Group 105:

RECOVERY AND STRESS METER

Project Report

First-year Hardware Project
School of ICT
Metropolia University of Applied Sciences
03 May 2023

Abstract

The project aimed to design and develop a software for an interactive embedded system capable of pulse detection, and with the collected data to analyse the heart rate variability parameters. For the detection of the heart signal, the optical photoplethysmography (PPG) technique was used to extract peak-to-peak intervals and the displayed parameters were partially calculated locally with created algorithms, and partially retrieved from Kubios Cloud Service based on PPI values gathered during the measurement.

For the implementation, Raspberry Pi Pico W was used with CrowTail Pulse Sensor v2.0 as the sensor of the hardware. In addition, further components of the protoboard were also utilized, such as rotary encoder, OLED display and LEDs. For the programming language MicroPython was selected and Thonny as IDE of the project.

The project was successfully executed by the team’s collective effort. The product meets the set requirements with few limitations. These limitations are for instance unsatisfactory measurements when PPG data is gathered from less vascular areas of the body, and the unreliable response from Kubios Server.

For future development, signal filtering could be implemented to the existing programme, and by improving the current peak detection algorithm, the accuracy could be increased.

Contents

Definitions, acronyms and abbreviations

ADC	Analog-digital converter
BPM	Beat per minute
ECG	Electrocardiogram
GPIO	General purpose input output
HR	Heart rate, unit given as beat per minute (BPM)
HRV	Heart rate variability, measures the variability of beat to beat, characterized by several parameters
Hz	Hertz, unit for frequency
IBI	Inter-beat-interval, unit: milliseconds
I2C	Inter-Integrated Circuit
LAN	Local area network
LED	Light emitting diode
ms	Millisecond
NN	Time interval between two consequential two peaks, unit: milliseconds
OLED	Organic light-emitting diode
Poincaré plot	recurrence plot used to quantify self-similarity in processes
PPI	peak-to-peak interval, time interval between two consequential peaks, unit: milliseconds
PPG	Photoplethysmography, optically heart signal detection from peripheral blood circulation
PNS	Parasympathetic nervous system
PWM	Pulse-width modulation
RMSSD	The square root of sum of squares of differences, unit: milliseconds
SD1	Poincaré plot index, estimation of the first ellipse shape parameter
SD1	Poincaré plot index, estimation of the second ellipse shape parameter

SDNN	Standard deviation of all NN intervals, unit: milliseconds
SDSD	Standard deviation of differences between adjacent NN intervals, unit: milliseconds
SNS	Sympathetic nervous system
SSID	Service Set Identifier, primary name associated with an 802.11 WLAN
WLAN	Wireless local area network

1 Introduction

The Recovery and Stress Meter project is implemented during Hardware 2 at Metropolia University of Applied Sciences. The project aims to create a hardware device capable of heart rate detection and, through the analysis of the heart rate variability, provide feedback on stress and recovery status indexes. The purpose of the project is to introduce the basic concepts of hardware engineering to first year ICT students, which results in a product that can be utilized in non-clinical environments by both medically trained and untrained users.

The documentation describes the purpose and the structure of the project, details the necessary technical features for successful implementation, and provides an insight to the challenges and the outcome of the project. The intended audiences of the report are primarily lecturers and students. The report also takes into consideration readers with lesser technical background knowledge.

This report consists of five major chapters. Chapter 1 presents a very brief, general introduction to the recovery and stress meter. Chapter 2 describes the theoretical and scientific background of the measurements and indexes. Chapter 3 outlines the methods and materials applied for this project. The result is detailed in Chapter 4 and Chapter 5 draws the conclusion of the project.

2 Theoretical Background

The following section explains the scientific background related to this project in greater detail in two subchapters. Firstly, the applied medical terms, such as heart rate and heart rate variability, are introduced in the first subchapter. The second subchapter focuses on the measurement of the heart rate, including the explanation of the technical aspect.

2.1 Heart Rate and Heart Rate Variability

Heart rate defines the number of heart's muscle contractions per minute and is measured with the unit of beat per minute (bpm). Each of these contractions, triggered by the sinoatrial node, sends a pressure wave through the arterial walls which can be detected in a form of pulse even by palpation. [1] According to the American Heart Association, the normal resting adult heart rate is between 60 to 100 bpm. [2] However, besides individual differences, it is also affected by numerous other factors, such as age, gender, physical condition or medication. [1][2] According to the Guinness World Records, the lowest heart rate ever recorded is 27 bpm and the predicted maximum heart rate is calculated by subtracting the age of the subject from 220. [3][4] Based on these values, the range of heart rate measurement is set between 20-220 bpm for this project.

The sinoatrial node and therefore the heart rate is regulated by the sympathetic (SNS) and parasympathetic (PNS) nervous system. Heart rate variability is the variation of the time intervals between two sequential heartbeats, which unit is defined in milliseconds (ms), and it is used as an index for various aspects of psychology, such as recovery and stress indexes. [5]

Recovery and stress indexes have been developed to represent the parasympathetic and sympathetic cardiac activities. The index value of the normal average population is marked with zero and the deviation from this value towards either direction is scaled in a logical numerical way as shown in Figure 1.

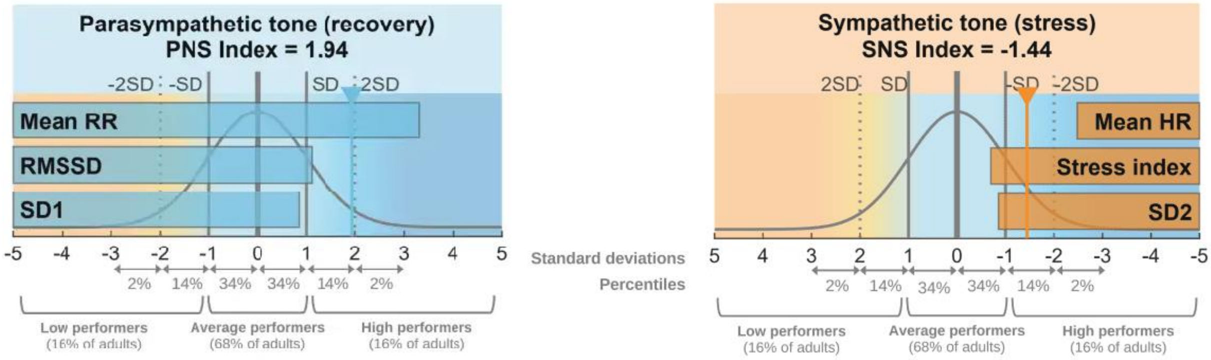

Figure 1: Parasympathetic and Sympathetic Nervous System Index [6]

As Figure 1 illustrates, both in Parasympathetic (PNS) and Sympathetic Nervous System (SNS) Index, the positive index reflects that the person performs better compared to population average, whereas a negative index describes the below average performance, and the numerical value of the index presents the magnitude of the deviation. [6]

Generally, stress factors activate the sympathetic nervous system which increases the heart rate. However, the magnitude of the change varies. Sympathetic nervous system or stress index is interpreted as the reaction to stress indicators. Once the trigger of the stress is no longer present, the parasympathetic nervous system aims to restore the sinoatrial node's activity to the physiological level. Thus, the parasympathetic nervous system or recovery index shows the subject's ability to recover from prior stress, which scale also varies within a given population. [6]

2.2 Detection of the Heart Rate

The detection of the heart rate via electrical signal, which electrocardiography is based on, is considered the gold standard, and widely used in both clinical and non-clinical environments. [2] However, to monitor the heart rate and the HRV for this project, a different, optical detection technique called photoplethysmography (PPG) is selected. The photoplethysmography measures the changes of the microvascular tissue indirectly through light emitting diodes (LEDs). The photodiodes then detect the amount and the direction of the light reflected from the arteries.

Figure 2 shows the electrical signal or electrocardiogram (ECG) generated by the sinoatrial node as a thinner and the photoplethysmogram (PPG) as the bolder line. This image describes the difference between the two signals in terms of amplitude within a given time domain.

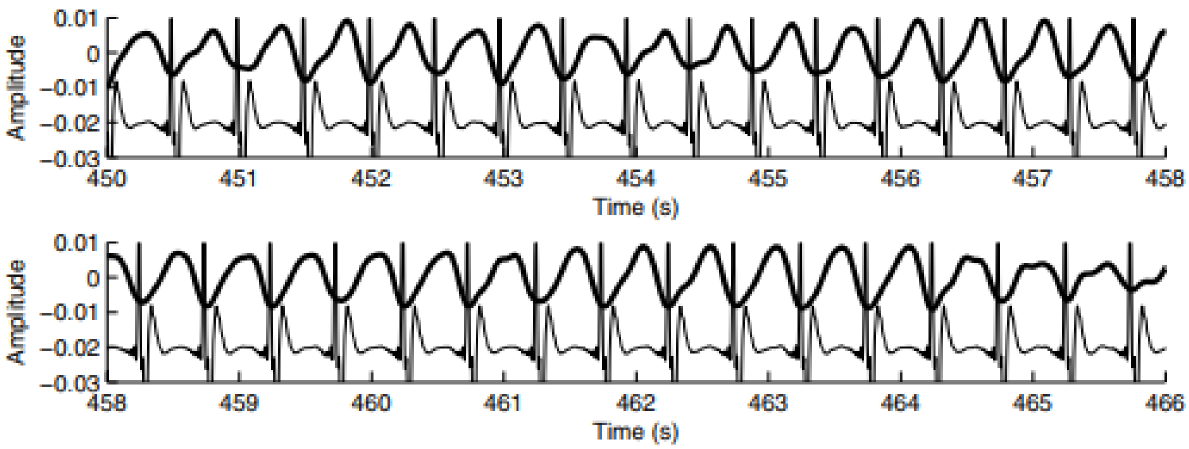

Figure 2: Photoplethysmogram (upper bold line) and electrocardiogram (lower normal line) of an infant's normal heart sinus rhythm [7]

As Figure 2 illustrates, both signals contain periodic, distinguishable peaks, whose frequency within a minute is the heart rate and the time interval between two sequential, respective peak points is the heart rate variability. In case of the PPG signal, the inter-beat-interval (IBI) is calculated usually from the negative peaks of the signal.

Crowtail Pulse Sensor v2.0, as illustrated in Figure 3, is designed to measure the heart rate of the human and is chosen to fulfil this exact purpose for this project. The hardware consists of an optical heart rate sensor (LED, photodiode), analog amplifier, noise cancellation circuitry and analog signal output. [8]

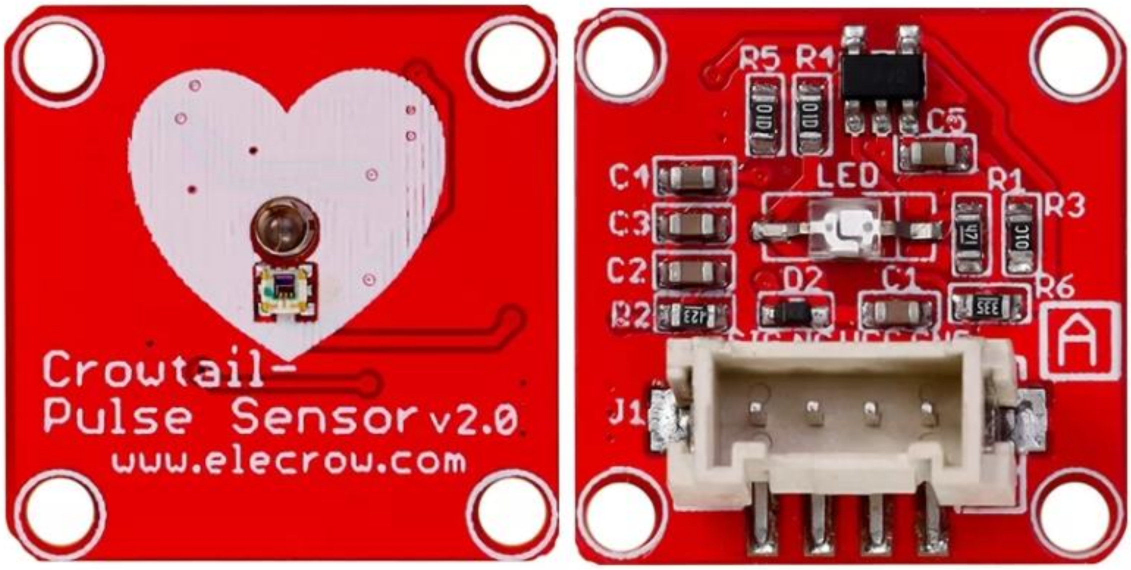

Figure 3: Crowtail Pulse Sensor v2.0 from the sensor (left) and component (right) side [8]

The left side of Figure 3 shows the sensor side of the hardware, which board has the LED in the middle and directly under it resides the diode. It is recommended to place this sensor and photodiode primarily to the earlobe or to the finger with the sensor side for best measurement result. However, it is possible to gather measurements from wrist, hand palm, arm, chest, cheek, and forehead.

3 Methods and Material

The following chapter is dedicated for the explanation of the materials utilized for the project. Firstly, the topic is introduced from a hardware point of view in the first subchapter. Secondly, the software aspect is discussed in the second subchapter along with the applied methods. Lastly, the third subchapter sums up the operational principle of the project.

3.1 Hardware Requirements

For the development of this project, Raspberry Pi Pico W, displayed in Figure 4, is selected as the microcontroller board due to its support of a wide range of peripherals. In addition, it also offers a 2.4 GHz wireless LAN connection which is utilised for the transmission of the HRV data to Kubios for additional analytical purposes. [9]

Figure 4: Raspberry Pi Pico W model [10]

As it is visualized in Figure 4, the middle part of the board holds the RP2040 dual-core Arm Cortex-M0+ processor. The silver, rectangle shaped component, locating below the processor, supplies the wireless connection and the white protruding element on the opposite, upper end of the board supports USB connection. The board also consists of an onboard led, which is found to the left side of the USB socket, and directly below it resides the BOOTSEL button. [11]

The hardware's SSD1306 OLED display, as shown in Figure 5, visualises the output information for the user.

Figure 5: SSD1306 OLED display [12]

Figure 5 shows a 128x64 wide, monochrome digital display, which is a simple dot matrix display that communicates through I2C. [12] In the project, it allows the user the select the activity respectively from the menu and serves as the tool to follow the real time signal that the sensor detects.

The rotary encoder used in the project is designed by project engineer, Joseph Hotchkiss of Metropolia University of Applied Science. Figure 6 shows a generic purpose rotary encoder; however, it does not match entirely the project device one.

Figure 6: Example of rotary switch control encoder [13]

As Figure 6 displays, the rotary encoder consists of a metal encoder, that has three functions: Rot Switch, ROT A and ROT B. The Rot Switch acts like a standard push button switch. The ROT A and ROT B functions allow the user to rotate the encoder clockwise and anticlockwise. The Rot Switch button function enables the selection of the chosen activity in the project.

Crowtail Pulse Sensor v2.0 which functions as the sensor of the hardware is introduced in detail in Chapter 2 and its technical aspect is not discussed further here. The sensor connects to the protoboard through the 4-pin Grove-connectors and to the Pico W through the ADC_0 pin as Figure 7 shows.

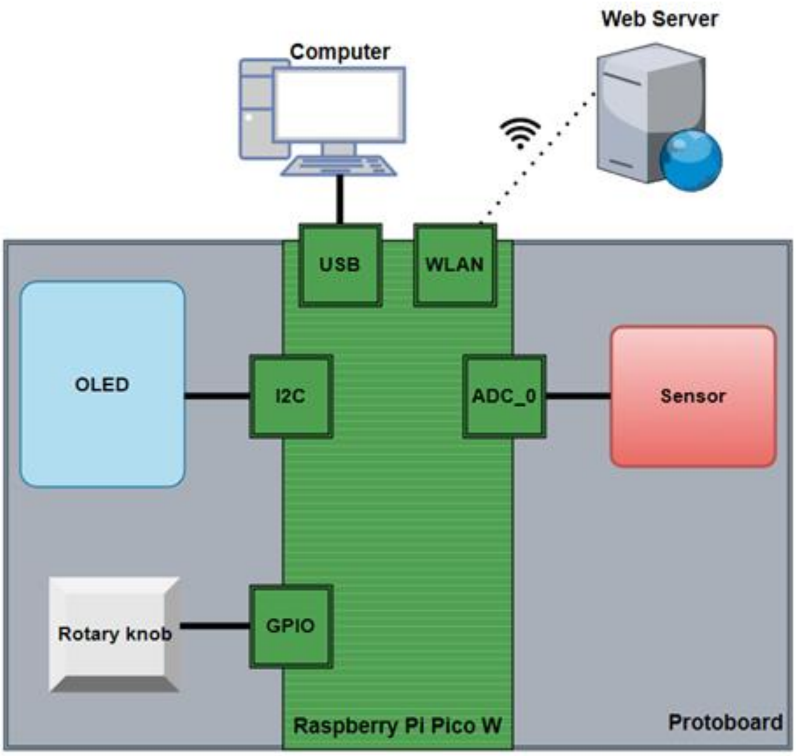

Figure 7: Visualisation of the hardware components and the connection

Figure 7 visualizes how the prior introduced hardware elements are connected and create the product that is capable the detection of the HRV parameters, recovery and stress indexes. As it is shown in Figure 7, the device is connected to the computer via USB connection, the rotary encoder is accessible through GPIO pins, similarly to the LED components of the protoboard. The OLED is available for programming via I2C interface. The external web server, Kubios is accessible with the wireless function of the Raspberry Pi Pico W.

3.2 Software Requirements and Methods

For programming the Raspberry Pi Pico W, Thonny IDE is used and MicroPython is the programming language. Thonny IDE is a Python IDE developed for embedded systems. It allows the MicroPython firmware to be downloaded to the Raspberry Pi Pico W, which enables its direct programming.

MicroPython has fewer libraries, however, more library options are developed specifically for embedded systems compared to the desktop Python. For instance, MicroPython's ssd1306 library, relying on MicroPython-specific library FrameBuffer, is used to control the OLED display. Additional libraries are utilized for the project, such as machine, array, network, socket, urequests and ujson.

Furthermore, two files, fifo.py and piotimer.py are provided by the lecturers of Metropolia University of Applied Sciences. The file named fifo.py provides the logical data storage in an interrupt safe way and the piotimer.py file contains a class for hardware timing.

3.3 Operating Principle

The Crowtail optical sensor detects the heart rate as an analog signal and transmits it to the Raspberry Pico Pi. This analog signal is converted into digital by the AD-converter of the microcontroller and the peak-detection algorithms, created for the project by the group members, determine the peak-to-peak intervals of the signal. To receive the most accurate measurement, the peak-detection algorithm is tested and fine-tuned with the test data created for this purpose, and filtering is implemented for an even more precise outcome.

With the collected PPI data, the mean peak-to-peak interval (mean PPI), mean heart rate (mean HR), standard deviation of successive interval differences (SDNN), root mean square of successive differences (RMSSD) and Poincaré plot shape parameters (SD1 and SD2) are calculated. The calculation of formerly mentioned values occurs locally on the development board, and the results are displayed on the OLED at the end of the measurement.

From Raspberry Pi Pico the collected peak-to-peak interval data is also transmitted wirelessly to the Kubios Cloud, where the data is further analysed to receive the recovery and stress indexes. The outcome of this analysis is then returned to the device and the results are presented through the OLED display for the user along with the locally calculated parameters.

During the measurement, the rotary encoder functions as the controller for this operation, that provides the user interaction for the hardware. The user can choose the activity respectively based on to the information displayed on the OLED, such as initialisation of the measurement.

4 Implementation

Chapter 4 introduces the programme and the algorithms utilized in the project. The first subchapter provides a summary of the implementation requirement. In the second subchapter, the software implementation is described, detailing the functions and the logic of the code respectively in their own subchapter.

4.1 Implementation requirement

The analog signal is collected with the Crowtail Pulse Sensor v2.0. For more detailed information about the sensor, please refer to Chapter 2.2. A direct human contact to the optical heart rate sensor is necessary to measure the heart rate signal. As the strength of the pulse varies individually, it is recommended to locate in advance the strongest pulse source for the most accurate measurement.

The heart rate measurement software is developed using a single-board computer Raspberry Pi Pico W. The coding of the programme is implemented in Thonny IDE using MicroPython programming language. The algorithms of the peak and the peak-to-peak interval detection are created by the group, along with the calculations of all heart rate variability parameters except for the recovery and stress indexes. As it was mentioned in Chapter 3.1, Raspberry Pi Pico W via the WLAN (Wireless Local Area Networks) connection transmits HRV data to Kubios for additional data analysis.

The project implementation is carried out in multiple steps, and to create a seamless team workflow and accessibility, the earlier versions are stored in the main branch of the group's Gitlab repository, along with the final version (main.py). Each team member has their own branch in the project repository, where the fragments of the codes are stored.

4.2 Software algorithms

Firstly, to execute the programme successfully, the required libraries are defined. Library “machine” is imported to access and control the GPIO (general purpose input output) pins, I2C (Inter-Integrated Circuit), ADC (analog-digital converter) and PWM (pulse-width modulation), and library ssd1306 is used to OLED display. For the timing purposes, libraries such as “piotimer”, “utime” are imported. Libraries “fifo” and “array” are utilized to store and analyse the digital data. To connect to the router and access successful to Kubios, the following libraries are required: “network”, “socket”, “urequests” and “ujson”.

Secondly, the pins related to ADC, OLED, LEDs and rotary encoders are defined in the programme. Then, the global variables are divided into four groups according to their purpose. Before each group there is an out-commented text to clarify the aim of each group, as shown in Figure 8.

```
# Sample Rate, Buffer
samplerate = 250
samples = Fifo(32)

# Menu selection variables and switch filtering
mode = 0
count = 0
switch_state = 0

# SSID credentials
ssid = 'KMD758Group5'
password = '105105105M'

# Kubios credentials
APIKEY = "pbZRUi49X48I56oL1Lq8y8NDjq6rPfzX3AQeNo3a"
CLIENT_ID = "3pjgjdmamlj759te85icf0lucv"
CLIENT_SECRET = "111fqsli1eo7mejcrlffbklvftcnfl4keoadrdv1o45vt9pndlef"

LOGIN_URL = "https://kubioscloud.auth.eu-west-1.amazoncognito.com/login"
TOKEN_URL = "https://kubioscloud.auth.eu-west-1.amazoncognito.com/oauth2/token"
REDIRECT_URI = "https://analysis.kubioscloud.com/v1/portal/login"
```

Figure 8: List of global variables

As Figure 8 displays, the first group of global variables defines the sample rate, which is set to 250 Hz, and the size of the Fifo. The variables of the second group contribute to debouncing effort and provides the seamless button press experience.

The third group consists of the necessary credentials to connect to the WLAN and the last group of variables contain the information to access Kubios.

In addition, the programme contains overall 11 functions and the main programme. The former is described in detail in Chapter 4.2.1 and the latter one in Chapter 4.2.2.

The final version of the programme (main.py) is stored in the Raspberry Pi Pico W, which allows the programme to be executed without the need of an IDE and the computer only serves as a power source of the Pico. Thus, the code is structured that way, that by using the hardware components of the Pico, the user can interact with the product, and restart the measurement.

4.2.1 Functions

As mentioned prior, the programme consists of overall 11 functions, which are listed in Table 1. The table defines the Function's name and the taken argument in the first column and explains their purpose in the second columns.

Function's name and taken argument	**Purpose and explanation**
read_adc(tid)	Samples the signal via ADC converter.
welcome_text()	Displays the Welcome text of the program: "Welcome to Group 5's project!" and draws the frame which consists of heart motives using two for loops for 3,75 seconds
press_to_start()	Displays the instruction of button press on OLED and draws an arrow with two EGC signals which point to the rotary encoder
connect()	Establishes WLAN connection with router
meanPPI_calculator(data)	Calculates the mean peak-to-peak interval, takes the collected PPI array as argument
meanHR_calculator(meanPPI)	Calculates the mean heart rate interval, takes the prior calculated mean PPI as argument

SDNN_calculator(data, PPI)	Calculates the SDNN, takes the collected PPI array and prior calculated mean PPI as arguments
RMSSD_calculator(data)	Calculates the RMSSD, takes the collected PPI array as argument
SDSD_calculator(data)	Calculates the SDSD, takes the collected PPI array as argument
SD1_calculator(SDSD)	Calculates the SD1 Poincaré plot index, takes the prior calculated SDSD as argument
SD2_calculator(SDNN, SDSD)	Calculates SD2 Poincaré plot index, takes the prior calculated SDNN and SDSD as arguments

Table 1: The functions of the programme

As Table 1 describes the functions “welcome_text” and “press_to_start” are related to display the output of OLED and takes no argument. The former one is called only once before the programme enters to the main part. The latter one is activated within the main programme and is displayed every time when it waits for a user input (button press) to start the measurement.

The function named “read_adc” serves the function of collecting sample data from the ADC and is activated when the user initiates the measurement. The function named “connect” establishes the internet connection with the prior defined SSID and passphrase when the measurement ends.

The rest of the functions from the list are related to the calculations of the HRV parameters. The mathematical formula for each calculation is provided during the lectures of the Project course at Metropolia. The “meanHR_calculator” function is called already in sync with the signal sampling and peak-to-peak analysis, once the number of values within the PPI array reaches 3, and the mean HR value is reviewed every time when a new value is added to PPI array. The meanHR_calculator is called again along with all the other HRV calculators once the programme has executed the measurement.

4.2.2 Main Programme

The flow chart, as displayed in Figure 9, intends to characterize the structure of the software. It divides the concept into smaller states for a more convenient understanding, outlines the main steps, and visualizes the decisive factors that define the path and the outcomes of the programme.

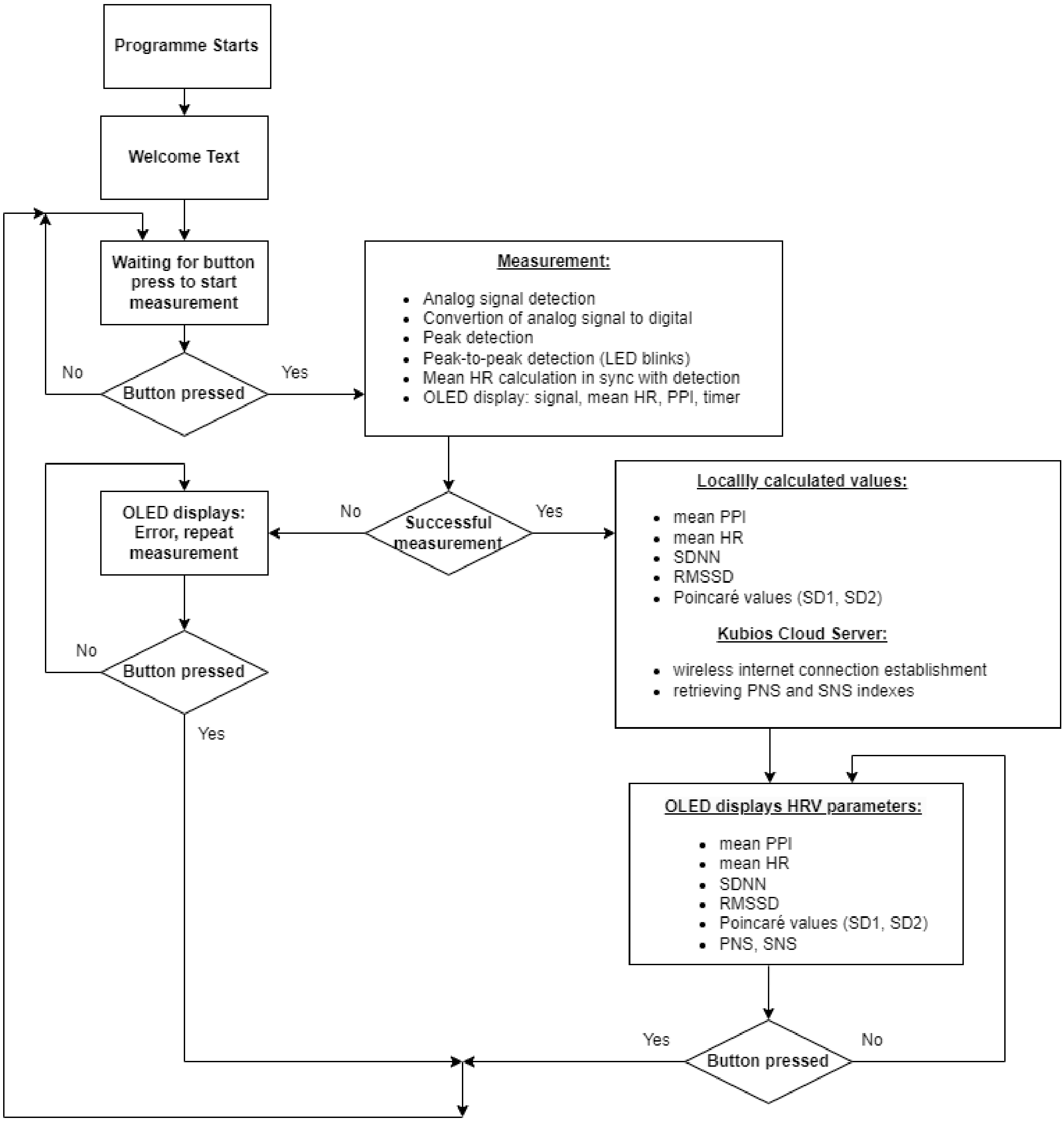

Figure 9: Flow chart of the programme

As shown in Figure 9, once the program begins, it displays a Welcome Text on the OLED screen for 3,75 seconds. Once the set time of the Welcome Text expires, the programme enters the main programme, a while True loop and moves to the next state, where it waits for the rotary encoder to be pressed. During this state the OLED display is also updated, asking the user to press the rotary encoder to initiate the measurement.

As the button is pressed, the onboard LED blinks once and the program begins to sample the heart rate signal. For the demonstration of the project, the data capture length is set to 60 seconds. As the sampling frequency is set to 250 Hz, for 60 seconds, 15000 samples are processed. Each sample is stored in circular buffer, to calculate the running average in an ‘H’ type array to allocate less memory.

The analysis of the samples is carried out already during the measurement, the algorithm starts the peak detection once the value of the sample exceeds the value of the threshold. The index of the highest sample value is stored, and if a new peak is detected, the index of the new peak is saved similarly, and the peak-to-peak interval is calculated based on the difference of these two indexes and is then converted into milliseconds. This peak-to-peak interval is appended to the list, hereinafter referred to as PPI array, that stores all the detected peak-to-peak intervals.

Once PPI array has 3 values, the “meanHR_calculator” function is called every time OLED screen is refreshed. The calculated mean HR is displayed on the OLED screen on the top during the measurement state along with every detected PPI value. The middle part of the OLED screen displays the heart signal continuously and the bottom part of the OLED shows a Timer that displays the seconds that has passed since the initiation of the measurement.

Once the programme executed the measurement for the set time, and if the length of the PPI array exceeds the value of 3 (meaning the measurement is successful as shown in Figure 9), the programme attempts to connect to the WLAN and sends the PPI array to Kubios Cloud Service. The returned json data is extracted, the PNS and SNS indexes are saved as variables and the other HRV parameters are calculated locally. At the end of the programme the results are displayed on the OLED screen. If during the measurement less than 3 peak-to-peak intervals are detected, the measurement is not considered satisfactory, and the OLED screen displays the message to repeat the measurement. In both cases, the display on the OLED screen remains until the rotary encoder is pressed again. If the rotary encoder is pressed, all the variables are reset, the arrays are emptied, and the programme returns to the state where it waits for the button press to initiate the measurement.

5 Conclusions

The project aims to design and develop a software for an embedded system capable of measuring heart rate and heart rate variability parameters with high accuracy. The SNS and PNS indexes are calculated by Kubios Cloud Service, based on the provided PPI values, using Rest API. The project is successfully achieved through the collaborative efforts of all team members.

The most challenging part of the project is the development and implementation of the algorithm to filter raw PPG signals and determine effective peak-to-peak interval and heart rate variability parameters. The team reviewed lecture materials and research papers on heart rate variability and signal processing to gather necessary information for algorithm design. The team tested several algorithm design approaches and made progressive changes before producing the most efficient algorithm.

The developed prototype measurement system has some limitations, such as average accuracy of measured heart rate due to the implementation of a basic signal processing algorithm, and unreliable measurements when PPG data is gathered from less vascular areas of the body, such as fingertips (a limitation of the PPG sensor). The data transfer of Kubios Cloud Service is not always reliable, has high response times and times out sometimes.

There is potential for improvement in the developed prototype measurement system. The system can be made more accurate by implementing advanced signal processing techniques including machine learning and artificial intelligence, and upgrading hardware features such as processing power, and sensors.

References

1. British Heart Rate Foundation. 2021. All about heart rate. Accessed. 03.04.2023 https://www.bhf.org.uk/informationsupport/heart-matters-magazine/medical/ask-the-experts/pulse-rate

2. American Heart Association. 2015. All about heart rate. Accessed: 03.04.2023 https://www.heart.org/en/health-topics/high-blood-pressure/the-facts-about-high-blood-pressure/all-about-heart-rate-pulse

3. Guinness World Records. 2005. Lowest heart rate. Accessed: 03.04.2023 https://www.guinnessworldrecords.com/world-records/lowest-heart-rate

4. MedicineNet, Poonam Sachdev MD. 2021. What Is the Highest Heart Rate You Can Have Without Dying? Accessed: 04.04.2023 https://www.medicinenet.com/highest_heart_rate_you_can_have_without_dying/article.htm

5. International Journal of Cardiology, Alex Schafer, Jan Vagedes. 2012. How accurate is pulse rate variability as an estimate of heart rate variability? Accessed: 04.04.2023 https://www.internationaljournalofcardiology.com/article/S0167-5273(12)00326-9/fulltext

6. Kubios. HRV in Evaluating Autonomic Nervous System Function. Accessed: 04.04.2023 https://www.kubios.com/hrv-ans-function/

7. M R Grubb, J Carpenter, J A Crowe, J Teoh, N Marlow, C Ward, C Mann, D Sharkey and B R Hayes-Gill. 2014. Forehead reflectance photoplethysmography to monitor heart rate: preliminary results from neonatal patients. Accessed: 04.04.2023 https://iopscience.iop.org/article/10.1088/0967-3334/35/5/881/pdf

8. Elecrow. Crowtail - Pulse Sensor. Accessed: 04.04.2023 https://www.elecrow.com/wiki/index.php?title=Crowtail-_Pulse_Sensor

9. Raspberry Pi, Raspberry Pi Pico. Accessed: 19.04.2023 https://www.raspberrypi.com/products/raspberry-pi-pico/

10. Galaxus, Raspberry Pi Pico® Microcontroller RP-PICO-W. Accessed: 19.04.2023 https://www.galaxus.ch/en/s1/product/raspberry-pi-pi-microcontroller-rp-pico-w-development-boards-kits-24227536

11. Raspberry Pi Pico W Datasheet. 2023. An RP2040-based microcontroller board with wireless. Accessed: 19.04.2023 https://datasheets.raspberrypi.com/picow/pico-w-datasheet.pdf

12. NettiGo, OLED Display 0.96" I2C 128x64 SSD1306 white. Accessed: 19.04.2023 https://nettigo.eu/products/oled-display-0-96-i2c-128x64-ssd1306-white

13. AliExpress, 360 Degrees Rotary Encoder Module for Arduino Brick Sensor Switch Development Board KY-040 With Pins. Accessed: 19.04.2023 https://www.aliexpress.com/i/32969404416.html

14. S Lukkarinen. 2023. Hardware 2 project requirements. Accessed: 19.04.2023 https://metropoliafi.sharepoint.com/:w:/s/OPS2022Hardwarethemeprojectplan/EeSB6jp9bixMhhpQ5eBNhTsBB1spE1Dm92Q_qsqvTkEuzg