IMPERIAL

A DEDICATED RECEIVER FOR WEARABLE DEVICES

Author

ADVIK CHITRE

CID: 02019365

Supervised by

PROF. E. RODRIGUEZ-VILLEGAS

An Interim Report submitted in fulfillment of requirements for the degree of
Master of Engineering in Electronic and Information Engineering

Department of Electrical and Electronic Engineering
Imperial College London
2024

Declaration of Originality

I hereby declare that the work presented in this thesis is my own unless otherwise stated. To the best of my knowledge the work is original and ideas developed in collaboration with others have been appropriately referenced.

Copyright Declaration

Contents

List of Acronyms

CVD colour vision deficiency

GDPR General Data Protection Regulation

GPL General Public License

LED light-emitting diode

LTE long-term evolution

RFID radio frequency identification

BLE Bluetooth low energy

GPS Global Positioning System

GPIO general-purpose input output

PCB printed circuit board

SOC system on chip

RAM random access memory

OS operating system

RTOS real-time operating system

GPOS general purpose operating system

WCAG Web Content Accessibility Guidelines

HCI human-computer interaction

UI user interface

UX user experience

GUI graphical user interface

LVGL Light and Versatile Graphics Library

TTS text-to-speech

FPS frames per second

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

MQTT Message Queuing Telemetry Transport

OTA over-the-air

DMA direct memory access

List of Figures

1 Project Specification

Wearable medical devices for biological monitoring often depend on the patient's smartphone to upload data through an application. Maintaining releases for multiple differing operating systems and additional application layers, particularly during software updates, causes unreliability resulting in a negative user experience [1].

Design a dedicated receiver to deliver data from wearable devices to their manufacturer. The design must maintain the functionality, reliability, and ease-of-use of a mobile phone receiver and should secondarily minimise cost at scale.

Produce a hardware prototype, mainly focussing on software to provide an accessible and intuitive experience. The resulting prototype should be evaluated objectively to ensure no functionality is lost and to ensure no unnecessary complexity is added to the user experience (UX).

2

Background

Contents

This is a highly design-based project. Therefore, background is important to understand how the implementation needs to be used. Prof. Esther Rodriguez-Villegas acted as a consultant based on her industrial experience. She has previously worked as part of a team to develop the AcuPebble [2], a device used at home during sleep to detect obstructive sleep apnoea.

2.1 Wearables

In the wellness and medical industry, wearable devices are used to monitor and collect data for a single patient over a period of time. Data can include any physiological biomarkers [1] such as sounds generated by respiratory and cardiac functions [2].

If the period of observation is long, the patient is ideally able to take the device home. This is enabled as the patient can download an application on their smartphone so it can be used as a receiver. The receiver application typically has the following functionality:

1. Ask the patient diagnostic questions
2. Receive data from the wearable device
3. Upload data to the medical team

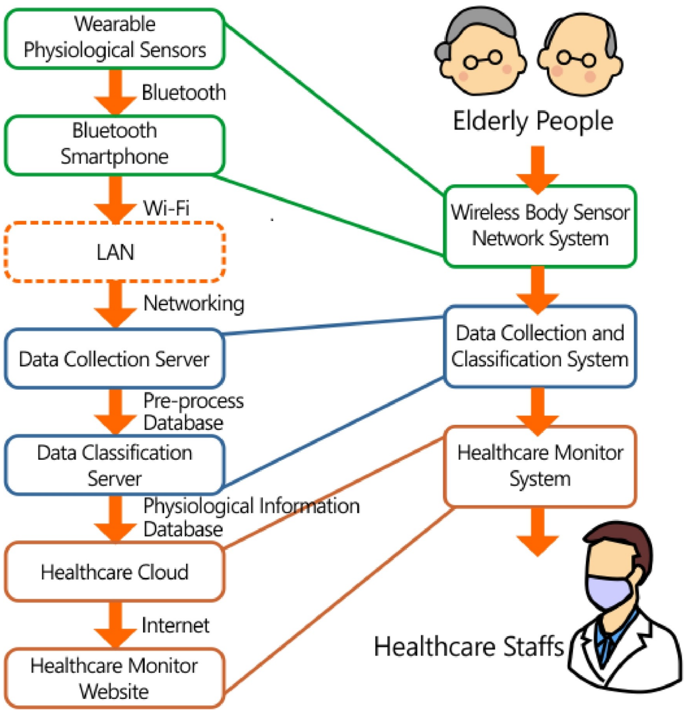


Figure 2.1: Mobile physiological sensor system and operating procedure, taken from [3].

Uploading collected data requires an internet connection. Typically the smartphone is already connected to the patient's home Wi-Fi network or mobile network. In the case that the patient does not own a smartphone or has no means of internet connectivity, some hardware can be provided.

The UX in the smartphone application is designed to maximise usability, minimising work required to teach patients proper use. The necessity of an interface with the patient would reduce the likelihood of project completion as teams would have to invest in app development.

Operating systems, such as iOS and Android, are constantly being upgraded which can break existing features [4]. Along with bug fixing and improvements, app developers typically release updates every 20-40 days [5]. This heavy maintenance burden can put significant strain on a small team that wants to focus more on developing new wearables and would limit new projects they can focus on.

Due to the different systems working together, there is some required networking to send data from the wearable to the medical staff. An example networking solution is shown in figure 2.1. This is not a unique structure, including an optional pre-processing database.

3
Design

Contents

This section describes the ideas that led to the implementation user interface (UI) designs shown in appendix A as well as the requirements for its accompanying hardware.

The proposed design is a generalised dedicated receiver, which wearable manufacturers can interface with to replace their smartphone application hence allowing them to focus solely on developing their wearable. As the receiver is generalised, it will be able to connect to a number different wearables. The manufacturer must only implement the 'Wearable' and the 'Defined Upload Location' shown in figure 3.1.

The design must include a responsive touchscreen to maintain high levels of interactivity, essentially resulting in a reduced smartphone.

3.1 Aims

At a high level, the design should improve the experience of both the doctor and the patient.

As stated in section 1, the device must be as accessible and easy to use as possible. Although the Web Content Accessibility Guidelines (WCAG) [6] are typically used for web pages, they are applicable to UI design and were hence useful. During the design phase, the following categories of disabilities should be considered:

1. Visual
2. Auditory
3. Cognitive
4. Physical

In order for the interface to be as usable as possible, especially for those with cognitive, learning, or neurological disabilities, designs should be predictable and simple [7].

As cost is a secondary objective function, types of components should be chosen by their function and then narrowed down by weighing its cost and effectiveness.

As any embedded system, the design should also rank efficiency and low-power consumption highly to prolong the observed battery capacity.

3.2 Device

The design should be transportable if necessary such as from the medical institution to the patient's residence. The majority of the time, the device will be located within the residence so should be designed to work well stationary, including independently standing easily. Inspiration was taken from screen-based home assistants such as Google's Nest Hub and Amazon's Echo Show. These home-based devices similarly use touchscreens to communicate with users but have a much greater focus on voice commands.

The receiver design centres around the largest and most constrained component, the touch-screen. The device should connect to the wearable, forwarding the data it receives in chunks. The forwarding destination endpoint should be customisable, with the restriction that it can receive the time series data. It was decided there is no data processing on the device for this prototype, but this could be an option in the future.

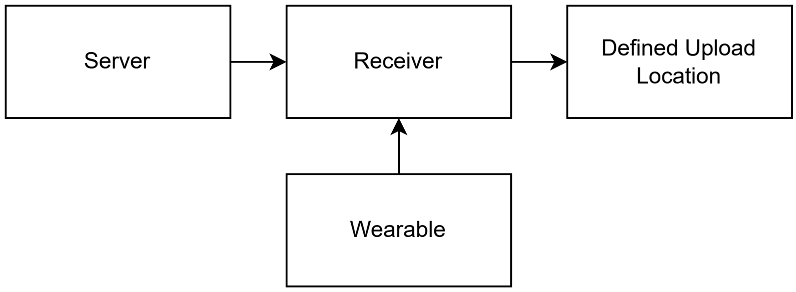


Figure 3.1: Communication between devices

The receiver should be remotely updatable. This will require compiled firmware to be stored on a server. As these are updates are not by smartphone manufacturers, they must be designed to uphold backwards comparability wherever possible.

This produces the constituent network diagram shown in figure 3.1

3.2.1 Multisensory

Multisensory design should be used wherever possible to account for patient's wide spectrum of possible disabilities, ideally never relying on a single sense [8]. Senses to consider are:

1. Visual
2. Auditory
3. Tactile

Sight is the primary form of human-computer interaction (HCI) through the touchscreen display. However some patients will be visually impaired so auditory interaction such as text-to-speech (TTS) should be used, at the very least to provide the option to narrate the diagnostic questions. Furthering this, common smartphone accessibility features include a screen reader and voice commands but these may be too aspirational for this computationally light device.

The device should send a reasonable amount of haptic feedback to confirm button presses. While screen-based and audio signals can be used to alert hearing-abled patients, the haptic motors can also be used to specifically alert deaf patients or simply find the receiver like a smartphone vibrating in a bag.

A light-emitting diode (LED) should also be included to alert the user, as the screen is only visible from a single side. The light should flash to get attention, but not so that its flashing becomes a nuisance, which can be described as greater than three times in any one second period [6, Success Criterion 2.3.2].

Alerts have previously been used to indicate abnormal vital signs [3]. If possible, support for this should be implemented so it can be activated if desired by the wearable manufacturer.

3.2.2 Icons

The device's language should be selectable. Displaying the language selection button as flag icons will make it language-agnostic.

An icon to display the connectivity of the wearable, as well as text, should be used and easily interpretable. Similarly an icon should also be used to display network connectivity.

3.2.3 Colours

Colours were chosen to display whether the device was functioning as expected. Red was chosen to indicate a problem and green otherwise. Only using red and green may be inaccessible for those with colour vision deficiency (CVD). Reinforced by WCAG, colours should be customisable [6, Success Criterion 1.4.8]. However, WCAG requires a minimum contrast ratio of at least 4.5:1 [6, Success Criterion 1.4.3] so there should therefore be some predetermined presets with a combination of colours that achieves this minimum.

3.2.4 Customisations

There should be a separate settings page in the software to control all customisable features, which should not require accessing for the device to function.

There should exist an option for the patient to disable haptics, flashing LEDs, and audio. There are common situations in which the device must be less pronounced to not become a nuisance to the patient, such as in a silent compartment of a train. This customisability is also supported by WCAG [6, Success Criterion 1.4.2] [6, Success Criterion 2.2.2]. Text should be resizeable [6, Success Criterion 1.4.4].

It was considered to not allow the disabling of all alert methods at the same time, but for each there may be scenarios in which this would be desired. The harm by allowing no notifications, possibly making the device not work until the patient realises, would also be equally possible with the original smartphone receiver.

3.2.5 Animations

Continuous rendering on icons when moving can improve the UX as showing continuity in transitions will be much easier for the patient to understand [9]. Animations shouldn't be too distracting and should strictly be used only when it adds information. There should be a customisation to disable all animations [6, Success Criterion 2.3.3].

3.3 Manufacturer Interface

As shown in figure 3.1, the receiver retrieves data from the server including the latest firmware for over-the-air (OTA) updates. The diagnostic questions which are asked to the patient should also be updatable. Therefore, the manufacturer needs a simple web-based interface to update their device-specific information.

4 Implementation

Contents

Based on section 3, implementation details were planned.

4.1 Platform

For embedded system, microcontrollers are common. Their lower computational abilities are usually a benefit as embedded devices typically require much less processing than general purpose computers and so extends battery life.

As the receiver is a single-purpose device, generalisation would introduce unnecessary cost and power consumption overheads. Furthermore, this prototype does not require onboard data processing further reducing computational requirements. Using a microcontroller for this design would be a great advantage over a smartphone receiver as it could result in a much longer battery life.

To choose how software is run on the receiver, the following models were considered:

1. real-time operating system (RTOS)
2. general purpose operating system (GPOS)

While they are able to be interchanged, RTOSes usually run on microcontrollers and GPOSes run on general purpose computers.

The requirements of the device very much suits an RTOS as it has time-sensitive low-level tasks such as Bluetooth low energy (BLE) communication. It also requires very little memory management, limited to only storing a buffer of data when the device has no internet connection.

As described in subsection 4.2.2, it is noticed that larger touchscreens tend to require a GPOS. A microcontroller with an RTOS is strongly preferable, but the choice of touchscreen component may limit this.

4.2 Hardware

To maintain the abilities of the smartphone, which the receiver must provide as described in section 1, standard features of a smartphone were listed. Their possible uses were determined in the dedicated receiver to inform the decision of what components to include, as shown in appendix B.

4.2.1 Controller

ESP32s are a popular microcontroller series for embedded projects due to their cost-efficiency for the features they provide. They also integrate Wi-Fi and Bluetooth modules but had limited general-purpose input output (GPIO) pins [10].

Other options such as the Raspberry Pi Pico were considered. The most competitive was the STM32 due to its high reliability and more configurable power optimisations. For the purpose of the intial prototype, the ESP32 was chosen. With adjustments including introducing Wi-Fi and Bluetooth modules, it can be swapped for the STM32 to meet stricter safety regulation requirements or more powerful variants if computation becomes a bottleneck.

For the prototype, a development board is acceptable but for the final product, the system on chip (SOC) should be soldered directly to a custom printed circuit board (PCB).

4.2.2 Touchscreen

An imperative feature of the receiver is a responsive touchscreen.

A touchscreen consists of two components: the display and the touch input. The selected touchscreen module should bundle both to ensure they work well together.

Touch input

There are two viable types of touchscreen input detection for this design: capacitive and resistive. Capacitive touchscreens are more durable, responsive, and can sense multiple concurrent touches. While capacitive touchscreens are more expensive, modern smartphones and other touchscreen devices come standard with them so has become expected by users.

Display Rates

To prevent screen tearing, a visual effect where the screen appears to be ripped, the refresh rate of the hardware must be greater than or equal to the frame rate of the software. The frame rate can partially be lowered in software to ensure no screen tearing occurs but if the frame rate is too low, the screen appears slow resulting in a bad user experience. The display should be able to provide a high enough frame rate for animations to no appear disjointed. However increasing frame rate also increases power consumption and humans have been shown to not notice large differences, even between 24 and 60 frames per second (FPS) [11].

Dimensions

The area of the touchscreen display should strike a balance between being large enough so it is easy to use and small enough so it is portable. In 2022, the average phones screen diagonal length was

6.3 inches [12]. To ensure usability of the receiver, its touchscreen should be larger than average. Therefore a display size criterion of between 6.3 and 10 inches was used.

The depth of the component was less restrictive, especially for this prototype. The depth should not make it difficult to hold the device, but a reasonable depth would improve durability by giving room for a more reinforced enclose.

Resolution

The number of pixels available to display the UI can cause a bad experience if too low. However, the more pixels that need to be rendered on every refresh, the more computation and memory are required. Therefore a balance must be found to minimise resolution without degrading the UX. The design shown in appendix A is simplistic and not reliant on high resolution so a lower resolution component would suffice.

Selected Component

Many available capacitive touchscreen components were either too too small or used HDMI ports to update the screen, requiring a GPOS likely because larger screens need more processing and would typically be used for more complex applications.

A leading component is Waveshare's ESP32 7" capacitive touchscreen because it is compatible with microcontrollers, including the ESP32-S3-N8R8 controller itself in the package. It is also packaged with a customised PCB, which may require modifications to repurpose GPIO pins that have been soldered to connectors. Even though the package is bundled with unneeded components, it is still a much more competitive price point than available products. The package is claimed to run Light and Versatile Graphics Library (LVGL)'s benchmark [13] at 41 FPS [14]. A limitation of this package is a significant number of pins are used by the touchscreen display [14].

4.2.3 Memory

Small amounts of quick-access memory may be required to store data when there is no internet connection to ensure no data is lost. The memory should be non-volatile as long periods without connection might be followed by a total discharge of the battery. The secondary storage will be much slower than random access memory (RAM), so should therefore only be used as a secondary measure if the free space, allocated to buffered data waiting to be uploaded, runs out.

4.2.4 Power

Non-rechargeable batteries remove the need for a mains electricity supply to recharge. While this could be advantageous for a few situations, it is not an unreasonable design constraint to assume the patient can find a mains supply after a full discharge cycle.

Using a rechargeable battery, a charging circuit is required to enable the battery to be charged while the device still functions.

4.2.5 Actuators

A vibrating motor disc should be used to produce haptic feedback. Along with an LED and speaker, the components can be connected through the I2C protocol.

4.2.6 Auxiliary Sensors

Additional sensors should be added within the receiver to collect the patient's environmental data, possibly highlighting problems caused by environmental conditions. The usefulness of this data is questionable however, for example it might record non-relevant data if the patient transports the receiver in a bag. These sensors would likely require some signal processing to ignore irrelevant data.

Global Positioning System (GPS) location data was considered as potential auxiliary data but the hurdles introduced by consent required to collect it as well as its little usefulness meant it was decided to not be included.

A light sensor should be used to adjust the screen brightness, with a control loop ensuring the screen is bright enough to be seen against background light.

There should be physical volume buttons to align with common touchscreen UXs.

There should be no power button as patients might inadvertently press it to disable the device. Instead, the device should be in a low power mode

4.2.7 Communication

As stated in 4.2.1, the ESP32 contains integrated modules that enable Wi-Fi and BLE communication. An additional long-term evolution (LTE) module will be required to allow mobile network connection.

It is a stretched goal to include an radio frequency identification (RFID) reader within the receiver to easily register and connect compatible wearable devices.

4.3 Software

C++ was selected as the preferred development programming language as the author had experience with it from previous modules. C++ is commonly used within embedded programming as it is fast and efficient while allowing for faster prototyping than C. It is also compatible with the Xtensa-based ESP32-S3 microcontroller.

4.3.1 Updates

Similar to a mobile device's receiver application being updated after release through the mechanism of the respective operating system (OS), such as Google's Play Store, it is possible to update a microcontroller's firmware virtually.

Using the esp32FOTA library [15], a large number of receivers can be remotely updated by downloading the new binary stored on a centralised server.

For a receiver to become aware of an available update, two options were considered:

1. Receiver very infrequently polls a Hypertext Transfer Protocol (HTTP) endpoint of the server

2. Server publishes binary to a Message Queuing Telemetry Transport (MQTT) topic which the receiver is subscribed to

It was decided to use option 1 because, while MQTT does allow for immediate updates, it uses excess energy by requiring a persistent connection and adds complexity by communicating through a broker. Furthermore, immediate updates are not a requirement of the system, so infrequent and flexible updates are acceptable.

4.3.2 Tasks

In an RTOS, functions of the microcontroller are separated into tasks. Tasks each have different priorities so more important processes can interrupt less important executions. Tasks for this design as designated priorities in table 4.3.2.

Task	**Priority**
Touch input	6
BLE connection	5
Render graphical user interface (GUI)	4
Upload	3
Screen Brightness Control	2
Update	1

4.3.3 Interface

The designs shown in appendix A should be implemented. LVGL is preferred for this as it has low processing requirements and can be run on the ESP32, including a working example for the selected touchscreen component [14].

If flickering becomes noticeable, a technique called double buffering can be used to reduce it. By pre-loading the next frame, visual defects can be smoothed however additional memory is used. The technique is supported by LVGL [16]. Furthermore, direct memory access (DMA) can be used to increase refresh rates.

To implement a full prototype, an interface to allow updates can be developed. This is not the main focus of the project so implementation should be in a frontend framework that minimises setup time.

5
Project Plan

Contents

So far, the receiver has been designed. As this project uses common techniques, lots of online resources were used. The remaining parts of the project are to implement and evaluate.

5.1 Organisation

Modules were strategically selected to finish the majority of other work in the first term with no summer exams.

The selected components in section 4 should be ordered once confirmed after a consultation with Dr. E. Stott. Ordering will need to be through multiple online vendors to be able to select the desired components.

The software logic described in section 4.3 can be implemented before the components arrive to efficiently utilise time. Functionality can not be tested without the components but basic errors can be fixed to ensure compilation is successful. Development efficiency should increase once testing is possible.

Shipping components could result in delays of up to a month since the time of purchase. Once a component arrives, it should be tested. if it perceived that a component does not work as

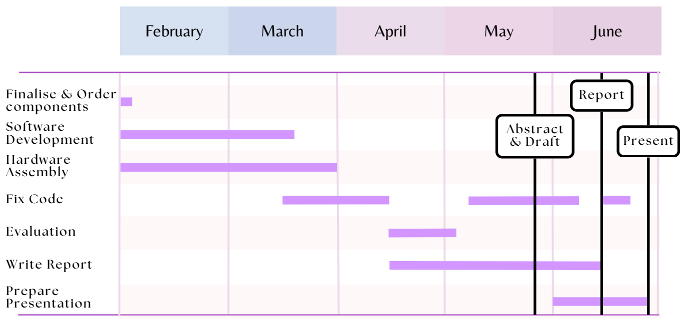


Figure 5.1: Gantt Chart

advertised or has unexpected behaviour, it may be necessary to order an alternative component. If a component is faulty, it may also be necessary to re-order the same component. Both of these scenarios will cause delays in the construction of the prototype. Completing the hardware assembly is a dependency to customise software to the prototype. If there are delays with the assembly, parts of the report can be done early.

5.2 Milestones

In order to ensure timely completion of work, specific milestones were set to accompany figure 5.1 (subject to delays described in subsection 5.1).

Milestone	**Date**
Touchscreen interface	10th Feb
BLE connection	24th Feb
Upload data	3rd Mar
Multisensory feedback	17th Mar
Auxiliary sensors	31th Mar
Integrated hardware prototype	7th Apr
Web interface prototype	21st Apr
Evaluation	12th May
Report	13th Jun
Presentation	26th Jun

6

Evaluation Plan

Contents

Mentioned in section 7, a full trial takes too long for this project. Therefore a mock environment must be setup to simulate the 'Wearable' and 'Defined Upload Location' as shown in figure 3.1.

6.1 Features

Testing must be carried out to determine if features function worse than, at the same level, or greater than those in a smartphone receiver.

6.1.1 Required

Connect to wearable and network

The receiver must be able to connect to the internet via Wi-Fi and LTE. The receiver must also connect to the wearable via BLE.

Reliably upload data received from the wearable

The use of this dedicated receiver should not reduce data that can be analysed. The device should be tested to ensure no additional data is lost compared to a smartphone receiver.

Diagnostic Questions

It should be possible to request and answer diagnostic questions.

Battery Life

The time in which the device is usable is less than or greater than a smartphone receiver.

6.1.2 Additional

Customisation

All customisable features, as described in subsection 3.2.4, should function as intended.

OTA updates

Updates to firmware should not affect the experience of the patient. They should occur in the background and in the event of an update, the receiver should maintain its state.

Auxiliary Sensors

Data recorded from auxiliary sensors should be accurate and reliably relate to the wearable's data at the same timestamp.

6.2 User Experience

Ideally, trials where patients could use the device over a prolonged period should be used and may be required, as discussed in section 7. For the purpose of this prototype's evaluation, a trial is too lengthy. Interviews can be conducted to ascertain the affect on short term user experience. The questionnaire should be designed to answer these questions:

1. Does the dedicated receiver improve or maintain the patient's experience?
2. Does the dedicated receiver improve or maintain doctors' experience ?

To design the questionnaire, influence is taken from the AcuPebble's usability survey [17] in which the following relevant questions were posed, with respondents answering with the degree of agreement to the statement:

1. I managed to follow all the steps on the mobile app without assistance.
2. I understood all instructions in the phone/tablet.
3. I felt confident using the app on the phone/tablet.

Although a greater number of participants is better, to remain realistic at least 10 English-speaking subjects should be selected to use the prototype and answer the above questions. The results would ideally be compared to a control group using a smartphone-based receiver with the same wearable to determine success. However this is not practical so results should be interpreted pragmatically in comparison to the AcuPebble's survey results [17] for the same questions.

7

Ethical, Legal, and Safety Plan

Contents

7.1 Design

As stated in section 1, the primary focus of the dedicated receiver is to maintain the ease-of-use of mobile receivers. As the device should be usable in all diagnostic contexts, the device must be accessible to a range of disabilities. The primary aim is therefore to reduce design bias against certain disabilities.

As IEC 62366-1:2015 [18] outlines, without using a usability engineering process, medical devices can be unintuitive. Consideration is essential to user perception as harmful errors can be minimised through considerate design.

7.2 Security

When collecting data, data protection laws must be followed such as the EU's General Data Protection Regulation (GDPR) and the UK's Data Protection Act 2018. As this dedicated receiver simply forwards data replacing the smartphone receiver, the medical institution should have existing permissions to record the private data. The main relevant regulation is to ensure data is "handled in a way that ensures appropriate security" [19]. To meet this requirement, data security best practices must be followed.

Data is transmitted from the wearable over BLE, known to be very secure except during the pairing process where it is susceptible to man-in-the-middle attacks unless proper pairing algorithms are implemented [20]. Data is also transmitted to the medical institution via HTTP. Hypertext Transfer Protocol Secure (HTTPS) should instead to be used to protect data.

In the event of theft, no personal information is able to be gained as the receiver is bound to a wearable. If the wearable is out of BLE range, no data is received. Authentication should be used for resetting the device, not as a theft prevention measure but to prevent the patient from inadvertently terminating data being uploaded and hence the wearables functionality.

GDPR also states data should be used transparently. Therefore, it is useful for the patient to receive a short overview of the device's function when they receive it.

7.3 Environment

Components should be sourced ethically, ensuring the suppliers are meeting regulations and not using exploitative practices.

There would be a negligible effect on energy consumption as the smartphone receiver carried out similar functions previously.

7.4 Testing

Functional testing should be carried to to ensure the main features of the device work as expected. Any basic failures detected would risk losing medical information and could harm patients.

As a separate auxiliary medical device, to be approved for use, clinical trials and likely required to prove the reliability of the device. Trials could measure statistical significance of hardware defects and more precisely measure the usability of the device than the methods stated in subsection 6.2.

When required, trials should be carried out in a way that ensures if the device is faulty, other methods are also considered to diagnose the patient. This can be done by using an approved method and statistically comparing the custom receiver alongside it, as conducted for the AcuPebble [17]. All trials must follow ISO 14155:2020's guidelines [21].

7.5 Hardware

The device should be safe for the patient to handle. The battery should be handled safely by avoiding extreme temperatures, overcharging, and short-circuiting. Circuitry should never be directly exposed, protected by an enclosure to prevent electrocution and fire risks. Waterproofing would be ideal but is not required for this prototype. Necessary warning labels, including electrical and battery safety, must be provided on packaging as described by ISO 15223-1:2021 [22].

Care should be taken during development to ensure safety by utilising protective equipment and proper precautions.

7.6 Intellectual Property

There are no obvious intellectual property infringement concerns as the device uses common techniques. Libraries should be carefully selected to ensure, if desired, commercial use is acceptable. General Public License (GPL) licences require the modified code must also be open sourced. Most libraries require crediting authors.

A
User Interface Designs

The design choices, described in section 3, were implemented as shown in figure A.2. Navigation between these screens is shown in A.1.

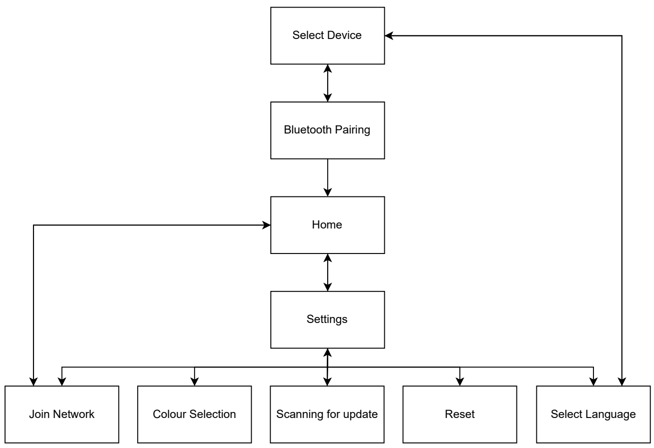
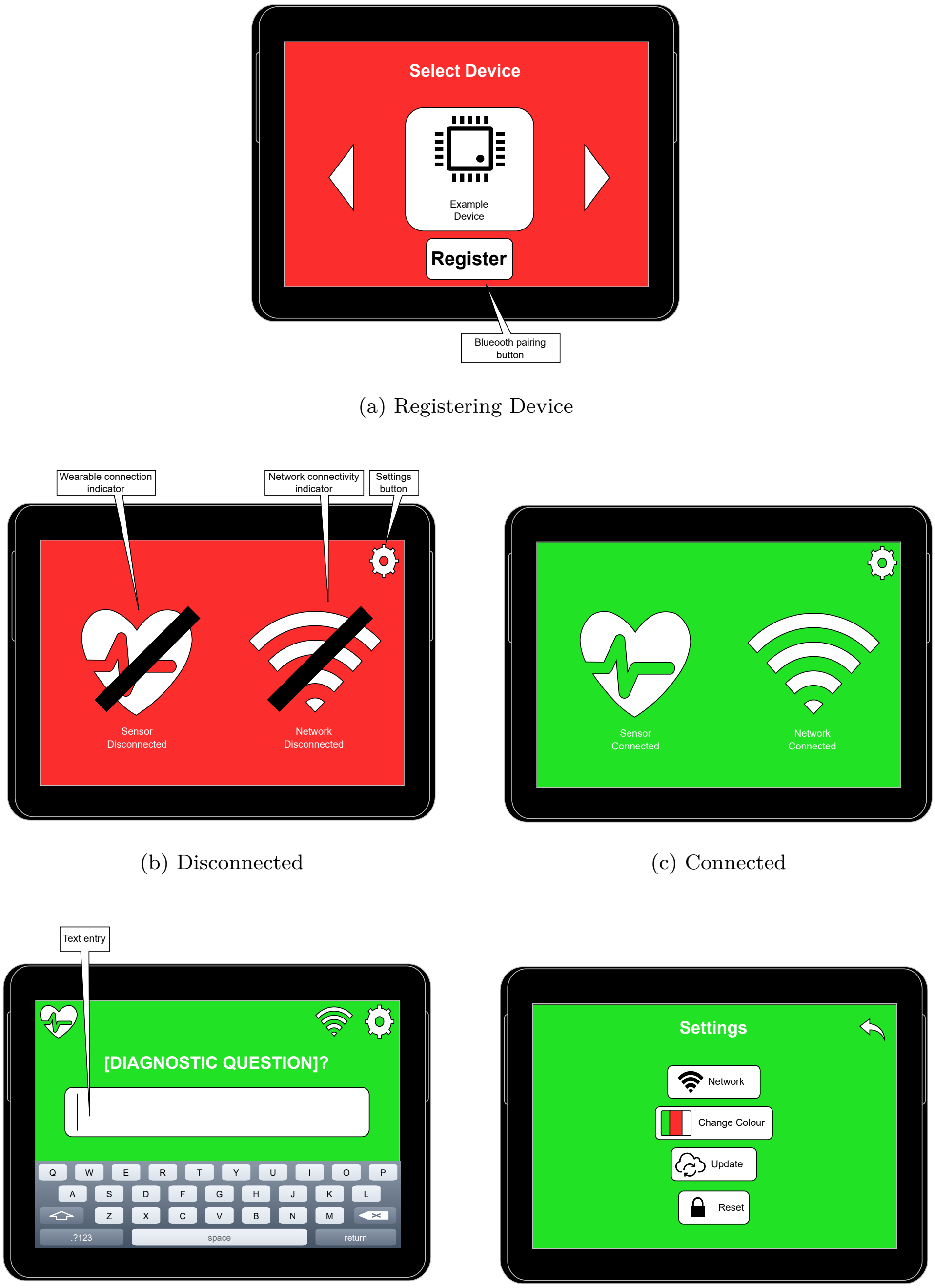


Figure A.1: Navigation between pages



(a) Registering Device

(b) Disconnected

(c) Connected

(d) Question

(e) Settings

Figure A.2: User Interface Designs

B

Smartphone Components

Feature	Receiver-Related Function	Include in receiver?
Touchscreen	HCI	✓
Processor	General computation	✓
Memory	Temporarily store diagnostic data	✓
Battery	Enable portability	✓
Haptic Actuator	Alerts for deaf patients	?
LED	Alerts for deaf patients	✓
Speaker	Alerts for visual impaired patients	✓
Buttons	Control volume	✓
Light Sensor	Auto-adjust screen brightness	✓
Camera	Gesture recognition	✗
GPS	Auxiliary diagnostic data	?
Accelerometer	Auxiliary diagnostic data	?
Barometer	Auxiliary diagnostic data	?
BLE	Communication with wearable	✓
Wi-Fi	Upload diagnostic data	✓
LTE	Upload when no available Wi-Fi	✓
RFID	Simple setup	?
Enclosure	Protect circuitry	✓
Waterproofing	Improve durability	?

Bibliography

[1] E. Rodriguez-Villegas, *Project description*, Accessed: 2024-10-16.

[2] Acurable Limited, *Acupebble sa100*, https://acurable.com/en/products/acupebble-SA100, Accessed: 2025-01-14.

[3] F.-Y. Leu, C.-Y. Ko, I. You, K.-K. R. Choo, and C.-L. Ho, "A smartphone-based wearable sensors for monitoring real-time physiological data," *Computers Electrical Engineering*, vol. 65, pp. 376–392, 2018, ISSN: 0045-7906. DOI: 10.1016/j.compeleceng.2017.06.031.

[4] J. Mitchell, *The New iPhone Operating System Will Break Your Old Apps*, https://www.standard.co.uk/news/tech/the-new-iphone-operating-system-will-break-your-old-apps-a3638566.html, Accessed: 2025-01-18, 2017.

[5] S. Gladun, *App Maintenance: Why and How Often You Should Update Your App*, https://agilie.com/blog/app-maintenance-why-and-how-often-you-should-update-your-app, Accessed: 2025-01-18, 2018.

[6] World Wide Web Consortium, *Web content accessibility guidelines*, https://www.w3.org/TR/WCAG21/, version 2.1, Accessed: 2025-01-14.

[7] Bureau of Internet Accessibility, *How Do We Perform Accessibility Testing for the Impact of Cognitive, Learning, and Neurological Disabilities?* https://www.boia.org/blog/how-do-we-perform-accessibility-testing-for-the-impact-of-cognitive-learning-and-neurological-disabilities, Accessed: 2025-01-18, 2023.

[8] International Electrotechnical Commission, "PD IEC/TR 62366-2:2016: Guidance on the application of usability engineering to medical devices," International Electrotechnical Commission (IEC), Tech. Rep. IEC/TR 62366-2:2016, 2016, Accessed: 2025-01-18.

[9] Jakob Nielsen, *Guidelines for Multimedia on the Web*, https://www.nngroup.com/articles/guidelines-for-multimedia-on-the-web/, Accessed: 2025-01-18, 2020.

[10] Espressif Systems, "ESP32-S3-WROOM-1/WROOM-1U Datasheet," Espressif Systems, Tech. Rep., version 1.2, 2023, Accessed: 2025-01-18.

[11] F. Pazhoohi and A. Kingstone, "The effect of movie frame rate on viewer preference: An eye tracking study," *Augmented Human Research*, p. 4, 2021. DOI: `10.1007/s41133-020-00040-0`.

[12] D. Hsieh, "Display dynamics – january 2022: Average smartphone display size stays at 6.3 inches while the resolution can be potentially enhanced," *Omdia*, 2022.

[13] LVGL, *Benchmark demo*, `https://github.com/lvgl/lvgl/tree/v8.3.6/demos/benchmark`, version 8.3.6, Accessed: 2025-01-18, 2022.

[14] Waveshare Electronics, *Esp32-s3-touch-lcd-7 documentation wiki*, `https://www.waveshare.com/wiki/ESP32-S3-Touch-LCD-7`, Accessed: 2025-01-14.

[15] C. Joyce, *Esp32fota*, `https://github.com/chrisjoyce911/esp32FOTA`, version 2.9, Accessed: 2025-01-14, 2024.

[16] LVGL, *Lvgl display buffering: Two buffers*, `https://docs.lvgl.io/8.2/porting/display.html#two-buffers`, Accessed: 2025-01-18.

[17] N. Devani, R. X. A. Pramono, S. A. Imtiaz, S. Bowyer, E. Rodriguez-Villegas, and S. Mandal, "Accuracy and usability of acupebble sa100 for automated diagnosis of obstructive sleep apnoea in the home environment setting: An evaluation study," *BMJ Open*, vol. 11, no. 12, 2021, ISSN: 2044-6055. DOI: `10.1136/bmjopen-2020-046803`.

[18] International Electrotechnical Commission, *IEC 62366-1:2015: Medical devices – Part 1: Application of usability engineering to medical devices*, Accessed: 2025-01-18, 2015.

[19] *Data protection act 2018*, `https://www.legislation.gov.uk/ukpga/2018/12/contents/enacted`, Accessed: 2025-01-18, 2018.

[20] T. Melamed, "An active man-in-the-middle attack on bluetooth smart devices," *International Journal of Safety and Security Engineering*, vol. 8, no. 2, pp. 196–204, 2018. DOI: `10.2495/SAFE-V8-N2-200-211`.

[21] International Organization for Standardization, *ISO 14155:2011: Clinical investigation of medical devices for human subjects — Good clinical practice*, Accessed: 2025-01-18, 2020.

[22] International Organization for Standardization, *ISO 15223-1:2021: Medical devices — Symbols to be used with medical device labels, labelling and information to be supplied*, Accessed: 2025-01-18, 2021.