Ryhmäyksi: Santtu Karhu, Sohaib Ebrahimi, Miko Heino, Mixu Koski-homi

Heart rate monitoring Device

Project Report

First-year Hardware Project

School of ICT

Metropolia University of Applied Sciences

5 May 2023

Contents

1 Introduction

This report presents the results of the first-year hardware project on the Hardware 2 course. The project aims to develop a heart rate measurement and analysis system using a sensor that produces an electrical signal proportional to the heart's mechanical activity.

The motivation for the project is to create a low-cost, easy-to-use, and reliable system that can monitor heart rate and heart rate variability (HRV) for personal health and fitness tracking.

One of the primary goals of our project was to develop proficiency in photoplethysmography (PPG) technology and signal processing algorithms for heart rate detection and analysis. To achieve this goal, we conducted research on PPG technology and signal processing algorithms, including reviewing scientific literature and consulting with experts in the field. We also explored different hardware and software tools that can be used to develop PPG algorithms and experiment with different approaches to heart rate detection and analysis.

Another important goal for our team was to gain experience in programming and developing algorithms that work on the device locally. To achieve this, we researched and read about our hardware components to understand the capabilities and limitations of the device and develop algorithms that can operate efficiently on the device. We used programming language MicroPython and experimented with different software frameworks to optimize our algorithms.

We also aimed to learn how to calculate heart rate variability (HRV) analysis and interpret HRV data. While learning about the HRV analysis we also learned about KubiosCloud and how provide their API with the required PPI data to receive necessary SNS and PNS values.

One of the project objectives was to learn coding with MicroPython and a microcontroller. This was achieved through lectures, online tutorials, and hands-on experimentation with the hardware and software. The team gained practical experience in programming and a deeper understanding of how microcontrollers work.

2 Theoretical Background

The heart is a vital organ responsible for pumping blood to deliver oxygen and nutrients to the body's cells. Heart rate is the number of times the heart contracts per minute, and HRV is the variation in the time interval between heartbeats.[1]

Heart rate and HRV are essential physiological parameters that reflect the autonomic nervous system's activity and provide valuable information about the cardiovascular system's health.[1]

The physiology of heart rate measurements is based on the electrical impulses generated by the heart's pacemaker cells. Heart rate and heart rate variability (HRV) are measures of the frequency and variation of these impulses, respectively, and are typically measured in beats per minute (BPM) and milliseconds [1]. These measures are indicators of the autonomic nervous system's control over the heart and can reflect changes in physiological states such as stress or exercise.

The typical resting heart rate for adults is between 60 and 100 beats per minute (bpm) [2]. However, it can be lower for athletes and higher for individuals with certain medical conditions. The range of values for HRV depends on the measurement technique used, but it is generally between 20 and 200 Ms.

The sensor used in the project is the Crow Tail 2.0 sensor, which measures photoplethysmography (PPG) data. PPG signals are generated by shining a light source onto the skin, which causes the blood vessels to expand and contract with each heartbeat. The resulting changes in light absorption can be used to measure heart rate and HRV.

In summary, heart rate and HRV are essential indicators of cardiovascular health, and PPG technology is a non-invasive way to measure them. The use of PPG sensors has become increasingly popular due to their ease of use and low cost.

3 Methods and Material

In the project, various materials were used including a sensor, software, and devices with their respective technical properties. The sensor used was a Crow Tail 2.0 sensor, which measures photoplethysmography (PPG) data. MicroPython was used as the software, which is a version of Python optimized for microcontrollers. The devices used for the function of the device were a laptop to run our website, SQL database and backend API, and Raspberry PI Pico microcontroller (Figure 1).

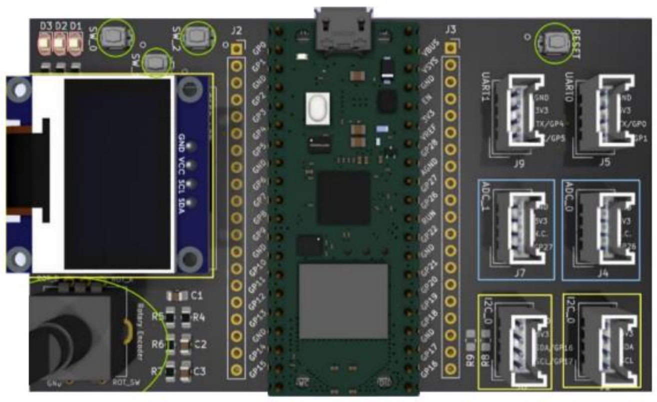


Figure 1. Raspberry Pi Pico w microcontroller with Custom connector board by J. Hotchkiss. [3]

Figure 2: Crowtail 2.0 sensor. [4]

The Crowtail Pulse Sensor is a wearable device that measures heart rate and blood oxygen saturation levels non-invasively by shining a light through the skin. The v2.8 version is an upgraded version with improved accuracy and compatibility with electronic platforms like Arduino boards (Figure 2). It can be used in health monitoring devices, fitness trackers, and wearable technology.

To test the system, we connected the sensor to the microcontroller and wrote code in MicroPython to read the data from the sensor then send that data to KubiosCloud for further analysis and display the results in the QLED display (Figure 3). Using the MicroPython-specific Framebuffer library, which provides support for graphics primitives, drawing text, and other methods.

Figure 3. SSD1306 compatible OLED-display used in the project. [5]

As our project was progressing on schedule, we had time to implement the planned additional features. First of these was the touch controls. We decided to go with DollaTek TTP223B Digital Touch Capacitive Sensors for their cheap price fast delivery so they would arrive in time (Figure 4).

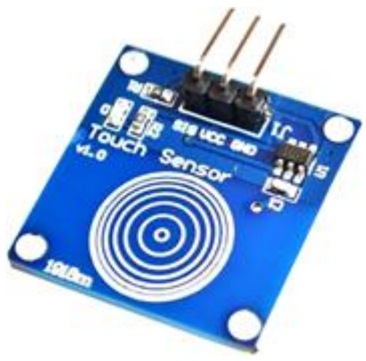


Figure 4: Capacitive sensor. [6]

These sensors are capable of detecting when a human finger goes into the electric field that they create. This allows us to hide them under the case’s lid which makes for cleaner design.

Another additional feature that we decided to go with was for the device to be battery powered and for it to be wirelessly charged.

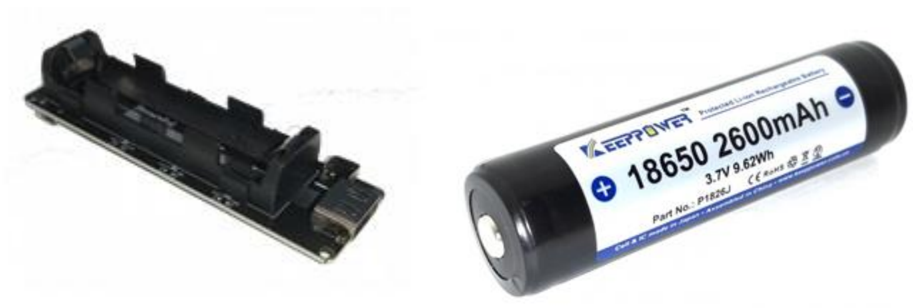


Figure 5-6: 18650 Battery holder [7] and 18650 Lithium cell [8].

For the safety of the students, we decided to go with Partco’s 18650 battery holder (Figure 5) and voltage converter which takes care of the charging of the battery and allows the output voltage to be either 3 volts or boosted 5 volts. For the battery we went with a Keeppower 18650 3,7V 2400mAh lithium cell (Figure 6) that has built in over/under voltage protection and short protection. This means that the battery has a circuit that prevents the overcharging and draw of power when the cell voltage goes too low. The short protection was for peace of mind as it would be hidden inside the case and could not be seen most of the time.

The wireless charging was implemented simply by attaching a micro-USB wireless phone charge receiver to the battery holder (Figure 7). To send the charge over, we needed a Qi wireless charger (Figure 8).

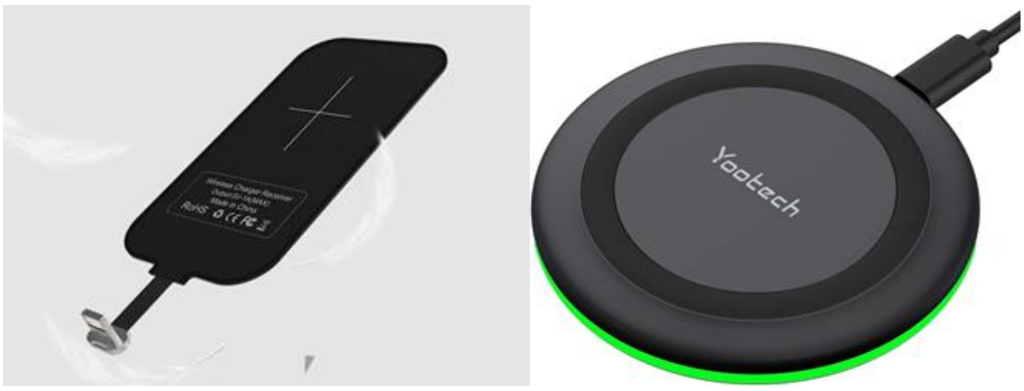


Figure 7-8: Qi enabled phone charge receiver [9] and a Wireless charger [10].

Our chosen charge receiver was a Nillkin Magic Tag Qi Wireless Charge Receiver that is capable of outputting 5 Volts at 1 Amp and the wireless charger that we chose was a Yootech Wireless Charger. We chose these for their low price, fast delivery, and ease of implementation.

4 Implementation

4.1 Code implementation

Implementation process of the project was successful as we were able to deliver a working device on time. Here we can see a system diagram that shows the simple version of how the project works (Figure 9):

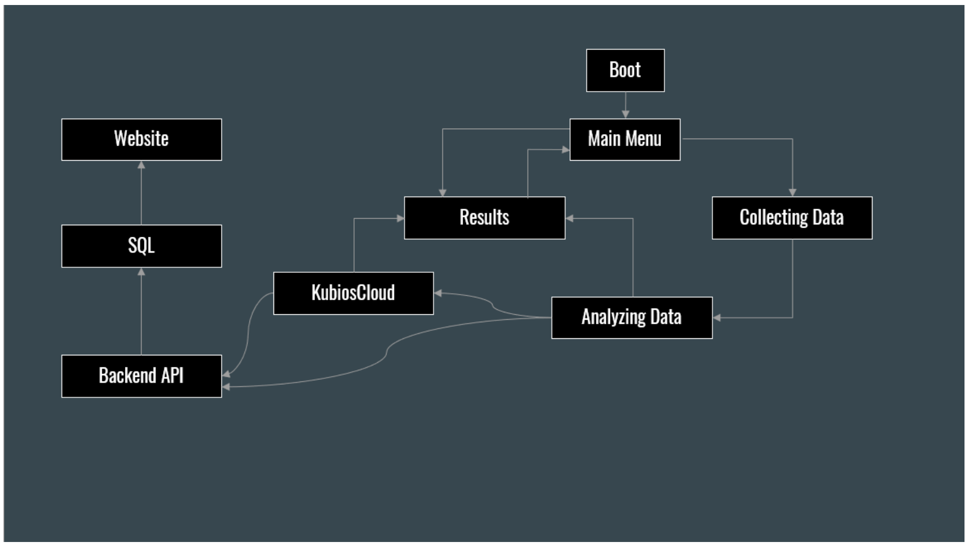


Figure 9: Template of the complete project system

First the device enters boot phase where it connects to the WI-FI using the SSID and password from the secrets file (secrets.py), after which the device enters main menu. From main menu the user can either start new data collection or if the device has already done one since it was booted, user can go back to check on those results.

In data collection phase, the device starts to save the data from the Crowtail sensor. It will then detect the peaks from said data and saves their timestamps into a list. From there it will count the intervals between the peaks and save them into another list.

After collecting the data for the required amount, the device enters analysing mode. Here the device calculates the BPM, SDNN and RMSSD values locally. Then the device sends the same PPI data over to the KubiosCloud to get stress and recovery indexes.

After the analysing phase is over, the data is sent over to our website’s backend API which will collect the received values to our SQL database. From there our website fetches the new values and displays them on the list of results. There the results can either be left as they are or be given a name to help with the identification. These results will also be displayed on the results screen where you can scroll through the values where page one is the local values, and the second page has the data from KubiosCloud.

Here is a simplified version of the loop, that the code goes through in operation from the moment it is turned on (Figure 10):

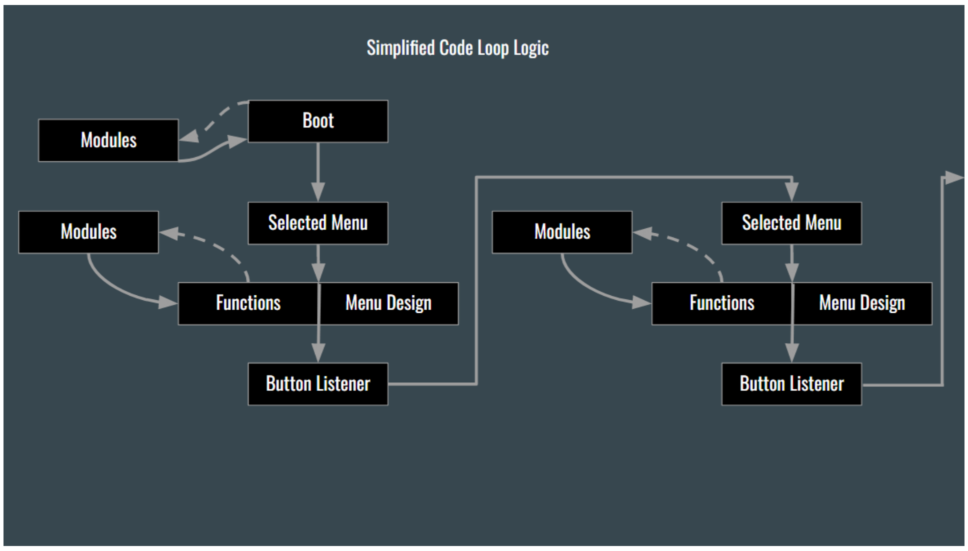


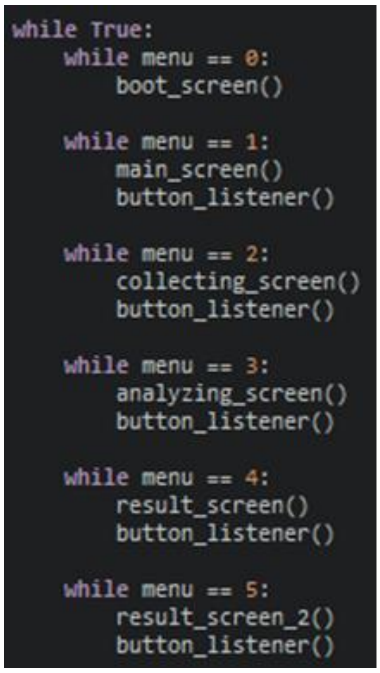
Figure 10: Simplified Code Loop Logic

When the device detects a menu change, it will change the loop it is in to the one that is chosen. When a new menu is entered, it will fetch an OLED design from the corresponding function. From here depending on which menu is being entered, required modules are imported to enable the desired functionality.

While the menu's loop is running, there will also be a button listener function running on the background to detected if a user input is given. When a button pressed, the function is going to detect which one of the two buttons is being used. From here the function will give a command depending on which menu loop its currently in.

Next, we will show an example of one such loops with explanations on what the code is going to do.

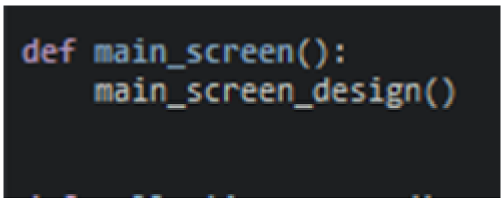
First our code will detect the loop its running in (Figure 11):



```
while True:
    while menu == 0:
        boot_screen()

    while menu == 1:
        main_screen()
        button_listener()

    while menu == 2:
        collecting_screen()
        button_listener()

    while menu == 3:
        analyzing_screen()
        button_listener()

    while menu == 4:
        result_screen()
        button_listener()

    while menu == 5:
        result_screen_2()
        button_listener()
```

Figure 11: Main “while True: loop”

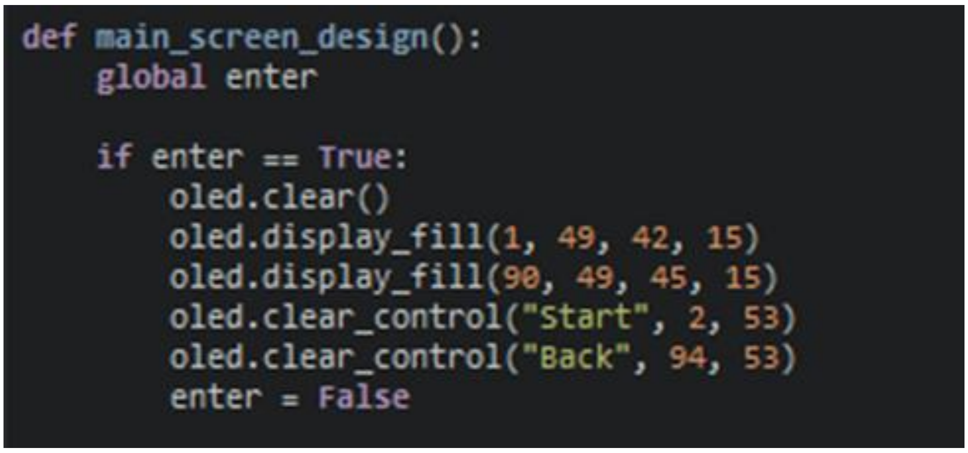
After this the code will select the required functions and modules that will be used in said menu (Figure 12):



```
def main_screen():
    main_screen_design()
```

Figure 12: Main menu’s loaded functions

As seen above, in main menu there is only one function being loaded which is the OLED screen design (Figure 13):

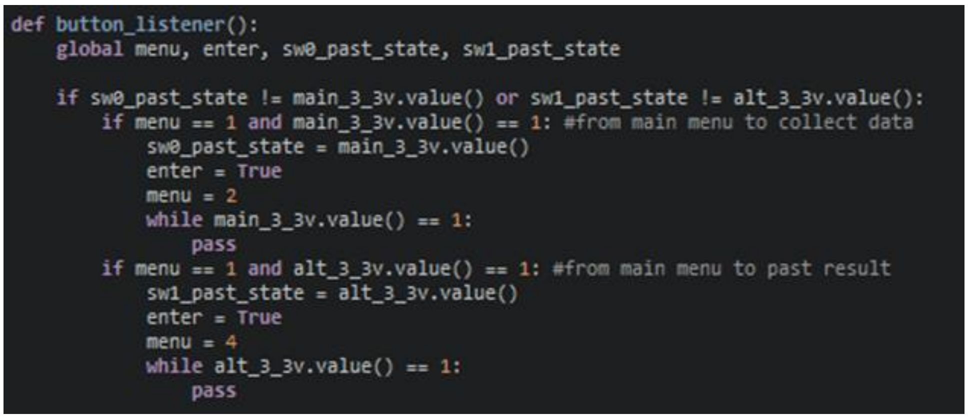


```
def main_screen_design():
    global enter

    if enter == True:
        oled.clear()
        oled.display_fill(1, 49, 42, 15)
        oled.display_fill(90, 49, 45, 15)
        oled.clear_control("Start", 2, 53)
        oled.clear_control("Back", 94, 53)
        enter = False
```

Figure 13: Main menu’s design

Our designs are using a custom OLED class that have our required commands built into it reduces the times we need to write the same code.

Last part is the button listener function that will decide what happens after a button is being pressed (Figure 14):



```
def button_listener():
    global menu, enter, sw0_past_state, sw1_past_state

    if sw0_past_state != main_3_3v.value() or sw1_past_state != alt_3_3v.value():
        if menu == 1 and main_3_3v.value() == 1: #from main menu to collect data
            sw0_past_state = main_3_3v.value()
            enter = True
            menu = 2
            while main_3_3v.value() == 1:
                pass
        if menu == 1 and alt_3_3v.value() == 1: #from main menu to past result
            sw1_past_state = alt_3_3v.value()
            enter = True
            menu = 4
            while alt_3_3v.value() == 1:
                pass
```

Figure 14: Snip from button listener code

4.2 Device implementation

From the start, our plan was to make a completely wireless device. This meant that there was going to be a lot of sensitive that components that needed to be protected. Because of this, we designed the device a custom case that the components would be mounted to.

The 3D models were designed using Microsoft’s free “3D builder” program, printed using a Creality Ender 3 v2 Neo and printed in PLA plastic. The case, lid and the finger sensor were designed to be detachable and secured in place with magnets. Unfortunately, we lacked slim and short enough USB-cable to keep the lid shut with just magnets so for the presentation, it had to be taped shut.

For the OLED screen, we attached it to the connector board using short cable extensions so it could be visible for the user when rest of the device is hidden (Figure 15).
The capacitive sensors were mounted on each side of the Crowtail sensor making it clear from the screen which control was on which side.

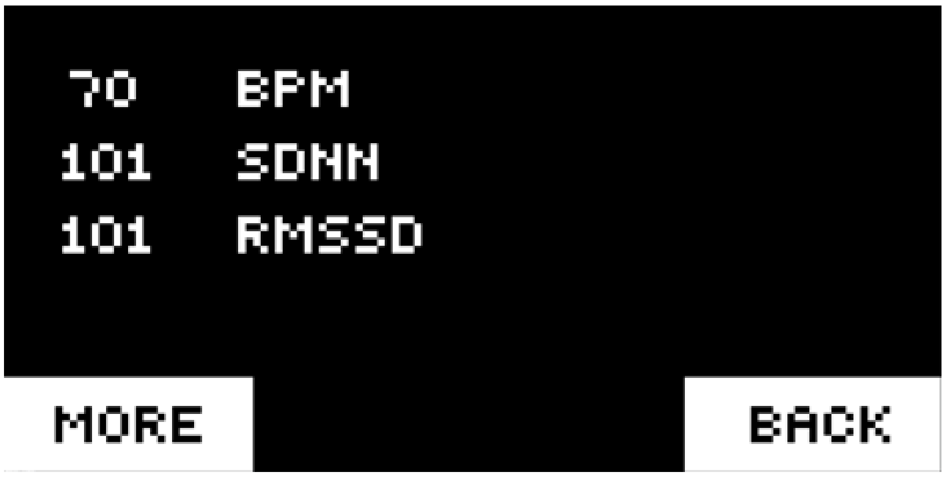


Figure 15: Example of controls on screen

These sensors were connected to a breadboard that is also connected to GP0, GP1, 3.3V and Gnd pins on the Pico’s Connector board. Here is a diagram how the connections were made on the breadboard (Figure 16):

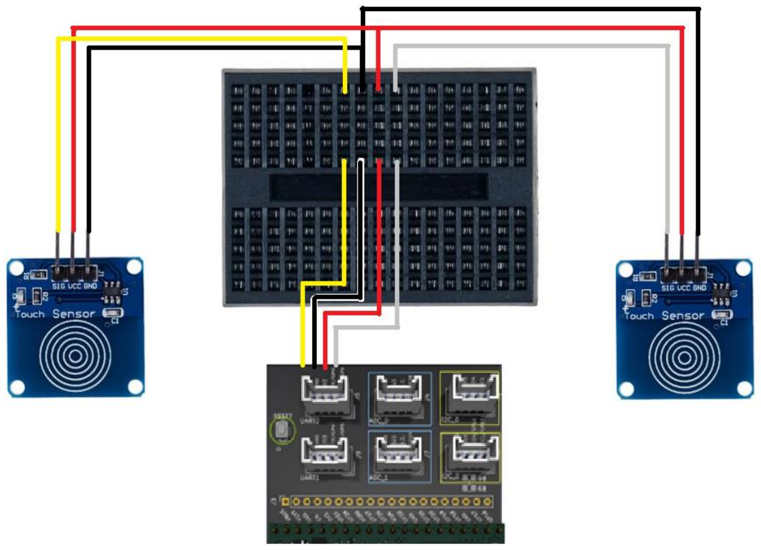


Figure 16: Capacitive touch sensor connection diagram. [11]

The Crowtail sensor was connected to the ADC connection on the custom board that had pin 26 assigned to it.

The testing process and comparison with medical device done by our team (Figure 17).

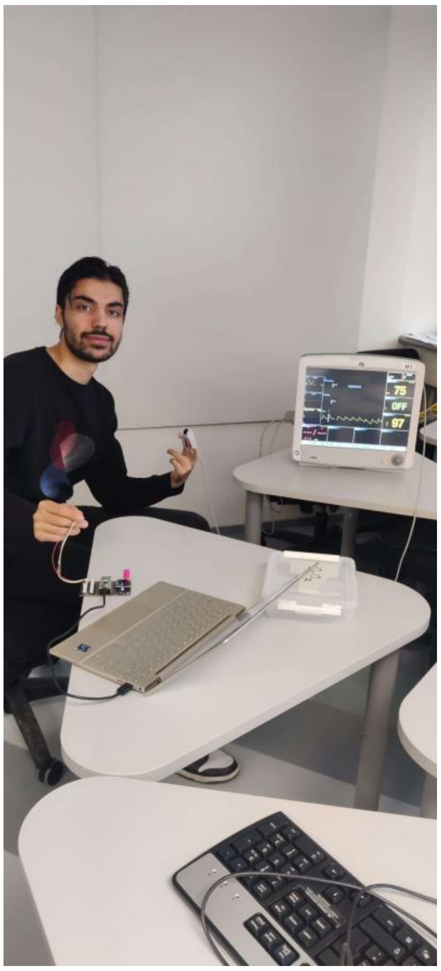


Figure 17: The system was tested by our competent team members

Our teammates should be commended for their diligent efforts in testing and documenting the results of our device. It was a pleasant surprise to discover that the readings obtained from our device were strikingly similar to those obtained from the fully functional health care machine used for comparison.

Now that all the parts were completed, we could assemble the device (Figure 18):

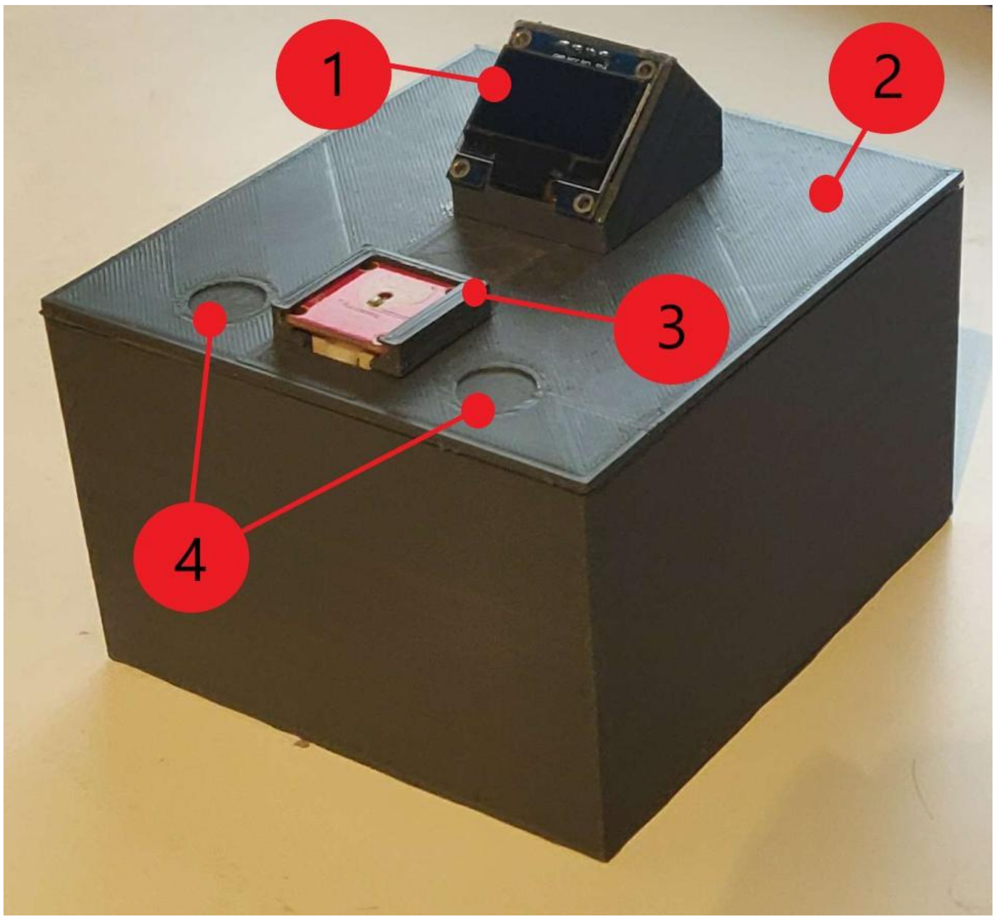


Figure 18. Picture of the completed project (see Table 1)

1. OLED display	Shows the UI of the project and provides the user with feedback.
2. 3D printed exterior	Holds all the components inside including wiring, the battery that powers the device and the raspberry pi Pico.
3. Crow Tail-pulse sensor	Allows the device to get accurate health data using an optical heart rate sensor.
4. Capacitive sensors	Touch sensitive buttons

Table 1. List of parts visible on the assembled device.

5 Conclusions

The project was successful in developing a low-cost and reliable heart rate measurement system using a PPG sensor and a Pico w board. The system can measure HRV, SNS and PNS accurately and in real-time.

The project achieved its goals of developing a functional prototype, validating its performance through experiments on human subjects and to learn and develop our skills.

Several problems occurred during the project, such as the difficulty in obtaining a stable PPG signal in some individuals, putting all the code and hardware together while everything being fully functional. We overcame these problems by optimizing the PPG signal acquisition, applying signal processing techniques, and testing the prototype several times.

The limitations of the prototype measurement system include the requirement for a stable and immobile fingertip, the limited accuracy and precision compared to professional medical devices.

The project can be improved by making the algorithm better, making the code easier to read and to add more high-quality processors and sensors to make results appear much faster and be more accurate. The current case can be made more user-friendlier and smaller.

6 References

1. Castaneda D, Esparza A, Ghamari M, Soltanpur C, Nazeran H. A review on wearable photoplethysmography sensors and their potential future applications in health care. Int J Biosens Bioelectron. 2018;4(4):195-202. doi: 10.15406/ijbsbe.2018.04.00125. Epub 2018 Aug 6. PMID: 30906922; PMCID: PMC6426305. [cited 2023 April 28]. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6426305/
2. Heart.org. All About Heart Rate (Pulse). [Internet]. [cited 2023 April 28]. Available from: https://www.heart.org/en/health-topics/high-blood-pressure/the-facts-about-high-blood-pressure/all-about-heart-rate-pulse
3. Metropolia University of Applied Sciences. Lecture slides: 03. Pins and LEDs, slide 5. [Internet]. [cited 2023 May 5]. Available from: https://oma.metropolia.fi/delegate/download_workspace_attachment/9140830/03.%20Pins%20and%20LEDs.pdf [adapted]
4. Peppe8o. Image of Crowtail-Pulse Sensor v2.0. [Internet]. [cited 2023 May 5]. Available from: https://peppe8o.com/pulse-sensor-with-raspberry-pi-pico-hearth-beat-chech-with-micropython/
5. Winstar. Image of product: Model WEA012864D-03. [Internet]. [cited 2023 May 5]. Available from: https://www.winstar.com.tw/products/oled-module/graphic-oled-display/4-pin-oled.html
6. Amazon. Image of product: DollaTek DIY 5PCS TTP223B Digital Touch Capacitive Sensor Switch Module for Arduino. [Internet]. [cited 2023 May 5]. Available from: https://www.amazon.co.uk/DollaTek-TTP223B-Digital-Capacitive-Arduino/dp/B07DJ4XDTV
7. Partco. Image of product: PARPID 18560USB. [Internet]. [cited 2023 May 5]. Available from: https://www.partco.fi/fi/akut-ja-paristot/paristopitimet/20272-parpid-18560usb.html
8. Partco. Image of product: Z-LI 18650N. [Internet]. [cited 2023 May 5]. Available from: https://www.partco.fi/fi/akut-ja-paristot/akkukennot/li-ion-akut/15873-z-li-18650n.html

9. Amazon. Image of product: Nillkin iPhone Wireless Charging Receiver Magic Tag Qi Wireless Charger Receiver 1000mAh for iPhone 7/6/6S/Plus. [Internet]. [cited 2023 May 5]. [Adapted]. Available from: https://www.amazon.com/Nillkin-Wireless-Charging-Receiver-Charger/dp/B01DLYF33K
10. Amazon. Image of product: Yootech Wireless Charger,10W Max Fast Wireless Charging Pad Compatible with iPhone 14/14 Plus/14 Pro/14 Pro Max/13/13 Mini/SE 2022/12/11/X/8. [Internet]. [cited 2023 May 5] Available from: https://www.amazon.com/Wireless-Qi-Certified-Charging-Compatible-Qi-Enabled/dp/B079KZ49PJ/ref=sr_1_3?crid=3K9BJ35UXW4Z3&keywords=Yootech%2BWireless%2BCharger&qid=1683962278&sprefix=yootech%2Bwireless%2Bcharger%2Caps%2C192&sr=8-3&th=1
11. Robotistan. Image of product: Black Mini Breadboard. Including references 1 and 4. [Internet]. [cited 2023 May 5]. [Adapted]. Available from: https://www.robotistan.com/back-mini-breadboard