SP: Jorganes Javier, Rikassaari Senni, Rappe Oskar, Pajunen Mira

Heart Rate Monitor

Project Report

First Year Hardware Project

School of ICT

Metropolia University of Applied Sciences

5th May 2023

Abstract

Heart rate can indicate a lot of different things about a user's health, a few of them being stress and recovery. During this project, a heart rate monitor was built using Raspberry Pi Pico W and a PPG sensor. This monitor can measure a user's heart rate, heart rate variability, as well as SNS and PNS values which give indications about the user's stress and recovery levels. This report goes through the other materials and software used to build and program it.

At the start of this paper, some theoretical background is discussed relating to heart rate and heart rate variability and their link to a person's health. The algorithm developed for this project is discussed in the implementation part and conclusions about the project can be viewed at the end of the paper with further development ideas for the monitor.

Version history

Ver	Description	Date	Author(s)
1.0	Wrote introduction and methods and material parts.	3.4.2023	Senni Rikassaari
1.1	Theoretical Backround part written	4.4.2023	Mira Pajunen, Senni Rikassaari
1.2	Theoretical Backround past finished.	5.4.2023	Javier Jorganes, Oskar Rappe, Mira Pajunen
1.3	First draft finished.	7.4.2023	Senni Rikassaari, Mira Pajunen
1.4	Grammar corrections	23.4.2023	Mira Pajunen
1.5	Draft 2 finished	26.04.2023	Javier Jorganes, Oskar Rappe, Mira Pajunen
1.6	References added. Grammar changes made.	28.04.2023	Senni Rikassaari
1.7	Final report draft done.	04.05.2023	Senni Rikassaari, Javier Jorganes, Oskar Rappe, Mira Pajunen
1.8	Final corrections.	05.05.2023	Senni Rikassaari, Mira Pajunen

1 Introduction

Heart rate monitoring is becoming increasingly popular as modern smartwatches and rings allow the tracking of pulse during sleep, rest, and exercise. While photoplethysmography (PPG) is the most common technology used to measure heart rate with wearable devices, the accuracy of this technology is disputed.

PPG sensors work by shining light into the skin and detecting the amount of light reflected back. By measuring the changes in blood volume, which affects the amount of light reflected back, the sensor can calculate heart rate. This project aimed to develop a device that could correctly measure a user’s heart rate using a PPG sensor and to develop an algorithm that detects heartbeats accurately from the data the sensor offers. Additionally, the project aimed to build a wireless connection to Kubios Cloud service to calculate more detailed HRV analysis from the data that the heart rate algorithm provided and display the results on the device’s screen. The project was completed during the last period of the first school year as part of the Project course and was worth 4 credits.

2 Theoretical Background

The physiology related to heart rate measurements is introduced in this section. General information concerning the heart rate, heart rate variability (HRV) and the sensor used in the project is provided.

2.1 Heart Rate

Heart rate reflects how often the heart beats and is commonly measured in units of beats per minute (BPM). Resting heart rate describes the heart rate when body is at rest, such as relaxed, laying down and sitting. Figure 1 illustrates the resting heart rates of men and women by age and physical condition. The maximum heart rate refers to the highest number of beats per minute that the heart can pump under maximum stress. This can be roughly calculated using a formula of 220 minus the person's age, although the results are only an approximation. (1)

Resting Heart Rate Chart

Men (beats per minute)

Age	18 - 25	26 - 35	36 - 45	46 - 55	56 - 65	65 +
Athlete	49 - 55	49 - 54	50 - 56	50 - 57	51 - 56	50 - 55
Excellent	56 - 61	55 - 61	57 - 62	58 - 63	57 - 61	56 - 61
Great	62 - 65	62 - 65	63 - 66	64 - 67	62 - 67	62 - 65
Good	66 - 69	66 - 70	67 - 70	68 - 71	68 - 71	66 - 69
Average	70 - 73	71 - 74	71 - 75	72 - 76	72 - 75	70 - 73
Below Average	74 - 81	75 - 81	76 - 82	77 - 83	76 - 81	74 - 79
Poor	82 +	82 +	83 +	84 +	82 +	80 +

Women (beats per minute)

Age	18 - 25	26 - 35	36 - 45	46 - 55	56 - 65	65 +
Athlete	54 - 60	54 - 59	54 - 59	54 - 60	54 - 59	54 - 59
Excellent	61 - 65	60 - 64	60 - 64	61 - 65	60 - 64	60 - 64
Great	66 - 69	65 - 68	65 - 69	66 - 69	65 - 68	65 - 68
Good	70 - 73	69 - 72	70 - 73	70 - 73	69 - 73	69 - 72
Average	74 - 78	73 - 76	74 - 78	74 - 77	74 - 77	73 - 76
Below Average	79 - 84	77 - 82	79 - 84	78 - 83	78 - 83	77 - 84
Poor	85 +	83 +	85 +	84 +	84 +	85 +

agelessinvesting.com

Figure 1. Resting heart rate chart (2).

The heart rate varies according to the body's physical needs, including oxygen uptake and carbon dioxide removal. Additionally, several factors affect the heart rate, such as a person’s age, genetics, diet, drugs, environment, and underlying medical conditions. Physical fitness is also a significant factor affecting heart rate, with well-trained athletes often having a lower resting heart rate than average individuals, sometimes as low as 40 BPM.

The most common physiological phenomena related to heart rate are:

- Exercise: Increases the heart rate so the muscles get more oxygen.
- Stress: Hearth rate increasement is a natural reaction in stressful situations. Stress can also reduce HRV levels.
- Sleep: Decreases the heart rate as the parasympathetic nervous system becomes dominant when asleep.

It is important to acknowledge that heart rate varies greatly among individuals and across different situations. Heart rate range from 60 to 100 bpm is considered a normal resting heart rate for adults.

2.2 Heart Rate Variability

Heart rate variability (HRV) is the variation of time intervals between heartbeats. The time between successful beats is measured in milliseconds (ms) and is called an “R-R interval” or “inter-beat interval” (IBI). Figure 2 provides a visual representation of HRV. A favorable heart rate variability is relative to every person since it is a highly sensitive metric. Heart rate variability that is normal for one person might not be that for someone else. Having said that, the HRV range can vary substantially from 25 ms to 150 ms. (3)

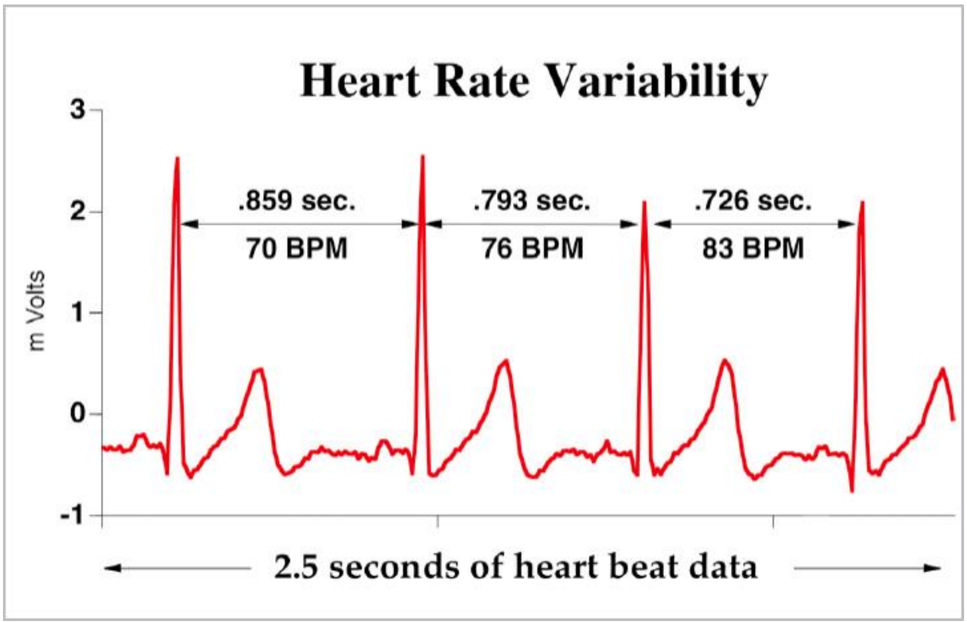


Figure2. Heart rate variability visualized in between heart beats and represented in seconds. (4)

Heart rate variability is controlled by the autonomic nervous system, which regulates the involuntary tasks of the body such as heart rate, blood pressure, breathing, and digestion. This system is divided into two major components: The sympathetic and the parasympathetic nervous system. The sympathetic nervous system activates various body functions when required, triggering the body's "fight-or-flight" response, which leads to dilated pupils, the release of adrenaline, and an increased heart rate. Conversely, the parasympathetic system calms the body’s functions down, constricting pupils and slowing the heartbeat. Examples of the different functions that these systems are responsible for are presented in greater detail in Figure 3.

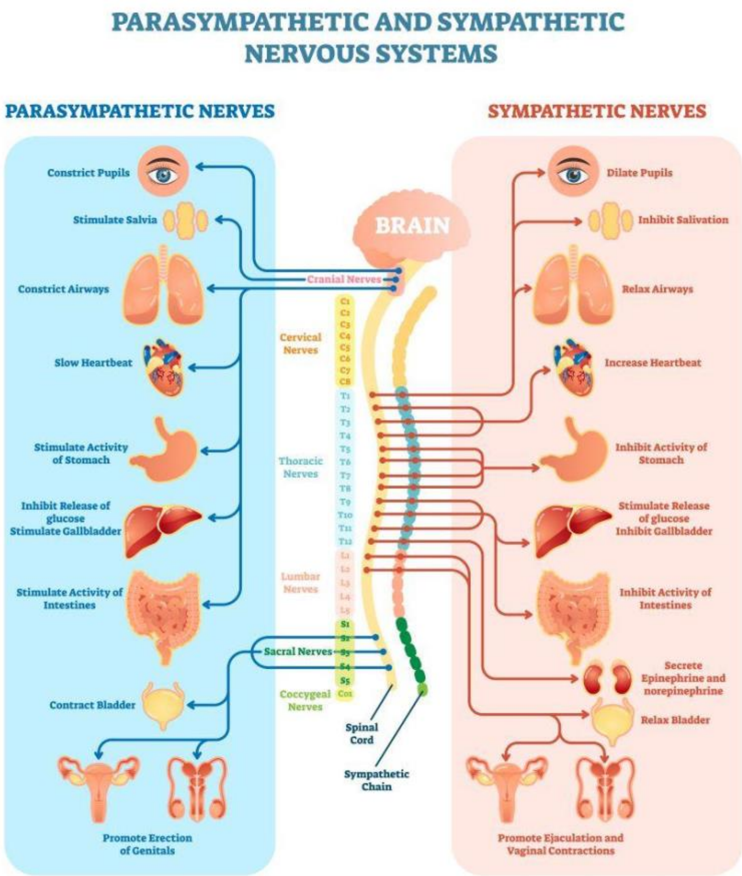


Figure 3. The functions of parasympathetic and sympathetic nerves. (7)

When the autonomic nervous system is in fight-or-flight mode, the variation between heartbeats tends to be lower. On the contrary, when the system is in a relaxed state, the variation between beats seems to be higher. As a result, a higher HRV is considered to be a sign of better health as it usually means that the body is more adaptable to changes. (5, 6)

2.3 Measuring Heart Rate and HRV

The sensor used in this project produces a signal called photoplethysmography (PPG). This signal measures the changes in blood volume in tissue, such as a finger, by detecting the amount of light absorbed or reflected by the tissue. When the heart beats, blood flow through the tissue changes, causing small variations in the amount of light absorbed or reflected, and this produces a pulsatile signal. Figure 4 demonstrates this.

By analysing the time intervals between successive heartbeats, it is possible to measure the HR and HRV. (8)

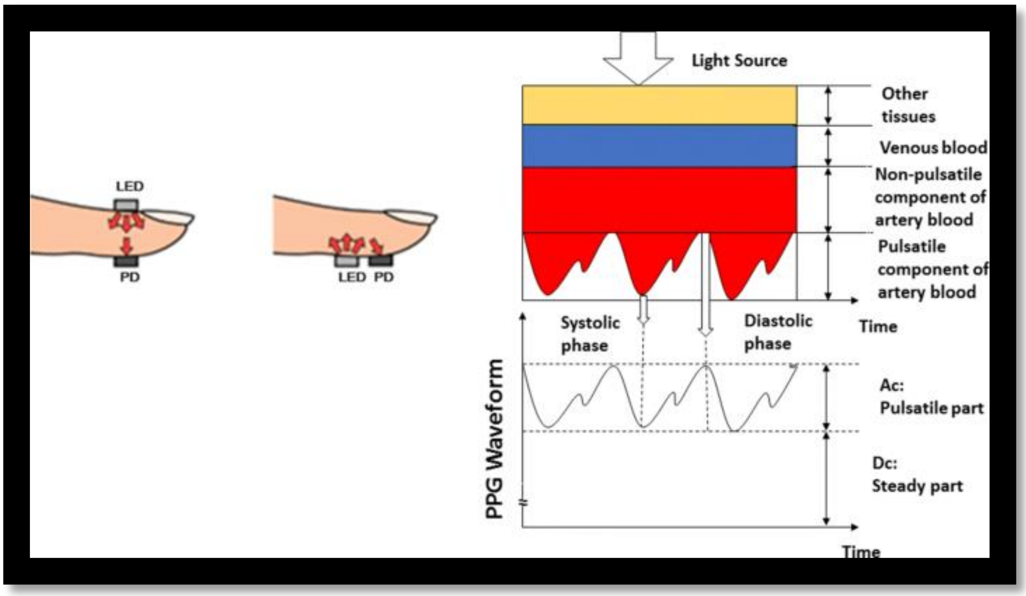
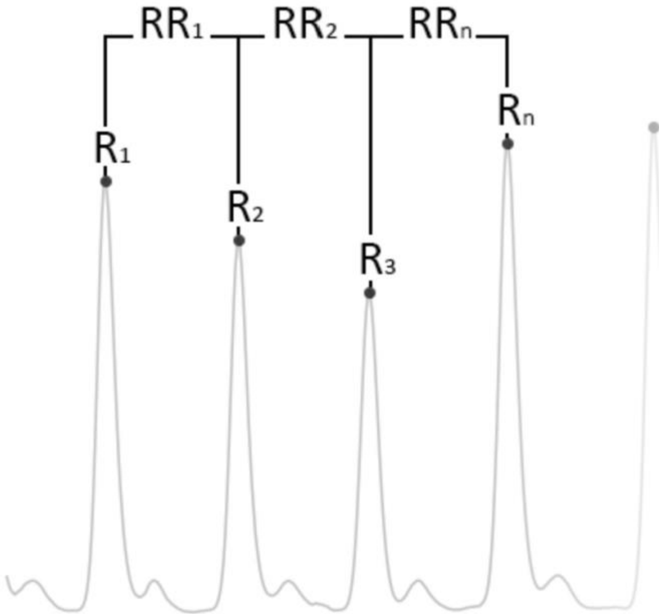


Figure 4. Representation of the photoplethysmography process. (8)

When extracting heart rate variability (HRV) measurements, the peak positions are crucial when obtaining an exact value of the signal. With these values the calculation of HRV is possible. An example of an often-used variability measurement is the RMSSD (root mean square of successive differences). Given a segment of heart rate data as displayed in the figure below, the RMSSD is calculated as shown in Figure 5. (9)



$$RMSSD = \sqrt{\frac{1}{n-2}\sum_{i=0}^{n-2}(RR_i - RR_{i+1})^2}$$

n = number of R-peaks used in analysis

Figure 5. Calculating RMSSD. (9)

Figure 5 gives a visual representation of the heart beats and how HRV is calculated using RMSSD measurement.

3 Materials and Testing

This section introduces the materials used in the project: a sensor, software, devices, and their most important technical properties. Furthermore, it presents the methods used in testing that the system produces valid readings for the measured parameters.

3.1 Materials

All the materials, hardware components, and software are presented in detail in this section.

3.1.1 Heart Rate Sensor

The sensor used in this project was a Crowtail pulse sensor v2.0. It is a PCB (Printed Circuit Board) sensor connected through the Grove-connector to the Raspberry Pi Pico’s analogue inputs. The electrical components of the heart rate sensor were soldered on the back side of the sensor PCB (Figure 15). The heart rate sensor is shown in Figure 15 and 16.

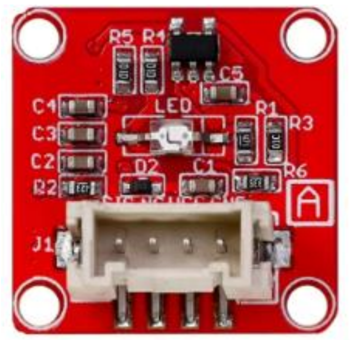


Figure 6. Crowtail pulse sensor v2.0 shown from the back (component) side. (10)

Figure 7. Crowtail pulse sensor v2.0 shown from the front (sensor) side. (10)

The location on which the heart rate was detected, be it a finger or some other place, is placed on the front side of the sensor PCB (Figure 7). In the middle of the board is a LED, which was used to transmit the light to the body tissue. Below the LED is an optical sensor, photodiode, which receives the reflected light and converts the signal to voltage. (11)

3.1.2 OLED Display

The OLED display has 128x64 pixels and can display both text and graphics as shown in Figure 8. The display is controlled by SSD1306 circuit which is embedded to the display. The SSD1306 compatible OLED display uses either a SPI or I2C interface. In this project the I2C interface was used.

Figure 8. SSD1306 compatible OLED display used in the project. (10)

MicroPython documentation gives examples how to use the library and get the most out of it. The OLED can display the heart rate as a number as well as some small graphic related to this topic. (11)

3.1.3 Rotary Encoder

A rotary encoder is a device that provides pulses to indicate the direction and speed of the rotation. The device has three outputs ROT A, ROT B, and Rot Switch. The Rot Switch behaves just like a standard push button tactile switch.

When the knob is pressed, the signal goes from high to low or low to high depending on how the encoder is wired up. Figure 9 illustrates the rotary encoder.

Figure 9. A rotary encoder with integrated push button switch. (10)

When the knob is rotated in one direction or the other, then the Rot A and Rot B outputs a series of pulses out of phase of one another. In one direction, Rot A phase leads Rot B, and when turning the other direction Rot A trails Rot B.

This means that one channel, either Rot A or Rot B, can be used as a clock signal and then check the state of the other pin to identify the direction of the rotation. (11)

3.1.4 Grove Connector

The hardware provides 6 modular, standardized Grove connectors for prototyping purposes. The Grove connector is also compatible with Crowtail connectors and cables. The Grove connector and Grove cables are shown in Figure 10 and 11.

Figure 10. A 4-pin 2.0 mm pitch DIP Grove female header. (10)

Figure 11. A 4-bin buckled Grove cables. (10)

The Grove connector has 4 pins. One of the pins is reserved for the operating voltage (Vcc) and the second for the ground. For digital connectors the remaining pins are usually the clock and data pin. If the connector is used for analogue signals, the remaining pins are usually reserved for the analogue inputs. (11)

3.1.5 Raspberry Pi Pico and USB-port

The Raspberry Pi Pico has a micro-USB-port that can be used by user code, to access an external USB device or host. Figure 12 shows the Raspberry Pi Pico board and its USB-port.

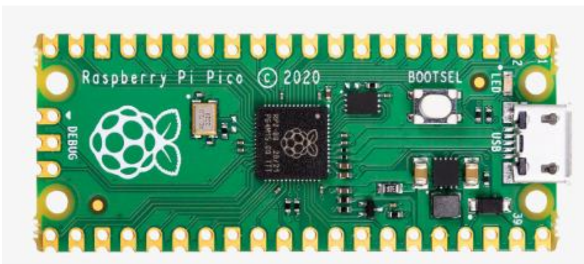


Figure 12. The Raspberry Pi Pico board and its USB-port. (10)

The port can also be used to access the USB bootloader stored in the RP2040 boot ROM. The port also works as a power source for the Raspberry Pi Pico. (11)

3.1.6 Software

The project development software used was Thonny IDE and MicroPython programming language for the Raspberry Pi Pico. MicroPython is a specialized version of Python for embedded systems, and Thonny IDE is a Python-integrated development environment focused, as well, on embedded systems.

MicroPython's "SSD1306" library was used to control the OLED display. This library relies on a MicroPython´s specific library called "FrameBuffer", which provided support for graphics primitives (fill, pixel, hline, vline, line, rect, ellipse, poly), drawing text (text), and other methods (scroll, blit).

This project also utilized Kubios Cloud, an Amazon Web Services-based cloud service using REST API interface, for full HRV analysis based on inter-beat interval calculations on the Raspberry Pi Pico. (11)

3.2 Testing

The accuracy of the system was confirmed by comparing the heart rate obtained from the Raspberry Pi Pico W monitor with those gathered from a CARESCAPE Monitor B650 (Figure 13), which is used in many different professional settings including, but not limited to, ICU (Intensive Care Unit), CCU (Cardiac Care Unit) and PACU (Post-Anesthesia Care Unit), so the results obtained from it are a good point of comparison.

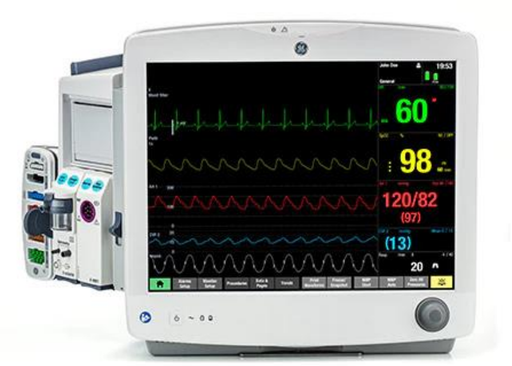


Figure 13. CARESCAPE Monitor B650.

The projects system was also tested with multiple subjects to reduce any possible human errors. The subjects were all between the ages of 20 and 35 and in good physical condition. The sensor was placed on either the finger or forehead of the subjects. Two types of measurement testing were conducted. First, the subjects were instructed to remain still while the sensor gathered data for the heart rate calculation. Second, the subjects were instructed to move the sensor slightly while measuring to determine if the heart rate algorithm could filter out any faulty signals caused by the movement.

4 Implementation

This section presents the final system and the data processing as well as the algorithms developed for the measurements of heart rate.

4.1 Final System

The system connects the Raspberry Pi Pico W to internet using a Wi-Fi connection. The OLED verifies a successful connection and displays the IP address on the screen. The program then waits for the user to press the rotary button to initiate the analysis of the user's pulse and to measure the time intervals between beats, which are stored in a list. Finally, the list is sent to Kubios Cloud service for further analysis of the HRV, which then sends the results back to the program. All these steps can be viewed from the Figure 14.

The results include the mean heart rate in beats per minute, as well as the stress (SNS) and recovery (PNS) indexes. The SNS and PNS indexes are obtained through analysing the heart rate variability, emphasizing the importance of measuring the time intervals between successive beats accurately.

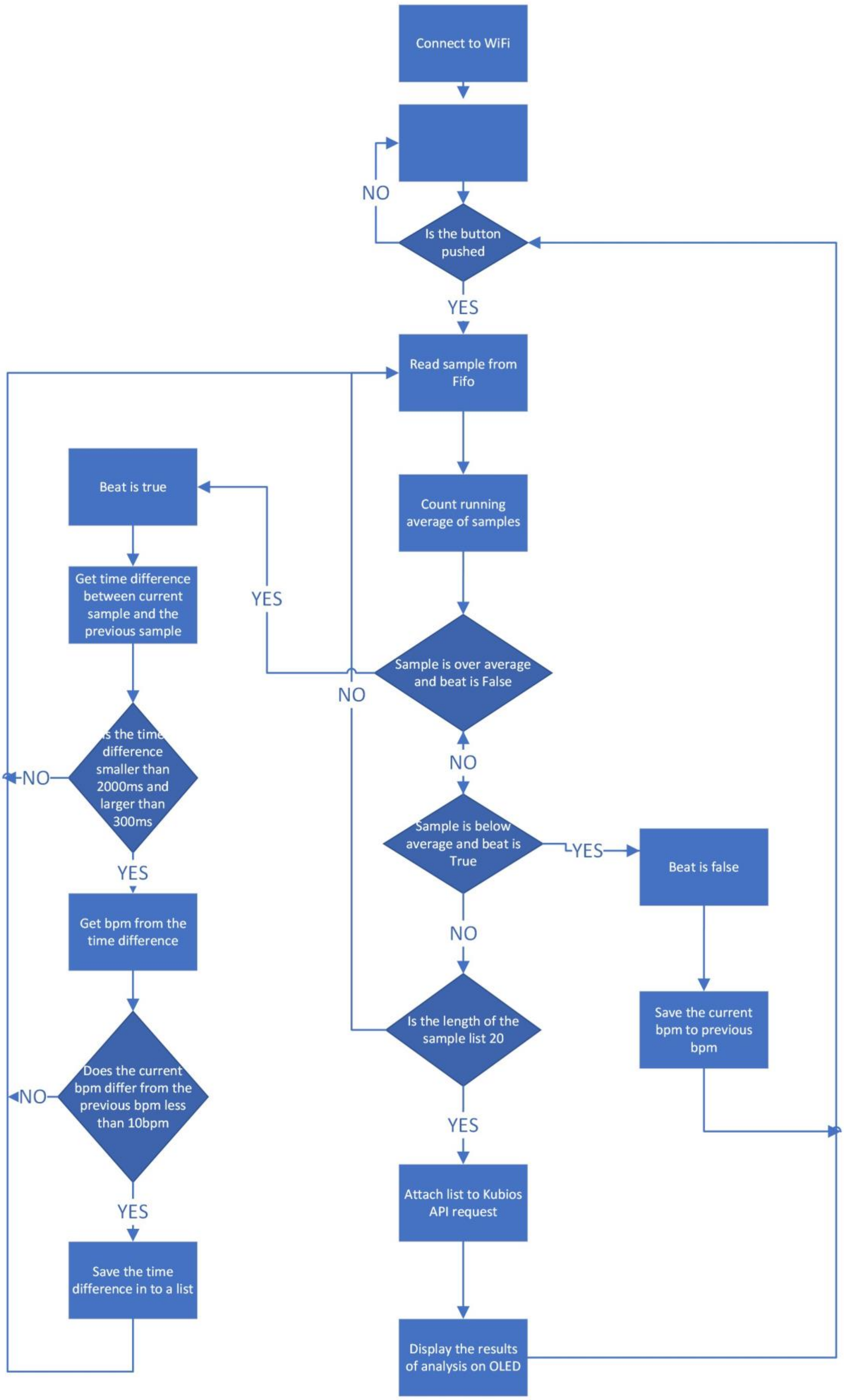

Connect to WiFi
NO
Is the button pushed
YES
Read sample from Fifo
Count running average of samples
Beat is true
YES
Get time difference between current sample and the previous sample
Sample is over average and beat is False
NO
s the time difference smaller than 2000ms and larger than 300ms
NO
NO
Sample is below average and beat is True
YES
Beat is false
YES
Get bpm from the time difference
NO
Is the length of the sample list 20
Save the current bpm to previous bpm
Does the current bpm differ from the previous bpm less than 10bpm
NO
YES
YES
Attach list to Kubios API request
Save the time difference in to a list
Display the results of analysis on OLED

Figure 14. System diagram of the prototype.

Once these results have been sent to the main program, they are displayed on the OLED. The user can then start the program again by pressing the button. All these steps visible on the system diagram (Figure 14).

4.2 Algorithms

The heart rate algorithm detects the rising edges of the signal obtained from the PPG sensor. Figure 15 below illustrates the function of this method. The algorithm calculates the average of 50 data samples to filter out some signal distortions. This small average is used to detect the heartbeat and is referred as a sample. Two thresholds are set, one to the average value of the data samples collected over 2 seconds (number 2 in the figure) and another slightly higher than the average (number 1 in the figure). If the sample value exceeds the higher threshold, a heartbeat is considered detected (number 3 in the figure).

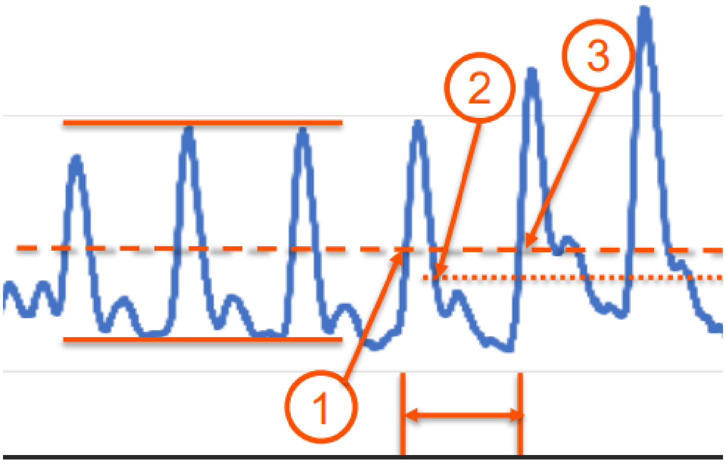


Figure 15. Detection of rising edges from the signal. (12)

However, the algorithm discards the sample if the value is below 75 or over 500 because it results in a heartbeat that is not within the range of 30 bpm to 200 bpm. If the sample value is within the desired range, the heart rate is calculated using an algorithm that divides 60 seconds by the product of the sample value and 4 ms/1000 ms.

The algorithm further filters the results by comparing the current heartbeat value to the previous heartbeat. If the difference is more than 10 bpm, the sample is considered faulty and discarded. This ensures that individual finger or sensor movements do not affect the measurement results. The time interval that leads to successful heart rate calculation is then stored in a list that will be sent to Kubios Cloud service for further analysis. The lower threshold is used to reset the search for next heartbeat and store the value of the latest detected heartbeat.

4.3 Data

The sensor samples the data every 4ms and stores it in a buffer that has a capacity of 500 samples. The main program retrieves the data samples from the buffer and calculates a running average of 50 samples to filter out some of the signal noise. In addition, a running average of the data over 2 seconds is calculated. The program further filters out the samples that does not produce a heart rate between 30 bpm and 200 bpm. Furthermore, the program disregards any sample that differs by more than 10 bpm from the previous heartbeat to ensure accurate and reliable results. The successive data samples are then transmitted to the Kubios Cloud Service for HRV analysis. The signal measurement time required for a successful HRV analysis varies depending on the number of the bad samples in the data. However, due to the time-constrations of the final presentation of the system, the Kubios HRV analysis was set to be performed after 20 successful samples, which typically took approximately 20 seconds of gathering data.

5 Conclusions

The algorithm running on the Raspberry Pi Pico W Heart Rate
Monitor effectively measures the user’s heartbeat, transmits the data to Kubios Cloud service wirelessly and then displays average BPM, SNS and PNS value on the OLED screen, thereby meeting the requirements and goals set.

Apart from the project goals themselves, personal goals were set to gain more experience in working with hardware components and understanding how sensors measure data. These objectives were successfully met. Another goal was to improve documentation skills, which was achieved through creating multiple versions of the project report.

The project encountered significant challenges related to the algorithm, particularly in developing a reliable method for detecting the user's heart rate. It was difficult to ensure that the algorithm worked accurately enough to filter out excessive noise picked up by the sensor, and this posed a significant hurdle in ensuring that the data sent to Kubios Cloud for analysis was of good quality.
Nevertheless, through effective teamwork and a positive attitude, we were able to overcome these obstacles and produce a functional algorithm.

The results of the system testing indicated that the algorithm was effective in filtering the majority of the faulty signals caused by subjects' movements or sensor movement, resulting in accurate heart rate calculation. However, the algorithm remains sensitive to prolonged sensor movement, which could potentially affect the accuracy of the results.

The system has some additional limitations. The Raspberry Pi Pico’s reliance on a constant power source limits its flexibility, so the implementation of a wireless power source would improve its functionality.
Additionally, the Wi-Fi router must be in close proximity to the Pico for reliable internet connection, further restricting the prototype's operational range. These limitations provide insight into potential areas for improvement in future.

The prototype could be improved in the future by implementing new features and by improving the algorithm even further. Here are a couple new features that could be implemented:

- PPG-signal on the OLED-screen: This feature would allow the user to see the heart’s PPG-signal on the Raspberry Pi’s OLED-screen. Though this would take up most of the screen it should be relatively straightforward to implement
- Menu: To solve the problem presented above a menu could be implemented. The menu could have different options shown on the OLED-screen. By rotating the rotary encoder and pressing a button on the Raspberry Pi, the user could choose between the different options. Possible options for the menu could include displaying the PPG-signal, the results from the Kubios cloud and the heartbeat with some heart figure animation.

Overall, the project was successful since the system developed is a fairly reliable tool for measuring heart rate and heart rate variability in a non-invasive manner using a PPG sensor and Raspberry Pi Pico W microcontroller. However, given the time limits of it, room for development was left.

References

1. Pulse & Heart Rate. Web Article. Cleveland Clinic. 15.6.2022. <https://my.clevelandclinic.org/health/diagnostics/17402-pulse--heart-rate>
2. Figure 1. <https://agelessinvesting.com/what-is-a-good-resting-heart-rate/>
3. Wikipedia contributors. (2023, April 4). Heart rate variability. In *Wikipedia, The Free Encyclopedia*. Retrieved 08:01, April 6, 2023, from <https://en.wikipedia.org/w/index.php?title=Heart_rate_variability&oldid=1148208181>(1)
4. Figure 2. <https://www.windsorchiro.com/blog/heart-rate-variability-hrv-8-week-challenge>
5. Harvard Health Publishing Staff. 1.12.2021. Heart rate variability: How it might indicate well-being. <https://www.health.harvard.edu/blog/heart-rate-variability-new-way-track-well-2017112212789>
6. Autonomic nervous system. Web Article. Cleveland Clinic. <https://my.clevelandclinic.org/health/body/23273-autonomic-nervous-system>
7. Figure 3. <https://www.thedeepsleepco.com/parasympathetic-nervous-system/>
8. Ismail, S., Akram, U. & Siddiqi, I. Heart rate tracking in photoplethysmography signals affected by motion artifacts: a review. *EURASIP J. Adv. Signal Process.* 2021, 5 (2021). https://doi.org/10.1186/s13634-020-00714-2(3)
9. Van Gent, Paul. 2022. Python Heart Rate Analysis Toolkit Documentation. P. 54-55(4)
10. Figures 6-12. Lukkarinen, Sakari. Hardware 2 Project Requirements. (2023).
11. Lukkarinen, Sakari. Hardware 2 Project Requirements. Pages 24-29. (2023).

12. Figure 15. Hardware 2 Course Project materials, Pulse Detection. Page 12. (2023)