HEALTH MONITORING SYSTEM

Robotics

Phebe Le

Planning Journal

Session 1.5: 6/5/2023-23/5/23

1. Introduction and project overview

This task is to create a health monitoring system that uses a pulse sensor, LCD display, SD card module and has external outputs such as sound and lights. The integration of a Bluetooth/Wi-Fi module to display a GUI is plausible.

2. Project objective and audience profile

The objective of this project is to create a forearm Holter monitor which measures one's heartrate. These inputs will be documented and stored within an SD card. The outputs will be a GUI, an LCD screen displaying the heart rate, a LED and piezo for alerts.

Scenario: The heart monitoring device will be a forearm and wrist wearable item. This location was chosen due to Cleveland Clinic stating there are 'two major arteries inside your forearm and wrist'. Whilst there are several major arteries elsewhere in one's body, I specifically chose this to allow easy access and monitoring. This device will be used solely in an imaginary hospital situation for cardiac monitoring. Specifically, for patients who are intending to exercise in a limited range. This scenario was chosen to adhere to the limited range that an ethernet shield cable may have. Cardiac monitoring will be displayed for the patient through an LCD screen and for the specialist through a monitoring device (an IOT device with access to webservers). Alerts will be displayed physically via a red light and beeping sounds, logically via a webserver.

3. Project scope

For this task, the resources required will be a pulse sensor, an LCD display screen, several LEDs, a SD card module, a SD card, a W-Fi module, a piezo module, an Arduino, a range of cabling, a breadboard and a smart device.

The logic of the program will require:

- Functions: To read heart rate values
- Pointers, vectors, malloc: To allocate memory dynamically
- Use of interrupts: to ensure consistency between heart rate values
- Use of SD card library: to store and write to SD card
- Use of LCD library: To constantly display measured users' heart rate
- Use of Ethernet library: To have a viewable web server
- Use of SPI library: To communicate with the SD module within the Ethernet shield
- Use of Vector library: to dynamically store heart rate data

The content which will need to researched is:

- The low and high range of heart rate for individuals
- Types of heartrate monitoring systems

The main criteria which need to be accomplished are:

- To read, write and be able to access data from an SD card

- To have memory allocation by defining a buffer for the memory allocation to be limited by, vector to store the BPM data and pointers to reference the location of different variables
- Alert system to check if the BPM exceeds a certain limit
- Use of LCD, serial monitor and webserver to display BPM data

4. Assumption

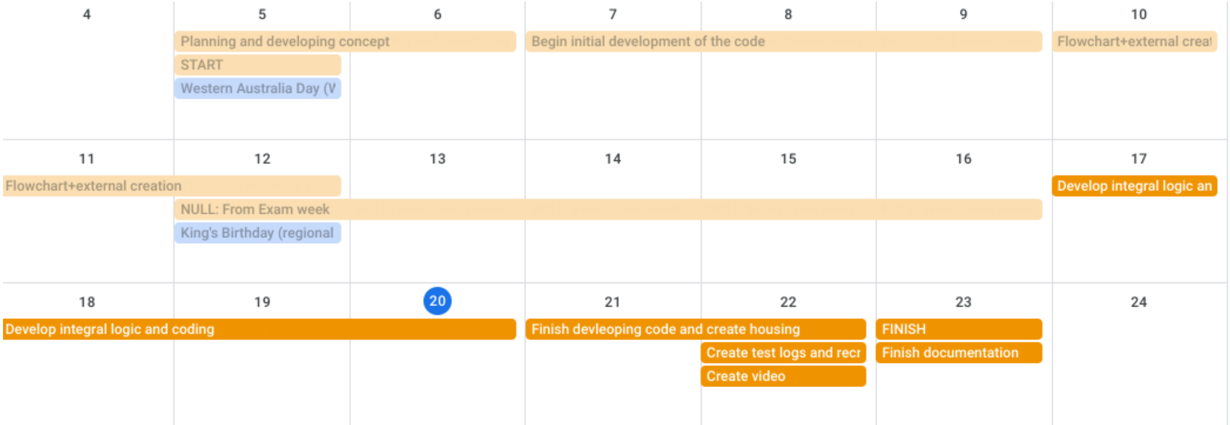
The assumptions of this task is that the main components and devices of the design will be readily provided. Other external equipment such as cabling, breadboard and housing will be created or brought. The timespan of this task will be three weeks. The weeks will be segmented into planning, designing, creating, implementing/debugging, testing and will eventually have a finished product. This is based on combining different stages from SLDC and SSLDC. The construction and testing process will be conducted through an iterative cycle to ensure a structural sound design. Final testing before release will be required.

5. Issues and risks

The risks that may occur will be electrocution, cuts and burns. Electrocution may be mitigated by careful checking power sources and reducing the amount of DC involved in the project. Cuts may be mitigated by cleaning and sealing them. Burns may be mitigated by using ice or water to relieve the burn. Other risks may occur, but many will be mitigated through careful handling and management of equipment.

Issues that may occur are lack of knowledge, incorrect test log results, loss of data and structural errors. These issues can be mitigated through research, debugging of code and physical analysis. Data loss will be mitigated by creating backup copies in different locations.

6. Project schedule



4	5	6	7	8	9	10
	Planning and developing concept START Western Australia Day (V		Begin initial development of the code			Flowchart+external creat
11	12	13	14	15	16	17
Flowchart+external creation	NULL: From Exam week King's Birthday (regional					Develop integral logic an
18	19	20	21	22	23	24
Develop integral logic and coding			Finish devleoping code and create housing	Create test logs and recr Create video	FINISH Finish documentation	

6.1. Phase 1: Research, Planning and Development

- **Full outline of design**
- Aim: The main design will be segmented into several different parts. The first being an armband which will store the Arduino and Ethernet shield. The second

section is a 'smartwatch' contraption which contains a breadboard, LCD, 2 LEDs (red and green), a pulse sensor and a piezo.

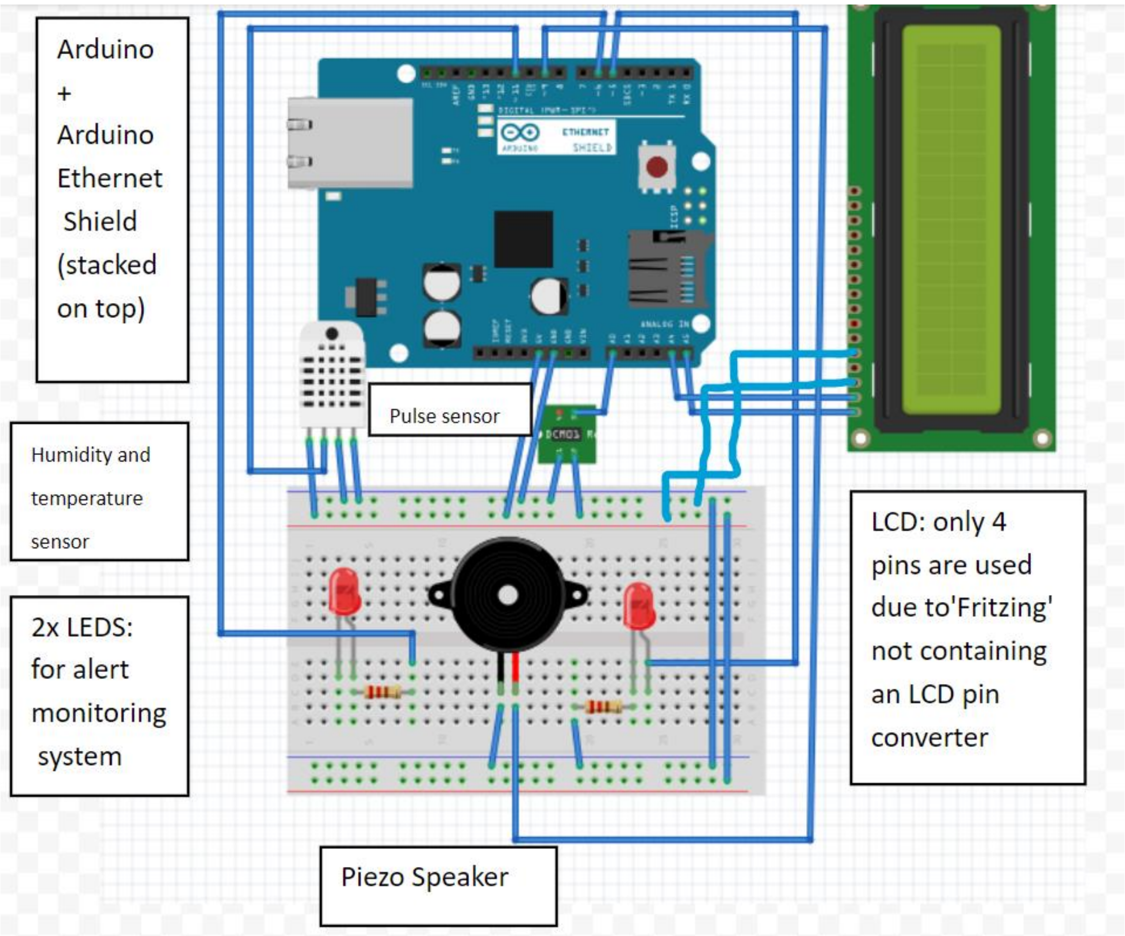
- ## Armband: Arduino and Ethernet Shield

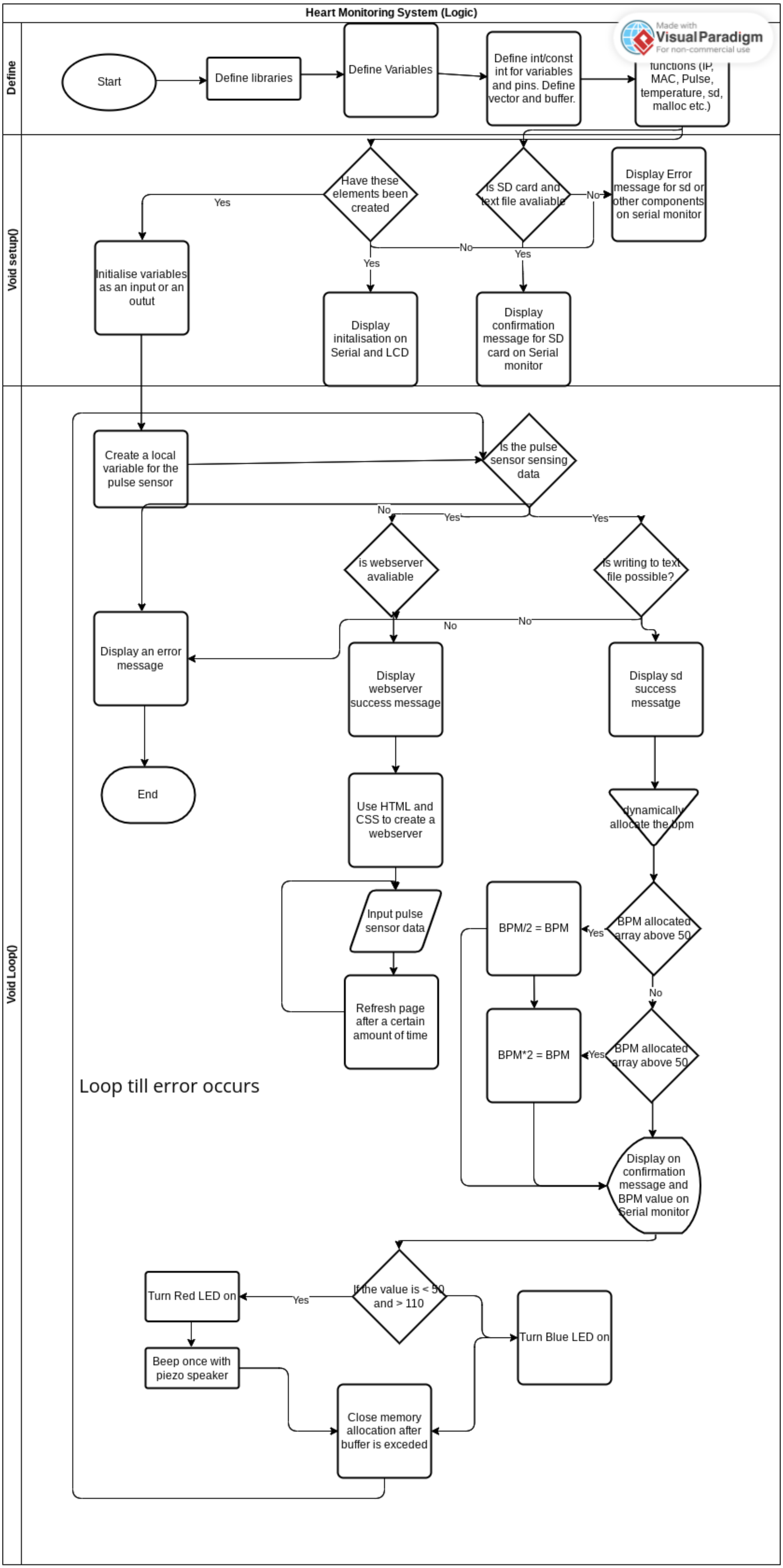
- Logical aim: To create the logical components of the armband, there will be several steps. These being:
- Implementation of the webserver. This will require the Ethernet shield to be configured to the network.
- Integration of smartwatch data. The data from the smartwatch will need to be viewable in a web server
- (If applicable) Creation of age profiles. Age profiles will be required to alert the user only when their age group BPM exceeds or beneath the average.

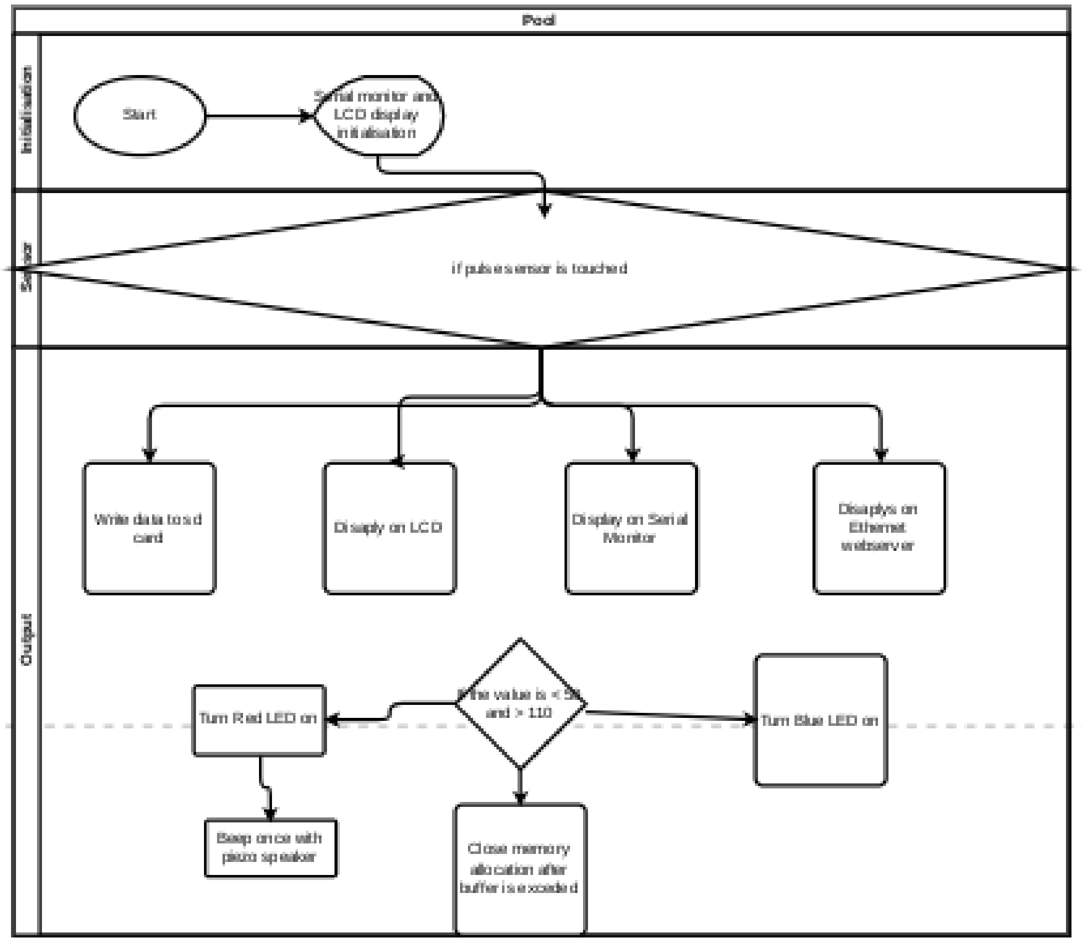
- ## Smartwatch: Other components

- Logical aim: To create the smartwatch, the logical design and implementation will be broken up into several different modules. These being:
- Integration of sensor
    - Pulse sensor: This will require the initial test to see if it works, changing the serial graph data to a set variable data (BPM)
- Integration of display components (LCD). The display should show the current BPM
- Integration of an alert system. The alert system will be triggered when an individual reaches a BPM that exceeds the average age range
    - If applicable, create custom profiles for each age range group

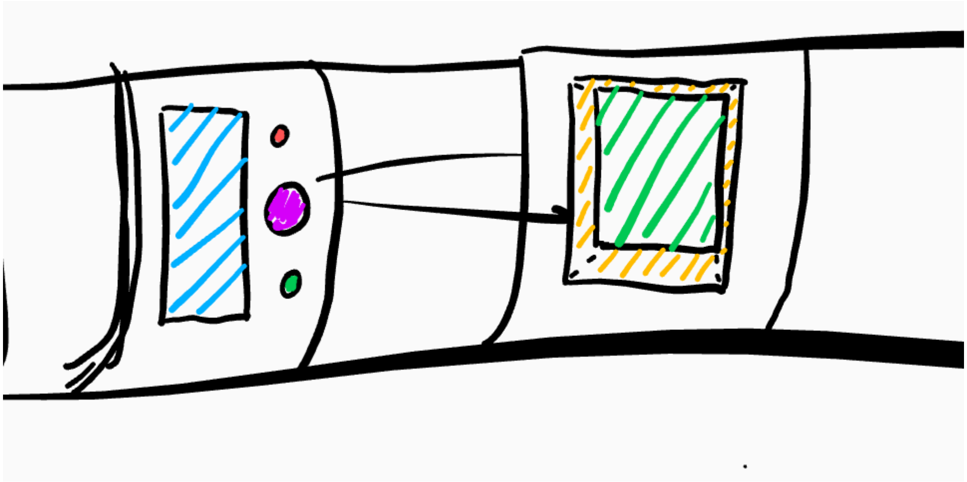
6.2. Diagram, Flowchart Chart and Initial Sketch


Arduino + Arduino Ethernet Shield (stacked on top)
Humidity and temperature sensor
2x LEDS: for alert monitoring system
Pulse sensor
Piezo Speaker
LCD: only 4 pins are used due to'Fritzing' not containing an LCD pin converter


Heart Monitoring System (Logic)
Define
Start
Define libraries
Define Variables
Define int/const int for variables and pins. Define vector and buffer.
functions (IP, MAC, Pulse, temperature, sd, malloc etc.)
Void setup()
Have these elements been created
Yes
Initialise variables as an input or an outut
Yes
Display initalisation on Serial and LCD
No
Is SD card and text file avaliable
No
Display Error message for sd or other components on serial monitor
Yes
Display confirmation message for SD card on Serial monitor
Void Loop()
Create a local variable for the pulse sensor
Is the pulse sensor sensing data
No
Yes
Yes
is webserver avaliable
Is writing to text file possible?
No
No
Display an error message
End
Display webserver success message
Display sd success messatge
Use HTML and CSS to create a webserver
dynamically allocate the bpm
Input pulse sensor data
BPM/2 = BPM
Yes
BPM allocated array above 50
No
Refresh page after a certain amount of time
BPM*2 = BPM
Yes
BPM allocated array above 50
Loop till error occurs
Display on confirmation message and BPM value on Serial monitor
Turn Red LED on
Yes
If the value is < 50 and > 110
Turn Blue LED on
Beep once with piezo speaker
Close memory allocation after buffer is exceded


Pool
Initialisation
Start
Serial monitor and LCD display initialisation
Sensor
if pulse sensor is touched
Output
Write data to sd card
Disaply on LCD
Display on Serial Monitor
Disaplys on Ethernet webserver
If the value is < 50 and > 110
Turn Red LED on
Turn Blue LED on
Beep once with piezo speaker
Close memory allocation after buffer is exceded

Physical implementation of the sketches



Initial planning sketch

6.3. Optimisation of code, major fixes, extensions

The code which has now been finished could have been optimised by using more functions and consistent use of datatypes. Through both sketches, whilst there were several variables for process like LCD, the important features such as the SD card writing file or the HTML file could be written in a function to reduce space in the main loop().

Major fixes that were conducted through this task was the changing of the vector libraries due to the C++'s lacking and varied inability to verify and run. This was changed from the C++ library(<vector>) to the Arduino vector library (<Vector.h>). The development of the HTML web page and constant iterations to ensure the fastest data retrieval method to be consistent with the main pulse sensor (through Serial Monitor). The optimisation of memory loss through consistently checking data entries and creating datatypes (byte to ensure no float variables could be inconsistent in other outputs).

Extensions to this task would be to change the final design of the housing due to its size, create pointers in the webserver, include the webserver as a library inside the main file to allow both codes to run simultaneously.

6.6. Tests and Fixes

Brief list of all tests conducted in both sections. To seem full implementation and process see Robotics Progress Report

Testing section 1 (the smartwatch):

- Error with storing data in SD card: overcome by adding a SD card close function
- Error with heartbeat values: overcome by adding a range and limitations
- Error with vector library: overcome by changing the library and removing concepts which were not required for the memory allocation
- Error with uploading sketch with Ethernet and smartwatch: overcome by removing the ethernet code and splitting the design into two sections

Testing section 2 (the webserver):

- Error with connection = overcome by finding personal Arduino IP and MAC address
- Error to sense BPM data: overcome by changing the order the BPM was
- Error with the range the BPM values were in: overcome by changing the values of the if statement for the BPM
- Error to update BPM data: overcome by adding refresh rate in HTML

Summary

The development of this task was significantly rushed due to the lack of time due to external exams and assignments. The development of the code was generally optimised to a certain degree yet could be improved by other functions and the inclusion of self-created libraries. The development of the design could be further improved to be smaller and closer to the concept of a smartwatch.