Team name: Siddarth Singotam, Fatemeh Ramezan Nasab Shomia, Mohammed Nadeem

Cardiovison

Project Report

First-year Hardware Project

School of ICT

Metropolia University of Applied Sciences

08 May 2023

Abstract – Siddarth Singotam

Heart rate detection is an important area of study that involves the measurement and analysis of a person's heart rate. The heart rate is a critical indicator of the health and fitness level of a person and can provide valuable information about their physical condition.

Heart rate variability (HRV) accurately assesses the autonomic nervous system (ANS) function. HRV is widely used by health and wellbeing professionals to objectively measure physiological and mental stress and recovery. In addition, HRV is a commonly used tool in the research of different cardiovascular and metabolic diseases and their risk factors.

The aim was to build a working concept of an HRV device. The Cardiovision module is intended to be used in home or office environments either by the end users themselves or together with health and wellbeing professionals such as physiotherapists, nurses and medical doctors.

The system consists of an Optical Sensor, Raspberry Pi Pico W, and an SSD1306 OLED display connected to a custom PCB framework board. The module could read sensor data and process it accordingly using a heart rate monitoring algorithm. The module could also communicate and process HRV data using the Kubios API software.

Version History

Ver	Description	Date	Author(s)
1.0	Added project title to the report (“Cardiovision”)	09/04/2023	Siddarth Singotam
1.1	Listed the Introduction, Implementation section	09/04/2023	Siddarth Singotam
1.2	Theoretical background – Added HRV measurement and physiological phenomena associated with heart rate and HRV.	09/04/2023	Mohammed Nadeem
1.3	Theoretical background – Listed the typical values for minimum and maximum rates for the measurements. Mentioned the type of sensor used in the project.	09/04/2023	Fatemeh Ramezan Nasab Shomia
1.4	Methods and Materials	23/04/2023	Fatemeh Ramezan Nasab Shomia
1.5	Updated Implementation	07/05/2023	Siddarth Singotam
1.6	Added abstract	08/05/2023	Siddarth Singotam

Table of Contents

1 Introduction – Siddarth Singotam

Heart rate detection is an important area of study that involves the measurement and analysis of a person's heart rate. The heart rate is a critical indicator of the health and fitness level of a person and can provide valuable information about their physical condition. In recent years, there has been a growing interest in developing non-invasive methods for measuring heart rate, such as the use of pulse sensors and microcontrollers. [1][2] Measuring stress levels and monitoring recovery are important in various fields, including sports, healthcare, and occupational safety.

Heart rate variability (HRV) is an accurate method to assess the autonomic nervous system (ANS) function. HRV is widely used by health and wellbeing professionals to objectively measure physiological and mental stress and recovery. In addition, HRV is a commonly used tool in the research of different cardiovascular and metabolic diseases and their risk factors.[3]

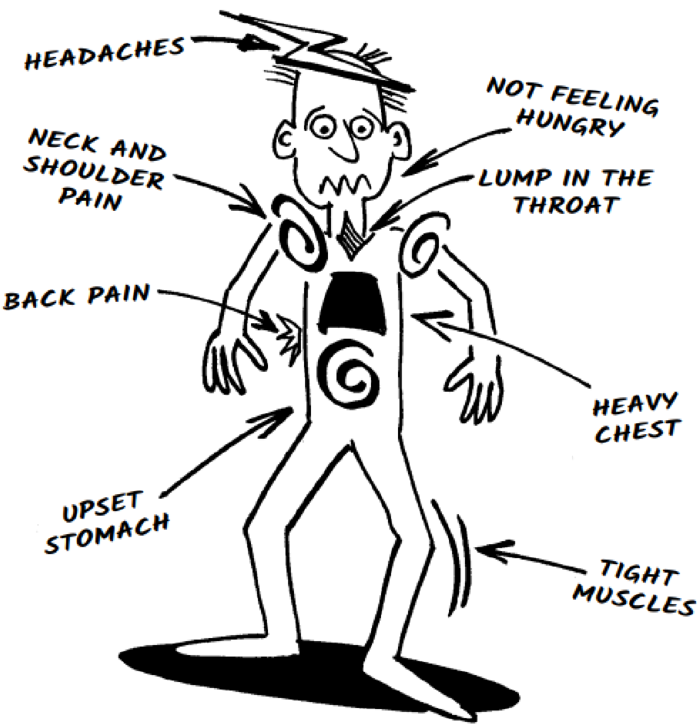

Figure 1.1 Stress and its physical effects on the human body.

The aim of the project is to design and implement a heart rate detection system using a custom PCB framework, a pulse sensor, and a Pico Raspberry Pi microcontroller. The system will be capable of measuring a person's heart rate and displaying the HRV data result on an OLED screen. The device is intended to be used in home or office environments either by the end users themselves or together with health and wellbeing professionals such as physiotherapists, nurses or medical doctors.

The project is motivated by practical applications in health technology, learning experience in hardware design, problem-solving in non-invasive heart rate measurement, innovation potential, and fulfilling course requirements. The motivation stems from creating a cost-effective solution for measuring heart rate, applying theoretical knowledge to a real-world project, and meeting the requirements of the Hardware 2 course.

The goals of the project include achieving a high-quality outcome that meets the requirements for a good grade in the Hardware 2 course. The team aims to design and build a fully functional heart rate detection system without expensive medical equipment, and gain practical experience in working with electrical components, sensors, and microcontrollers. Additionally, other aspects of the goals of the project are to develop project management and collaboration skills and showcase proficiency in designing and building a system from scratch. In Figure 1, a sample of the Cardiovision module is shown.

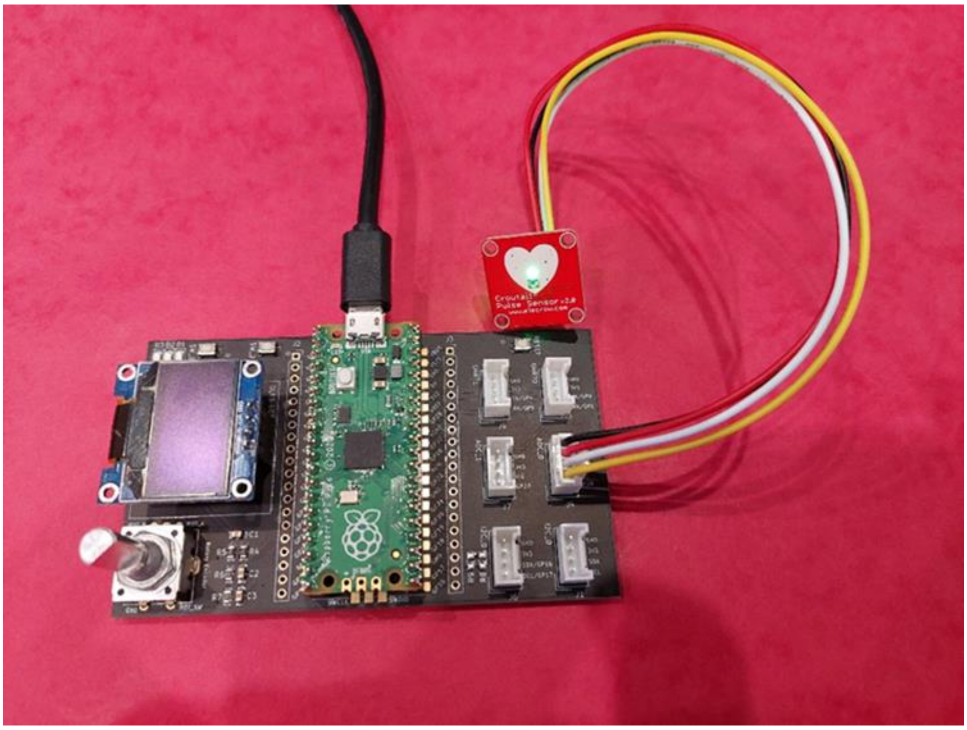

Figure 1.2 A sample of the Cardiovision Module

2 Theoretical Background – Mohammad & Fatemeh

Technology has done wonders when it comes to the health sector. The diagnosis done now a days in hospitals and health centers is always a combination of technologically advanced devices. Cardiovision is also an inspiration which is all about enabling people to measure their health status by using a handheld device.

Heart function can be the basis of many diseases and pre detection can help in saving precious lives. This device can also help in maintaining the healthy lifestyle of people.

Heart rate variability commonly known as HRV is a calculation of time between the heartbeat fluctuations. Some time the fluctuation can be undetected if not calculated with specialized devices . They have a direct link to the overall health of the human being if there is any fluctuations in the heart rate. The time is measured in milli seconds (ms) and is known as inter beat interval.[4]
In Figure 2, an example of HRV measurements graph is shown.

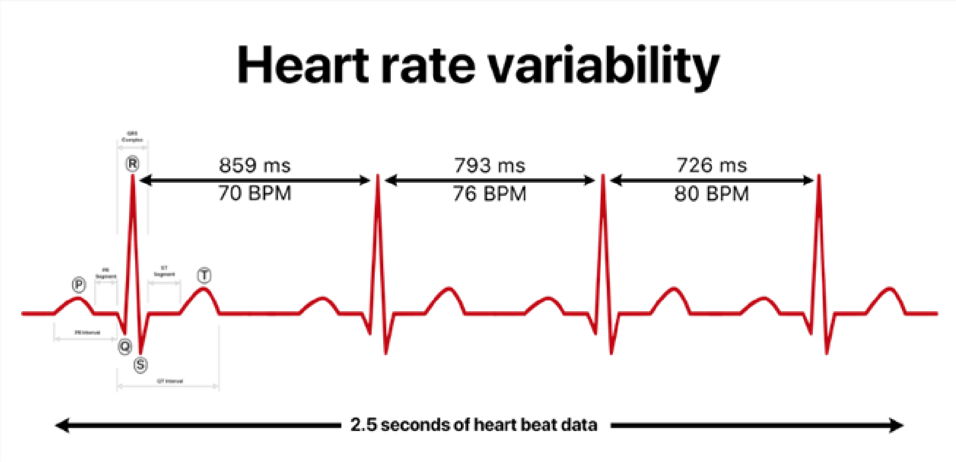

Figure 2.1 HRV measurements graph

HRV can be associated with a lot of physiological phenomena such as stress, anxiety and hypertension. People under emotional distress and elevated anxiety can also have fluctuating HRV values. This can also indicate several chronic diseases such as limited heart function, diabetes or coronary diseases. The healthy HRV is 100+ ms and any value below this is considered unhealthy. While below 50 is considered life threatening as shown in the below chart.[5]
In Figure 3, different scores of HRV is shown.

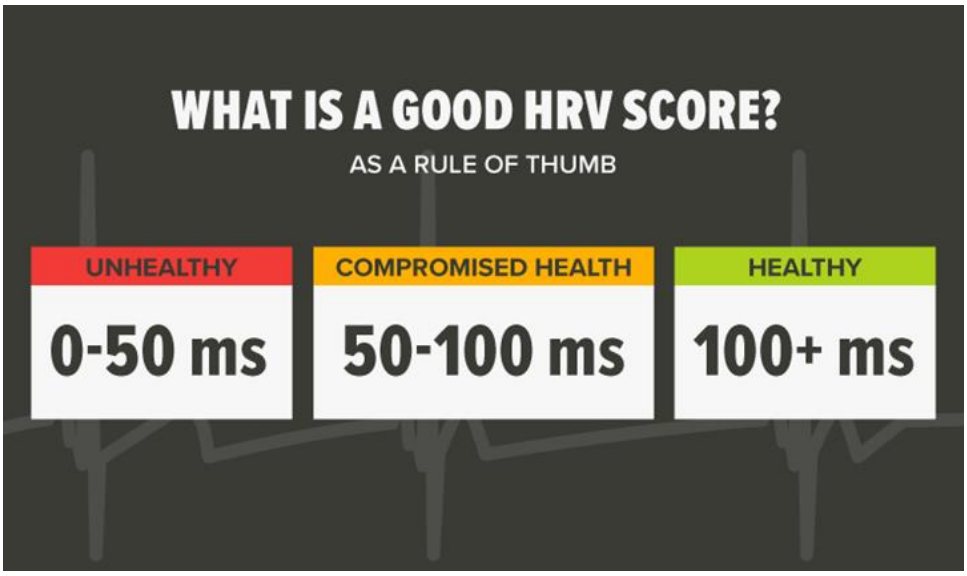

Figure 2.2 HRV score [6]

Heart rate and heart rate variability (HRV) are important indicators of cardiovascular health and fitness. The normal range for resting heart rate is generally considered to be between 60-100 beats per minute (bpm) for adults, with the average being around 72 bpm [7]. However, the range of values can vary depending on factors such as age, gender, fitness level, and health status. For example, highly trained athletes may have resting heart rates as low as 40 bpm, while individuals with certain medical conditions or taking certain medications may have higher resting heart rates [8]. The range of heart rate variability can also vary, with a higher variability generally considered to be a sign of better cardiovascular health and fitness [9].

The sensor used in this project is typically a photoplethysmography (PPG) sensor, which produces a signal by measuring changes in blood volume in the skin. Specifically, PPG sensors use light to measure the absorption of blood in the capillaries, and the signal produced is a series of peaks and troughs that correspond to each heartbeat [10]. By analyzing the time between these peaks, the heart rate can be calculated. HRV can also be calculated by analyzing the variability of the time between these peaks [11]. This is done by emitting light into the skin and measuring the amount of light that is absorbed or reflected by the blood vessels. As the heart beats, the blood vessels expand and contract, causing changes in the amount of light that is absorbed or reflected. These changes are detected by the sensor and used to calculate the heart rate and other parameters such as HRV [12].

The optical method (known as photoplethysmography) measures heart rate by sensing changes in blood flow through the index finger. A plot of this change recorded against time is called a photoplethysmographic waveform (PPG). In Figure 4, a sample of plot of the change recorded against time that is called a photoplethysmographic waveform (PPG) is shown.

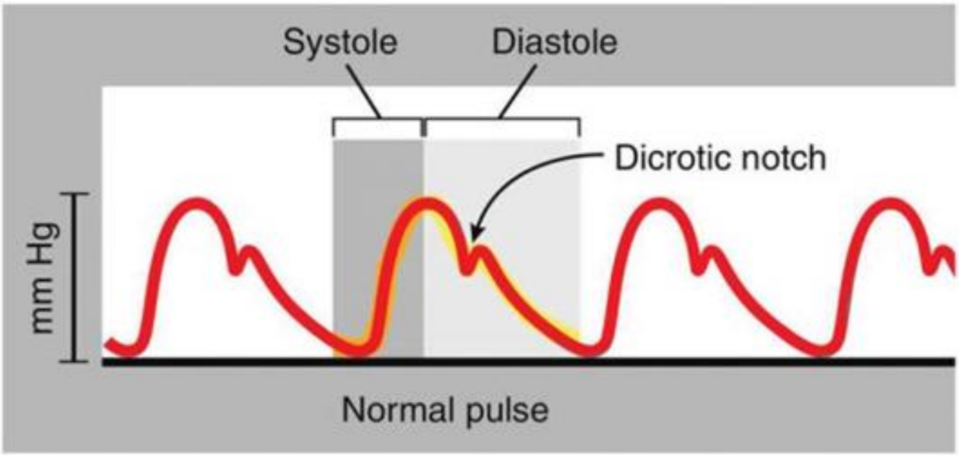

Figure 2.3 A sample of plot of the change recorded against time that is called a photoplethysmographic waveform (PPG)

In summary, heart rate and HRV are important indicators of cardiovascular health and fitness, with normal ranges varying depending on factors such as age, gender, fitness level, and health status. PPG sensors produce a signal based on changes in blood volume in the skin, which can be used to measure heart rate and HRV.

In the next section, the methods and materials used for the project will be introduced.

3 Methods and Material - Fatemeh

The Heart rate monitor was built using a Raspberry Pi Pico W microcontroller board, a Crowtail-Pulse Sensor v2.0, and an SSD1306 OLED display.

The Raspberry Pi Pico board is a powerful microcontroller board with several features, including a dual-core Arm Cortex processor, 264 kB on-chip RAM, 2 MB on-board Flash RAM, 26 multifunction GPIO, 3 analog inputs, 2x UART, 2 x SPI, 2x I2C controllers, 16x PWM channels, 1x USB, and 2.4 GHz wireless LAN (in the PicoW version). The Raspberry Pi Pico board was connected to the USB port of the computer through a micro USB B cable.

The Crowtail-Pulse Sensor v2.0 was used in the system to detect the heart rate. It is a plug-and-play heart-rate sensor module that uses a photoplethysmography

(PPG) technique to measure the heart rate. The sensor was connected to the ADC port of Raspberry Pi Pico board through connector includes 4 wires.
The SSD1306 OLED display was used in the system to display the heart rate and HRV results. It is a 128 * 64 OLED display that was connected to the Raspberry Pi Pico board through its GPIO pins.
The Thonny Python IDE was used to program the Raspberry Pi Pico board using the MicroPython firmware, which is a special version of the standard Python 3.4 that is designed to run on microcontrollers such as the RP2040, ESP32, and WiPy. MicroPython takes into consideration the hardware limitations of microcontrollers, such as limited memory and number representations.

Several libraries, such as filefifo.py, fifo.py, led.py, piotimer.py, and ssd1306.mpy were also used from https://gitlab.metropolia.fi/lansk/picow-mip-example to help with the development of the system.
All the required micropython codes are written in Tohny and to ensure the correctness of the program and due to the possibility of errors during sampling, instead of taking input from the sensor pulse by placing a finger on the sensor, from the data in the file capture03_250Hz.txt in the Data section Used at https://gitlab.metropolia.fi/lansk/picow-mip-example/-/tree/main/data. After running the program, the correctness of the heart rates printed by the program was measured by matching and comparing the results with the Heart Rate values capture03_hr file in the material posted for the project lesson on the Metropolia Oma website(oma.metropolia.fi).
The method used in this project is to calculate heart rate using the following formulas:

$$PPI = n \cdot t_s \text{ (in ms)} \rightarrow HR = 60/(PPI/1000) \text{ (in bpm)}$$

which requires time to be included in the program to calculate PPi. Also, for to reach to correct result for PPi , a program was written to identify the maximum peaks that are higher than the average of the samples.
To ensure the accuracy and reliability of the system, it was tested on human subjects with different heart rates and physical conditions. This helped to verify that the system produced valid readings for the measured parameters.

In the next step, it is necessary to send the obtained values for ppi to Kubios using the network connection.

Kubios Cloud is an ideal solution for projects that require daily heart rate variability (HRV) data from a group of people or for those who want to integrate the Kubios HRV analysis algorithms into their application. This analysis service features gold-standard HRV algorithms that have been scientifically validated and widely used by professionals and researchers worldwide. It includes an accurate pre-processing algorithm for detecting missed, extra, and misaligned beats, with a 97.0% accuracy rate in detecting ectopic beats and 99.9% accuracy in identifying normal sinus beats. The HRV data is also detrended to minimize the effect of slowly changing heart rate baseline on the short-term HRV analysis results. The detailed HRV analysis includes Parasympathetic nervous system (PNS) and sympathetic nervous system (SNS) indexes to assess changes in recovery and stress, time-domain, frequency-domain, and nonlinear HRV parameters, as well as an individual Readiness index when daily history is provided/available. Additionally, the service provides respiratory rate in breaths/min, evaluated using a validated algorithm. Kubios Cloud runs in Amazon Web Services and a documented RESTful API is available, making it reliable and easy to integrate. Users can use the Kubios HRV app and its Team functionality to start collecting HRV data or analyze data coming from their own application.[13]

Overall, the Cardiovison heart rate detection system was designed and tested to provide accurate and reliable heart rate and HRV measurements in a non-invasive and cost-effective manner using the Raspberry Pi Pico board, Crowtail-Pulse Sensor v2.0, SSD1306 OLED display, and several libraries and firmware specific for the PicoW version. Also, The system uses a formula to calculate heart rate, which requires time to be included in the program, and an algorithm and program was written to identify the maximum peaks for accurate results.

Finally, by writing codes to connect Raspberry Pi to the Internet and also call SNS and PNS values from Kubios site using .json codes the project was completed. In Figure 5, an overview of the materials and steps and methods of the heart rate monitoring project is shown.

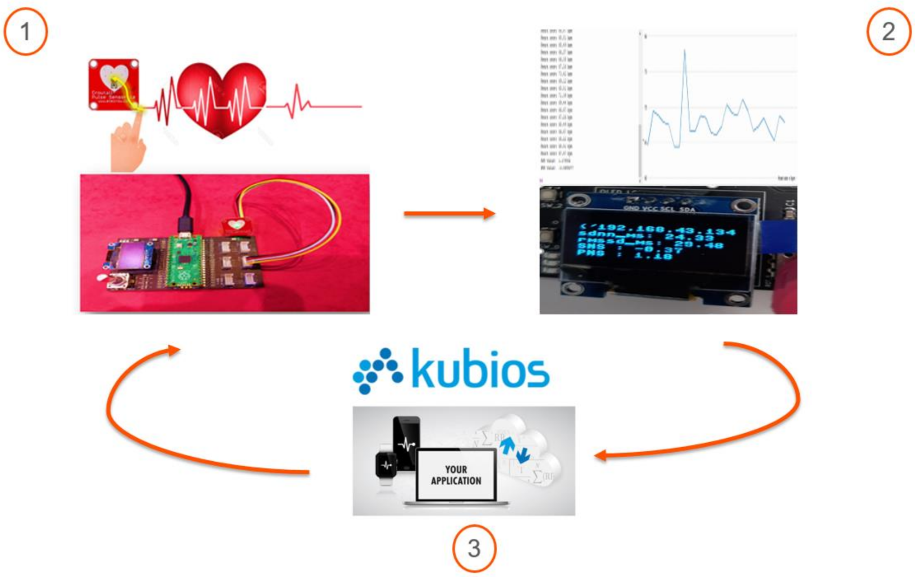

Figure 3.1 The overview of the materials and steps and methods of the heart rate monitoring project

Finally, the results obtained from Kubios for RMSSD and SDNN are compared with the calculations included in the program using the following formulas to check the correct percentage of the calculations.

$$\mathrm{SDNN} = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N}(\mathrm{RR}_n - \overline{\mathrm{RR}})^2}$$

Figure 3.2 SDNN formula

$$\mathrm{RMSSD} = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N-1}(\mathrm{RR}_{n+1} - \mathrm{RR}_n)^2}.$$

Figure 3.3 RMSSD formula

In the next section, the detailed explanation of the heart rate calculation algorithm and the implementation of the codes related to the project will be discussed.

4 Implementation – Siddarth Singotam

4.1 System and Operation

The Diagram below shows the simplified Operating Principle of the final system. The system consists of an Optical Sensor (1x), Raspberry Pi Pico W (1x), and an SSD1306 OLED display (1x) connected to a custom PCB framework board (1x).

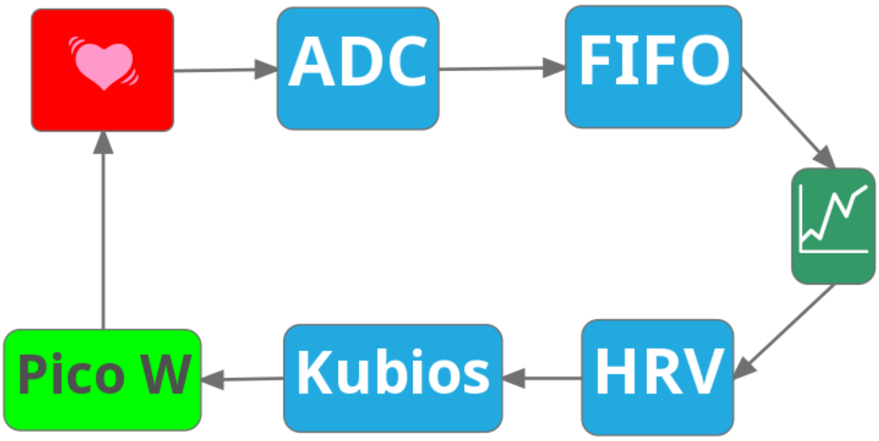

Figure 4.1.1 A simplified system of the implementation process. Start from Pico W

The Pico W is connected to the internet. The system uses MicroPython[14] to run the algorithms. The Pico W is connected to the Kubios Cloud API software service.

The Pico W is first connected to the Internet. The Sensor connected to Pico W collects the analog signals measured from the part of the body where the pulse is observed. The analog signals are then converted into digital signals with the help of an ADC [15] converter run in the code. The converted digital signal values are then put in a FIFO buffer [16], where the values are stored/ deleted in the order they are collected and processed. These collected values are then run in a Heart Rate Detection algorithm to measure the heart rate and Peak-to-Peak Interval (PPI).

With The Help of the measured heart rate and PPI, Heart Rate Variability (HRV) data can be calculated locally. The PPI measured and collected is also sent to Kubios Cloud API, which analyzes and processes the data. The output is printed on the OLED display.

4.2 Algorithms and Data Processing

The diagram below shows the simplified workflow structure of the algorithm in the final version of the project code. The Pico W is connected to the internet when the main program is run.

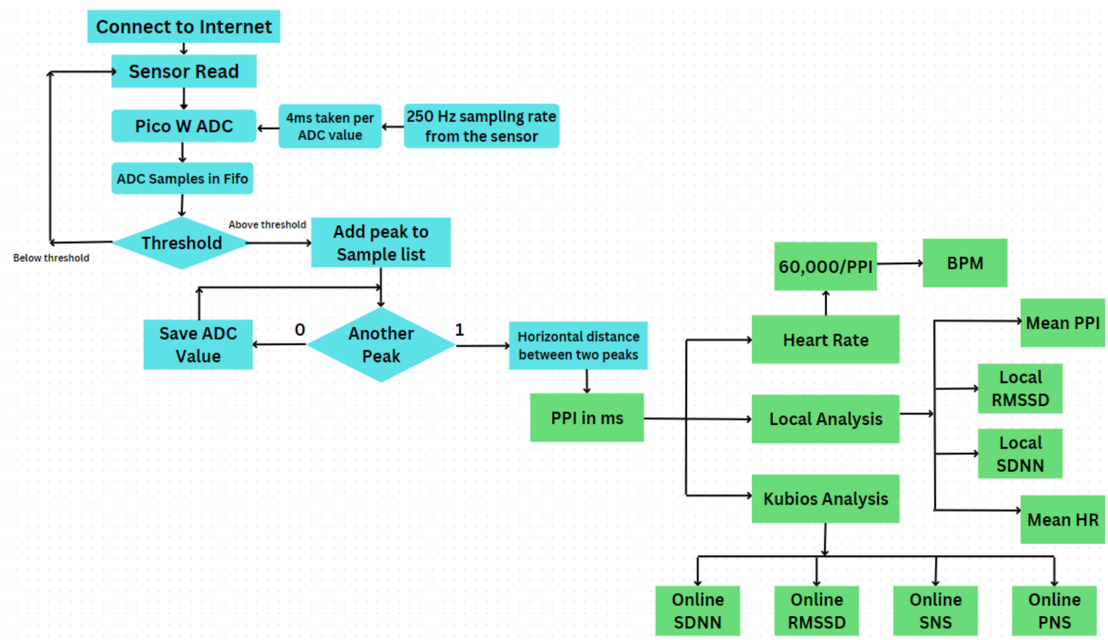

Figure 4.2.1 Workflow of the algorithm

As seen from the above diagram, the program constantly collects all the values from the sensor. The values are then collected in a fixed sample rate of 250 Hz or every 4ms from a callback function using a hardware interrupt and are put in a FIFO framebuffer. The voltage values are converted into digital values (ADC count) using an ADC converter.

The ADC count values are then appended to a list of the size of 750 samples. From the sample list the maximum and the minimum sample values present in the list are extracted to scale up the threshold. The formula for scaling the threshold is given below. Note that the threshold value in fig 4.2.3 is the simplified version of the formula in fig 4.2.2 below.

$$threshold = \frac{(max + min)}{2} + \frac{(max - min)}{10}$$

Figure 4.2.2 Threshold value formula used

```
if len(samples) >= 750:
    max_sample = max(samples)
    min_sample = min(samples)
    threshold = (3*max_sample + 2*min_sample)/5
```

Figure 4.2.3 Scaling the threshold value

Now the rise in peaks are detected in the algorithm using the threshold. A counter is run to find the calculate the sample distance between two successive rise peaks. Once the rise in peak is detected, The code automatically calculates the Peak-to-Peak Interval (PPI) and stores them in a list. PPI in is the product of the counter run between the rise of peaks times the time taken for each interval. Since the hardware interrupt is called every 4ms, it is the time taken for each interval to measure.

```
for s in samples:
    if s >= threshold and prev < threshold:
        peaks.append(counter)
    counter += 1
    prev = s

for i in range(1, len(peaks)):
    delta_gap = peaks[i] - peaks[i-1]
    ppi = delta_gap * interval_gap_ms
```

Figure 4.2.4 Calculating PPI between two rises in peaks

The PPI values can be used to calculate the heart rate in Beats per minute (BPM). Heart rate is calculated by 60,000 divided by PPI in milliseconds (ms). The range for recording heart rates is between 30 and 240 BPM, which is the average range of heart rate for an average human being. This was done so to prevent irregular readings.

```
if ppi > 0:
    heart_rate = 60000/ppi
    if heart_rate > min_hr and heart_rate < max_hr:
        print(f'Heart Rate: {round(heart_rate)}')
```

Figure 4.2.5 Calculating the heart rate using PPI in ms

When the program has collected enough PPI samples, the loop for collecting the samples is interrupted, and the Heart Rate Variability (HRV) data analysis begins. With The Help of the collected PPI list, HRV data can be calculated locally. The PPI measured and collected is also sent to Kubios Cloud API, which analyzes and processes the data sent and sends a large amount of data as a JSON response, which includes the SNS and PNS values. After all the data is processed the Program prints the local and online HRV data which includes the mean PPI, heart rate, SDNN, RMSSD and online SDNN, online RMSSD, PNS, and SNS index values respectively. This whole process can be repeated by re-running the main program.

Figure 4.2.6 HRV local and Kubios analysis output sample

4.3 Demo and Results

The Image sample below shows the overall demo and results printed on the display of the module, included in the implementation in the final project code. The demo of the module is based on the logic workflow of Fig. 4.2.1.

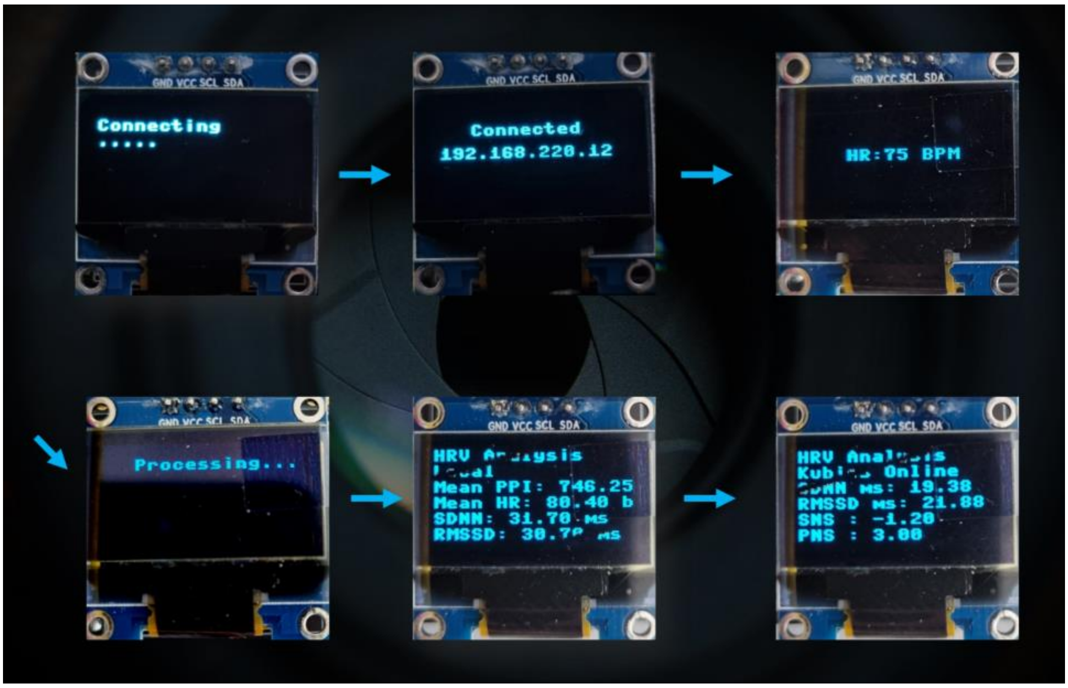

Figure 4.3.1 Demo and Results of Implementation in the module

5 Conclusions – Mohammed Nadeem

Instructions (REMOVE WHEN READY): In the conclusions section, you should wrap up the report and evaluate how well the original goals of the project were achieved.

The section should answer the following questions:

1. How well did the project go?

2. Were the goals reached?
3. What kind of problems occurred during the project? How did you handle them?
4. What limitations does your prototype measurement system have?
5. How could the work be improved or continued in the future?

References & Links

1. "Heart rate measurement." MedlinePlus, U.S. National Library of Medicine, 1 Mar. 2021, medlineplus.gov/ency/article/007253.htm.

2. Narkiewicz, K. "Heart rate as a therapeutic target in hypertension." Journal of Hypertension, vol. 35, no. 1, 2017, pp. 7–14., doi:10.1097/hjh.0000000000001142.
3. Kubios, "Wellbeing and health application of HRV," Kubios Oy, [Online]. Available: https://www.kubios.com/health-and-wellbeing/. [Accessed 23 09 2022].
4. https://my.clevelandclinic.org/health/symptoms/21773-heart-rate-variability-hrv#:~:text=Heart%20rate%20variability%20is%20where,issues%20like%20anxiety%20and%20depression.
5. https://en.wikipedia.org/wiki/Heart_rate_variability
6. https://blog.mymetabolicmeals.com/hrv-guide/#:~:text=HRV%20is%20a%20highly%20sensitive,%2C%20resilience%2C%20and%20increased%20fitness.
7. American Heart Association. (2022). Heart rate. https://www.heart.org/en/health-topics/high-blood-pressure/changes-you-can-make-to-manage-high-blood-pressure/how-to-monitor-and-record-your-blood-pressure/heart-rate
8. Eijsvogels, T. M. H., Molossi, S., Lee, D. C., Emery, M. S., & Thompson, P. D. (2016). Exercise at the Extremes: The Amount of Exercise to Reduce Cardiovascular Events. Journal of the American College of Cardiology, 67(3), 316–329. https://doi.org/10.1016/j.jacc.2015.11.034
9. HeartMath Institute. (2022). Heart Rate Variability (HRV) and Coherence. https://www.heartmath.com/science/heart-rate-variability/
10. Allen, J. (2007). Photoplethysmography and its application in clinical physiological measurement. Physiological Measurement, 28(3), R1–R39. https://doi.org/10.1088/0967-3334/28/3/R01
11. Shaffer, F., & Ginsberg, J. P. (2017). An Overview of Heart Rate Variability Metrics and Norms. Frontiers in Public Health, 5. https://doi.org/10.3389/fpubh.2017.00258

12. Allen, J. (2007). Photoplethysmography and its application in clinical physiological measurement. Physiological Measurement, 28(3), R1–R39. https://doi.org/10.1088/0967-3334/28/3/R01
13. https://www.kubios.com/kubios-cloud/
14. https://micropython.org/
15. https://docs.micropython.org/en/latest/esp8266/tutorial/adc.html
16. https://gitlab.metropolia.fi/lansk/mp-lib/-/blob/main/fifo.py