the program prompts the user to press the rotary encoder button to start the measurement.

The values collected from the pulse sensor are stored in fifo. With a sampling rate of 250Hz, 250 samples are obtained per second. Each measurement option reads samples from the fifo and dynamically calculates a threshold of 90% between the highest and lowest samples within the past second. If the current sample is over the threshold, the program starts acknowledging the samples and sets a current peak. If the sample falls under the threshold, the interval between the previous and the current peak is calculated (RRI). The program calculates the heart rate using the RRI value, and updates the previous peak to be the current peak, as well as confirms the rationality of the heart rate. If the heart rate is deemed rational, the program calculates the PPI and stores it in a list, an LED blinks to indicate a peak has been found, and the program stops acknowledging the samples until they go over the threshold again.

4.2.1 Measure HR

The Measure HR option measures the average heart rate of the user. The initial measurement takes five seconds, after which the average heart rate is displayed on the screen. The display then updates every five seconds, and the measurement can be stopped whenever by pressing the rotary encoder.

4.2.2 Analysis

The Analysis option measures the heart rate of the user for 30 seconds and draws a live PPG signal on the display. Once the measurement is complete, the program calculates and displays the date and time, mean PPI, mean HR, SDNN, and RMSSD values. The timestamp for the measurement is obtained from the time module. Pressing the rotary encoder saves the results to history and returns the program back to the menu.

4.2.3 Kubios

The Kubios measurement option operates similarly to the Analysis measurement, requiring 30 seconds for measurement and drawing the PPG signal on the display. However, once the measurement is complete, the Raspberry Pi Pico establishes a Wi-Fi connection to send the list of PPI values to Kubios. The data received from Kubios includes the date and time of the measurement, the mean HR, SDNN, RMSSD, SNS and PNS values. The mean PPI is calculated locally. These results are displayed on the OLED display and published with the MQTT protocol to the topic “kubios”. In case the Wi-Fi connection can’t be established or the Kubios data retrieved, the program informs the user and prompts them to press for the menu. If everything went successfully, the results are saved to history.

4.2.4 History

If available, the History option displays the most recent four measurements. Upon the first save, the program creates a .txt file, ensuring that the results are stored locally on the Raspberry Pi Pico.

4.3 Algorithms for HR and HRV Measurement

Algorithm	Function/Method	Description	Related System Component
Heart Rate Measurement	`basic_hr()`	Measures heart rate by calculating the RRI, and then dividing the amount of seconds in a minute (60) by the RR interval value.	ADC (`Isr_adc`), FIFO, LEDs
Heart Rate Variability Analysis	`basic_hr()`	Measures HRV by analyzing time intervals between successive peaks. The PPI is calculated by multiplying the intervals with the amount of time it takes to obtain one sample (4ms). The PPI values are then used to calculate HRV metrics such as SDNN and RMSSD.	ADC (`Isr_adc`), FIFO

Peak Detection	`basic_hr()`	Identifies significant peaks in the ADC data for heart rate and HRV calculations by adjusting a threshold.	ADC (`Isr_adc`), FIFO
Mean PPI Calculation	`calc_mean_ppi(ppi_values)`	Calculates the mean peak-to-peak interval by dividing the sum of the PPI values with the total number of PPI values.	Data Processing
Mean HR Calculation	`calc_mean_hr(average_ppi)`	Calculates the mean heart rate by dividing the amount of milliseconds in a minute (60000) with the mean PPI.	Data Processing
RMSSD Calculation	`calc_rmssd(ppi_values)`	Computes the root mean square of successive differences between heartbeats to estimate parasympathetic activity.	Data Processing
SDNN Calculation	`calc_sdnn(ppi_values, average_ppi)`	Calculates the standard deviation of NN intervals, a measure of HRV that indicates overall heart rate variability.	Data Processing

Table 1: Explains the algorithms for measuring heart rate and variability, describing their functions, operations, and the related system components.

5 Group Work Summary

This section outlines the contributions of each member during the project course. Midway summary is done half way through the course and describes what each member has completed so far, be it programming, background research, or writing the project report. Final summary explains the same ideas but is written after the project is finished.

5.1 Midway Summary

Ade Aiho: Attended the project lectures. Completed the assignments from weeks 1 and 2. Found a relevant research article which could be helpful for the project. Took part in writing the project report introduction.

Topias Aho: Read through the first four weeks of course material and participated in lectures. Completed assignments to utilize the learned course material in practice. Found a research article which could be useful for future project work. Took part in writing the project report and project planning.

Jonne Roponen: Participated in project lectures and workshops to study and learn the weekly project course material. Completed the first two project assignment sets. Looked into research articles that could potentially be beneficial for the project and project report. Useful articles could be cited in the final project report. Also partook in writing the project report introduction and weekly plan.

The group will finish the remaining weekly assignments and consider how they are applicable to the actual project. Project report will be written alongside the project work. To plan the upcoming weeks the project will be broken down into separate weekly sections.

Week 1. Interface development

Coding the user interface and testing it with the OLED display. Implementing rotary knob controls to navigate through the UI. Making sure icons and animations work correctly.

Week 2. Health metrics algorithms

Developing and testing algorithms for calculating health metrics like heart rate and heart rate variability. Creating functions to calculate different health metrics from the extracted PPG signals. Test different methods of noise filtering to ensure accurate data collection.

Week 3. Main program construction

Building a main program that ties all the components together. The main loop will use a rotary encoder to navigate between different metrics displayed, start measurements, and display results.

Week 4. Cloud connectivity and final testing

Establishing a connection with Kubios Cloud for more advanced HRV analysis and finalizing the project and project report. Implementing the connection to send IBI data to Kubios Cloud and receiving the analysis results. Preparing for the project presentation.

The group has yet to run into anything that would hinder or stop the progress and development of the project.

5.2 Final Summary

Ade Aiho: Partook in planning the project. Developed the initial algorithm for finding the peaks in the PPG signal and calculating the heart rate. Made the parts of the program that handle the connection with Kubios Cloud and MQTT messaging. Tested and verified the overall functionality of the program. Wrote sections of the report and prepared the slides for the presentation.

Topias Aho: Took part in planning the project and writing the report. Coded the parts of the program that handle measurement history and live graph drawing. Designed the overall structure of the program, combining the work of all group members into a working product. Made improvements to the data filtering and HR detection by comparing results against a patient monitor. Implemented new parts of the code into the provided mp remote script making code testing more accessible.

Jonne Roponen: Partook in the overall design and planning of the project. Tried to delegate tasks to each member according to their own skillset. Did research for articles relating to the theoretical background, materials, and methods. Wrote sections of theoretical background, materials and methods, conclusions, and group performance evaluation. Prepared the presentation slides with Ade. Tried to keep the reference list manageable. Made the display and rotary encoder classes, wrote and tested the functions for HRV calculators, HRV metric display, LED blinking, and introduction.

5.3 Group Performance Evaluation

The team got along very well and was able to come to an understanding quickly how to approach tasks and how to assign workforce. After the initial HR detection and processing algorithm was finished the team was able to start working on other functionalities. Each member was assigned to finish a certain part of the program according to their own skills and desires.

Every team member had constantly something to work on and team meetings were held regularly to keep up what each team member had accomplished. These meetings were also to see if there were any difficulties or problems that could have potentially arisen. Team members were open to feedback and nobody was afraid to voice their concerns or ideas if they differed from the proposed ones.

There might have been slight room for improvement when it comes to planning project work in tandem with other studies. Also the workload could have been spaced more evenly in the leading weeks before the project presentation, which would have led to a less workload intensive weekend before the presentation.

As the program gradually began to be developed the goals of the group expanded and were soon realized as well. As a team of quite ambitious people the group had constantly a great level of motivation. Everyone performed to the best of their ability and beyond. The team itself is quite pleased with how everyone performed and what became of the project device.

6 Conclusions

After reviewing the end product and comparing it to the project assignments from which the project was derived, it can be said that the project went really well. Looking back at the knowledge and skills the team had before the project in contrast with the present it is fair to say the project was a success.

The team set out to reach all the grade 5 requirements and gradually built up the device software to meet the requirements of each individual grade. Due to no previous experience with embedded systems the team had some doubts about the end product at first, but as soon as the project assignments were completed the team had a better picture of the project as a whole. The goal was to create a working prototype of a heart rate detection and analysis system, which was successful.

The obstacles faced by the team were relatively minor. Most of the headache was caused by the development of the heart rate detection and processing algorithm. Dealing with live data and having it be thresholded to produce sensible data was more difficult than anything else. In order to surpass this obstacle the team patiently and rigorously tested the program and benchmarked it against a patient monitor until a satisfactory level of accuracy was achieved. All other minor problems were handled during project classes or in discord meetings amongst the group, when a member faced something they would bring it up and it would be looked at together. Thus all the issues and misunderstanding were weeded out.

Currently the device does not have any real limitations. The features of the device meet all the requirements of the evaluation criteria for grade 5. Small improvements could be made in the graphical layout for more intuitive user experience. Also the drawing of the live PPG signal could be more refined.

Some development ideas for the future are: more HRV calculations locally (SD1, SD2, SDSD). A More sophisticated algorithm for the detection and processing of the live analog signal. Machine learning for a predictive measurement and trends in overall health of the subject. The data collected could be integrated and utilized by other health applications to further the insights into health metrics. Lastly, the device could have a wearable version for continuous measurement and monitoring of heart rate.

References

1. Lukkarinen, S. 2024. Hardware 2 Project Requirements, Version 2.9. School of ICT, Metropolia University of Applied Sciences. Online [Accessed 16 April 2024].
2. Tiwari R, Kumar R, Malik S, Raj T, Kumar P. 2021. Analysis of Heart Rate Variability and Implication of Different Factors on Heart Rate Variability. Current Cardiology Reviews. Available online: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8950456/ [Accessed 20 April 2024].
3. Booth, Frank W. Christian K. Roberts, and Matthew J. Laye. 2012. Lack of exercise is a major cause of chronic diseases. *Comprehensive Physiology*. Available online: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4241367/ [Accessed 29 April 2024].
4. Young, Hayley, and David Benton. 2015. We should be using nonlinear indices when relating heart-rate dynamics to cognition and mood. *Scientific Reports*. Available online: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4643265/ [Accessed 29 April 2024].
5. Lief. 2024. How to Measure Heart Rate Variability: ECG vs PPG. Available online: https://blog.getlief.com/how-to-measure-heart-rate-variability-ecg-vs-ppg/ [Accessed 21 April 2024].
6. Jay W. Mason, Douglas J. Ramseth, Dennis O. Chanter, Thomas E. Moon, Daniel B. Goodman, Boaz Mendzelevski. 2007. Electrocardiographic reference ranges derived from 79,743 ambulatory subjects.Journal of Electrocardiology. Available online: https://www-sciencedirect-com.ezproxy.metropolia.fi/science/article/pii/S0022073606003530 [Accessed 20 April 2024].
7. Shookster, Daniel; Lindsey, Bryndan; Cortes, Nelson & Martin, Joel R. 2020. Accuracy of Commonly Used Age-Predicted Maximal Heart Rate Equations. *International Journal of Exercise Science*. 13(7): 1242-1250.

Available online: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7523886/ [Accessed 21 April 2024].

8. WHOOP. 2024. Resting Heart Rate by Age and Gender. Available online: https://www.whoop.com/us/en/thelocker/resting-heart-rate-by-age-and-gender/ [Accessed 21 April 2024].
9. He, Zonglu. 2020. The control mechanisms of heart rate dynamics in a new heart rate nonlinear time series model. Scientific Reports. Available online: https://www.nature.com/articles/s41598-020-61562-6 [Accessed 20 April 2024].
10. Kubios. 2024. HRV and Autonomic Nervous System Function. Available online: https://www.kubios.com/hrv-ans-function/ [Accessed 20 April 2024].
11. Shaffer, F. & Ginsberg, J.P. 2017. An Overview of Heart Rate Variability Metrics and Norms. Frontiers in Public Health. Available online: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5624990/ [Accessed 20 April 2024].
12. Choi, Jin-Ha; Shin, Hye-Min. 2023. Comprehensive Analysis of Multi-Wavelength Photoplethysmography: Techniques and Applications. *Electronics*. Available online: https://doi.org/10.3390/electronics12132923 [Accessed 21 April 2024].
13. Hinde, Katrina; White, Graham & Armstrong, Nicola. 2021. Wearable Devices Suitable for Monitoring Twenty Four Hour Heart Rate Variability in Military Populations. Sensors. Available online: https://doi.org/10.3390/s21041061 [Accessed 21 April 2024].
14. Wójcikowski, Marek & Pankiewicz, Bogdan. 2020. Photoplethysmographic Time-Domain Heart Rate Measurement Algorithm for Resource-Constrained Wearable Devices and its Implementation. Sensors. Available online: https://www.mdpi.com/1424-8220/20/6/1783 [Accessed 21 April 2024].
15. ResearchGate. 2024. Principle of photoplethysmography (PPG): (a) reflective mode, (b) transmitting mode. Available online: https://www.researchgate.net/figure/Principle-of-photoplethysmography-P

PG-104-a-reflective-mode-b-transmitting_fig7_338723696 [Accessed 20 April 2024].

16. Shaffer, Fred & Ginsberg, J.P. 2017. An Overview of Heart Rate Variability Metrics and Norms. *Frontiers in Public Health*. Available online: https://doi.org/10.3389/fpubh.2017.00258 [Accessed 21 April 2024].
17. Elecrow. 2024. Crowtail Pulse Sensor. Available online: https://www.elecrow.com/crowtail-pulse-sensor-p-1673.html [Accessed 21 April 2024].
18. Cytron Technologies. 2024. Grove 4-Pin Straight Through-Hole Socket. Available online: https://sg.cytron.io/c-robotics-electronics/c-electronic-components/c-connectors/c-pcb-connector/p-grove-4-pin-straight-through-hole-socket [Accessed 21 April 2024].
19. eBay. 2024. Grove Cable 4 Pole 20cm 5pk - 110990027. Available online: https://www.ebay.co.uk/p/693174108 [Accessed 21 April 2024].
20. Electronic Wings. 2024. SSD1306 OLED Display. Available online: https://www.electronicwings.com/sensors-modules/ssd1306-oled-display [Accessed 21 April 2024].
21. Raspberry Pi Foundation. 2024. Raspberry Pi Pico W Datasheet. Available online: https://datasheets.raspberrypi.com/picow/pico-w-datasheet.pdf [Accessed 22 April 2024].
22. Raspberry Pi Foundation. 2024. Raspberry Pi Pico Product Brief. Available online: https://datasheets.raspberrypi.com/pico/pico-product-brief.pdf [Accessed 23 April 2024].
23. P. D. George. 2014-2023. “MicroPython” George Robotics Limited. Available Online: https://micropython.org/ [Accessed 29 April 2024].
24. Hotchkiss, Joseph. 2024. GPI interrupts and Fifo. Online [Accessed 29 April 2024].
25. Hotchkiss, Joseph. 2024. Timers and ADC. Online [Accessed 29 April 2024].

26. MicroPython. 2024. SSD1306 Driver. MicroPython Documentation. Last updated on 09 May 2024. Available online: https://docs.micropython.org/en/latest/esp8266/quickref.html#ssd1306-driver [Accessed 10 may 2024]

Flow Chart of the Final System