

École polytechnique de Louvain

Smartphone repurposing for open-source medical applications

HealthyPi 5 open-source multi-parameter monitor:
smartphone integration and validation

Author: **Adrien ANTONUTTI**

Supervisors: **David BOL, Benoît HERMAN**

Readers: **Jean-Pierre RASKIN, Thomas ANTOINE, Régis LOMBA**

Academic year 2025–2026

Master [120] in Electrical Engineering

Acknowledgements

I would like to express my gratitude to the people who helped and supported me throughout the completion of this master thesis.

First and foremost, I would like to thank Benoît Herman and David Bol, my supervisors, for their guidance throughout this work and their constructive feedback. Their advice kept me moving forward and helped clarify my vision at every stage.

I would like to thank Régis Lomba for his advice and remarks during the year, particularly regarding open-source practices.

I would also like to thank Jean-Pierre Raskin for agreeing to be a reader and member of the jury for this thesis.

I would also like to thank Thomas Antoine for his advice on academic writing, his careful proofreading, and his help with the smartphone-related tools, as well as for his constant availability throughout this work.

I am particularly grateful to Ludovic Gérard and Christophe Carboni, as well as the Cliniques Universitaires Saint-Luc, for granting me access to the hospital infrastructure, which proved invaluable for the validation of the prototype. I would like to extend a special thanks to Ayhan Isyapan for his warm welcome and his thorough explanations of the various devices.

I would also like to thank the administrative staff of the IMMC for handling the ordering of the necessary components.

Finally, I would like to express my sincere gratitude to Laura and Jean-Michel for their careful proofreading, and to my parents for their support and trust throughout my studies.

Abstract

Access to medical devices remains a critical challenge in low or middle income countries, where supply chain disruptions, high costs, and regulatory barriers limit the availability of essential diagnostic equipment. Open-source approaches have been proposed as a possibility to address these issues, but their practical viability remains insufficiently demonstrated. This thesis investigates to what extent smartphone repurposing can serve as a viable platform for more equitable healthcare access, across three objectives: cost reduction, supply chain resilience, and adaptation to local needs. Two complementary approaches are explored. First, a literature review of existing smartphone-based medical applications demonstrates that clinically relevant diagnostic performance can be achieved for parameters such as heart rate, SpO₂, jaundice and Parkinson's disease screening, using mostly sensors already embedded in the device. Second, the HealthyPi 5 open-source multi-parameter monitor is selected as a case study for smartphone integration. After identifying and resolving critical bugs in the firmware, ECG, respiration and PPG acquisition are brought to a functional state. A design space exploration of possible smartphone integration approaches leads to the implementation of USB and UART communication channels, enabling a cost reduction of approximately 30%. Experimental validation carried out at the Cliniques Universitaires Saint-Luc confirms the functional acquisition of ECG, respiration and heart rate, while SpO₂ estimation remains unreliable and requires further investigation. The results highlight both the promise and the current limitations of open-source medical platforms: cost reduction is achievable, firmware adaptability enables customisation that commercial devices do not allow, but supply chain resilience is not improved for platforms relying on specialised integrated circuits, and the path to clinical certification and its associated cost remain uncertain.

Contents

1	Introduction	1
1.1	Context	1
1.2	Objectives	2
1.3	Outline	3
2	Medical device in open-source context	5
2.1	Open-source	5
2.1.1	Definition of hardware and software open-source	5
2.1.2	Challenges for open-source hardware	6
2.1.3	Evaluation metrics for OSH project quality	7
2.2	Medical device	9
2.2.1	Definition	9
2.2.2	Regulatory context	10
2.3	Opportunities of open-source medical devices	11
2.4	Challenges of open-source medical devices	11
3	Literature review of smartphone repurposing for medical applications	13
3.1	Heart rate measurements	13
3.1.1	Heart rate measurement technique	13
3.1.2	Comparison of smartphone based HR measurement	14
3.2	Blood oxygen saturation measurements	15
3.2.1	Pulse oximetry principle	16
3.2.2	Comparison of smartphone implementation	17
3.3	Jaundice detection	19
3.4	Parkinson disease detection	21
3.4.1	Audio based PD diagnostic	22
3.4.2	Accelerometer based PD diagnostic	24
3.5	Hear loss detection	24
4	Multi-parameter monitor	27
4.1	Definition	27
4.2	Physiological parameters	28
4.3	General requirements	31
4.4	Suitability for smartphone integration	32
4.5	Previous prototype at UCLouvain	32

5	HealthyPi 5 open-source project modification	33
5.1	Description of HealthyPi 5 open source monitor	33
5.1.1	Features	33
5.1.2	HealthyPi basic kit content	34
5.1.3	System hardware overview	36
5.1.4	Software overview	37
5.2	Preliminary functional evaluation of the system	39
5.2.1	Voltage regulation check	39
5.2.2	First test of the system	39
5.2.3	Overall integration	41
5.3	Troubleshooting and modifications	42
5.3.1	Communication issue	42
5.3.2	PPG acquisition troubleshooting and modification	44
5.3.3	MAX30001 Troubleshooting	50
5.3.4	Minor modifications	56
6	Smartphone integration to the system	57
6.1	Design space exploration	57
6.1.1	Cost repartition of HP5 system	57
6.1.2	Comparison of possible connection between smartphone and AFEs	59
6.1.3	Cost saving for each solution	61
6.2	Setup of USB communication	62
6.3	Setup of UART communication	63
6.3.1	Implementation on Google Pixel 6	64
6.3.2	OpenView modification to access UART port	67
7	Results and analysis	69
7.1	Characterisation of different communication channels	69
7.1.1	Bitrate	69
7.1.2	Power consumption	70
7.2	System performance analysis	72
7.2.1	Validation methodology	72
7.2.2	Signal acquisition	73
7.2.3	Vital metrics	76
7.3	Maturity assessment of HealthyPi 5	80
7.3.1	Openness of HealthyPi 5	80
7.3.2	Technical maturity	81
7.3.3	Documentation maturity	82
8	Discussion	83
8.1	Return of experience of HP5	83
8.2	Reduced cost	83
8.3	Mitigation to supply chain disruption	84
8.4	Adaptation to local needs	84
8.5	Perspectives	84
9	Conclusion	85

Appendix	87
A.1 Use of AI tools	87
A.2 Code	87
A.2.1 Communication	87
A.3 Logs from Google Pixel 6 UART	89
A.4 Screenshot of OpenView application	91
A.5 Picture of Philips IntelliVue MP70	92
A.6 Reproducible preliminary test for MAX30001 communication	92
A.7 Firmware flash procedure	93
A.8 Price study information	94
A.9 Requirements for multi-parameter monitor	95
A.10 Description of Blood oxygen saturation (SpO ₂) and HR PPG-based algorithm	95

List of Acronyms

ADC Analog-to-Digital Converter

AFE Analog Front-End

AUC Area Under the Curve

BLE Bluetooth Low Energy

BoM Bill of Materials

CSI Channel State Information

ECG ElectroCardioGram

HP5 HealthyPi 5

HR Heart Rate

IC Integrated Circuit

IEC International ElectroTechnical Commission

ISO International Organisation for Standardisation

LOA Limit Of Agreement

MAE Mean Absolute Error

MD Medical Device

MDR Medical Device Regulation

ODRL Open Documentation Readiness Level

OSH Open-Source Hardware

OSS Open-Source Software

OTRL Open Technology Readiness Level

PD Parkinson's Disease

PPG PhotoPlethysmoGraphy

PTA Pure-Tone Audiometry

QSPI Quad Serial Peripheral Interface

RMSE Root Mean Square Error

RR Respiration Rate

SCG SeismoCardioGraphy

SPI Serial Peripheral Interface

SpO₂ Blood oxygen saturation

TcB TransCutaneous Bilirubinometer

TRL Technology Readiness Level

TSB Total Serum Bilirubin

UPDRS Unified Parkinson's Disease Rating Scale

List of Figures

3.1	Molar absorption coefficients of HbO ₂ and RHb in function of incident light wavelength, extracted from [42]	17
3.2	Typical PPG signal, extracted from [47]	17
3.3	Bhutani nomogram used for risk-assessment of bilirubin level, extracted from [52]	19
3.4	AUC definition, extracted from [74]	23
4.1	Normal ECG waveform, extracted from [102]	28
4.2	PPG waveform characteristics, extracted from [104]	29
5.1	Components present in the HealthyPi 5 (HP5) basic kit	35
5.2	Schematic overview HP5 hardware	36
5.3	RP2040 firmware architecture	38
5.4	ECG (green), PPG (yellow) and respiration (blue) signals with original firmware and OpenView app on desktop, when using electrodes.	40
5.5	ECG (green), PPG (yellow) and respiration (blue) signals with original firmware and OpenView app on smartphone.	41
5.6	Effect of finger passing near the SpO ₂ finger probe on the PPG (yellow) signal.	41
5.7	AFE4400 schematic bloc, extracted from [125]	45
5.8	AFE4400 TIA schematic bloc, extracted from [125]	47
5.9	Gain of high pass filter for PPG signal	49
5.10	PPG signal acquired with modifications, with ambient light variation	50
5.11	PPG signal motion artefact	50
5.12	Frequency content of V from ideal simulation	51
5.13	MAX30001 ECG channel description, extracted from [132]	51
5.14	MAX30001 Respiration channel, extracted from [132]	53
5.15	Recommended ECG electrodes placement, extracted from HealthyPi 5 (HP5) documentation[135]	55
5.16	Alternative ECG electrodes placement	55
5.17	Obtained ECG trace with placement of figure 5.16	55
6.1	Cost repartition for the complete system	58
6.2	Cost repartition for electronic components	58
6.3	Cost repartition for ICs	58
6.4	Homepage displaying available Bluetooth devices	62
6.5	Application drawer for USB and UART streaming interface access, main page is used to access Bluetooth	62
6.6	Diagram of OpenView Flutter application logic before modifications	63
6.7	Diagram of OpenView Flutter application logic after modifications	63

6.8	USB-C pins, extracted [146]	64
6.9	USB-C breakout board	64
6.10	Setup for UART communication between HP5 board and Google Pixel 6 smartphone	64
6.11	Program logic diagram for UART integration to OpenView app	68
7.1	Throughput comparison of communication channel	69
7.2	Setup for power consumption measurements	70
7.3	Current and power consumption of HP5 with BLE, USB and UART communication	71
7.4	Prosim 8 : ECG electrodes are placed on ECG contact and optical emitter and detector emulates the finger for PPG and Blood oxygen saturation (SpO ₂) testing, extracted from [157]	73
7.5	ECG, PPG and respiration signal acquisition from HP5, with additional signal processing for PPG waveform	75
7.6	Picture of the monitor screen, taken with smartphone and background reflection removed with AI tool, original in figure 3	75
7.7	Scatter plot of measured HR in function of true HR	77
7.8	Mean and standard deviation of error for different true HR	77
7.9	Detected PPG peaks with identical algorithm used on the HP5	78
7.10	SpO ₂ measurement with reference monitor and HP5	79
7.11	SpO ₂ difference between HP5 and Philips monitor SpO ₂ estimation	79
7.12	RR scatter of HP5 and Philips monitor estimation, with additional points for corrected calibration factor on HP5	80
1	Screenshot of OpenView application, reading data from the UART port of the Google Pixel 6	91
2	Screenshot of OpenView application, with final application and firmware while tested on healthy subject	91
3	Raw picture of the monitor screen, background was removed for better visualisation	92
4	ECG (green) and respiration (blue) signals when randomly touching LA and RA probe. Orange arrows correspond to the timing when fingers come into contact with the probe and purples arrows when contact ends.	93
5	Complete general technical requirements for advanced multi-parameter monitor.	95

List of Tables

2.1	Summary of Open Technology Readiness Levels (OTRL)	9
2.2	Summary of Open Documentation Readiness Levels (ODRL)	9
3.1	Comparison of correlation and MAE for multiple paper comparing ECG derived HR and smartphone derived HR	14
3.2	MAE of proposed method against medical graded pulse oximeter	18
3.3	Comparison of correlation r and LOA (95% confidence interval) for studies using different ROI, calibration methods and algorithms. Methods are grouped by ROI.	20
3.4	Binary classification for PD patient and healthy patient. *PD/HC repartition not reported.	23
3.5	Binary classification for PD patient and healthy patient using smartphone's accelerometer	24
4.1	Summary of general technical requirements for basic, intermediate, and advanced multi-parameter patient monitors. Extracted from <i>Medevis</i> database.	31
5.1	Operating modes of HP5	34
5.2	Voltage measurements at different test points on the HP5 board	39
5.3	Functionality of HP5 features, on desktop USB streaming and smartphone BLE streaming	41
5.4	Packet structure used by OpenView	43
5.5	HP5 payload format for packet type 0x02	43
5.6	Summary of AFE4400 output data channels	45
5.7	Input low pass filter configuration	48
5.8	ECG channel configuration	52
5.9	BioZ/respiration channel configuration	53
6.1	Estimated cost reduction for the different smartphone integration approaches	61
6.2	Achieved baudrate on Linux and Android OS on Google Pixel 6	67
7.1	Actual and theoretical maximum required bitrate between Google Pixel 6 and HP5	70
7.2	Configuration parameters used for the signal acquisition validation . . .	74
7.3	Extracted characteristics of the acquired ECG, PPG and respiration signals	76
7.4	Open-o-meter evaluation of the HealthyPi project	81
7.5	Estimated OTRL level for each physiological parameter acquisition implemented on HealthyPi	82

1	Estimated hardware costs and suppliers	94
2	Examples of I2C-compatible AFE alternatives considered for communication between AFEs and smartphone using I2C bus. All integrated indicate if the Integrated Circuit (IC) contains all necessary components for PPG acquisition : a photodiode and LED to emit light. ECG AFE is necessary not "all integrated" since it requires external electrodes.	94

Chapter 1

Introduction

1.1 Context

Health is one of the most important aspect of human life and one of the first components for a fulfilled life. It is also a human right, as stated in the World Health Organisation constitution, *"the enjoyment of the highest attainable standard of health is one of the fundamental rights of every human being without distinction of race, religion, political belief, economic or social condition"*[1]. In an increasingly unstable world, it is important to remember that *"health of all peoples is fundamental to the attainment of peace and security"*[1], as also stated in the WHO constitution.

One of the components to achieve this goal is access to medical infrastructure for all human beings, and Medical Devices (MDs) are part of it. Unfortunately, there is a strong healthcare access inequality, as *"only 13% of the global population accounts for 76% of the global MD use"*[2]. One illustration of this is that *"40% of healthcare equipment in low or middle income countries (LMICs) is out of service, compared with less than 1% in high income countries"*[3]. As 80% of the MD market shares is held by USA, Europe and Japan [2], the majority of resulting MDs are inappropriate in low resource context. LMICs rely mostly on donation for access to MDs: *"up to 70% of medical equipment in sub-Saharan Africa is donated, only 10% to 30% of donated equipment becomes operational"*[3]. Operational and environmental condition such as dust, humidity and temperature are different in LMICs than in the origin countries of these donations, and the lack of technician and spares parts to correctly maintain MDs makes the traditional medical supply chain ineffective in these contexts.

The COVID-19 pandemic further highlighted these vulnerabilities; when global supply chains collapsed, many regions were left without essential equipment like ventilators or intensive care unit beds [4]. This crisis demonstrated the urgent need for locally maintainable and adaptable medical technologies, where open-source designs and hardware repurposing could play a life-saving role.

Simultaneously, the unprecedented democratisation of high-performance computation devices has been accelerated by the ubiquity of smartphones. In 2025, it is estimated that 6.9 billion people globally use smartphones [5]. This represents 84% of the world population. Smartphones are now equipped with high performance hardware such as high resolution image sensors, quality microphone, accelerometer, energy storage

through the battery, quality screen and impressive computational capabilities.

Smartphone repurposing - the act of finding a new use for a smartphone - is an opportunity for a more equitable healthcare access. Using hardware already present in the phone such as battery, display, sensors and computational capabilities as base in the conception of MDs could lead to medical equipment more suited to low resource settings. Existing applications already propose the reuse of smartphone for applications such as Heart Rate (HR) measurements, blood oxygenation or jaundice detection. As MDs are heavily regulated, the question of how those implementations perform compared to standard reference MDs arises.

Furthermore, the *open-source* philosophy provides a novel approach for the design of MDs, shifting the paradigm from proprietary black-boxes to transparent, collaborative ecosystems. Open-source ensures that medical tools are not only accessible but also re-appropriable: by providing public access to hardware schematics, firmware, and software source code, it empowers local engineers to inspect, modify, and repair devices without reliance on the original equipment manufacturer. This is particularly critical in LMICs, where the right to repair is often hindered by restrictive intellectual property or the absence of official technical support. Open-source methodologies therefore allow for rapid iterative improvements and the adaptation of devices to specific local clinical needs, thereby increasing the resilience of healthcare systems against crisis or inappropriate design.

In conclusion, the ubiquity and affordability of high-performance smartphones and the maturity of open-source collaborative models create a unique window of opportunity. While the traditional MD industry often overlooks the specific constraints of low-resource settings, a bottom-up approach—leveraging existing consumer electronics and Open-Source Hardware (OSH)—could bridge the widening gap in global health equity. However, transforming a ubiquitous device, such as the smartphone, into a reliable clinical tool requires rigorous technical validation and a clear understanding of the regulatory landscape. This work explores this path by investigating how repurposed smartphones, associated with open-source engineering, could provide a robust, cost-effective, and adaptable alternative for MDs.

1.2 Objectives

The goal of this master thesis is to investigate the following research question : **To what extent can smartphone repurposing serve as a viable platform for a more equitable healthcare by addressing these key objectives :**

- **Reducing the initial cost of MDs.**
- **Accelerating the development of MDs adapted to the constraints of low resource environment.**
- **Mitigating reliance on conventional supply chains through local or open-sources alternatives.**

To fully explore the potential of smartphone hardware, a literature review of existing use cases for smartphones as MDs is conducted. For each application, various

methodologies and results are benchmarked against medical gold standards. Then, an existing open-source project—a multi-parameter monitor—is adapted to integrate a smartphone as a core component. This practical adaptation serves as a case study to assess the system’s technical maturity, adaptability, and openness, while highlighting the specific engineering challenges encountered during the integration of a consumer smartphone to an existing medical monitoring system.

1.3 Outline

Chapter 2 defines open-source and the regulatory context around the design of MDs. It highlights the opportunities and challenges that open-source methodologies bring to the MD landscape.

Chapter 3 provides a literature review of existing smartphone repurposing projects for medical applications. For each use case, the smartphone-based implementation is compared against the accepted clinical reference standard, allowing to characterise what performance can be expected from smartphone repurposing across a range of diagnostic applications.

Chapter 4 introduces multi-parameter monitors: their clinical definition, the physiological parameters they acquire, and their general requirements. It motivates the suitability of multi-parameter monitors as a target application for smartphone integration, and presents the previous prototype developed at UCLouvain.

Chapter 5 presents the HealthyPi 5 open-source platform selected for this work. An overview of its hardware and software architecture is provided, followed by a preliminary evaluation of the system. The troubleshooting and modifications done to establish a functional baseline are documented in detail.

Chapter 6 explores the design space from a cost perspective for smartphone integration and details the implementation of USB and UART communication channels.

Chapter 7 presents the experimental results: signal acquisition and vital metrics are validated against a clinical reference device, and the technical and documentation maturity of the platform is assessed using open-source evaluation metrics. A comparison of each communication channel in terms of bitrate and power consumption is also given.

Finally, Chapter 8 synthesises the findings, discusses the system’s performance in light of the three research objectives, and proposes future perspectives. Chapter 9 concludes this work with a summary of the main contributions and results.

Chapter 2

Medical device in open-source context

2.1 Open-source

2.1.1 Definition of hardware and software open-source

Originally emerging from the software development community, the open-source philosophy promotes transparency, collaboration, and knowledge sharing between users and developers. To define Open-Source Software (OSS), the *Open Source Initiative* established ten fundamental criteria that a project should satisfy in order to be considered open-source. Those criteria are the following[6]:

1. **Free Redistribution** : The license must allow free distribution or sale of the software without requiring royalties or additional fees.
2. **Source Code** : The source code must be accessible and distributed in a form that allows modification and understanding of the software.
3. **Derived Works** : Users must be allowed to modify the software and redistribute modified versions under the same licensing conditions.
4. **Integrity of the Author's Source Code** : Modified versions may be restricted from using the original source directly, but distribution of modified software must remain permitted.
5. **No Discrimination Against Persons or Groups** : The license must grant equal rights to all individuals and groups.
6. **No Discrimination Against Fields of Endeavor** : The software must be usable in any application domain without restriction.
7. **Distribution of License** : The rights associated with the software must automatically apply to all redistributed copies.
8. **License Must Not Be Specific to a Product** : The license rights must remain valid independently of a particular software distribution or product.
9. **License Must Not Restrict Other Software** : The license must not impose conditions on other software distributed alongside the OSS.

10. **License Must Be Technology-Neutral** : The license must not depend on a specific technology, platform, or interface type.

Over time, the open-source philosophy has progressively expanded beyond software and are now also applied to hardware systems, leading to the development of OSH. Due to its physical nature, OSH introduces its own additional challenges related to manufacturing and reproducibility[7]. According to [8], the development of OSH has been facilitated by two major technological evolutions. First, the increasing accessibility of relatively low-cost manufacturing technologies, such as 3D printers, enables users to fabricate mechanical parts directly from shared digital design files. Second, the emergence of standardised and accessible electronic prototyping platforms, such as *Arduino*, has simplified the development of embedded systems by providing reusable hardware and software ecosystems for sensor acquisition, signal processing, or actuator control.

OSH is defined by the *Open Source Hardware Association* as "*hardware whose design is made publicly available so that anyone can study, modify, distribute, make, and sell the design or hardware based on that design. The hardware's source, the design from which it is made, is available in the preferred format for making modifications to it. Ideally, open source hardware uses readily-available components and materials, standard processes, open infrastructure, unrestricted content, and open-source design tools to maximize the ability of individuals to make and use hardware. Open source hardware gives people the freedom to control their technology while sharing knowledge and encouraging commerce through the open exchange of designs.*"[9].

2.1.2 Challenges for open-source hardware

Unlike OSS, the physical nature of hardware projects introduces multiple additional challenges that software projects do not have to address [7]:

- **Access to infrastructure** is required, access not only to digital resources such as source files and internet connectivity, but also to physical infrastructure like electronic equipment, manufacturing tools, prototyping platforms, etc, is needed [7]. Community workshops like "FabLabs" are often proposed as a solution to improve access to infrastructure by providing shared tools and collaborative prototyping spaces[10].
- **Technical skills** required for hardware development may also limit participation in OSH projects. Unlike software projects, hardware systems often require multi-disciplinary competencies in electronics, embedded programming, mechanical design, and manufacturing processes, which can reduce the number of potential contributors and increase the need for knowledge transfer between experienced and novice contributors[11].
- **Complex and iterative process** : The hardware design process is generally more complex and iterative than software development because it involves multiple interconnected stages, each one depending on the previous one. For the case of electronic product, it includes electronic schematic design, PCB layout, embedded

firmware integration, mechanical design, prototyping, and manufacturing validation. Errors introduced during one stage may require costly redesign and physical re-fabrication of components [11].

- **Reproducibility** represents another major challenge for OSH projects. To ensure that designs can be replicated by external contributors, manufacturing processes should preferably rely on standardised and accessible fabrication tools, such as equipment commonly available in FabLabs or makerspaces. Similarly, electronic components and materials should remain globally available, interchangeable, or replaceable by locally sourced alternatives when possible. Production costs must also remain compatible with the targeted accessibility objectives, including the cost of raw materials and fabrication processes such as PCB manufacturing or 3D printing. Comprehensive documentation, including schematics, bill of materials, assembly instructions, manufacturing files, and testing procedures, must also be provided to guarantee reproducibility of the hardware platform [11].

2.1.3 Evaluation metrics for OSH project quality

Evaluating the success of an OSH project is not straightforward, as it involves multiple technical, social, and organisational dimensions. [12] identified three major dimensions that characterise the success of OSH projects, each encompassing several related aspects summarized below[12]:

1. Successful OSH projects create value :

- Create value for contributors through gratification, recognition, academic or professional impact, and long-term engagement.
- Provide useful and accessible hardware to users, with sustained usage and adoption by diverse communities.
- Enable derivative projects and remixes, fostering innovation and reuse.
- Contribute to society by improving access to knowledge, advancing science and technology, and supporting public health.
- Generate commercial value through sustainable business models, hardware sales, and commercial adoption.
- Remain sustainable over time through funding, community continuity, and reduced dependence on the original creators.

2. Successful OSH projects create high-quality outputs :

- Develop performant, reproducible, modifiable, and accessible hardware.
- Provide complete, accessible, and editable documentation.
- Use open standards, open formats, and transparent design practices.
- Produce hardware that effectively solves a practical problem or fulfills a specific need.

3. Successful OSH projects have effective and open processes :

- Follow good product design and development practices.

- Use effective project management, version control, and traceable contributions.
- Foster active communities of users, contributors, and developers.
- Maintain transparent workflows and openly share knowledge, lessons learned, and design decisions.
- Support openness through open-source toolchains, licensing, and collaborative practices.

For each success characteristic, metrics have been proposed to measure the progress toward it. A total of 101 metrics was identified, making their practical use difficult. The next sections therefore present three commonly used metrics to evaluate the openness, technical maturity, and documentation maturity of an OSH project.

Measure of openness

To measure to which degree an OSH project is open, a rating scheme called "**Open-o-Meter**" has been proposed by [13]. This scale consists of 8 criteria that are rated to 0 if not met, 0.5 if partially met and 1 if fully met, resulting in a scale ranging from 0 to 8. Criteria are the following and are separated in 2 categories :

- **Product openness :**

- Hardware/software must be published under relevant OSH/OSS licences.
- Design files of all components (software/electronic/non-electronic) designed by the originator are made publicly available.
- A bill of material is available.
- Assembly instructions are available.
- All files mentioned above are published in the original and editable format.

- **Process openness :**

- A version control system is used (e.g. Git like software and Github collaborative platform).
- A contribution guide showing how to contribute to the project is available.
- An issue tracking or equivalent systems is present to manage tasks and improvement proposition.

Measure of technical maturity

Introduced by NASA during the 1970s, the Technology Readiness Level (TRL) scale allows technologies to be classified according to their level of technological maturity[14]. Its main advantage is its simplicity[11], however it is not fully adapted to OSH projects as it does not provide indications regarding openness or reproducibility. To address these limitations, the Open Technology Readiness Level (OTRL) was proposed by [15] to assess the maturity of open-source technologies. Unlike the traditional TRL scale composed of nine levels, the OTRL framework only defines six levels in order to "*fit the faster development cycle of OSH*"[11]. The different OTRL levels and their associated outcomes are summarized in table 2.1.

Level no.	Short level description	Main outcome
OTRL-1	Needs identification	Product idea, needs and initial specifications are defined
OTRL-2	Conception/definition of product architecture	Mature product concept has been formulated
OTRL-3	Design and modelling	Product model is developed
OTRL-4	Prototyping and testing	Full functional prototype is built and tested
OTRL-5	Manufacturing development	Fairly reliable processes identified and characterised
OTRL-6	Product qualification	Certificate marking conformity assessment or comparable

Table 2.1: Summary of Open Technology Readiness Levels (OTRL)

Measure of documentation maturity

Alongside the OTRL framework, the Open Documentation Readiness Level (ODRL) was also proposed by [15] to evaluate the maturity and completeness of the documentation associated with an OSH project. ODRL evaluates how well the project documentation enables transparency, reproducibility, and collaborative development. The different ODRL levels are summarized in table 2.2.

Level no.	Short level description	Main outcome
ODRL-1	Documentation process commenced	Published information under free open source licence
ODRL-2	Collaborative documentation in progress	Provision of documentation files and editable formats enabling asynchronous collaborative development
ODRL-3	Full documentation published	Complete documentation as intended by DIN SPEC 3105-1 requirements for technical documentation
ODRL-4	Full documentation for product qualification	Product qualification documents published enabling decentralised commercial distribution

Table 2.2: Summary of Open Documentation Readiness Levels (ODRL)

2.2 Medical device

2.2.1 Definition

As the precise definition of a MD varies depending on national regulatory frameworks, no universally accepted definition exists. However, definitions provided by major regulatory bodies, such as the European Medical Device Regulation (MDR)[16], provide a useful reference:

" A medical device means any instrument, apparatus, appliance, software, implant, reagent, material or other article intended by the manufacturer to be used, alone or in combination, for human beings for one or more of the following specific medical purposes:

- *diagnosis, prevention, monitoring, prediction, prognosis, treatment or alleviation of disease,*
- *diagnosis, monitoring, treatment, alleviation of, or compensation for, an injury or disability,*
- *investigation, replacement or modification of the anatomy or of a physiological or pathological process or state,)*
- *providing information by means of in vitro examination of specimens derived from the human body, including organ, blood and tissue donations,*

and which does not achieve its principal intended action by pharmacological, immunological or metabolic means, in or on the human body, but which may be assisted in its function by such means. "

2.2.2 Regulatory context

When a manufacturer wants to place a MD on the market, it must comply with strict regulations. Their purpose is to "*limit the risk of a product causing harm or not fulfilling its intended purpose or not complying with standards of quality*"[17]. According to WHO, a regulatory framework includes at least the following components[17] :

- a set of regulatory rules;
- a government-approved regulatory authority that has the role to enforce the rules;
- one or more "conformity assessment bodies", they are accredited by state institution and may issue market approval;
- a classification scheme ranking devices by level of potential risk, in European Union, devices are classed from *Class I* to *Class III*, with *Class I* devices are with the lowest risk and *Class III* with the highest[18];
- a quality assurance or management system, managed by the manufacturer;
- a system for evaluating the clinical safety and performance of a device;
- a system for granting market entrance;
- a surveillance system capable of detecting and investigating adverse events associated with actual use of the a device on the market.

Implementation of MDRs can vary depending on the country or world region considered. However, international standards organisations play a central role in harmonizing technical and safety requirements across markets. In practice, compliance with standards published by these organisations is widely used by manufacturers to demonstrate conformity with MDRs[19], [20]. The role of the 2 most important organisations of this type is summarized in the next section.

ISO role

The International Organisation for Standardisation (ISO) is a non-governmental organisation gathering national standards bodies from more than 160 countries[21]. In the medical-device field, ISO publishes internationally recognized standards related to quality management, risk management, biological evaluation, and clinical investigations [21]. Among the most important standards are ISO 13485, which defines requirements for quality management systems for medical-device manufacturers, and ISO 14971, which focuses on risk management throughout the device lifecycle [22]. Compliance with these standards is commonly used to demonstrate conformity with regulations such as the European MDR [19], [20].

IEC role

The International ElectroTechnical Commission (IEC), established in 1906, develops international standards related to electrical, electronic, and electrotechnical systems [23]. In the medical-device domain, IEC standards address the safety and performance of medical electrical equipment. The IEC 60601-1 standard and its collateral standards are particularly important, as they define "*general requirements for medical electrical equipment and devices covering standards for basic safety and essential performance*" [24]. These standards are widely adopted internationally and constitute a major technical reference for the design and certification of medical electronic systems.

2.3 Opportunities of open-source medical devices

Open-source approaches have the potential to reshape the medical-device industry [25]. During the pre-marketing phase, transparency and collaborative development can improve device design, safety, reliability, and security, as a larger number of contributors are able to review and contribute to the project [25]. Open-source development results are also publicly accessible, independently of the financial resources of the users or manufacturers [26]. In some cases, the overall development cost of MDs can also be reduced [25]. However, it remains unclear whether open-source approaches still lead to lower overall costs once certification and regulatory-compliance expenses are taken into account [27]. Open-source MDs may additionally improve the availability and maintainability of healthcare technologies. Since designs and manufacturing files are openly available, suitable replacement parts or equivalent devices can more easily be produced when commercial products become obsolete or unavailable [25]. This could also improve resilience during crises such as pandemics by allowing distributed and local manufacturing of essential MDs [28].

2.4 Challenges of open-source medical devices

Despite these opportunities, open-source MDs also face important challenges. One of the major limitations is the compatibility of current regulatory frameworks with open and collaborative development models. Existing regulations require certified quality-management systems and clearly identified regulatory responsibilities, which are difficult to implement in decentralised open-source communities [26]. In addition,

open-source medical-device development requires competencies not only in hardware and software engineering, but also in risk management, usability engineering, regulatory affairs, and post-market surveillance [26]. The absence of a traditional seller-buyer relationship can also complicate communication with users and make post-market monitoring and risk management more difficult [26]. Finally, legal questions regarding the responsibility and protection of contributors remain insufficiently addressed. According to [28], dedicated legal and regulatory frameworks would be necessary to both protect contributors and allow rapid but safe deployment of open-source MDs when needed.

Chapter 3

Literature review of smartphone repurposing for medical applications

Modern smartphones integrate a rich set of embedded sensors — cameras, microphones, accelerometers, gyroscopes, and wireless modules — associated with computing power and near-universal availability, making them an attractive platform for low-cost medical applications. This chapter reviews existing projects that incorporate a smartphone into a MD or diagnostic workflow. Articles were selected through keyword searches in Google Scholar using terms such as "smartphone repurposing MD" and application-specific terms for each use case, supplemented by references found in the retrieved articles. Selection was not based on a systematic protocol; the review is intended to illustrate the range of possible applications and typical performance levels rather than provide an exhaustive and systematic review. Each use case is introduced with a brief description of the underlying measurement principle, followed by a comparison of smartphone-based implementations against the clinical reference standard in terms of accuracy and practical feasibility.

3.1 Heart rate measurements

HR measurement is a critical indicator of a patient's condition. It consists of the number of heart beats done in a minute. HR can be too fast (above 100 bpm at rest), which is the symptom of tachycardia. It can also be too slow : below 60 bpm is considered as bradycardia. However, for a physically fit person, the heart might be healthy and pump enough blood under 60 bpm [29]. Heart rhythms problem can indicate underlying problem, such as atrial fibrillation, high blood pressure or coronary heart disease [30].

3.1.1 Heart rate measurement technique

Measuring the HR is therefore critical to prevent cardiovascular disease or follow patient state. ElectroCardioGram (ECG) based HR is recognized as the gold standard for HR measurements [31]. PhotoPlethysmoGraphy (PPG) is "*a simple and low-cost optical technique that can be used to detect blood volume changes in the microvascular bed of tissue*"[32]. This periodic change of blood volume in peripheral area such as finger or face, is directly correlated to the HR. The principle of PPG is to emit light and to use a photo-sensor

to record the reflected light from the skin. With the variation of blood volume, the amount of reflected light will change and so the HR can be determined. The accuracy of commercially available PPG sensors is at present similar to standard ECG [33].

As smartphone typically contains a flash and a camera, PPG can be done : the flash is the light emitting device and the camera the photo-sensor. The most common approach is to place the finger of the patient on the flash and camera of the smartphone. This is categorised as contact PPG. Non-contact PPG has also been tested, it simply consist of taking a video of the skin of the patient. In this case no flash is used, and the light source is ambient light.

Another non invasive HR measurement technique is SeismoCardioGraphy (SCG) : it records the micro mechanical vibration produced by the heartbeat on the chest which is used to determine HR[34]. The accelerometer and gyroscope sensors of smartphone can be used to measure the heartbeat of the patient.

HR measurements using Wi-Fi is also possible. *Pulse-fi* system has demonstrated HR measurements from the Channel State Information (CSI) of the Wi-Fi. "*CSI describes the propagation process of the wireless signal and therefore contains geometric information of the propagation space*"[35] and therefore enable retrieval of HR using Wi-Fi. Using the CSI of ESP32 microcontrollers, researcher have achieved a Mean Absolute Error (MAE) of 0.46 bpm [36]. This method could be adapted to smartphone by exploiting the wireless chip of the phone. However, one major obstacle will be accessing channel state information, as manufacturer might not give access to it for security or proprietary reasons.

3.1.2 Comparison of smartphone based HR measurement

Ref,year	Subjects	App name	Method	r	MAE ($\pm$ SD)
[37],2012	14	Custom app	Contact PPG	0.99	/
[38],2017	108	Instant HR	Contact PPG	0.83	4.52 (1.12)
		Heart Fitness	Contact PPG	0.96	1.96 (1.39)
		Whats My HR	Non-contact PPG	0.62	7.08 (1.39)
		Cardiio Version	Non-contact PPG	0.61	8.11 (1.38)
[39],2018	110	Instant HR	Contact PPG	0.98	0.8 (1)
		Cardiio : HR Monitor	Contact PPG	0.97	0.7 (0.9)
		Runtastic HR Monitor	Contact PPG	0.92	1.0 (1.3)
[40],2024	220	Custom app	SCG	/	0.03 (5.35)
[41],2026	57	Kubios	SCG	0.99	0.06 (0.19)

Table 3.1: Comparison of correlation and MAE for multiple paper comparing ECG derived HR and smartphone derived HR

Table 3.1 compares the different methods and application found in the literature and provides the correlation and MAE between the estimated HR and the one measured with a gold reference. It highlights that non-contact PPG methods performs worse than contact PPG applications : correlation between gold ECG HR measurements and non-contact PPG is low, and the MAE is higher for both non-contact PPG applications than others methods. One hypothesis is that this method is exposed to ambient light

variation, which will be seen as noise on the PPG signal, therefore making it more difficult to derive HR. It could be also caused by the fact that non-contact apps use the front camera[38], which are typically less performant than the main camera of smartphones. Contact PPG application performs all quite well : correlation is between 0.92 and 0.97. MAE is between 0.7 and 4.52. This error is comparable to the one made by a pulse oximeter, which is a dedicated PPG device for HR measurements. According to [38], the pulse oximeter has a MAE of 2.0 when compared to ECG based HR measurements.

Results present in table 3.1 are for healthy subjects with no heart rhythm issue. [39] tested their application with people afflicted by tachycardia or bradycardia. Their finding was that all 3 apps (*Instant HR*, *Cardio : HR Monitor*, *Runtastic HR Monitor*) were poorly correlated when patient has tachycardia ($r = \pm 0.25$). For cases of bradycardia, only *Runtastic HR Monitor* was poorly performing[39]. This highlights the need for more robust method for HR measurements in case of higher/slower HR than a normal one at rest.

One limitation is that only *iPhone5s* [39] and *iPhone 4* and *5* for [38] were used. Using others device or platform might influence the accuracy as hardware and software will differ. Characterising the sensitivity of the model to parameters like camera, post processing, flash properties is needed to know how much must the model be adapted.

SCG HR measurements seems promising, since it has the highest correlation of all methods and the lowest MAE. However, the standard deviation of the error in [40]'s custom app indicate a less reliable measurements. It should also be noted that for *Kubios* app, two of the three authors are working for the company of the same name. Those results might be valid but there is some conflict of interest. While this method seems to have potential, it will be sensitive to movement done by the patients, known as "movement artefact". *"Even minor body movements or changes in chest position can introduce disturbances that result in missed or irregularly detected beats requiring correction. Reliable HR metrics can be obtained when the user remains completely still."*[41]. The requirements needed for SCG are also to keep in mind. Both [40] and [41] had the strict requirements of sampling accelerometer and gyroscope at 100 Hz at a sufficient precision. On some older Android device, the recorded data from the accelerometer was too noisy to be exploitable [41]. [40] mentions that on the 220 subjects of the study, only 138 owns a smartphone meeting those requirements.

3.2 Blood oxygen saturation measurements

Blood oxygen saturation (SpO_2) is an indicator of how well the blood is oxygenated. Hemoglobin (Hb) is the protein responsible of transporting the oxygen present in red blood cell. Hemoglobin can be oxygenated (oxy-hemoglobin, HbO_2) or deoxygenated (deoxy-hemoglobin, RHb) [42]. SpO_2 is defined in equation 3.1 as the "percentage of the amount of oxy-hemoglobin to total hemoglobin".

$$SpO_2 = 100 \cdot \frac{C[HbO_2]}{C[HbO_2] + C[RHb]} \quad (3.1)$$

SpO_2 is a key indicator for the health of a patient. Low oxygen saturation is associ-

ated with increased all-cause mortality and pulmonary disease [43]. If the amount of oxygen in the blood is insufficient, individual cells can be damaged or organs might malfunctions [44]. To measure SpO_2 , 2 methods are available :

1. **Gas chromatography** consists of using a sample of arterial blood, extracted by a trained clinician [45] and analyse it with a gas chromatograph. It provides a direct measurement of gas concentration and therefore HbO_2 and Rhb . This method is the gold-standard for oxygen saturation level.
2. **Pulse oximetry** is a non invasive method, light is sent trough body tissue at different wavelength. SpO_2 can be recovered from the amount of light transmitted/reflected. While this method is non invasive, it is less precise than gas chromatography.

3.2.1 Pulse oximetry principle

Pulse oximetry relies on the *Beer-Lambert* law describing the attenuation of light based on material properties. It is described in the following equation, where I is the received light intensity, I_0 is the incident light intensity, A is the absorption, ϵ is the molar extinction coefficient, C is the concentration of the material and d is optical path length [46].

$$I = I_0 e^{-A} \quad \text{where } A = \epsilon(\lambda)Cd \quad (3.2)$$

Absorption coefficient can be rewrite when taking into account the tissue composition [42] :

$$A = d (\epsilon_{HbO_2}(\lambda)C[HbO_2] + \epsilon_{RHb}(\lambda)C[RHb] + \epsilon_{other}(\lambda)C[other]) \quad (3.3)$$

As HbO_2 and RHb have different absorption coefficient for multiple wavelength, as shown in figure 3.1 : RHb is more absorbed than HbO_2 at 660nm (red), and it is the opposite at 880nm (IR). Those wavelength are typically used for pulse oximetry application. The resulting signal, for both channel (IR and red) has an AC and a DC component, as illustrated in figure 3.2. The DC component is related to the different tissue component reflecting light and do not vary with time. The AC component is the light absorbed by the pulsatile arterial blood[42]. The rapid decrease of transmitted light correspond to the systole, as shown in figure 3.2. Systole is the moment "*when blood is ejected from the left ventricle into the vascular system, thereby increasing the peripheral arterial blood volume*" [47]. It is followed by an increase of the transmitted light, corresponding to the diastole where the blood leave the peripheral area, resulting in smaller peripheral arterial blood volume.

In practice, *R ratio*, defined in equation 3.4 is often used. If the amount of oxygenated hemoglobin is high, the IR light will be more absorbed. When fresh oxygenated blood arrives in the peripheral area, the AC drop related to systole will be bigger for the IR light. For the red light, it is the opposite, the red light is relatively less absorbed than IR light, resulting in a bigger AC drop for the red channel. As AC_{RED} is relatively bigger than AC_{IR} , the *R ratio* will therefore be small. It is the opposite in the case of poorly oxygenated blood : the *R* will be high. The division of both AC components AC_{RED} and

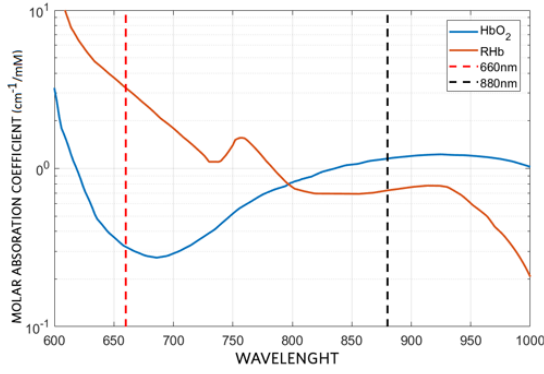


Figure 3.1: Molar absorption coefficients of HbO₂ and RHb in function of incident light wavelength, extracted from [42]

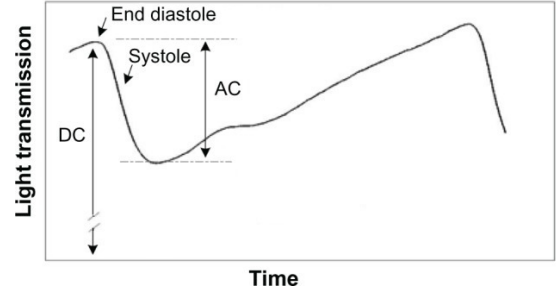


Figure 3.2: Typical PPG signal, extracted from [47]

AC_{IR} by the DC one (DC_{RED} and DC_{IR} respectively) can be seen as a normalisation, to reduce the impact of the skin tone and body part thickness.

$$R = \frac{AC_{RED}/DC_{RED}}{AC_{IR}/DC_{IR}} \quad (3.4)$$

The link between oxygenation level and the R ratio, if assuming that both light have the same optical path length, is given by (derivation can be found in [44])

$$SpO_2 = \frac{\epsilon_{RHb}(\lambda_{RED})(1 - R)}{\epsilon_{RHb}(\lambda_{IR}) - \epsilon_{HbO_2}(\lambda_{RED}) + [\epsilon_{HbO_2}(\lambda_{IR}) - \epsilon_{RHb}(\lambda_{IR})] R} \quad (3.5)$$

However, due to ideal assumptions and imperfections of the optical hardware, the theoretical relation is not used and calibrations is typically done for the relation between SpO_2 and R ratio. For example, [44] used the second equation of equation 3.6 that is closer to 3.5. Empirical data, combined with optimisation technique, determines the different coefficient of the relation used. Quadratic equation are also used, and [48] suggest that some commercial pulse oximeter used a simple linear relation (effectively setting $a = 0$).

$$SpO_2 = aR^2 + bR + c \quad \text{or} \quad SpO_2 = \frac{A - BR}{C - DR} \quad (3.6)$$

3.2.2 Comparison of smartphone implementation

Hospital-based oximeter are expensive and bulky : this limits their use to hospital or healthy clinic and make them not suitable to low resource settings. Smartphone implementation using the embedded camera and flash of the phone have been proposed by researchers to reduce the cost and allow individuals to track their oxygen level. This approach might be more suited to low resource environment. However, the absence of infrared light in the flash of the smartphone, and the inability of the camera to capture infrared light, is a major hurdle. Proposed solution consist of using the different color channel (Red, green and blue) in different manners, results are summarised in table 3.2. [49] has shown that the combined response curve allow to differentiate oxyhemoglobin and hemoglobin, when taking into account the followings factors :

- The spectral power distribution of the flash (in function of wavelength)
- The spectral sensitivity of each channel (R,G,B) of the smartphone camera
- Absorption spectra of oxyhemoglobin and hemoglobin

Ref,year	Subjects	Color channels	Methods	MAE ($\pm$ SD) [%]
[45],2015	20	R and G	Calibration-free methods	0.45 (1.95)
[50],2018	39	R,B and G	1D CNN model	2.02 (/)
[44],2017	6	R and NIR with optical filter	R ratio with non-linear model	2.5 (1.62)
[49],2021	5	R,B and G	Multi-linear model	0 (1.07)

Table 3.2: MAE of proposed method against medical graded pulse oximeter

All the proposed implementations do not need extra hardware, except [44] that uses optical filter and optical spatial divider, so that a part of the camera field of view is exposed to only red light, and the other to near infrared light. Other implementation use red, green and sometimes blue channel to have information at different wavelength. The quality of the PPG signal can be degraded with the possible finger movement from the patient by introducing higher frequency content compared to the PPG signal. To mitigate this, [45] used a digital filter, while [50] has a dedicated algorithm to remove motion artefact from the PPG signal. Overall, good results are obtained : the MAE for [50] and [49] is around 2% of error compared to a medical graded pulse oximeter.

However, for all implementation, the amount of subjects is not sufficient to make conclusion about the reliability of this method. Calibration has been done on small sample of people and might be over-fitted to the chosen population. However, [45] used a calibration free method proposed by [51] that doesn't require fitting any model to data. It relies on the shape of the PPG signal, by using the slope of the systole part of the PPG signal and considering it linear. However, [44] claims that is is *"an optimistic assumption and usually results in reduced performance in practice"* and therefore justify the use of a convolutional neural network. While all the proposed methods seems promising, a bigger scale validation should be done to assess the reliability of this method. The Food and Drug Administration, the American federal agency responsible of public health protection, enforce the use of standard that requires [42]:

- manufacturer to declare calibration range, used reference and methods of calibration,
- performance assessment with 200 data points, equally distributed between 70% and 100%, with test subjects from different ages, gender and skin tone.

The Root Mean Square Error (RMSE) of the system should be below 3% of error to be accepted[42]. Also, the impact of using different device has not been characterised. For method with calibration, it is not clear what would be the impact of not adapting the calibration to the device. Moreover, FDA require the performance assessment to be done on the complete system[42]. In this case, this means that for a proposed application to be approved by FDA, performance assessment should be done on each smartphone present on the market. Without sufficient resources and time, it is unclear how this would be possible.

3.3 Jaundice detection

Jaundice is defined as *"the yellow discolouration of the skin caused by excess bilirubin, a chemical by product of recycling old blood cell"*[52]. It is one of the conditions that requires attention as it affects 60% of newborns and 80% of early newborns[53]. The presence of bilirubin is normal even for healthy neonate, however too high concentration of bilirubin in the blood can be fatal or lead to irreversible brain damage. Treatment are available and consists of phototherapy, in more severe case blood transfusion is necessary [54]. To determine the risk that a newborn encounter, a *Bhutani nomogram* is used : it consist of low, intermediate and high risk zone of bilirubin level in function of the age of the baby, as depicted in figure 3.3.

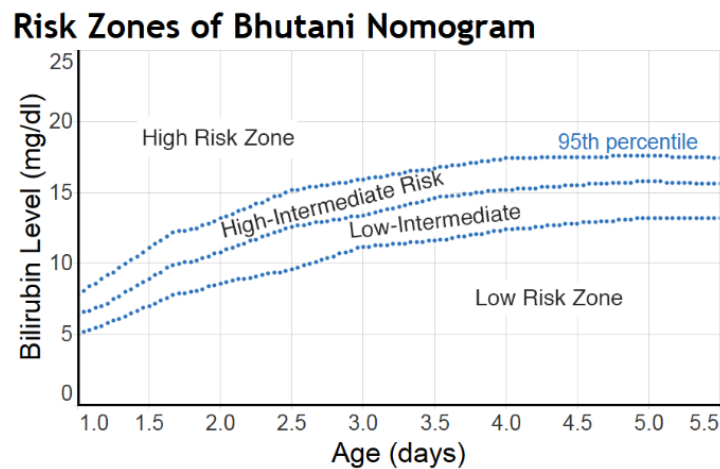


Figure 3.3: Bhutani nomogram used for risk-assessment of bilirubin level, extracted from [52]

To assess the risk, a measurements or at least estimation of bilirubin level is needed. Two methods are classically used :

1. **Total Serum Bilirubin (TSB)** is direct measure of bilirubin level from blood sample. This method is invasive but is the most accurate and is considered gold standard[52].
2. **TransCutaneous Bilirubinometer (TcB)** *"is a specialized meter for a non-invasive alternative that indirectly measures bilirubin levels"*[52]. It use the same principle as PPG : light is emitted at specific wavelength and bilirubin level is estimated with the measure of absorbed and reflected light.

Since TSB is invasive and can lead to undesirable consequences such as blood loss or infection at sampling site[55], TcB is used first. However, TcBs are known to be unreliable for bilirubin level above 14.5mg/dl [52]. For high measure obtained with TcB, a TSB measure must follow to correctly assess the risk.

Usually, bilirubin level hits its peak when the baby has already left the hospital [52]. As a TcB cost between 200 and 10 000\$ [56], visual assessment of jaundice presence is preferred. Parents and clinicians are able to identify the presence of jaundice but not its severity [52]. In the context of such low-resources environment, bilirubin level

evaluation based on smartphone's camera might be a good tool to replace TcB screening as it can be used at home with minimum material needed. The principle is to use the camera of the smartphone associated to machine learnings algorithms to predict a bilirubin level value.

Ref, year	Subjects	ROI	Calibration	r	LOA [mg/dL]
<i>Skin-based methods</i>					
[55], 2018	113	Forehead skin	Colored card	0.479	/
[53], 2024	546	Forehead, sternum & abdomen	Colored card	0.84	4.0
[52], 2014	100	Sternum	Colored card	0.84	5.35 ^a
[54], 2025	341	Extracted skin regions	Colored card	0.90	4.8
[57], 2017	530	Sternum skin	Colored card	0.91	3.6
<i>Sclera-based methods</i>					
[58], 2017	70	Sclera	/ (Box+flash)	0.89	5.41
[59], 2023	102	Sclera	/	0.97	12.55

^a Only the 80% interval (10th–90th percentile) was reported. The 95% LOA was estimated assuming normally distributed errors: $LOA_{95} = 1.96 s_d = 1.96 \cdot \frac{LOA_{90}}{1.282}$.

Table 3.3: Comparison of correlation r and LOA (95% confidence interval) for studies using different ROI, calibration methods and algorithms. Methods are grouped by ROI.

Table 3.3 shows the correlation and the Limit Of Agreement (LOA) between smartphone's estimation and TSB reference measurements. LOA is defined as the interval within which 95% of the differences between the predicted and reference bilirubin values are expected to fall [60]. Unlike correlation, which measures the strength of the linear relationship between two variables, LOA quantifies the actual agreement between 2 methods of measurement. High correlation between two different measure can still lead to poor agreement when the dynamic range of the dataset is large, as a wide spread of true values inflates the correlation coefficient regardless of prediction error. This is illustrated by [59], which achieves a high correlation of 0.97 on a dataset spanning 0.5 to 25 mg/dL, yet reports a LOA of ± 12.55 mg/dL, meaning that for an individual patient the prediction could be off by nearly 12 mg/dL. [55] obtained significantly weaker results with a correlation of only 0.479. This poor performance can be attributed to the use of a simple univariate linear regression on averaged color values, whereas other studies employing more complex machine learning algorithms consistently achieve higher correlations, highlighting that the relationship between skin color and bilirubin level is too complex to be captured by a linear model.

Comparison is also done when the bilirubin level prediction is done using image of the skin or the sclera, which is "*the white layer that covers the outside of the eye*" [61]. While it is difficult to identify if one method is more performant than the other, sclera based methods should be more race independent than skin based method, as "*typical sclera is race-agnostic*" [58]. However imaging the skin of a newborn is easier than the eyes as the baby could be sleeping or simply not willing to open them.

For skin based methods, all studies used a calibration methods using calibration cards containing different color and attempt to remove the effect of ambient light. While this method is quite effective, further studies should be done on the impact of color degradation of the calibration card. In the case of paper card, the impact of using a different printer should be characterised. For sclera based methods, [58] has shown

than using a box to mask ambient light and using the flash of the smartphone is more effective than using a similar method to color card with glasses containing multiple color. [59] wasn't using calibration, which might also explain the poor performance of the model.

Another limitation is that most studies rely on a single smartphone model. Although [59] attempts to address this by testing across image sensors of different resolutions, the evaluated devices are not clearly identified. A systematic evaluation across well-identified, diverse smartphone models would be necessary to ensure reliability for the general public. Furthermore, none of the reviewed studies characterise inter-operator variability, meaning that the impact of how different users capture images on prediction accuracy remains unknown.

3.4 Parkinson disease detection

Parkinson's Disease (PD) is a fast-growing neurodegenerative condition. It affects mainly the motor feature but also non-motor features like cognitive decline, depression and pain [62]. According to [63], diagnostic is done when bradykinesia is detected, which is "slowness of initiation of voluntary movements with progressive reduction in speed and amplitude of repetitive actions"[63] and one additional symptom such as muscular rigidity, resting tremor or postural instability.

Despite the improvement of modern medicine, there is currently no cure for PD, but medication and surgical operation may slow down the disease's progress or alleviate patients' pain [64]. Therefore, an early diagnostic is needed to mitigate the disease and improve quality of life of patient. However the clinical diagnosis of PD is not always valid : a study done in London has found that 15% patients diagnosed with PD do not meet the strict criteria for the disease. Moreover, approximately 20% of patients that actually meet those criteria have not been diagnosed with PD[65].

The most used criteria to assess PD severity is the **Unified Parkinson's Disease Rating Scale (UPDRS)**. It was developed in the 1980s and consist of a scale to 176 to 0 : 0 represents an healthy state and 176 total disability [64]. To compute the score, both physical observations and survey completed by the patient are done. UPDRS is constituted of four part [66] :

1. *Behavioural problems such as : intellectual decline, hallucinations and depression*
2. *Perceptions of their ability for daily task (dressing, walking and eating).*
3. *Motor evaluation : include rating for tremor, slowness (bradykinesia), rigidity and balance.*
4. *Motor complications : include rating for involuntary movements (dyskinesias), painful cramps (dystonia) and irregular medication responses (motor fluctuations).*

With the goal of improving the diagnosis of PD and make it more accessible, multiple works have been conducted to use smartphone for PD diagnostic. Smartphone sensors like microphone and accelerometer could be used to predict if a patient has PD or a UPDRS score.

3.4.1 Audio based PD diagnostic

One approach to try to diagnose or measure PD severity with a smartphone is to use the microphone. There is evidence that PD cause vocal impairment that can become worse with PD progression[67]. It might also be one of the earliest indicator of PD [68], [69]. In 2009, [64] claimed that they were the first to *"demonstrate rapid, remote replication of UPDRS assessment with clinically useful accuracy using only simple, self-administered, and noninvasive speech tests."*. This was done with a custom audio recording equipment. Their results had a MAE of 7.5 from the clinician's UPDRS estimate. As UPDRS ranges from 0 to 176, the relative error is 4.26%. [70] have shown that speech features recorded with a professional microphone and smartphone had a strong correlation, indicating that smartphone could be used to diagnose PD.

Binary classification metrics definition

To better understand the results of others article trying to do binary classification to dissociate PD patient from healthy control-group, definition of the used metrics are needed. When binary classification is done, in this case predicting if a patient has PD or not, 4 groups are obtained [71] :

1. *True positive (TP) : samples correctly classified as positive.*
2. *True negative (TN) : samples correctly classified as negative.*
3. *False positive (FP) : samples incorrectly classified as positive.*
4. *False negative (FN) : samples incorrectly classified as negative.*

Accuracy is the number proportion of correct prediction, it is therefore given by

$$accuracy = \frac{(TN + TP)}{(TN + TP + FP + FN)} \quad (3.7)$$

Precision indicates how reliable a classifier's positive predictions. Recall (also called sensitivity) indicates how well actual positives are detected. Specificity depicts how well actual negatives are classified. Those different metric's formula are defined by [72]

$$precision = \frac{TP}{TP + FP} \quad recall = \frac{TP}{TP + FN} \quad specificity = \frac{TN}{TN + FP} \quad (3.8)$$

Another metric used is the Area Under the Curve (AUC), defined as the area under the curve of pairs (recall, 1-specificity) across multiple classifier thresholds, as depicted in figure 3.4. Unlike single-threshold metrics, AUC evaluates classifier performance across multiple thresholds. A perfect model yields $AUC = 1$, while a random classifier yields $AUC = 0.5$ [73].

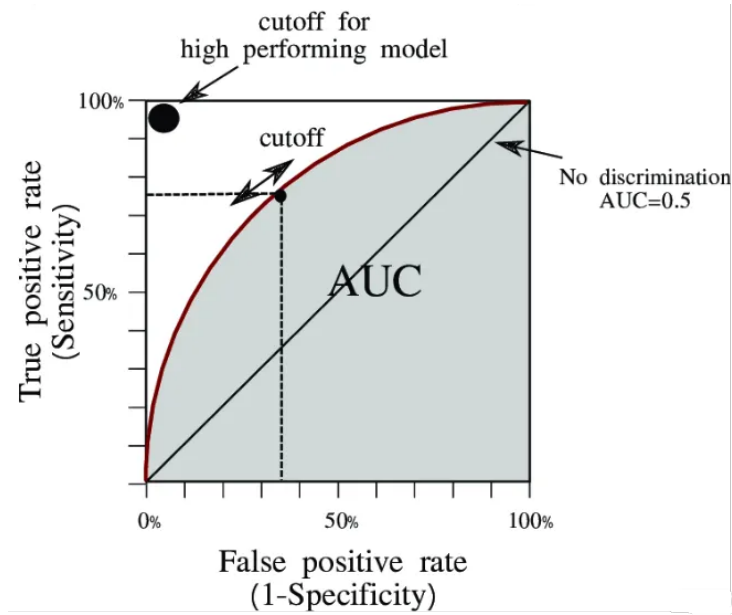


Figure 3.4: AUC definition, extracted from [74]

Research's result comparison

Table 3.4 shows the results from different studies using audio data from smartphone. All results are convincing and highlights the fact that a diagnostic based on audio data is possible. [75] and [76] have showed that good results are also obtained when using more consequent dataset (samples from more than 500 participants). Both sustained vowels and natural speech recording allow the models to reach good performance. However, for natural speech recording, [75] have shown that language aware model are needed in order to reach good classification performance.

Ref, year	Cohort PD/HC	Data type	Val. method	Results
[70], 2017	30/30	Sustained vowels	Leave-one-out CV	AUC of 0.85
[77], 2017	23/8	Natural speech	Test set of 50%	96% accuracy
[76], 2020	5826*	Sustained vowels	Test set of 20%	99% accuracy
[78], 2022	36/36	Sustained vowels	Leave-one-out CV	100% accuracy
[75], 2022	145/416	Natural speech	Test set (24/39)	AUC above 0.8

Table 3.4: Binary classification for PD patient and healthy patient. *PD/HC repartition not reported.

[78] have obtained better results with model trained on multiple vowels (/a/+ /o/+ /m/) than when using a single phoneme. One benefit of natural speech recording is that they don't require specific task to record a certain vowel or sound. It can be done during phone calls, as done in [75]'s work, or when speech is detected in the surrounding. This make it easier for the patient but raise privacy concerns.

One major limitation of all results present in table 3.4 is that the audio data comes from the same device for each dataset. If a model has been trained on data coming from a specific smartphone and is given as input data from another smartphone, performance

might be degraded. The recorded audio will differ because hardware like microphone and filter, and audio signal processing is different on each device. [79] highlighted this problem : their results showed that accuracy is decreased when using data coming from another smartphone as used for the training of the model. By using multi-condition training, which is "*the incorporation of data sourced from a diverse range of conditions into the training dataset*"[80], mean-accuracy was improved and standard deviation of accuracy was reduced, indicating that the system is more robust. The impact on performance of the smartphone position relative to the speaker was also demonstrated.

3.4.2 Accelerometer based PD diagnostic

Another approach to detect PD is to use the accelerometer to record the gait of the patient. Gait is defined as *the walking pattern in humans* [81]. Patients affected might suffer from parkinsonian gait, characterised by "*rigidity of joints, stooped posture, shuffling small steps, freezing episodes and short rapid bursts* " [81].

Table 3.5 shows results obtained by different paper using machine learning to identify this characteristic gait and predict if a patient is healthy or not. [82] was trying to predict step length and frequency, while making the study they discovered that one of the participant was developing PD and that the accelerometer data from the participant was separable from the healthy participant. [83] and [84] confirms the possibility to identify PD presence and, so do [85] and [86] on larger dataset using deep learning methods.

Ref,year	Cohort (PD/HC)	Model	Val. method	Results
[82],2016	1/16	SVM	No test set	94% accuracy
[83],2014	10/10	Features extraction + RF	10-fold CV	Sens. and spec. above 97.5%
[84],2017	28/22	Features extraction + RF	10% test-set	87.03% accuracy
[85],2021	1274/3525	Data augmentation and DL	41% test-set	0.87 AUC
[86],2024	119/467	Data augmentation and DL	5-fold CV	0.883 AUC

Table 3.5: Binary classification for PD patient and healthy patient using smartphone's accelerometer

[86] mentions that multi-modal model could be used to make a more robust diagnostic. Multi-modal means using data from different sources. In 2025, [87] proposed a model combining voice, hand movements (recorded through tapping on smartphone screen) and gait data. It reported only a slightly better AUC on the overall dataset : 0.82 versus 0.8 for the single-modality model. However, for early-stage PD patient, it demonstrated better accuracy and is inherently more robust to confounding factors, as a single modality such as voice is not PD-specific and may be affected by unrelated conditions [86].

3.5 Hear loss detection

Hearing loss is one of the most common sensory impairments worldwide and can significantly affect communication, social interaction and quality of life[88]. Untreated hearing impairment is also associated with cognitive decline[89], [90]. The

estimated global cost of unaddressed hearing impairment is greater than 980 billion annually[88]. Pure-Tone Audiometry (PTA) is considered the gold standard for hearing evaluation[91], but it requires dedicated equipment, trained personnel and controlled acoustic environments[92]. These constraints limit access to hearing screening, particularly in low-resource settings and rural areas[88], [92]. Smartphone-based hearing tests have therefore been proposed as a low-cost and portable alternative allowing better accessibility to hearing healthcare[91], [92].

Most smartphone-based hearing screening methods rely either on PTA or speech-in-noise recognition tests[91]. PTA consists of tones of different frequencies and volumes played through headphones or earphones, and the patient indicates whether the sound is audible or not for each frequency. Speech-recognition method evaluates the ability of the user to distinguish spoken digits or words in noisy environments[92]. Compared to pure-tone tests, speech-based methods are generally less dependent on precise calibration and may better represent real-world hearing difficulties[92].

One challenge for smartphone-based audiometry is the variability of the audio response between smartphone models. Differences in speakers, headphones, audio processing and calibration may affect the accuracy of the measured hearing threshold. [93] investigated this issue on more than 8500 Android devices representing over 2000 smartphone model. Their results showed significant differences between smartphone models, confirming the need for model-specific calibration procedures. However, the variability between smartphones of the same model remained limited, suggesting that *"it is feasible to conduct hearing screening on devices calibrated with the predefined reference sound level"* (predefined for each specific smartphone model)[93].

A systematic review by [92] and a meta-analysis conducted by [91] evaluated the performance of smartphone-based hearing screening applications by analysing respectively 17 and 25 studies, the latter including a total of 4470 patients. Both studies reported generally good diagnostic performance, with sensitivity and specificity values ranging between 75% and 100%. In particular, [91] reported a pooled sensitivity of 89% and specificity of 93% for smartphone-based PTA, while speech-recognition-based tests achieved a sensitivity of 91% and specificity of 88% to detect hearing impairments. Applications such as *uHear*, *Digits-in-Noise*, *HearTest* and *HearScreen* were identified as particularly promising solutions for large-scale and low-cost screening. Both reviews also highlighted the strong influence of testing conditions on diagnostic performance. Headphone-based systems generally performed better than earphones, while tests conducted in noisy environments showed reduced specificity compared to acoustically controlled conditions. Elderly and paediatric populations also tended to achieve lower performance. The reduced performance in elderly populations may not only be related to the accuracy of the application, but also to usability challenges associated with smartphone-based testing. According to [94], older adults with mild cognitive impairment reported only moderate usability. *"Usability was strongly shaped by factors, such as ease of use, clear instructions, available support, age relevance, and trust."*[94]. These findings suggest that usability and accessibility considerations are as important as diagnostic performance for large-scale deployment of smartphone-based hearing screening applications.

Overall, smartphone-based hearing screening appears promising for improving accessibility to hearing healthcare, particularly in low-resource settings or for self-assessment screening. Current studies report generally good diagnostic performance. However, important challenges remain regarding environmental noise, standardisation of testing conditions, and usability, especially for elderly and young populations.

Summary

The use cases reviewed in this chapter demonstrate that smartphones can achieve clinically relevant performance across a wide range of diagnostic applications, relying on sensors already embedded in the smartphone with sometimes additional hardware like earphone or calibration card required. Contact PPG-based HR monitoring reaches accuracies comparable to dedicated pulse oximeters for healthy subjects, SpO₂ estimation shows promising but preliminary results, and applications targeting jaundice, Parkinson's disease and hearing loss all report encouraging diagnostic performance. However, a set of limitations is common to all reviewed works: validation is typically conducted on small populations using a single smartphone model, making generalisation uncertain. Device variability, lack of standardised validation protocols, and the complexity of obtaining regulatory approval remain the main obstacles to an eventual clinical deployment.

Chapter 4

Multi-parameter monitor

The previous chapter has shown that smartphones alone can address a variety of diagnostic use cases. However, this thesis investigates a broader question: whether a smartphone can be integrated to an existing open-source medical platform to extend its capabilities. In that scope, and in the continuity of Louis Breels' master thesis on open-source multi-parameter monitoring (and more generally about MD and open-source), multi-parameter monitor was chosen as the target application, as they represent a clinically useful and technically challenging use case. This chapter introduces its clinical definition, the different monitored physiological parameters, its general requirements, and concludes by motivating the suitability of multi-parameter monitors for smartphone integration.

4.1 Definition

Multi-parameter monitors are essential MDs used for the continuous observation of a patient's vital physiological functions in different healthcare units such as intensive care, emergency care, general surgery, medical imaging or specialized diagnostic/treatment[95]. According to the Medevis database[95], a general-purpose multi-parameter monitor is defined as *"an electrically-powered device/assembly designed for the continuous assessment of several vital physiologic parameters (e.g., ECG, blood pressure, HR, temperature, cardiac output, apnoea, and respiratory/anaesthetic gas concentrations) of one patient in intensive or general healthcare settings. It typically set-up by the patient bedside and often includes programmable alarms, portable radio transmitters, receivers, and antennas (telemetry systems) to allow monitoring during patient ambulation or transportation; it is not dedicated to neonatal use and is not intended to be worn on the body."*

The Medevis database is *"an open access WHO electronic database of Medical Devices"*[96]. Its goal is to *"establish and update an evidence web-based health technologies database to serve as a clearing house which will provide guidance on appropriate MDs according to the levels of care, setting and intended health intervention, which can be tailored to the specific needs of country or region"*[96]. For each MD, general requirements and standards that should be fulfilled by the manufacturer are provided.

4.2 Physiological parameters

The current section summarizes the physiological parameters typically monitored by multi-parameter monitor and briefly introduces their clinical relevance and acquisition principles. This overview is based on the previous work conducted in Louis Breels' master thesis[27], related to the open-source implementation of a multi-parameter monitor, and is included here to provide the necessary context for the present work.

Electrocardiogram

ECG records the electrical activity of the heart [97] and is widely used for cardiac monitoring and arrhythmia detection [98]. ECG signals are obtained by measuring electrical potential differences between electrodes placed on the body surface[99]. Diagnostic systems commonly use 12 leads, requiring 10 electrodes,[100] whereas portable monitoring systems generally use 3 or 5 lead configurations sufficient for HR monitoring[101]. A typical ECG waveform is composed of several characteristic waves, illustrated in figure 4.1. "The P wave, QRS complex, and T wave respectively correspond to atrial depolarization, ventricular depolarization, and ventricular repolarization." [102]. In practice, the interval between multiple R peaks of the QRS complex can be used for HR estimation due to its high amplitude and ease of detection[103].

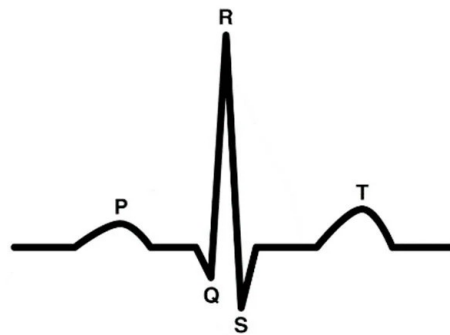


Figure 4.1: Normal ECG waveform, extracted from [102]

Photoplethysmography

As defined in section 3.1.1, PPG is a measurement technique that allows recording a signal reflecting blood volume variations in peripheral tissues. This signal will be called PPG in the rest of this work. A typical PPG waveform contains the following characteristics[104], illustrated in figure 4.2 :

- Onset denotes the beginning of the pulse in the systolic phase, where blood with oxygen flows at the measuring site.
- The systolic peak is the maximum peak following the onset during systolic ejection. The rising edge from onset to systolic represents increasing blood volume in peripheral tissue.
- The diastolic notch and diastolic peaks are wave reflections of the periphery, influenced by arterial stiffness or vascular resistance and compliance. They tend to disappear when people start aging.

- The systolic phase corresponds to the rapid increase in blood volume following ventricular contraction.
- The diastolic phase corresponds to the progressive decrease of blood volume before the next cardiac cycle.

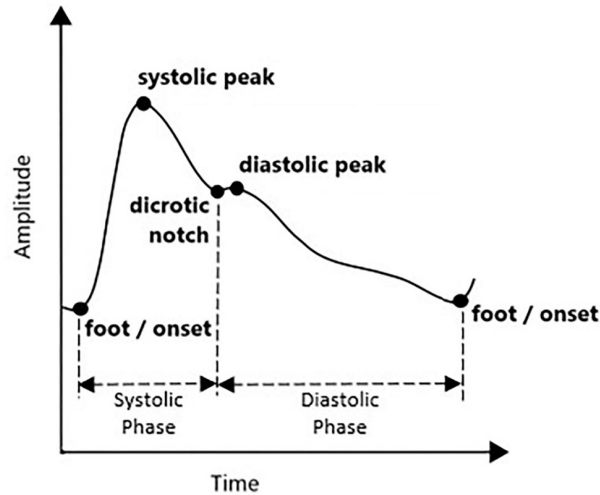


Figure 4.2: PPG waveform characteristics, extracted from [104]

Respiration rate

Respiration Rate (RR) corresponds to the number of breaths per minute and is an important indicator of respiratory and physiological condition [105]. Abnormal RR may indicate respiratory distress, hypoxia, or cardiac complications[106]. Normal RR is between 12 to 20 breathings per minute [105]. RR can be measured using several techniques, including airflow sensors or sound, thoracic impedance measurements, or extraction from PPG signal[99]. In monitoring systems, thorax impedance and PPG-based estimation methods are commonly preferred because they are non-invasive and easy to integrate[99]. The signal extracted from PPG or thorax impedance is usually displayed on the monitor screen. From now, this signal will be called *respiration signal*. By extracting the peak of the respiration signal, RR can be determined.

Heart rate

HR represents the number of heartbeats per minute [107] and is one of the most commonly monitored vital signs. Variations in HR may indicate cardiovascular abnormalities or changes in physiological state[101]. HR can be derived from ECG signals or from PPG signals by analysing blood volume pulse variations[108]. Both techniques are widely used in continuous monitoring systems.

SpO₂

SpO₂ estimates the proportion of oxygenated hemoglobin in arterial blood, as defined in section 3.1. It is used to assess blood oxygenation. According to [109] normal SpO₂ value range from 95% to 100%, while a value below 95% indicate insufficient

blood oxygenation, defined as hypoxemia. SpO₂ value under 90% correspond to the limit of respiratory failure. SpO₂ is commonly measured non-invasively using pulse oximetry, which principle has been detailed in section 3.2.1. Usually, at least one of the the PPG channels (red or infrared) is displayed on the monitor screen.

Temperature[110]

Body temperature reflects the thermal state of the organism and is regulated by human thermoregulatory mechanisms. Abnormal temperatures may indicate hypothermia, fever, infection, or heat stress. Fever is diagnosed when surface body temperature, measured at the level of the armpit is above 37.2°C. In clinical monitoring, core body temperature provides the most accurate estimation but requires invasive measurements. Multi-parameter monitoring systems commonly rely on non-invasive surface temperature measurements using contact temperature sensors placed on the skin. Despite being less accurate and generally smaller than core measurements, surface temperature remains useful for continuous patient monitoring and fever detection. However core body temperature is preferred in the case of operating room settings, where accuracy and reliability is critical.

Blood pressure

Blood pressure (BP) represents the force exerted by circulating blood on arterial walls and is generally expressed through systolic and diastolic pressure values [23], [111]. Systolic pressure corresponds to the maximum arterial pressure reached during heart contraction and blood ejection from the left ventricle, whereas diastolic pressure corresponds to the minimum arterial pressure measured while the heart relaxes between beats [23], [111]. BP monitoring is essential for detecting hypertension, hypotension, and cardiovascular disorders [112]. Conventional measurements are performed using cuff-based oscillometric or auscultatory methods [23], [112]. Recent wearable approaches investigate cuffless estimation methods based on PPG signals, although these methods are still under validation for clinical use [23].

CO₂

Carbon dioxide (CO₂) monitoring evaluates the concentration of carbon dioxide in exhaled respiratory gases and is mainly used to assess ventilation efficiency and respiratory status [113]. The most commonly monitored parameter is end-tidal carbon dioxide (EtCO₂), corresponding to the maximum CO₂ concentration measured at the end of expiration, which provides an indirect estimation of arterial CO₂ partial pressure [114]. Continuous CO₂ monitoring is widely used in intensive care, emergency medicine, and anaesthesia to detect hypoventilation, apnea, airway obstruction, or ventilatory failure [114]. CO₂ is generally measured non-invasively using capnography, a technique based on infrared spectrometry, since carbon dioxide absorbs infrared light at specific wavelengths [115]. Multi-parameter monitors usually implement either mainstream capnography, where the sensor is directly placed in the breathing circuit, or sidestream capnography, where a gas sample is continuously aspirated toward an external sensor module [114].

4.3 General requirements

The *Medevis* database defines three categories of multi-parameter monitors: basic, intermediate, and advanced monitors. Table 4.1 summarizes the main general technical requirements associated with each category. Basic monitors only monitor non-invasive blood pressure (NIBP) and SpO₂ monitoring, whereas intermediate monitors additionally include ECG, RR, HR, and temperature monitoring capabilities. Advanced monitors further integrate CO₂ monitoring functionalities. In terms of display capabilities, basic monitors are generally limited to numerical value visualisation, while intermediate and advanced monitors must also provide real-time waveform display. More detailed technical requirements are provided in Appendix A.9. *Medevis* specifications also reference the IEC and ISO standards that manufacturers should comply with for each monitor category.

Requirement	Basic	Intermediate	Advanced
Displayed Parameters	NIBP, SpO ₂ ; optional HR and RR	ECG, RR, HR, NIBP, temperature, SpO ₂	ECG, RR, HR, invasive and non-invasive BP, temperature, SpO ₂ , CO ₂
ECG Capability	Not required	3 to 12 leads ECG	
Temperature Monitoring	Not included	30–40 °C, minimum gradation of 0.1 °C	
Respiration Monitoring	Optional	0–100 bpm, minimum gradation 1bpm	
CO ₂ Monitoring	Not included		Mainstream or sidestream CO ₂
Display Type	Display with numeric values	Display with real-time waveform visualisation and numeric values	Display with real-time waveform visualisation and numeric values, minimum 10-inch screen
Memory / Trend Storage	Preferable	Included	
Common Requirements			
HR Range	30–250 bpm, minimum gradation 1 bpm, accuracy better than ±5 bpm		
SpO ₂ Range	70–99%, minimum gradation 1%, accuracy better than ±3%		
Blood Pressure Range	30–270 mmHg, minimum gradation 1mmHg		
Alarm Functions	Audio-visual alarms for all monitored parameters and system status (power failure, probe connection state, etc.)		
Power Supply	Main power with internal rechargeable battery backup with at least 1h of autonomy in case of power failure.		
Design	Shock/vibration resistant, IPX1 or better, with protection against defibrillator discharge and electrosurgical unit (ESU) interference		

Table 4.1: Summary of general technical requirements for basic, intermediate, and advanced multi-parameter patient monitors. Extracted from *Medevis* database.

4.4 Suitability for smartphone integration

Multi-parameter monitors are particularly well suited for smartphone integration and repurposing approaches because they already aggregate several physiological sensing modalities within a compact and portable system. As discussed in Chapter 3, smartphones can achieve clinically relevant performance for parameters such as HR or SpO₂ using their embedded sensors. However, a recurring limitation across virtually all reviewed works is the lack of robustness and reproducibility when changing smartphone models. Differences in camera sensors, flash spectral characteristics, microphones, accelerometers, embedded signal processing or software pipelines may significantly affect the acquired signals and degrade measurement reliability. This was illustrated for instance by the device-dependency of PPG-based pulse oximetry, where calibration curves developed for one smartphone offer no guarantee of validity on another.

In contrast, a multi-parameter monitor implementation relying on dedicated medical-grade sensing hardware would keep signal acquisition independent from smartphone-specific hardware variations. The smartphone can then be repurposed for communication, visualisation, power supply and storage, while physiological acquisition is performed by a controlled and reproducible external platform.

4.5 Previous prototype at UCLouvain

In his UCLouvain master thesis, Louis Breels [27] identified two fully open-source multi-parameter monitoring projects, namely *HealthyPi 4* and *Pulse Express*, for which both hardware and software resources are publicly accessible. Among these projects, *HealthyPi 4* was selected because its functionalities and architecture were considered closer to the requirements of an intermediate monitor. As part of his work, the *HealthyPi 4* platform was reproduced and experimentally evaluated. However, the obtained results were not satisfactory, as the prototype was only capable of monitoring body temperature. Since then, a new version of the platform, *HealthyPi 5* (HP5), has been released. The present work therefore focuses on the evaluation and adaptation of HP5, with particular attention given to the integration of smartphone to the system using multiple approaches.

Chapter 5

HealthyPi 5 open-source project modification

5.1 Description of HealthyPi 5 open source monitor

HealthyPi 5 (HP5) is an open-source multi-parameter monitor developed by *ProtoCentral*, an India-based company “strongly committed to the development of open source hardware”[9]. Beyond the conception of projects like the HealthyPi, *ProtoCentral* serve as a platform for the OSH community, by offering access to development boards, sensors, and other electronic components, primarily focused on Arduino and Raspberry Pi platforms. To fund the HP5 development, a crowd supply was launched on the *Crowd Supply* website in June 2023, raising 46 800\$. As the project is completely open-source, all the components of the system can be found on Github :

- **Hardware** : Eagle design files (.brd, .sch) containing the schematic and PCB design, released under the *CERN-OHL-P v2* license, are available on : https://github.com/Protocentral/protocentral_healthy_pi_5/tree/main.
- **Software** : The software containing the firmware running on the ESP32 is available on : https://github.com/Protocentral/protocentral_healthy_pi_5_zephyr. Firmware running on the ESP32 can be found at https://github.com/Protocentral/healthy_pi_5_esp32c3_zephyr/tree/main. Finally, the code base for the *OpenView 2*, the cross-platform application used for data visualisation and logging, can be found at https://github.com/Protocentral/protocentral_openview2. All software is released under the MIT license.
- **Documentation** : Most of the useful documentation is on the *Protocentral* website <https://healthypi.protocentral.com/>. It explains how to setup the device, the different mode and how build the firmware and flash the board. Documentation is also released in the before mentioned repository under the *Creative Commons Share-alike 4.0 International*, but it actually doesn't include any content present on the website, only one image of the physical board.

5.1.1 Features

Protocentral claims that the HP5 platform is able to monitor HR, RR, SpO₂ and temperature evolution in real time. It also displays the trace of ECG, PPG and respiration

signal in real time. It is therefore closer to the intermediate monitor defined in section 4.3, however, it lacks the monitoring of non invasive blood pressure, which is required for the intermediate type of monitor.

The system can be operated in different modes, table 5.1 summarizes the available operating modes and their intended uses, as described in the official HP5 documentation. By default, the system operates in USB streaming mode, allowing physiological data to be transmitted to a computer through a USB cable. However, when a Bluetooth Low Energy (BLE) connection is detected, the device automatically switches to BLE streaming mode and interrupts USB data transmission. In both case, the application running on the user device is *OpenView 2*, which is the application provided by *Protocentral* to display analog signals and physiological parameters acquired by the HealthyPi board. The application main features are visualisation of the data in real-time and storage of the data on the device.

Operation Mode	Communication / Interface	Typical Usage
USB Streaming	Desktop application via USB cable	Bench testing, signal validation, and CSV data acquisition on a computer
BLE Streaming	Mobile application via BLE	Mobile or portable monitoring without wired connection
Data Logging	On-board microSD card	Long-term unattended recordings and off-line data storage
Display Mode	On-board TFT screen	Standalone operation without external smartphone or computer

Table 5.1: Operating modes of HP5

Two characteristics of the HP5 platform can be discussed with respect to the general requirements previously identified for multi-parameter monitors. First, although *Medevis* specifications require the implementation of audio-visual alarms for monitored parameters and system status, HP5 does not currently provide an alarm management feature. No visual or audible alert is generated in the event of abnormal physiological values, patient condition degradation, sensor disconnection, or system malfunction. Consequently, the platform does not presently satisfy the alarm-related requirements expected for monitoring systems. Second, in accordance with the common power supply requirements identified in table 4.1, the device can operate either through USB power or using its integrated rechargeable battery, allowing operation during temporary external power interruptions.

5.1.2 HealthyPi basic kit content

The HP5 can be purchased in two versions: the complete kit and the basic kit. The complete kit contains the same core components as the basic version, but additionally includes a 3.5-inch display, a 3D-printed enclosure, and a 3.7 V 2000 mAh battery. It also includes a carrier board that can be used to extend the system and provide an isolated power supply[116]. For this work, only the components included in the basic kit were required.

The basic kit was purchased on *Mouser* for 257.14 €. The components are delivered in a custom and well-protected package featuring the *Protocentral* logo. The basic kit, illustrated in figure 5.1, is constituted of the following content :

- HealthyPi main board: the different sensors are connected to this board.
- 3-electrodes ECG cable: this cable connects to the main board through a 3-connector jack interface.
- Finger-clip SpO₂ probe: this finger clip integrates two LEDs and one photodiode, and is used for PPG signal acquisition. It is connected to the board using a DB9 connector. This PPG sensor is in a transmissive configuration : the photodiode and LEDs are on each side of the sensors, therefore, only transmitted light will hit the photodiode.
- Temperature sensor: this cable includes the temperature sensor and connects to the board through a Qwiic connector.
- 30 single-use ECG electrode patches: these patches are used to provide good electrical contact between the electrodes and the skin. The electrodes include a conductive gel intended to improve conductivity.

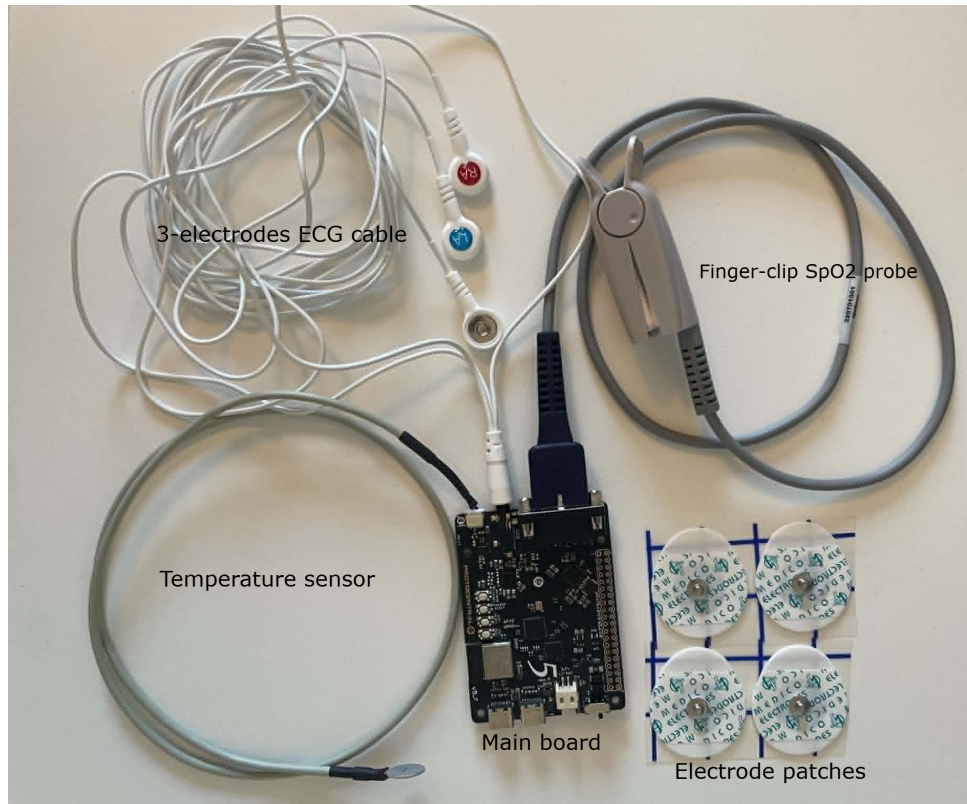


Figure 5.1: Components present in the HP5 basic kit

5.1.3 System hardware overview

Figure 5.2 provides an overview of the hardware present on the HP5 board. The main control unit is a RP2040, which is the first microcontroller from the *Raspberry Pi* foundation. It features a Dual ARM Cortex-M0+ running at 133MHz with 264kB on-chip RAM. This microcontroller has the particularity of not having built-in non volatile memory, therefore, in order to store the firmware, an external flash is used and connected through a Quad Serial Peripheral Interface (QSPI) interface. The RP2040 does not feature wireless capability, this is the role of the ESP32-C3 wireless controller : it features Bluetooth and Wi-Fi capabilities. Data from the RP2040 are transferred through a UART interface between the 2 MCU. The ESP32-C3 handles the wireless data transmission to the host. Both controller are connected to an USB connector for power supply, communication, and for the flashing procedure.

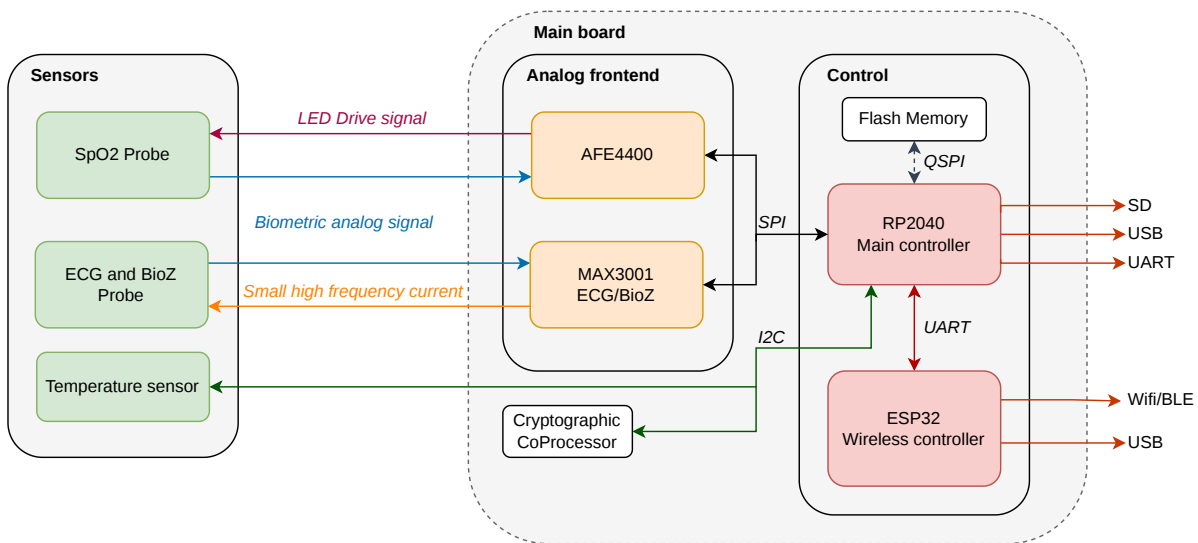


Figure 5.2: Schematic overview HP5 hardware

The main microcontroller is connected to two analog frontend Integrated Circuits (ICs) through an Serial Peripheral Interface (SPI) bus. Their purpose is to condition the signals by amplifying and filtering them, and to convert them into digital values through sampling and quantization. The system contains two Analog Front-Ends (AFEs) :

- **AFE4400**, used for the acquisition of the PPG, which is done by :
 - handling the red and infrared LEDs currents intensity and timing,
 - amplifying, filtering and sample the analog photodiode current induced by LEDs.
- **MAX30001** handles the ECG and respiration signal acquisition by :
 - amplifying, filtering and sampling the potential difference induce by heart's electrical activity corresponding to the ECG signal,
 - measuring the thorax impedance, corresponding to the respiration signal, by injecting a small high frequency current between 2 electrodes.

More details about the operations of both AFEs will be discussed in section 5.3.2 and 5.3.3. The HP5 also features an I2C bus connected to two peripherals:

- The external temperature probe included in the kit communicates through this interface using a **MAX30205** temperature sensor. The probe is another product designed by *Protoncentral* that can be purchased separately and is also open-source.
- An **ATECC608B** cryptographic coprocessor is also connected to the bus. This chip provides hardware-based cryptographic functions such as secure key storage, encryption, and device authentication. In the current HealthyPi firmware implementation, this component is not used. However, it could be useful in future developments to implement secure communication, authenticate connected devices, or protect sensitive patient data during wireless transmission.

Finally, the board features a Qwiic connector, which is a standardised four-wire connector system developed by *SparkFun* for I2C communication. It includes the I2C lines *SCL* and *SDA*, as well as *GND* and *VDD* lines used to power external peripherals[117]. The Qwiic ecosystem allows rapid integration of additional sensors and peripherals without soldering, thereby simplifying hardware expansion and prototyping. Through this connector, the HealthyPi platform can easily be extended with additional I2C slave devices.

5.1.4 Software overview

Firmware

Firmware running on the RP2040 and ESP32 are built with the real-time operating system *Zephyr*. *Zephyr* RTOS has a small footprint kernel intended for resource-constrained systems such as embedded systems. It supports multiple processor architecture and provide all necessary tools to build software for an embedded system. It includes multi-thread management, inter-thread synchronisation and communication mechanism. It also provides a memory allocation service, file system management and a well defined device model for driver configuration. One major benefit of using *Zephyr* is that it also offers support for *Bluetooth* natively.

As no documentation describing the firmware architecture existed, or at least describing the role of each thread, it was needed to reverse-engineer the code before modifying it. Figure 5.3 provides an overview of the architecture of the RP2040 firmware, each thread has the following purpose :

- **Hardware thread:** This thread reads the temperature sensor and the battery gauge level every second and publishes the corresponding values to dedicated Zbus channels. Zbus is a communication mechanism provided by *Zephyr* RTOS that enables data exchange between software modules through publish/subscribe communication channels[118]. In this case, each Zbus channel has a callback that sends the numerical value through Bluetooth.
- **Sampling thread:** Every 7 ms, a timer triggers the acquisition of one PPG sample from the *AFE4400* and one ECG and thorax impedance sample from the *MAX30001*. The acquired samples are stored in a dedicated data structure and pushed into the queue *q_hpi_data_sample*.

- **Data thread:** This thread consumes the items stored in the queue `q_hpi_data_sample`. It executes the PPG-based SpO₂/HR algorithm as well as the RR algorithm. The SpO₂/HR algorithm is executed every second, while the RR algorithm is executed every fifth sample. The derived physiological metrics are then published to dedicated Zbus channels. In addition, a USB packet is transmitted each time a sample is consumed containing ECG, PPG and respiration analog value and numerical value of HR, RR, SpO₂ and temperature. If Bluetooth is enabled, a Bluetooth notification is sent every 8 samples containing only the ECG, PPG and respiration analog value. A Bluetooth notification consists in transmitting a short message to the client without acknowledgement from the client[119]. The BLE notification concerning PPG is in red since it was incorrectly implemented, more details are available in section 5.3.1.
- **CMD thread:** This thread handles commands received from the host while a Bluetooth connection is active. It is mainly used to control the on-board SD-card data logging functionality. For example, the smartphone application can send a Bluetooth command to trigger the start of a data logging operation.

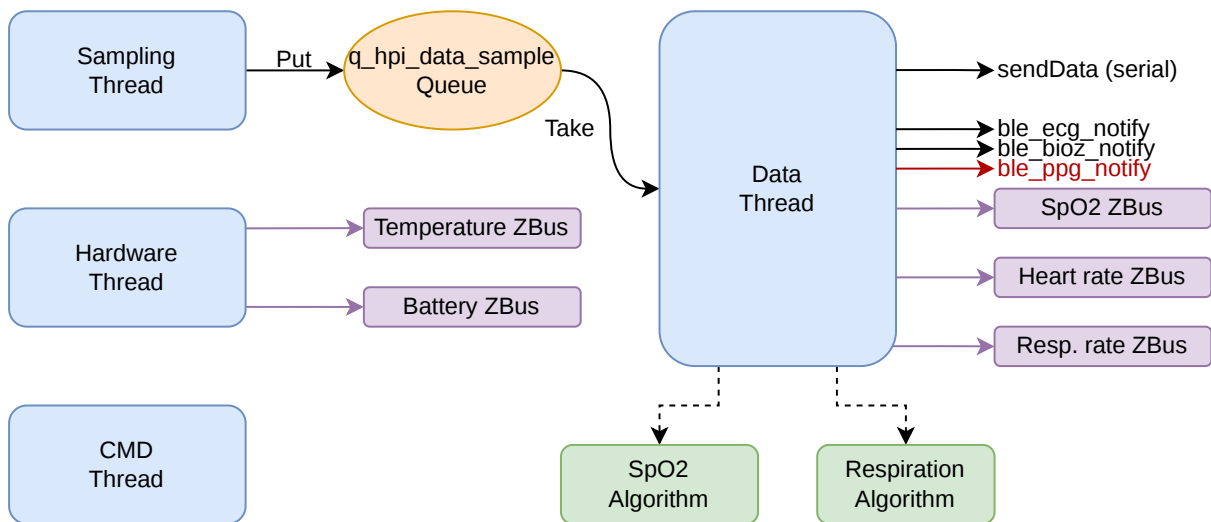


Figure 5.3: RP2040 firmware architecture

The ESP32 module runs dedicated firmware responsible for low-level Bluetooth controller functionalities. Zephyr provides support for the *HCI UART* protocol, that "exposes Bluetooth controller support over UART to another CPU using the H:4 HCI (Host controller interface) transport protocol"[120]. In this configuration, the Bluetooth host stack is executed by *Zephyr* on the main microcontroller, the RP2040 in this case, while the ESP32 mainly handles radio communication and Bluetooth packet transmission.

Application

The application *OpenView 2*, designed to work with HealthyPi, is developed using the Flutter framework. Flutter is a cross-platform application development framework developed by Google that uses the Dart programming language and allows the deployment of applications on multiple platforms from a single code base[121]. In the current implementation, the desktop and smartphone versions of the application are completely

separated: the desktop application only supports USB communication, whereas the smartphone application exclusively relies on BLE communication.

5.2 Preliminary functional evaluation of the system

5.2.1 Voltage regulation check

Before testing the system, a first validation of the power supply circuitry and voltage regulation is needed in order to test the system on a human body safely. The HealthyPi USB port was connected to a regular desktop USB port to supply the board. Even if the board is faulty and has a short circuit, the computer shouldn't be damaged as the USB standard state that USB port is *"required to withstand a continuous short circuit of data and/or power supply line, or the cable shield at the connector, for a minimum of 24 hours without degradation"*[122]. A multimeter was used to probe the different test-points : V_{BUS} , V_{DD} , 1.8V, 3.3V and CLK, as well as the voltage V_{USB} provided by the computer, directly measured on the USB port of the board. Results, presented in table 5.2, suggest that power supply circuitry and voltage regulation is functional as all voltage are close to the intended one. Output voltage of the 1.8V and 3.3V voltage regulators are 0.02V below their nominal values. Such deviations remain consistent with the specifications present in the data-sheet of the TCR2EF voltage regulators, used on the board. The indicated output voltage accuracy is of ± 18 mV[123]. CLK voltage, which provide access to one clock of the system, is also nominal, as the duty cycle of a clock is 50% and the multi-meter measure the average voltage. A value close to 0.9V is expected, if the clock voltage is 1.8V. In this case 0.88V has been measured.

Testpoint	Measured voltage
V_{BUS}	4.92 V
V_{DD}	4.62 V
3.3V	3.28 V
1.8V	1.78 V
CLK	0.88 V
V_{USB}	4.96 V

Table 5.2: Voltage measurements at different test points on the HP5 board

5.2.2 First test of the system

For this first test, a minimal approach was chosen to have the same experience as a person without programming experience, meaning that the pre-compiled firmware provided by *Protocentral* was used. On the HealthyPi website, links to the firmware are available, however none of them works : it lands on a 404 error Github webpage. The firmware can be found in the *"Releases"* tab on the Github repository of the project, where the UF2 files to update the firmware are available. Fixing the links could make the access to the firmware a bit more difficult for uninitiated people. Once the firmware is downloaded, the RP2040 is flashed by following the procedure described in annex A.7.

The last available version of the RP2040 firmware was used: 2.1.0. For the host, the system was tested on a desktop computer and a smartphone. The version used

of OpenView was the 2.1.17. On PC, it was downloaded on the *Releases* tab of the OpenView Github repository. Results on computer and smartphone are shown in figure 5.4 and 5.5 respectively. The conclusions for the acquisition of the different analog signals are the following:

- For both smartphone and computer, the **ECG signal** appears to be random : the R event is not visible as it should be and the trace is not exploitable.
- Also for smartphone and computer, the **respiration signal** is coherent : when inhaling the signal increases and when exhaling it decreases. It should be noted that the time scale is not the same for figure 5.4 and 5.5. The x-axis on computer has a duration of 12 seconds while on smartphone the x-axis last 25 seconds.
- The **PPG signal** is present on computer but not on phone. The signal displayed on the computer, in figure 5.4, seems coherent and has the same shape as classic PPG signal presented in 3.2. A lower frequency component is present in the signal : the signal increases over time. However, it should be noted that this signal was acquired by placing the thumb into the clip. This is not a recommended practice since it is explicitly stated in the "Don't" in the instruction provided by *Protocentral* : "Do not use the thumb as it may lead to inaccurate Spo2 readings"[124]. When testing with other fingers, only a flat curve is obtained. At this point of the work no explanation was found. Moreover, the PPG trace is extremely sensitive to ambient light : by simply having a hand moving between the ambient light source, a window in this case, and the sensor, a massive swing in the PPG trace is observed, as illustrated in figure 5.6.

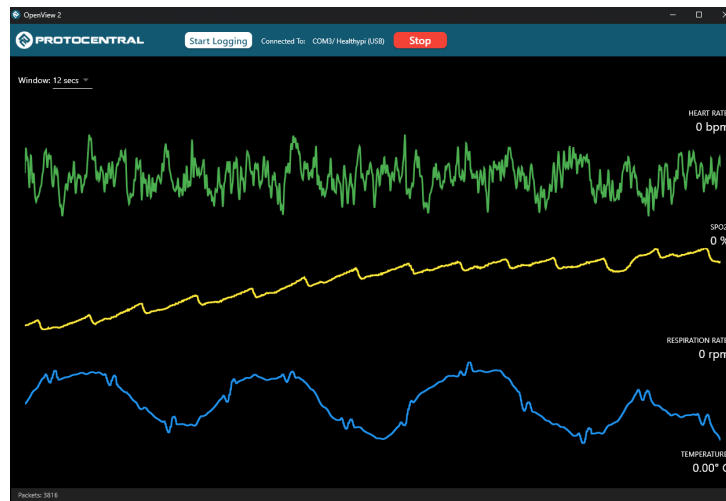


Figure 5.4: ECG (green), PPG (yellow) and respiration (blue) signals with original firmware and OpenView app on desktop, when using electrodes.

For the values of HR, SpO₂, RR and temperature it was observed that :

- No value is displayed on desktop, as shown in figure 5.4 : all displayed values stays at zero and no information except signal traces is provided. This could either be a bug in the OpenView application, where the data is correctly received but not displayed, or a communication bug when data is transferred from the microcontroller trough the USB port.

- More values are displayed on smartphone, as illustrated in figure 5.5 : all values are provided except the oxygen saturation level. The HR measurement is completely off, as the given value is 202BPM and that the user was at rest. The HR measurement is based on the ECG, which is noisy and not exploitable, it is therefore normal for the HR computation to be faulty. When the temperature probe is placed under the armpit, the temperature reading displayed in the app steadily increases before stabilizing at 34.1°C. This temperature is realistic but indicates a significant difference with the normal body temperature of $\approx 37^{\circ}\text{C}$. RR is also displayed on the smartphone screen, but the displayed value, in the case of figure 5.5 of 39 respiration per minute, seemed a bit high compared to the estimated RR during this first test (± 20 to 30 respiration per minute).

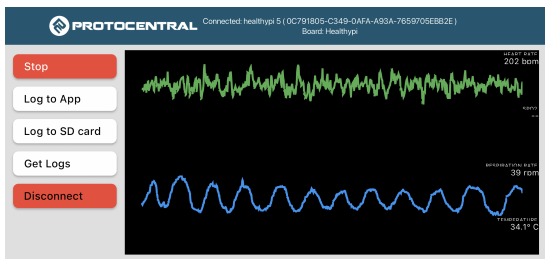


Figure 5.5: ECG (green), PPG (yellow) and respiration (blue) signals with original firmware and OpenView app on smartphone.

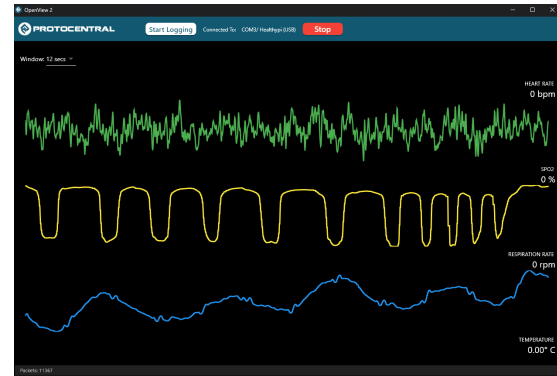


Figure 5.6: Effect of finger passing near the SpO₂ finger probe on the PPG (yellow) signal.

5.2.3 Overall integration

Functionality	On desktop USB streaming	On smartphone BLE streaming
ECG trace	Noisy	Noisy
Respiration trace	Coherent	Coherent
PPG trace	OK with the thumb, very sensitive to light	No trace
Temperature value	Stuck at 0	OK
HR value	Stuck at 0	Value based on noisy ECG
RR value	Stuck at 0	Value obtained seems too high
SpO ₂ value	Stuck at 0	No value displayed

Table 5.3: Functionality of HP5 features, on desktop USB streaming and smartphone BLE streaming

Table 5.3 summarizes the results at this stage in a more readable way. It highlights that most of the signal acquisition is faulty and that the computed metrics are either not displayed or inaccurate. In order to integrate the smartphone on a working platform, next section explains the debugging done and modifications made in order to have a more functional prototype. The goals are the following :

- display correctly vital metrics on desktop,
- display the PPG trace and metric on smartphone via BLE,
- have an exploitable ECG trace,
- make the SpO₂ measurement possible with the intended finger (index) and make the measurement less sensitive to light,
- make the different vital metrics more reliable.

5.3 Troubleshooting and modifications

All modifications described in this work are publicly available on the following *GitHub* repositories, which are forks of the original *ProtoCentral* repositories:

- Zephyr-based RP2040 firmware : https://github.com/beMang/protocentral_healthypi5_zephyr
- OpenView 2 application : https://github.com/beMang/protocentral_openview2
- All files related to this work that are not included in the previous *GitHub* repositories are available on the following UCLouvain Forge repository: : https://forge.uclouvain.be/aantonutti/master_thesis_smartphone_repurposing.

5.3.1 Communication issue

Fix USB communication debug

The first step to solve the communication issue was to investigate the content of the packet sent over the USB. A python script was used to receive and decode the packet. The packet structure used by OpenView is described in table 5.4. The content of the payload will vary in function of the *packet type byte*, which can take multiple values :

- **0x02** : this indicates a "standard packet". No information about the payload structure of this packet type has been found on the different repository of the project or on the *ProtoCentral* website. However, by retro engineering the OpenView code, the packet payload structure is easily retrieved and presented in table 5.5.
- **0x03** : this indicates a payload containing ECG and BioZ data. The payload contains 12 32 bits ECG samples, as well as HR and RR value encoded in 8 bits.
- **0x04** : this indicates a PPG related packet. It contains 8 PPG samples with 16 bits and a temperature value in 16 bits.

The README of the OpenView repository explicitly states that "*Healthypi 5 firmware version 1.0.0 and above, the data (ECG, Bioz and PPG) are sent as separate packets*" and uses packet type **0x03** and **0x04**. However, when checking the effectively used *packet type* bytes with the debug script, all the received packets are of type **0x02**. The documentation provided by *ProtoCentral* is inconsistent with the current publicly available implementation.

Position	Value	Description
0	0x0A	Start bytes
1	0xFA	
2	PL len LSB	Payload length
3	PL len MSB	
4	0x02	Packet Type
5	Payload 0	Payload
...	...	
i + 5	Payload i	
...	...	
PL len + 4	Payload end	
PL len + 5	0x0B	Stop byte

Table 5.4: Packet structure used by OpenView

Index	Size	Data Description
0-3	32 bits	ECG Sample
4-7	32 bits	BioZ / Respiration Sample
8	8 bits	BioZ skip flag
9-12	32 bits	PPG Red Channel
13-16	32 bits	PPG Infrared Channel
17-18	16 bits	Body Temperature
19	8 bits	SpO ₂ (%)
20	8 bits	HR (BPM)
21	8 bits	RR (RPM)

Table 5.5: HP5 payload format for packet type 0x02

```

1 void sendData(int32_t ecg_sample, int32_t bioz_sample, int32_t raw_red,
    int32_t raw_ir, int16_t temp, uint8_t hr, uint8_t rr, uint8_t spo2,
    bool _bioZSkipSample)

```

Listing 5.1: sendData function signature

By decoding the packet according to table 5.5, it is observed that the board simply fill bytes 17 to 21 with zeroes. This means that the firmware of the board is faulty and only sent values equal to zero. The bug is not related to the code of the application. In the firmware, the function that sends the data trough the USB port is **sendData**, its signature is provided in listing 5.1. When this function was called, the value of the 5 last arguments was equal to zero, explaining why zero was observed in the retrieved packages. To fix this the following modifications were done :

- Introduction of a new variables for each value (HR, SpO₂, RR and temperature).
- When a Zbus channel is updated, values are sent trough BLE if BLE is activated, now the corresponding previously created variables are updated too.
- Instead of using zero for the 5 last arguments, use the newly introduced and updated variables.
- For the temperature value, as it is published by the *Hardware* thread on a Zbus channel, and not from the *Data* thread like other variables, a new listener to the corresponding Zbus channel was added to update the local temperature value in the *Data* thread.

With those modifications, the different values computed by the *Data* thread are now transmitted as expected to the computer via USB.

Fix Bluetooth PPG trace absence

The reason why the PPG trace was absent when using the smartphone is simple : the function intended to send the PPG buffer via Bluetooth is simply never called. Functions are implemented to send the respiration, ECG and PPG trace via Bluetooth, but only the last one is never called. There is no clear reason of why this is the case, it is more probably an oversight. To fix this, a buffer was added to store the samples, similarly to how it was done for the ECG and respiration trace. Once the buffer is full, the already implemented function *ble_ppg_notify* is used to send the buffer to the smartphone via Bluetooth.

The nature of the before mentioned issues suggests that the firmware may not have undergone sufficient validation prior to publication. Although these flaws are technically simple to address, they were nonetheless critical for the system's functionality and user experience.

5.3.2 PPG acquisition troubleshooting and modification

AFE4400 description

The AFE4400 is an IC typically designed for PPG application. It is connected to an external photodiode that will measure light, as well as 2 LEDs emitting light at different wavelengths. The role of the chip is to drive the LEDs, filter and sample the signal from the photodiode and communicate those samples to the processor.

Figure 5.7 is the schematic bloc provided by the manufacturer, *Texas Instrument*, in the product datasheet. Its essential features are performed by the receiver (Rx) and transmitter (Tx) channel :

- **Rx channel** filters and amplifies the photodiode current, resulting from the transmitted light emitted by the LED. This is done with the block **TIA** presents on figure 5.7. After the signal has been filtered and amplified, an Analog-to-Digital Converter (ADC) samples and quantizes it. The ADC output digital samples are 22 bits wide and can be accessed by the processor through the SPI interface.
- **Tx channel** handles the current delivered to the LEDs. It controls when each LED emits light. The current can be programmed up to 50 mA with 8 bits resolution.

The timing of when each LED emits light and when the sampling process starts and ends is handled by the timing controller. The timing of each step can be programmed precisely by changing configuration register of the chip. The timing configuration of the AFE4400 has not been changed and will therefore be not discussed.

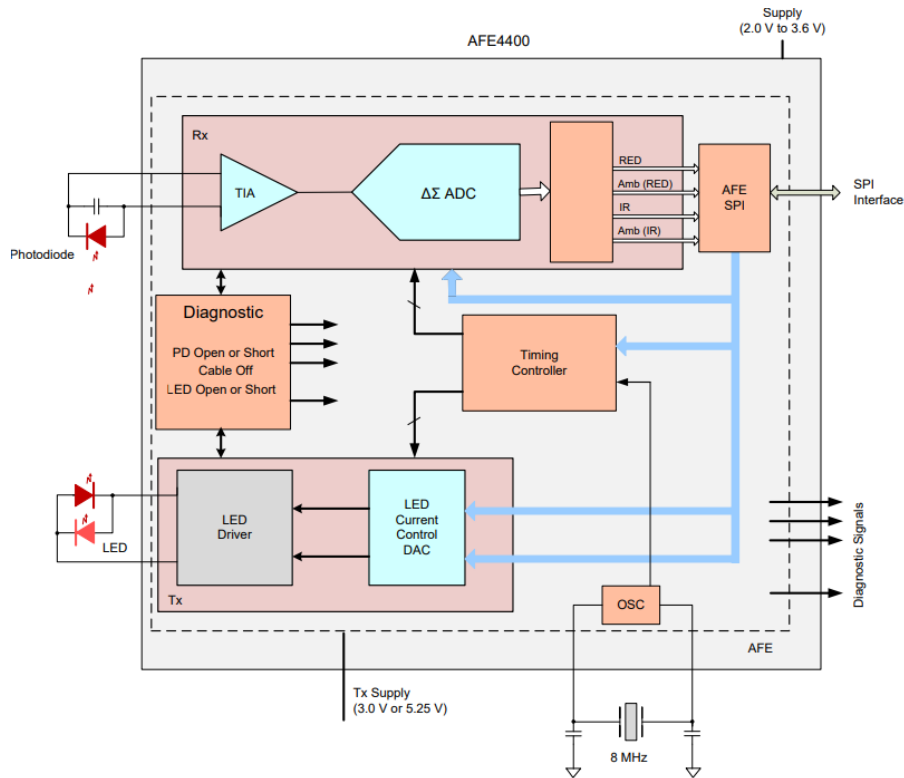


Figure 5.7: AFE4400 schematic bloc, extracted from [125]

The output registers available through the SPI interface are presented in table 5.3.2. Measurements are available when both LEDs are off, corresponding to the measurements of the ambient light. Registers *LED1* and *LED2* correspond to the light intensity when *LED1* and *LED2* are on respectively. Additionally, the subtraction between those 2 measurements is made. Those registers allow to remove the component of the ambient light intensity and to isolate the useful PPG signal. In the original implementation, only register *LED1* and *LED2* are used. Those values are affected by the ambient light variation and explain the observed sensitivity to ambient light in figure 5.6.

Channel	Description
LED1	Voltage measured during LED1 pulse.
AMB LED1	Ambient light voltage measured during LEDs off-phase.
LED1 - AMB LED1	Ambient-subtracted voltage for LED1 channel.
LED2	Voltage measured during LED2 pulse.
AMB LED2	Ambient light voltage measured during LEDs off-phase.
LED2 - AMB LED2	Ambient-subtracted voltage for LED2 channel.

Table 5.6: Summary of AFE4400 output data channels

Critical bug fix

The HealthyPi firmware provides logs through the UART port. How the UART port is read with the smartphone is detailed in section 6.3. When using the PPG sensors, the provided logs indicate that the AC components of the PPG signal is too small. The log warning is triggered when the AC amplitude is below 100. In order to increase this AC component, increasing the current flowing into the LED was tested. The default

current value was 20 mA. An additional log was added to display the raw value of the PPG signal. It was observed that the logged value was always 8191, except in the case where the thumb was used to cover the photodiode sensor. Since the output register of the AFE4400 is 22 bits wide, using a two's complement representation, the expected range is from -2^{21} to $2^{21} \approx 2 \cdot 10^6$. This saturation at $8191 = 2^{13} - 1$ was unexpected. Investigations resulted in the finding of a critical bug in the driver of the AFE4400.

To convert a N bits binary number $a_{N-1}a_{N-2}...a_0$ using two's complement representation into decimal d , the following formula is used[126] :

$$d = -a_{N-1}2^{N-1} + \sum_{i=0}^{N-2} a_i 2^i \quad (5.1)$$

In this representation, the most significant bit (MSB) functions as the sign bit, representing the negative decimal component. The RP2040 architecture utilizes 32-bit integers; therefore, to correctly interpret the 22-bit data stream from the AFE4400, the MSB must be properly aligned with the 32-bit word. In the driver implementation shown in listing 5.2, the raw data is left-shifted by 10 bits to align the sign bit with the 32nd bit of the register. An arithmetic right shift is subsequently required to restore the original scale; this operation preserves the sign bit because of the `int32_t` cast. However, the current implementation incorrectly shifts the data by 18 bits instead of 10. This error causes 8 data bits to be discarded, truncating the effective dynamic range to 14 bits. As a result, the signal is constrained to a range of $[-2^{13}, 2^{13} - 1]$, which directly explains the saturation observed in the output.

```
1 uint32_t led1val = _afe4400_read_reg(dev, LED1VAL);
2 led1val = (uint32_t)(led1val << 10);
3 int32_t led1val_signed = (int32_t)led1val;
4 *raw_ir_sample = (int32_t)led1val_signed >> 18;
```

Listing 5.2: Critical bug in AFE4400 driver

A first explanation for this bug is that the reduction of the effective resolution was intentional in order to reduce the payload size during transmission, as PPG samples are transmitted as 16-bit values in packet type 0x03. However this would mean that 2 bits would be always zero, since the incorrect driver output is 14 bits. Another possible explanation is related to the effective resolution of the AFE4400 itself. Although the output register is 22 bits wide, the data-sheet specifies a noise-free resolution of 13 bits[125]. The implemented bit truncation therefore approximately matches this effective resolution. However, preserving the complete 22-bit output remains preferable, as averaging and filtering operations performed by the AFE4400[125] can still improve the effective resolution. Regardless of the original intention, the current implementation results in a significant truncation of the available signal dynamic range.

Fixing this bug only requires to change the number 18 by 10 in listing 5.2 to correctly do the conversion between the AFE4400 22 bits wide register and RP2040 32 bits integer. However, the output was still saturated, at the expected saturation value ($2^{13} - 1$), when the index was used in presence of ambient light. When the thumb is used, because it is thicker, the amount of light that hit the photodiode is reduced and the sensor output is not saturated. When working in an environment with almost no ambient light, the PPG signal was also not saturated and observable. It can be concluded that the contribution of the ambient light saturates the input of the analog to digital converter.

AFE4400 configuration modification

In order to remove the saturation observed when the index finger is used, the configuration of the analog frontend integrated circuit AFE4400 was modified. Figure 5.8 shows the analog front end present in the AFE4400 before the ADC. The voltage V_{DIFF} at the input of the ADC is given by equation 5.2. The photodiode current is $I_{PLETH} + I_{AMB}$, I_{PLET} is the current related to the blood volume variation and I_{AMB} is the current resulting of the ambient light. I_{CANCEL} is a programmable current to reduce the contribution of the ambient light. The original implementation configured this cancellation current to zero. Additionally, R_F and R_G can be programmed to set the gain of the first and second amplification stage respectively. Choosing the right value is crucial to fulfil the following goals :

- The output V_{DIFF} , during LEDs on phase, should ideally be around 0.6 V, as recommended by the manufacturer[125], this will ensure to be in the dynamic range of the ADC.
- Reduce the sensitivity to ambient light, this will avoid unintended saturation of the ADC input.

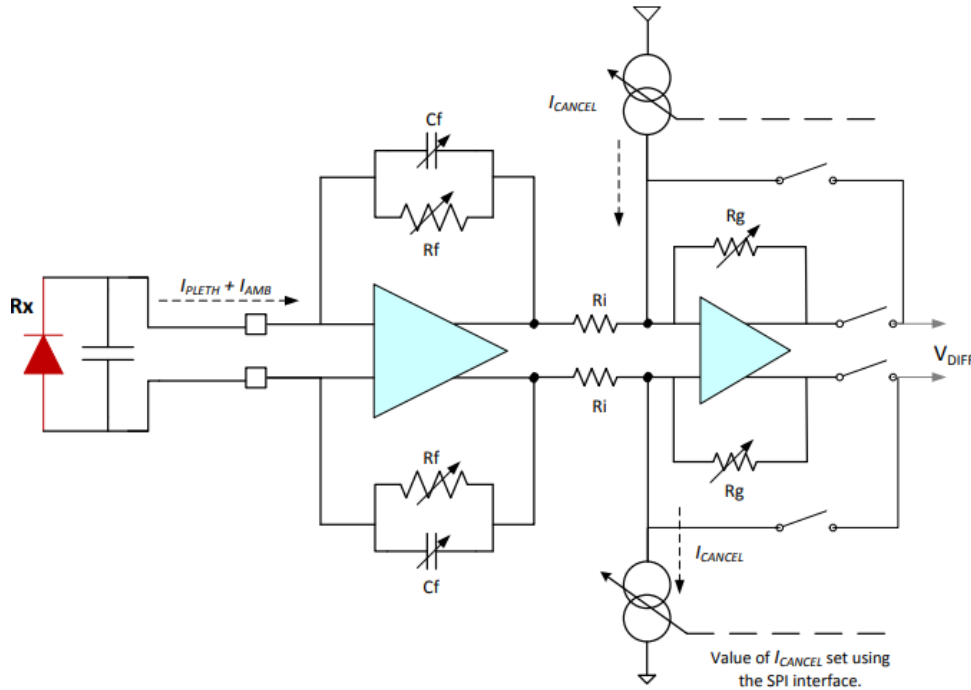


Figure 5.8: AFE4400 TIA schematic bloc, extracted from [125]

$$V_{DIFF} = 2 \cdot \left(I_{PLETH} \frac{R_F}{R_I} + I_{AMB} \frac{R_F}{R_I} - I_{CANCEL} \right) \cdot R_G \quad (5.2)$$

The first adjustment to achieve those goals was to set a cancellation current. The available current range from 0 to 10 μA , with a resolution of 1 μA . The following feedback loop was implemented in the AFE4400 driver :

- Every 5th samples, put (AMB LED2 + AMB LED1)/2 in a buffer. It has been choose to do this measurement not on every sample to reduce the additional computation effort done by the microcontroller.

- Once the buffer has 10 samples, average ambient light value is computed. This average ambient light value will serve as error for a simple proportional-integrator controller.
- A variable *integral* accumulates the error over time. The cancellation current is given by : $I_{CANCEL} = 0.01 \cdot error + 0.001 \cdot integral$.
- If the output current is out of the limit, adapt it to the possible range (0 to 10 μA).
- Write the final I_{CANCEL} computed into the corresponding register.

That change alone did not solve the issue : saturation was still observed when the index was used indoor with normal daylight. The contribution of the term $I_{AMB} \frac{R_F}{R_I}$ in equation 5.2 was bigger than the maximum available cancellation current. It was observed that the current value chosen by the controller was always the maximum one of 10 μA . To mitigate this, R_F was reduced. In the initial implementation R_F had a value of 500k Ω , reducing it by a factor of 5 resulted in good performance. To keep term $I_{PLETH} \frac{R_F}{R_I}$ around the same order of magnitude, the LED current was multiplied by five. Since the resistor R_F and the feedback capacitor C_F form a low pass filter for the input current[125], the capacitor value need to be adapted. Frequency content above the cut-off frequency, at $f_c = \frac{1}{2\pi R_F C_F}$, will be attenuated, resulting in less noise in the signal. To remove as much noise as possible, the cut-off frequency should be as low as possible. However, since the LED current and; therefore; the photodiode current consists of pulse of current, the bandwidth of the filter should be sufficiently high to not filter them. Equation 5.3 is the recommendation given by the manufacturer to have a sufficient bandwidth.

$$R_F \cdot C_F \leq \frac{RX_sample_time}{10} \Leftrightarrow C_F \leq \frac{RX_sample_time}{10 \cdot R_F} \quad (5.3)$$

Table 5.7 compares the original filter configuration and the modified one. The advised capacitor value is the one obtained using equation 5.3. In the original implementation, C_F value was lower than the advised one, resulting in a higher cut-off frequency. In the modified implementation, the maximum available C_F value was used : 250pF to be as close as possible to the advised capacitor value. The resulting cut-off frequency is one order of magnitude lower, resulting in less parasitic high frequency content at the input of the ADC.

Implementation	R_F (k Ω)	C_F (pF)	Advised C_F (pF)	f_c (kHz)
Original	500	5	100	63.662
Modified	100	250	500	6.366

Table 5.7: Input low pass filter configuration

The last configuration modification was to change which register is read for each sample. The original implementation was reading registers LED1 and LED2 (see table 5.3.2 for output register description), which is sensitive to the ambient light variation. Reading registers (LED1 - AMB LED1) and (LED2 - AMB LED2) that removes the value of measured ambient light results in a much less sensitivity to ambient light. The large swing observed in figure 5.6 when a hand passes between the sensor and ambient light source is removed, as shown in figure 5.10, and the PPG signal remains exploitable.

DC filter implementation

The last step to improve the acquisition of the PPG signal was the addition of a high pass filter to remove the lower frequency component present in the PPG signal. A simple infinite impulse response filter, proposed by [127], is used to remove this lower frequency component of the signal before sending the data. The HR estimation and oxygen level algorithm still uses the original signal, only the signal sent to the host is modified.

$$y[n] = \frac{1 + \beta}{2} (x[n] - x[n - 1]) + \beta y[n - 1] \quad (5.4)$$

Equation 5.4 describes the filter and figure 5.9 shows the amplitude response of the filter for multiple β . β is a parameter that will shift the cut-off frequency : a smaller β leads to more DC attenuation, but cut-off frequency becomes higher. As the frequency range of interest for the PPG stands between 0.01 to 2Hz [128], $\beta = 0.995$ was chosen, resulting in a cut-off frequency at ≈ 0.01 Hz.

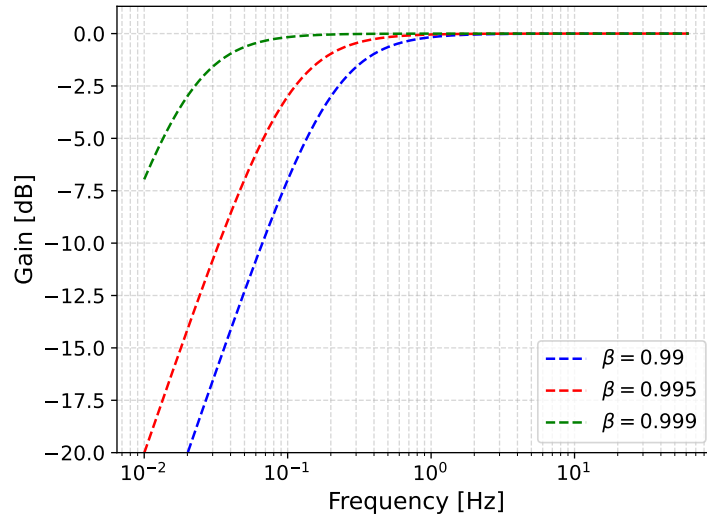


Figure 5.9: Gain of high pass filter for PPG signal

Results and further improvements

The improvements of the PPG signal acquisition is clearly visible when comparing the recorded PPG trace before (figure 5.4) and after (figure 5.10). The removal of the low frequency component allows a much better signal visualisation as the peak of the signal are much more stable with time. This signal has been recorded with the same ambient light variations presents for figure 5.6, by moving a hand between the sensor and the ambient light source. The swing observed in figure 5.6 is not observed with the modifications: the sensitivity to ambient light has been removed. However, despite the improved signal acquisition, multiple improvements are still possible:

- Even with the reduced R_F , the chosen value by the controller for the cancellation current I_{CANCEL} was the maximum one of $10 \mu A$. The term $I_{AMB} \frac{R_F}{R_I} - I_{CANCEL}$ was not sufficiently small to use the second stage gain R_G . If the gain was greater, this term would be amplified and result in saturation of the ADC input. Further

reducing R_F would lead to have $I_{AMB} \frac{R_F}{R_I} \approx I_{CANCEL}$ and enable use of the second stage gain.

- With the second stage gain, a possible reduction of the used LEDs current could be experimented to reduce the power consumption of the system.
- The PPG acquisition is sensitive to motion of the finger. Even small movement of the index can lead to significant movement artefacts, as present in figure 5.11. A motion removal algorithm, as proposed by [50], could be implemented.
- Diagnostic signals are available to detect if the photodiode or LEDs are in open or short circuit. Those signals are not used in the current implementation, they could help to detect sensor connection status or hardware failure in case of short-circuit.

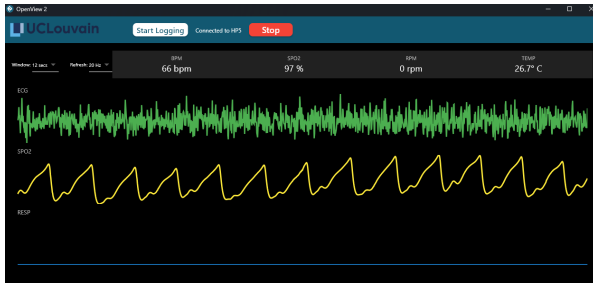


Figure 5.10: PPG signal acquired with modifications, with ambient light variation

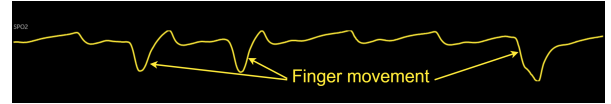


Figure 5.11: PPG signal motion artefact

5.3.3 MAX30001 Troubleshooting

Working principle

To check the configuration of the MAX30001, it is needed to understand the working principle for ECG and respiration signal acquisition. The respiration signal is derived from the variation of the thorax impedance. Thorax impedance has two components : a relatively constant value Z_B and a varying value Δ_R . It can be written $Z_T = Z_B + \Delta_R$. According to [129], "there is a good correlation between the impedance Δ_R and the volume of respiration air". Δ_R typically ranges from 0.1 to 1Ω[129]. To measure the impedance, a small high frequency current I_{MOD} is injected through the body by electrodes, in this case at 8 kHz and by electrodes RA and LA. The measured voltage measured by the electrodes, with the voltage V_{ECG} induced by the heart's electrical activity :

$$V = V_{ECG} + Z_T \cdot I_{MOD}$$

The injected current is done at high frequency to avoid the ECG signal represented by V_{ECG} and the respiration signal to be at the same frequency range. This allows the system to measure thorax impedance and the ECG signal at the same time on the same electrodes. The respiration signal contained in Z_T is present around the frequency of the modulating current I_{MOD} , at 8 kHz in the case of the HealthyPi. Figure 5.12 illustrates that by showing the frequency content of V . It has been simulated with a RR purely sinusoidal at 30 respiration per minute, and an ECG signal of 60 BPM, and an injection

current I_{MOD} at 8 kHz. It can be seen that the ECG frequency content, ranging from 0.5 to 70 Hz[130] is isolated from the respiration signal that has been modulated around the higher frequency of I_{MOD} . Present harmonics are the odd one (1,3,5...) since the injected current has a square wave shape. The code for this simulation is available in listing 2.

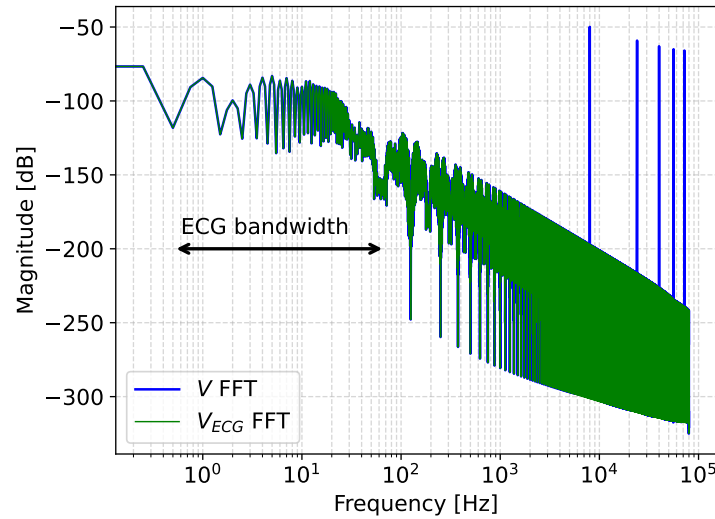


Figure 5.12: Frequency content of V from ideal simulation

ECG channel description

Both bio-impedance and bio-potential channel have ESD and EMI protection at their input, as shown in figure 5.13 and 5.14. ESD protection prevent electrostatic discharge from damaging electric component, by clamping too high voltage at the input. The EMI protection consist of a low-pass filter removing high frequency content of electromagnetic interference that can be generated by nearby devices. It reduces noise and prevents interference with other devices[131].

ECG channel (or biopotential channel as named in the data-sheet) is described in figure 5.13. The input amplifier has a fixed gain of 20 V/V and also acts as a first-order high-pass filter. The cut-off frequency is set by the capacitor placed between the $CAPP$ and $CAPN$ pins. In the case of the HealthyPi, the capacitor placed on the PCB has a value of 1 μF , setting the cut-off frequency to 0.5 Hz. A second-order low-pass

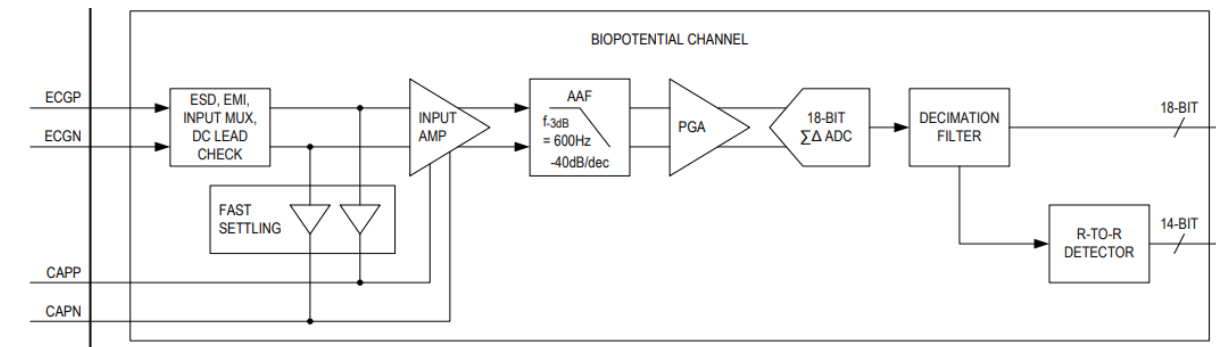


Figure 5.13: MAX30001 ECG channel description, extracted from [132]

filter with a cut-off frequency of 600 Hz is then applied. This filter has two goals: acting as an anti-aliasing filter by removing frequency components higher than the Nyquist frequency, and removing the high-frequency component $Z_T \cdot I_{MOD}$ related to the respiration signal. The ADC sampling rate is approximately 32 kHz, and this filter provides a 57 dB attenuation at 16 kHz. Following this filter, a second amplifier with programmable gain ranging from 1 to 8 is used, resulting in an overall gain from 20 to 160 V/V, before the signal is quantized by an 18-bits ADC. A decimation filter then reduces the ADC sampling rate from 32 kHz to the desired sampling rate configured by the controller. This digital filter can also be configured to act as a band-pass filter, with low and high cut-off frequencies defined by $DHPF$ and $DLPF$. By analyzing the configured register values together with the information provided in the data-sheet, the original ECG channel configuration could be reconstructed and is summarized in table 5.3.3.

Parameter	Value
Sampling rate	128 Hz
Analog HPF	0.5 Hz
Overall gain	80 V/V
$DLPF$	28.35 Hz
$DHPF$	0.5 Hz

Table 5.8: ECG channel configuration

Samples are recovered with the SPI interface. To convert a sample ADC to the voltage between the electrodes V_{ECG} related to heart's electrical activity, the following formula is used, where ECG_GAIN is the overall¹ gain :

$$V_{ECG} = \frac{ADC}{2^{17} ECG_GAIN} \quad (5.5)$$

The *MAX30001* also features an integrated *R-to-R detector*, which detects the R event, which consist of the high peak observed in the ECG waveform. A register is available containing the interval between the two last R event. This register is used to estimate the HR in the HealthyPi firmware. However, since the ECG appeared noisy, the HR estimation from the PPG signal will be used for the rest of this work.

Respiration channel description

The respiration channel (named BioZ channel in the data-sheet), presented in figure 5.14, consists of a high-pass filter whose goal is to remove the frequency content of the ECG before demodulating the impedance signal. Its cut-off frequency can be programmed from 125 to 7200 Hz or bypassed. The signal is then amplified by an input amplifier and subsequently demodulated using a mixer and a low-pass filter. During the demodulation stage, a programmable phase offset can also be applied in order to compensate for phase shifts introduced by the body impedance and acquisition chain. This filter also acts as an anti-aliasing filter, ensuring that no frequency content is present above the Nyquist frequency. The demodulated signal is then amplified with

¹By overall, it means the gain of the whole amplification chain, with the input amplifier taken into account.

an overall programmable gain ranging from 10 to 80 V/V. The resulting analog signal is then quantized and sampled by a 20-bit ADC. Finally, a decimation filter with the same purpose as in the ECG channel is applied before the samples are transmitted through the SPI interface. The original BioZ channel configuration can be found in table 5.9.

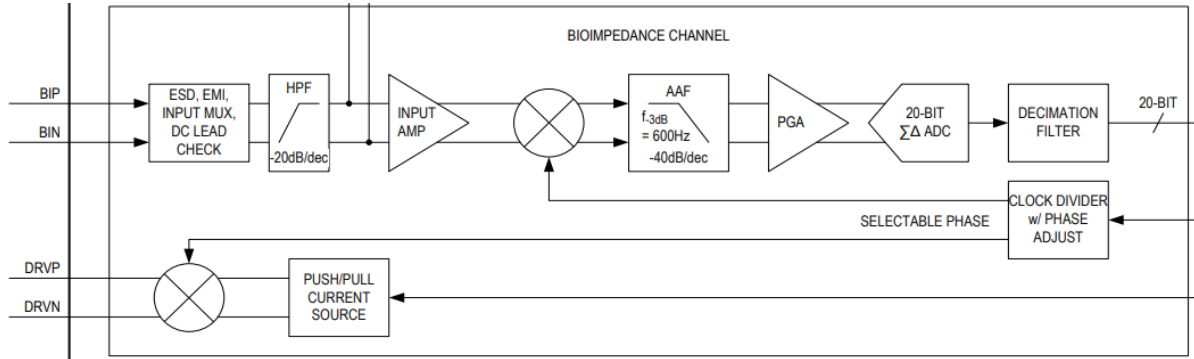


Figure 5.14: MAX30001 Respiration channel, extracted from [132]

Parameter	Value
AHPF	125 Hz
Current amplitude	48 μ A
Current frequency	8 kHz
Overall gain	40 V/V
DHPF	0.05 Hz
DLPF	4 Hz
Phase offset	33.75°

Table 5.9: BioZ/respiration channel configuration

Samples can be converted to impedance using the following formula, where ADC is the sample value, $BIOZ_CGMAG$ is injected current amplitude and $BIOZ_GAIN$ the overall gain :

$$Z_T = \frac{ADC}{2^{19} \cdot BIOZ_CGMAG \cdot BIOZ_GAIN} \quad (5.6)$$

Note that, depending on the digital filter configuration, Z_B , the DC part of Z_T , might be filtered.

Troubleshooting and modifications

Regarding the ECG and respiration signal acquisition, as no obvious configuration mistake of the MAX30001 was found, no extensive implementation modification was done. However following minor modifications were done :

- **Change sample rate :** The original sample rate configured on the MAX30001 was 128 Hz. To be consistent with the rest of the system, it was modified to 125 Hz.
- **Digital low pass filter configuration :** The $DLPF$ is bypassed, as the ECG signal can contain useful information above 28.35Hz[130]. Moreover, if filtering is actually useful on the ECG signal, it can be done at later stage of processing, either on the microcontroller or the smartphone CPU.

- **Bit shifting issue :** The same bit shifting issue observed in the AFE4400 driver was also observed here. The function converting the 24-bit ECG register to a 32-bit integer on the controller side was sub-optimally implemented. The retrieval of the raw two's complement ECG samples, performed at line 1 of listing 5.3, is consistent with the MAX30001 datasheet: the 18-bit ECG value is located from bit 23 to bit 6 included. The 6 least significant bits contain tags related to the validity of the current ECG sample[132]. This explains the presence of the bit mask `0xC0 (0b11000000)`, used to remove the bits associated with the tags. The resulting variable is then shifted 8 bits to the left to correctly align the sign bit of the two's complement representation. Then, an arithmetic right shift of 6 bits is applied. This value was modified to 14 in order to remove the bits set to zero by the mask and recover the original ECG sample value. Unlike the AFE4400 case, where bits were lost, no bits are lost here and only the scaling of each sample is affected. With the modified bit shifting, all samples become 2^6 times smaller. This modification is therefore not expected to change the ECG signal acquisition results, but only to provide a more consistent scaling.

```

1 uint32_t uecgtemp = (uint32_t)((((uint32_t)buf_ecg[i * 3] << 16 | (
    uint32_t)buf_ecg[i * 3 + 1] << 8) | (uint32_t)(buf_ecg[i * 3 + 2] & 0
    xC0)));
2 uecgtemp = (uint32_t)(uecgtemp << 8);
3
4 int32_t secgtemp = (int32_t)uecgtemp;
5 secgtemp = (int32_t)secgtemp >> 6;
6
7 ecg_samples[valid_ecg_samples] = (int32_t)(secgtemp);
8 valid_ecg_samples++;

```

Listing 5.3: Driver code for ECG samples retrieval with incorrect bit shifting

As expected, the corrected shifting modification of the driver did not solve the noisy ECG trace. Moreover, multiple users reported noisy ECG acquisition on the *GitHub* repository of the project [133], [134], suggesting the presence of either a hardware implementation issue or a software-related problem. However, it was observed that the electrode placement strongly affects the signal quality, and under a less conventional electrode placement, the prototype can produce a functional ECG signal, as discussed in the next section.

Impact of electrodes placement

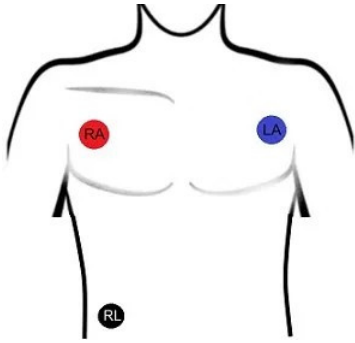


Figure 5.15: Recommended ECG electrodes placement, extracted from HP5 documentation[135]

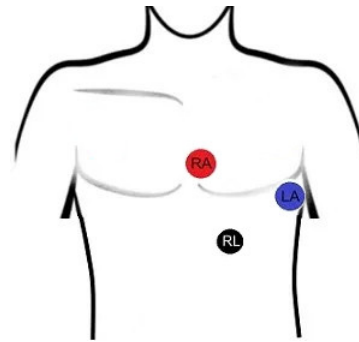


Figure 5.16: Alternative ECG electrodes placement

Previous tests of the system were performed using the electrode placement recommended in the HealthyPi documentation[135], shown in figure 5.15. This electrode configuration corresponds to the standard placement used for 3-electrode ECG acquisition[136], however it resulted in a noisy ECG trace. According to [137], better results can be obtained by "*placing electrodes on the chest wall equidistant from the heart*". This configuration, shown in figure 5.16, was therefore tested. The voltage amplitude was retrieved using equation 5.5. The resulting ECG trace, presented in figure 5.17, is significantly more exploitable: the R peak is clearly visible, as well as the Q wave. Other ECG waveform characteristics are however not distinguishable. No definitive conclusion regarding the quality of the reconstructed ECG waveform can be drawn at this stage, since only one subject was tested and no comparison with a gold-standard reference was performed. Furthermore, other electrode configurations should be evaluated to determine which placement provides the best signal quality. The main conclusion at this stage is therefore that the system can become functional when using a different electrode placement. However, the HP5 documentation should be adapted to better account for the sensitivity of the ECG acquisition to electrode positioning, as multiple users also reported difficulties obtaining a usable ECG trace[133], [134].

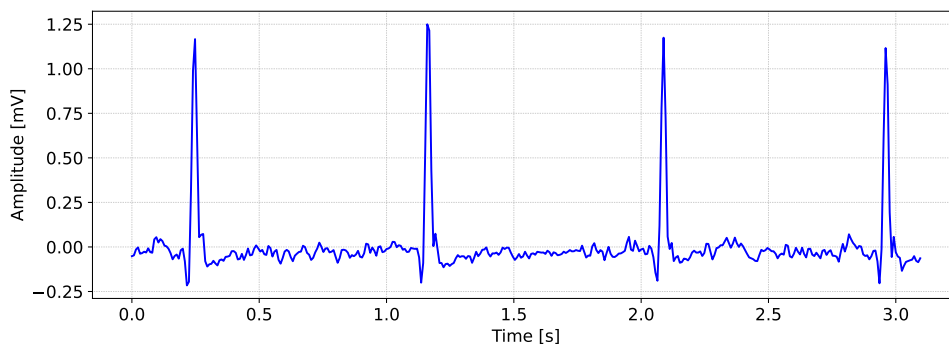


Figure 5.17: Obtained ECG trace with placement of figure 5.16

5.3.4 Minor modifications

Modified sampling rate

The timer setting the sampling rate in the *Sampling thread* was modified. Originally it has a value of 7 ms, which corresponds to a sampling frequency of ≈ 142.85 Hz. However, the configured sampling frequency of the AFE4400, and now also of the MAX30001, is 125 Hz. Moreover, the SpO₂ estimation algorithm assumes a sampling frequency of 125 Hz. In order to keep the sampling frequency consistent across the system, the value of the timer in the *Sampling thread* was changed to 8 ms, which correspond exactly to 125 Hz.

Spo2 and PPG based HR algorithm estimation modification

The main steps of the HR and SpO₂ estimation algorithm are detailed in appendix A.10. Step 3 of the algorithm, which detects ADC saturation, was modified. In the original implementation, saturation was detected by comparing the DC values of both channels with a threshold of approximately $4 \cdot 10^6$. This value is not consistent with the actual ADC output range, which is $[-2^{21}, 2^{21} - 1]$. Negative values are clamped to zero by the *Data thread*. This should not occur in normal operation, as the AFE operating point is configured to provide positive output values, as recommended by the manufacturer. The saturation threshold was therefore changed to 1 990 000, corresponding to approximately 95% of the positive full scale. This ensures a more consistent detection of ADC saturation.

Chapter 6

Smartphone integration to the system

This chapter explores the integration of a smartphone into the HealthyPi platform to reduce hardware cost and simplify the system architecture. First, a cost analysis of the existing HealthyPi implementation is conducted to evaluate the potential savings enabled by smartphone integration. Different communication approaches between the smartphone and the biomedical sensing hardware are then investigated, namely I2C, UART, USB and Bluetooth interfaces. Finally, USB and UART communication implementations details are explained.

6.1 Design space exploration

6.1.1 Cost repartition of HP5 system

To quantify the potential cost reduction brought by integrating a smartphone in the HealthyPi systems, the cost of a functional HealthyPi platform has been analysed. This functional platform includes a battery and a display screen, similar to the complete kit provided by *Protocentral*, so the system is functional on its own. To determine the percentage of each part of the system, unit prices were used, in the context of a bigger scale purchase, the price of electronic components and PCB manufacturing can be significantly decreased as degressive pricing is a common practice in the industry. The cost of assembly and labour work associated to it was not taken into account. Results of this analysis are shown in figure 6.1, it provides the cost repartition between sensors, screen, battery, PCB manufacturing and electronics components present on the PCB

The total estimated cost of the system is 134.38\$, the most costly part of the system is the PCB manufacturing and the prices of the electronic components. The cost of the sensors probe (the ECG cable and SpO₂ finger-clip) accounts for 21% of the cost of the system, the use of a smartphone will not be able to shrink the cost associated to the sensors. However, the integration of a smartphone removes the need of screen and battery for the system and leads to a 21.1% cost reduction.

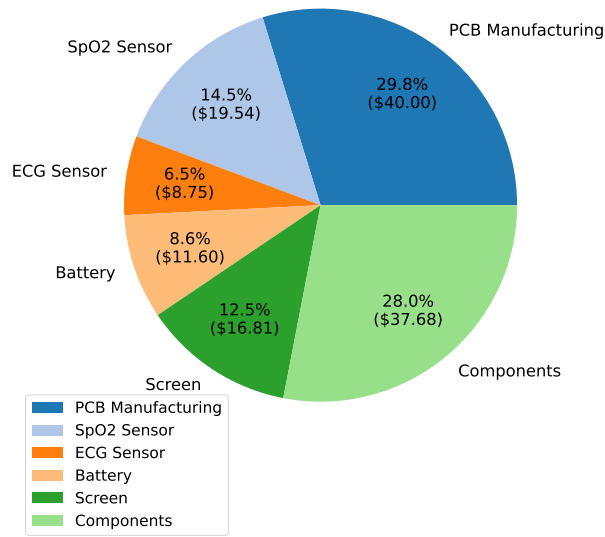


Figure 6.1: Cost repartition for the complete system

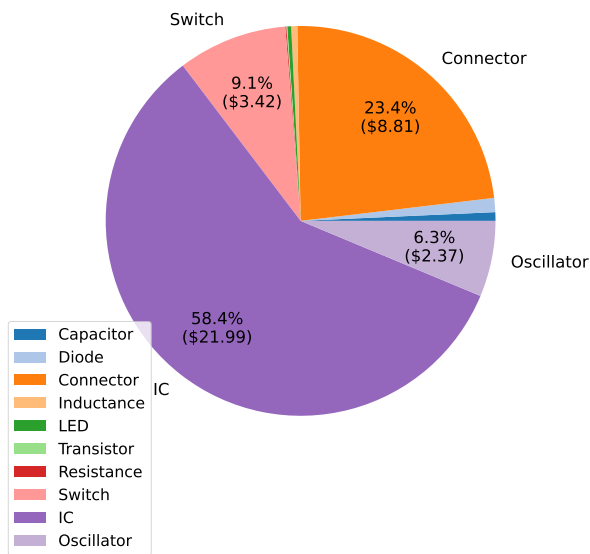


Figure 6.2: Cost repartition for electronic components

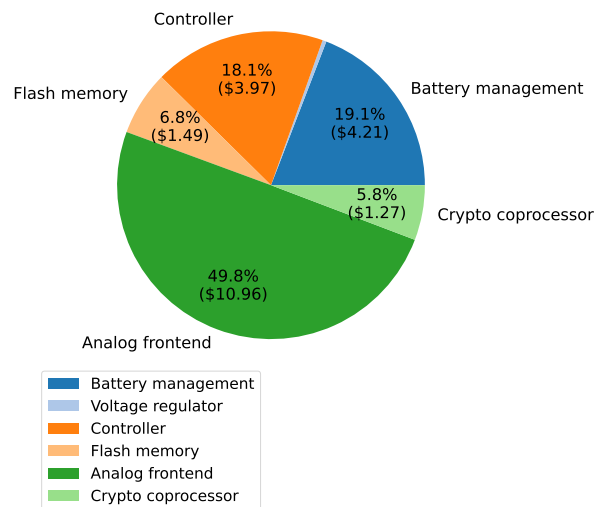


Figure 6.3: Cost repartition for ICs

Figure 6.2 details the cost repartition of the different electronic components present on the HealthyPi board. Integrated circuits and connectors holds for most of the cost. The cost of the different type of ICs is provided in figure 6.3. In the ICs, the AFEs are the most expensive part and smartphone do not have this circuit type, this cost will be incompressible. However, the cost of battery management ICs can be removed since the system will be supplied by the smartphone. Also, as the smartphone has internal storage, the SD card slot present in the *Connector* category can be removed. In total, this leads to a 33.60\$, or 25%, cost reduction of the system.

6.1.2 Comparison of possible connection between smartphone and AFEs

To further reduce the cost, different connection type were considered. The only hardware that the smartphone does not have is the AFE and the connectors for the sensors, in figure 5.2, it corresponds to the *Sensors* and *Analog frontend* section. All considered connection options are explored in next sections.

SPI

A first possibility to reduce the hardware cost is to directly interface the AFEs with the smartphone communication bus. In such a configuration, the smartphone would communicate directly with the AFEs, removing the need for an intermediate micro-controller or dedicated communication hardware. This approach would provide the highest potential cost reduction, since most of the external communication hardware could be eliminated.

The AFEs used in the HealthyPi platform communicate through an SPI interface. However, SPI communication requires a dedicated Chip Select (CS) signal for each peripheral device connected to the bus [138]. In a system integrating multiple AFEs, several independent CS lines would therefore be required. One option would be to remove an existing peripheral already connected to the SPI bus and reuse its CS line for an AFE. However, smartphones do not generally expose multiple low-level CS lines, making the direct integration of multiple SPI AFEs impractical.

I2C

Unlike SPI, I2C protocol allows multiple devices to share the same communication bus without requiring dedicated selection signals for each peripheral. To identify the target device during communication, an addressing mechanism is directly integrated into the I2C protocol [139], allowing each peripheral connected to the bus to be uniquely identified by its address. As the AFEs currently used in the HealthyPi platform communicate through SPI, a list of possible I2C-compatible AFE alternatives was identified and is presented in table 2.

Both approaches (I2C and SPI) present several practical limitations :

- Accessing the I2C or SPI bus of a smartphone generally requires direct modifications of the smartphone printed circuit board. Such modifications are technically challenging, require specialized equipment and soldering skills, and would be difficult to reproduce reliably in a large-scale deployment context.
- Significant software challenges may arise, since accessing low-level communication interfaces on commercial smartphones will probably require modifications of the operating system, custom drivers, or elevated system privileges.
- Once modified, the smartphone would likely become dedicated to the medical acquisition system and could no longer be conveniently used for general-purpose applications

For these reasons, despite its strong cost reduction potential, the direct AFE interface, through SPI or I2C, approach appears difficult to implement in practice.

UART

The UART debug interface available on smartphones can be used for communication. UART, standing for *Universal Asynchronous Receiver-Transmitter*, is a serial communication protocol allowing data exchange between two devices using only two communication lines, namely a transmit (TX) and a receive (RX) signal [140]. Unlike protocols such as SPI or I2C, UART communication does not require a dedicated clock signal [140], which simplifies the hardware implementation. The original purpose of the smartphone UART debug interface is to *"get kernel or bootloader logs on a system without a working display/full USB configuration"* [141]. The RP2040 microcontroller used in the HealthyPi platform also provides UART communication pins, primarily intended for debugging purposes. These interfaces could also be used to transmit biomedical measurements from the AFEs to the smartphone through the UART connection.

Access to the smartphone UART interface is generally possible on most smartphones, although its physical implementation varies depending on the manufacturer and smartphone model. According to [141], UART access can typically be found through dedicated pins exposed on the USB connector using a specific cable, through the headphone jack on older devices, or directly on specific pins on the smartphone main board.

In this configuration, a microcontroller would still be required to collect data from the AFEs and transmit it to the smartphone through UART communication. A simpler microcontroller than the RP2040 could potentially be considered for this purpose. However, given the relatively low cost of the RP2040 and its existing integration within the HealthyPi platform, replacing it may not provide a significant economic advantage.

USB

Another possible solution is to use the USB interface available on smartphones. Unlike low-level communication buses such as I2C or UART debug interfaces, USB connectivity is natively supported on all modern smartphones, making it a widely accessible and standardised communication method. It can also be easily connected to a computer. In addition, low-cost microcontrollers such as the RP2040 provide native USB communication capabilities, allowing integration with the sensing platform. This approach also avoids direct hardware modifications of the smartphone, considerably simplifying the implementation and improving the reproducibility of the solution. While UART communication can also not require hardware modification on certain devices through the audio jack or USB connector depending on manufacturer implementation, USB remains significantly more standardised and universally accessible. The USB communication between a computer and the HealthyPi is already implemented, but not between the HealthyPi board and a smartphone.

Bluetooth

Bluetooth is widely supported on modern smartphones and provides a wireless communication interface that may improve usability and user comfort by removing the

need for a physical cable connection. However, this approach requires a microcontroller supporting Bluetooth communication, either through an integrated radio interface or through the addition of dedicated wireless communication hardware. Consequently, part of the communication hardware duplicates functionalities already available in the smartphone, which may reduce the potential cost savings compared to wired communication approaches.

6.1.3 Cost saving for each solution

For the cost saving estimation, only scenarios keeping a single communication method were considered. In all configurations, as previously stated, the integration of a smartphone allows the removal of the screen, battery, and associated power-management circuitry, resulting in a baseline cost reduction of 25% (33.60\$). The cost of the smartphone itself was not included in the estimation, as it is assumed that the end user already owns a compatible smartphone or that an existing unused device could be repurposed for this application. The detailed results for each communication approach are presented in table 6.1.

For the USB and UART approaches, only the RP2040 microcontroller is required, and the removal of the ESP32 microcontroller and its associated oscillator, switches and connectors was considered. Passive components such as resistors, capacitors and diodes associated with these circuits were neglected because of their negligible contribution to the total cost. Since the same hardware is removed in both cases, UART and USB provide the same additional estimated cost reduction, namely 4.14% (5.56\$). For the Bluetooth scenario, the removal of the RP2040 microcontroller and its associated components was considered. This configuration assumes that the ESP32 alone has sufficient computational capability to handle both Bluetooth communication and biomedical signal acquisition. The resulting additional cost reduction is estimated at 2.59% (3.48\$). Finally, the I2C approach provides the highest theoretical reduction, with an additional estimated saving of 6.73% (9.04\$).

Communication	Additional savings [\$]	Additional savings [%]	Total cost reduction [\$]	Total cost reduction [%]
I2C	9.04	6.73	42.64	31.73
UART	5.56	4.14	39.16	29.14
USB	5.56	4.14	39.16	29.14
Bluetooth	3.48	2.59	37.08	27.59

Table 6.1: Estimated cost reduction for the different smartphone integration approaches

The obtained results show that the largest cost reduction is achieved by removing the screen, battery and associated circuitry, leading to a baseline reduction of 25% of the total system cost. In comparison, the additional savings obtained by removing hardware component, at the expense of supporting only one communication, remain relatively limited, with a maximum additional reduction of only 6.73% for the I2C approach. Because of these relatively marginal gains, it was decided to use the original HealthyPi hardware without major modifications and to experimentally investigate the USB, UART and Bluetooth communication approaches. However, the I2C/SPI integration approach was not implemented, as it would require technically challenging hardware modifications on the smartphone with no guarantee of successful operation.

It is also very specific to a smartphone implementation and would make the platform not usable with other devices like a computer. The following sections therefore focus on the software modifications required to support USB communication with a smartphone and on the implementation of the UART communication. Section 7.1 provides a basic characterisation of each communication channel in term of data-rate and power consumption.

6.2 Setup of USB communication

The original implementation done by *Protocentral* only supports USB for desktop devices (Windows, Linux and Mac). Only Bluetooth is originally supported for smartphone device. To detect on which platform the application is running, Flutter gives access to internal variables like *Platform.isAndroid*. This disables the access to the USB streaming interface when using a smartphone. To allow to debug both Bluetooth and USB streaming in the app, a page dedicated to the USB streaming has been added, as shown in figure 6.5.

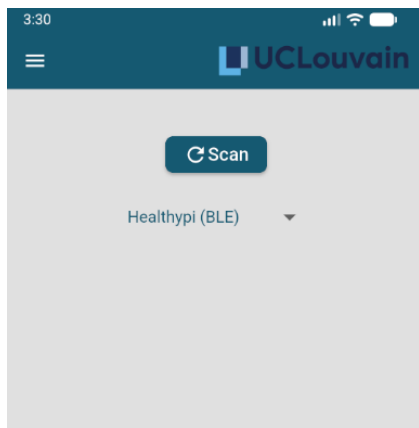


Figure 6.4: Homepage displaying available Bluetooth devices

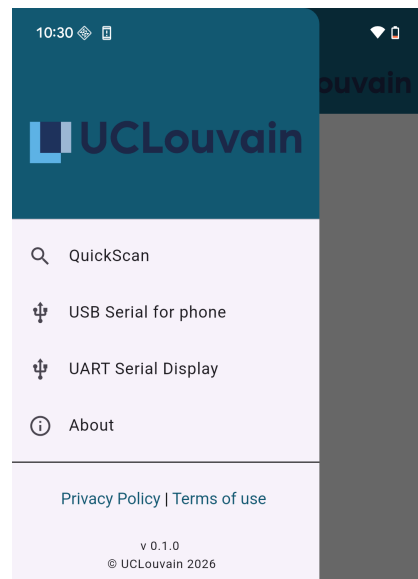


Figure 6.5: Application drawer for USB and UART streaming interface access, main page is used to access Bluetooth

The original implementation, illustrated by figure 6.6, used the library *flutter_libserialport* that allow access and communication trough USB port. It relied on the flutter library *libserialport*, which is a simple wrapper that “helps to make the libserialport Dart package work “out of the box” without the need of manually building and deploying the libserialport C-library”[142]. However, while testing on a Google Pixel 6 running Android, when the library tries to access the serial port located in */dev/*, the access is denied. Running the application as root should solve the problem; however, requiring a rooted smartphone is not desirable as it reduces accessibility, since most consumer devices are not rooted by default and rooting procedures are device-specific and technically complex. It also weakens system security by bypassing Android’s application sandboxing and

permission isolation mechanisms. A solution is to use another Flutter library, *usb_serial* design to "allow Flutter code to interact with USB serial devices connected to your Android device"[143] can be used when the detected platform is Android. Instead of trying to access `/dev/` devices, it communicates with the *Android USB Host API*. When trying to access to USB device, a permission dialog window appear so that the user grant access to the USB device. Another issue in the code base is the duplication of some part of the code, it makes the codebase harder to understand and to modify. For example, the code to display the different vital signals is present two times : once in the Bluetooth page (BLE_Page) and once in the page dedicated to USB serial communication (Serial_Page).

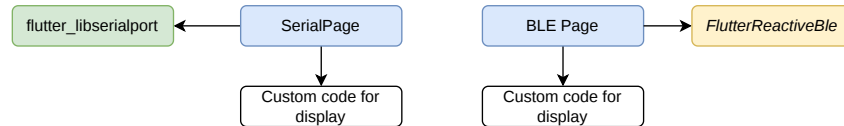


Figure 6.6: Diagram of OpenView Flutter application logic before modifications

To solve the duplication issue, the class *GenericPlot* has been added. The role of this class is to take the ECG, PPG and respiration signal as well as HR, SpO₂, RR and temperature values and to display it on the screen. This class replace the *custom code for display* code as illustrated in figure 6.6 and 6.7 and remove the duplication. Slight user interface modification was done : a button to choose which window duration of the signals should be displayed and another to set the refresh rate of the user interface.

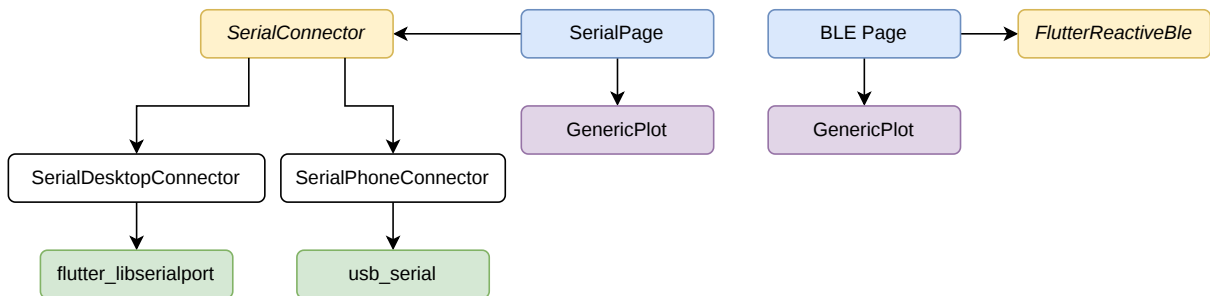


Figure 6.7: Diagram of OpenView Flutter application logic after modifications

As the library *libserialport* only support Android and to keep the app functional on desktop, an abstract class was used to describe a USB connection. "Abstract classes cannot be instantiated and abstract methods are methods that are declared without an implementation"[144], in this case the abstract class *SerialConnector* will enforce child class to have a method to handle connection and disconnection. The constructor take as argument a callback that is called when a message is received. The use of an abstract class allow the serial page to not adapt itself in function of the platform. The platform dependant implementation details are hidden in the child of *SerialConnector*, *SerialDesktopConnector* and *SerialPhoneConnector*. The separation of the platform dependant details and user interface implementation make the code more readable and maintainable.

6.3 Setup of UART communication

The objective of this section is to demonstrate UART communication between the HealthyPi board and a smartphone, in this case a Google Pixel 6. The functionality was

evaluated on two different operating systems running on the smartphone: a standard Android distribution and *Yocto*, a Linux distribution designed for embedded systems deployment across a wide range of hardware platforms[145].

6.3.1 Implementation on Google Pixel 6

Physical setup

In the case of the Google Pixel 6, it is documented that "*SideBand Use/SBU*" pins of the USB-C connector (see figure 6.8) of the smartphone can be used as UART port. SBU1 correspond to the TX pin and SBU2 to the RX pin.

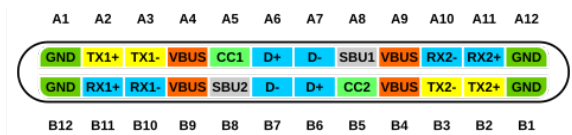


Figure 6.8: USB-C pins, extracted [146]

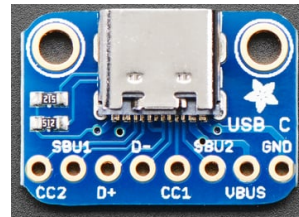


Figure 6.9: USB-C breakout board

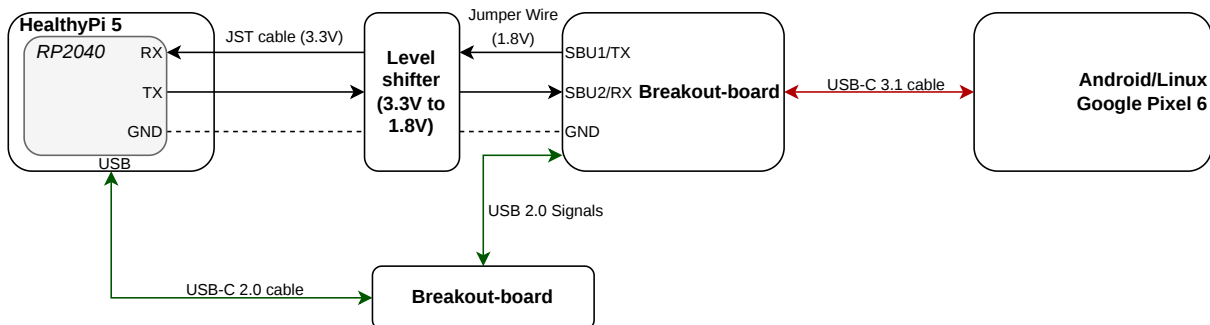


Figure 6.10: Setup for UART communication between HP5 board and Google Pixel 6 smartphone

The proposed setup for the UART communication between the HealthyPi micro-controller and the phone is described in figure 6.10. The purpose of this setup is to physically connect the RX and TX pins of the RP2040 from HealthyPi to the SBU pins on the USB-C connector of the Google Pixel 6. The following components are used, presented from the left to the right :

- A JST (Japan solderless terminal) cable compatible with one end compatible with the female JST port present on the HealthyPi board. More precisely, a 3-pins *JST SH* cable with a pitch of 1mm between the connectors is needed to be physically compatible (SH refer to this specific type of cable[147]). The other end is simply 3 female jumper wires that can be connected to any male header.
- A bi-directional level shifter to convert the different voltage. The UART port of the RP2040 has a logic level of 3.3V, while the phone has one of 1.8V. The purpose of this level shifter is to convert the digital signal to the right voltage for each device.

- 3 jumper wires to connect the level shifter to the breakout board.
- Two identical breakout board, presented in figure 6.9 are used to provide an easy access to USB 2.0 signals (V_{BUS} , GND , $D+$ and $D-$) and sideband use pins ($SBU1$ and $SBU2$). The first is used to connect the SBU pins to the Google Pixel 6. The second one take the USB 2.0 signals and connect them to the HealthyPi USB port with a regular USB-C cable. This is done to provide power to the board, the data signals are not used.
- A USB-C cable to connect the SBU pins to the Google Pixel 6. This USB-C cable need to have the SBU line to be physically present in the cable. For those reasons, a Thunderbolt 4 cable, although expensive (around 50€) was used to guarantee the presence of the SBU lines in the cable. The cost of the cable was not taken into account in table 6.1, the UART scenario is in fact the most expensive solution when the cost of the adapted USB cable is taken into account.

The setup required for UART communication is more complex and less convenient than the USB approach. The user must use a specific USB-C cable providing access to the SBU lines, which are not necessarily implemented in standard low-cost cables. In addition, a dedicated breakout board is required to physically connect the UART signals from the USB-C connector to the RP2040 UART pins. In this work, this connection is achieved using breakout boards and external wiring. However, a more integrated solution could be obtained by directly routing the RP2040 UART pins to the USB-C connector on the PCB itself. However this does not remove the cost of the specific USB cable needed.

Software details and achieved baudrate

In order to debug the communication, a command line access to the smartphone is needed. On Android, the *Android Debugger Bridge* was used. It allows to interact with the smartphone by providing an command line access, from a computer connected with USB or a wireless Wi-fi connection. In this case Wi-Fi was used as the USB connector of the smartphone is already in use in the setup. On Linux, using *gs101_wlan*¹, developed by Thomas Antoine, enable the user "to use the Wi-Fi on a Google Pixel 6 running linux". A bash script, also given by Thomas Antoine, launch a Wi-Fi access point from the smartphone. Command-line access is achieved by connecting to the Wi-Fi access point and using SSH to interact with the smartphone.

Next step is to identify which peripheral present in folder */dev/*, which "contains device files that provide access to peripheral"[148], corresponds to the UART serial interface. Looking at the device tree source can provide interesting information. The device tree source is "a data structure and language for describing hardware, it provides a description of hardware that is readable by an operating system so that the operating system doesn't need to hard code details of the machine"[149]. In this case the device tree source of the processor used by the Google Pixel 6, the GS101 chip, is accessible to everyone on android.googlesource.com. By searching the keyword "uart" in the device tree source used by Android 16 to describe the GS101², the node presented in listing 6.1

¹Github repo : https://github.com/TEhhhA/gs101_wlan

²Available at https://android.googlesource.com/kernel/devices/google/gs101/+refs/tags/android-16.0.0_r0.10/dts/gs101.dtsi#165

is found. Different information such as the address of the UART registers, buffer size and clock source are provided. The field *compatible* tells the kernel to load the driver corresponding to "samsung,exynos-uart". By looking at the different definition at the start of the corresponding driver, shown in listing 6.2³, the name of the device to access to the UART is found and defined as "ttySAC".

```

1 serial_0: uart@10A00000 {
2     compatible = "samsung,exynos-uart";
3     samsung,separate-uart-clk;
4     samsung,usi-serial-v2;
5     samsung,dbg-uart-ch;
6     samsung,dbg-uart-baud = <115200>;
7     samsung,dbg-word-len = <8>;
8     reg-io-width = <4>;
9     reg = <0x0 0x10A00000 0x100>;
10    samsung,fifo-size = <256>;
11    interrupts = <GIC_SPI IRQ_USIO_UART_PERICO IRQ_TYPE_LEVEL_HIGH>;
12    clocks = <&clock_GATE_PERICO_TOP1_USIO_UART>, <&clock
VDOUT_CLK_PERICO_USIO_UART>;
13    clock-names = "gate_uart_clk0", "ipclk_uart0";
14    status = "disabled";
15 };

```

Listing 6.1: Device tree node corresponding to the UART port

```

1 /* UART name and device definitions */
2 #define S3C24XX_SERIAL_NAME "ttySAC"
3 #define S3C24XX_SERIAL_MAJOR 204
4 #define S3C24XX_SERIAL_MINOR 64
5 /* Baudrate definition*/
6 #define MAX_BAUD 4000000
7 #define MIN_BAUD 0
8 #define DEFAULT_SOURCE_CLK 200000000

```

Listing 6.2: Definition found in driver samsung,exynos-uart code

By default the UART port is disabled, it can be enabled by booting the phone in *fastboot* mode and using the fastboot command "*fastboot oem uart enable*", note that the bootloader must be unlocked beforehand. To test the communication, a simple Arduino program was flashed on the RP2040. For this first test, the RP2040 was connected with USB to a desktop computer to check the program output. The program

- sends a message every second to phone trough the UART serial interface,
- check if data is received from the phone on the UART serial interface, if data is present, it is forwarded to the computer trough USB.

By using the default baudrate of the Google Pixel 6 of 3000000 on the Arduino, successful communication has been achieved. To display content sent by the HealthyPi, *cat* command is used and allow "*writing the content of a file or input stream to standard output*"[150]. Communication is also obtained in the other direction, from the phone to the HealthyPi board : obtained logs are accessible in the appendix in listing 3.

³Available at https://android.googlesource.com/kernel/exynos+/refs/tags/android-wear-15.0.0_r0.11/drivers/tty/serial/samsung.c#93

The command *stty* is used to configure the serial port and can be used to configure other baudrate. When testing a small baudrate compared to the default baudrate, like 9600, the phone is stuck in a loop where it keeps crashing and trying to reboot. The logs provided by the phone (see listing 4) indicates that *"a watchdog detected a hard lockup"*. A watchdog timer (or watchdog) is *"an electronic timer that is used to detect and recover from errors"*[151]. The principle is to reset the watchdog timer regularly at a given period. If the system detect that the watchdog has not been reset after the given period, it indicates a possible failure and the system takes action. In this case, the smartphone keeps restarting. There is no clear cause for this. One suggestion is that the task of writing the logs to the UART takes too much time as the baudrate is slow and trigger the watchdog. This behaviour has not been observed for baudrate of 50 000 and above.

Smartphone OS	Achieved baudrate
Android	4 000 000
Linux (Yocto distribution)	3 000 000

Table 6.2: Achieved baudrate on Linux and Android OS on Google Pixel 6

The highest achieved baudrate on each operating system is given in table 6.2. When trying to configure a baudrate higher than 4 000 000, *stty* was failing to do so : when verifying the effective configured baudrate, it was not changed. Neither *stty* nor direct *iotcl* serial port configuration was able to configure a baudrate higher than 4 000 000. By inspecting the driver code, it has been found that a variable *MAX_BAUD* is defined, as shown in listing 6.2. This variable limits the maximum baudrate accepted by the driver. The driver vary slightly in function of the branch and version of the Android project. For some version, *MAX_BAUD* is equal to 3 000 000, and for some it is equal to 4 000 000. In the driver defined in the Linux kernel⁴, the max baudrate is fixed to 3 000 000. This explain the difference observed in table 6.2. The baudrate is here limited by the software, and it is not clear whether the software limit corresponds to the actual hardware limit. No further investigation was conducted, but building a custom kernel with a modified version of the driver could be a possible approach to answer this question. However, attempting to exceed the operating conditions for which the hardware was designed could potentially lead to unstable behaviour or hardware damage.

6.3.2 OpenView modification to access UART port

The OpenView app was modified in order to display the data coming from the UART port. An additional page was added, as shown in figure 6.5. When the user goes on this page *"UartSerialPage"* the app does the following :

- Launch a sub-process that will read */dev/ttySAC0*, corresponding to the UART port, with root privileges. When an app runs a root command, Magisk intercept it and, the first time, asks the user if he wants to grant the app root access. By granting the app root access, it will be able to read the UART port.
- The standard output (stdout) of the sub-process is forwarded to the page and parsed, using the same *PacketFramer* class as the USB Page. Then data is displayed

⁴Available at https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/drivers/tty/serial/samsung_tty.c?h=v7.1-rc1

with the *GenericPlot* class. This modification highlights the benefit of a generic class to display the graph and avoid duplication in the codebase.

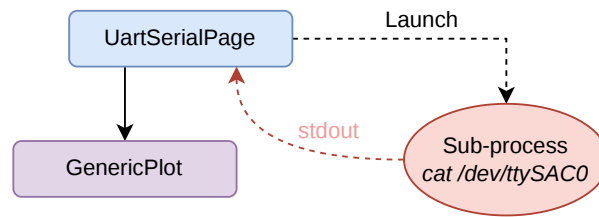


Figure 6.11: Program logic diagram for UART integration to OpenView app

On the microcontroller, a small modification was done in order to send data through the UART. An additional configuration variable was added to enable or not UART streaming. In the **sendData** function, if the configuration variable is enabled, the data is also sent through the UART port. In fact, the possibility of streaming data through UART might have been done at some point by the *Protocentral team*. Comments were present in the The function **sendData** making a call to **send_rpi_uart** to send data to the UART port. However, this function was not yet implemented but was easily done since *Zephyr* provides a function to write to the UART port. It is unclear why this portion of the code was commented out.

Those two modification allow data from the RP2040 to be transmitted using UART, and to retrieve it from the *Flutter* application to display it to the user, as shown in figure 1.

Chapter 7

Results and analysis

7.1 Characterisation of different communication channels

7.1.1 Bitrate

Bitrate was evaluated by transmitting a 10 Mbit message and measuring the time required for the transmission to complete. Synchronisation calls (function *flush*) were used to obtain reliable timing measurements. In practice, only the USB-C and UART throughput could be experimentally measured. The firmware provided by *ProtoCentral* for BLE testing could not be compiled due to a persistent missing-file error. Multiple attempts were made to resolve the issue but were unsuccessful within the timeframe of this work. However, the ESP32 documentation provides the maximum theoretical BLE throughput achievable with the ESP32 [152]. This maximum throughput depends on whether the communicating device supports the Bluetooth LE 2M PHY mode, which is a configuration that allows to double the symbol rate at the physical layer of Bluetooth, compared to 1M PHY[153]. The obtained results are presented in figure 7.1, horizontal black and blue dashed lines are the BLE theoretical bitrate.

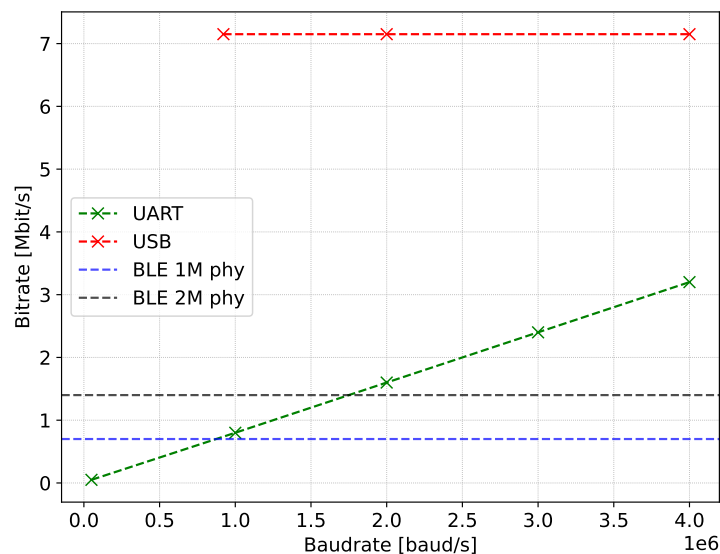


Figure 7.1: Throughput comparison of communication channel

The USB throughput remains constant regardless of the configured baudrate. This is because the baudrate parameter is only kept for legacy software compatibility and does not affect the actual USB communication speed[154]. USB therefore provides the highest throughput among the tested interfaces, reaching approximately 7.1 Mbit/s. For UART communication, the measured throughput scales linearly with the configured baudrate. The obtained values are consistent with the theoretical throughput expected for UART communication using one start bit and one stop bit per transmitted byte[140], resulting in an effective bitrate equal to 80% of the baudrate. The theoretical BLE throughput provided by the ESP32 documentation is significantly lower than both USB and high baudrate UART communication. In all tested configurations, provided that a sufficiently high UART baudrate is selected, the available communication bitrate largely exceeds the requirements of the HealthyPi. Table 7.1 presents the bitrate required both for the current configuration and for the maximum sampling rates supported by the AFEs. The sampling rates selected for temperature, SpO₂, RR, and HR were arbitrarily limited to 5 Hz, as higher update frequencies do not appear beneficial for these slowly varying physiological parameters. The obtained bitrates are lower than 25 kbits/s, which is significantly lower than bitrate obtained for all communication channel.

Channel	Width (bits)	Actual rate (Hz)	Max rate (Hz)
PPG	22	125	500
ECG	18	125	512
Resp. WF	20	62.5	64
Temp.	8	1	5
SpO ₂	8	1	5
RR	8	1	5
HR	8	1	5
Total bitrate		6282 bit/s	21656 bit/s

Table 7.1: Actual and theoretical maximum required bitrate between Google Pixel 6 and HP5

7.1.2 Power consumption

Setup

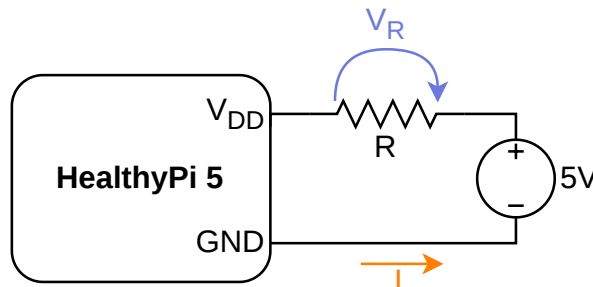


Figure 7.2: Setup for power consumption measurements

The power consumption of the system was measured using a resistor placed in series with the power supply, as shown in figure 7.2. The current was first measured with a multimeter in order to obtain an order of magnitude of the average current

consumption. The measured current was approximately 80 mA, with board in BLE mode. The value of the series resistor must be sufficiently large to produce a measurable voltage with an oscilloscope, in this case a *PicoScope 2000 Series*, while remaining low enough to avoid an excessive supply voltage drop. A resistor of $R = 1\Omega$ was used, although its effective resistance measured with a multimeter was approximately 0.7Ω . The expected voltage drop is around 56mV. The current consumed by the system is $I = \frac{V_R}{R}$, with V_R the voltage drop across the resistance. The power consumption of the system is given by $P = V_{DD} \cdot I$, with V_{DD} the supply voltage.

Only the power consumption of the HP5 board, without the smartphone, was measured. The power consumption of the complete system, including the smartphone and with the board supplied by the smartphone, was estimated using binary programs available in the `/sys/class/power_supply/` directory on the smartphone, which provide real-time battery current measurements. These programs provide both average and instantaneous current values. However, the reported measurements fluctuated rapidly, even when the system configuration remained unchanged. These fluctuations can be explained by the dynamic power-management algorithms implemented in the smartphone[155]. Background tasks, wireless activity, CPU frequency scaling and battery-management algorithms continuously modify the current consumption, even when the tested smartphone and HP5 configuration remain unchanged. As a result, obtaining stable and reproducible power measurements directly from the smartphone software interface proved difficult.

Results

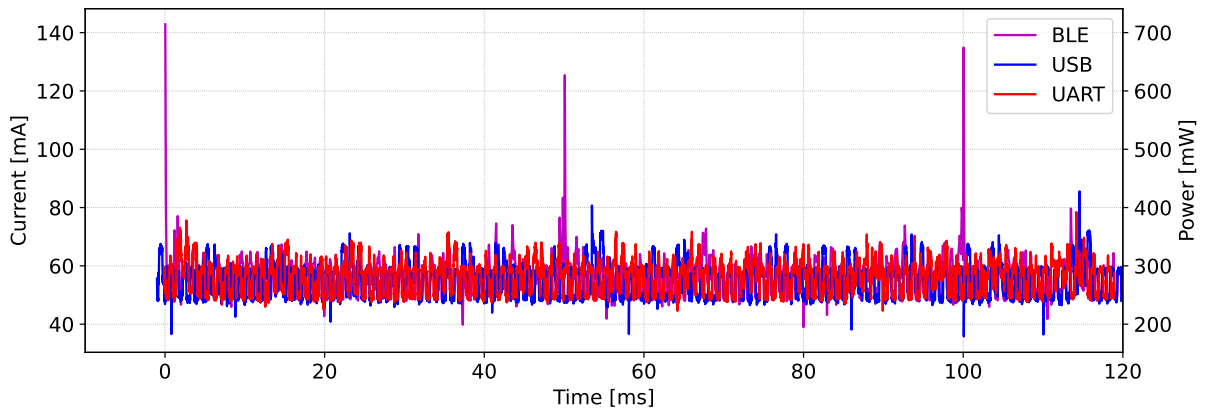


Figure 7.3: Current and power consumption of HP5 with BLE, USB and UART communication

The obtained current/power trace for each communication channel is shown in figure 7.3. BLE consumption trace clearly shows that the BLE radio is activated periodically every 50 ms as a peak is observed at this precise interval. This interval is defined by the Bluetooth stack, which schedules data transfers every 30 to 60 ms interval depending on the negotiated notification interval between both devices. Despite these peaks, the average current consumption remains very close to the other communication methods. The measured mean currents are 54.82 mA for USB communication, 56.06 mA for UART communication, and 56.19 mA for BLE communication. These results

indicate that the additional consumption introduced by BLE communication is almost negligible compared to the baseline power consumption of the system. The low amount of data transferred during each BLE notification is small and, as a result, the radio is only activated for a very short duration during each transmission interval. The mean current is therefore almost not impacted. However, the current measured using the multimeter, approximately 80 mA, is significantly higher than the mean current obtained using the shunt resistance measurement method, which is around 55 mA. This discrepancy may result from the limited precision of the multimeter, as well as possible inaccuracies or errors in the shunt resistance measurement setup.

7.2 System performance analysis

The validation of the system was made possible through a collaboration with the Cliniques Universitaires Saint-Luc. This collaboration enabled the characterisation of the prototype in a controlled environment by providing access to a clinical multi-parameter monitor used in intensive care, which served as a reference device for performance assessment. A vital signs simulator was also made available, allowing the generation of controlled physiological signals with well-defined parameters.

7.2.1 Validation methodology

Reference device

The reference monitor used for the validation is a **Philips IntelliVue MP70** multi-parameter monitor. The IntelliVue MP70 is a clinical monitoring system "*optimized for the surgical, cardiac, medical and neonatal care environments*"[156]. It provides continuous monitoring of multiple physiological parameters such as ECG, HR, SpO₂, RR, invasive and non-invasive blood pressure, and body temperature. The monitor features a modular architecture allowing the integration of additional measurement modules and can display multiple physiological waveforms simultaneously on a touchscreen interface. The *IntelliVue* series also features networking capabilities, alarm management system and advanced clinical monitoring functionalities [156].

Vital signs simulator

The vital signs simulator used during the validation phase was the **ProSim 8**, manufactured by *Fluke Biomedical* and illustrated in figure 7.4. The ProSim 8 is a portable multi-parameter patient simulator designed for testing and validating physiological monitoring equipment. It is commonly used in hospitals and biomedical engineering departments for the maintenance and performance verification of multi-parameter patient monitors. It is capable of generating a wide range of physiological signals, including ECG waveforms, respiration signals, invasive blood pressure signals, temperature measurements and SpO₂ simulations. The device also supports advanced cardiac simulations such as arrhythmias, pacemaker waveforms and ECG performance testing functions [157]. The simulator allows controlled adjustment of several physiological parameters including HR, ECG amplitude, RR and waveform characteristics[157], enabling reproducible testing conditions for medical monitoring devices. In the context of this work, the ProSim 8 was used to generate controlled ECG, PPG and respiration

signals with well defined vital parameter values in order to evaluate the acquisition and estimation capabilities of the HP5 platform under reproducible experimental conditions.

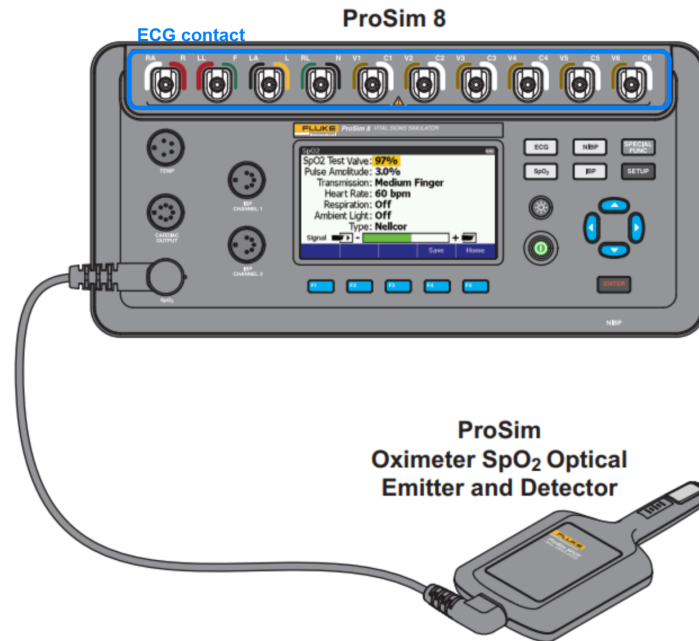


Figure 7.4: Prosim 8 : ECG electrodes are placed on ECG contact and optical emitter and detector emulates the finger for PPG and SpO₂ testing, extracted from [157]

Validation procedure

The validation procedure consisted in generating controlled physiological signals using the Fluke Biomedical ProSim 8 simulator and comparing the measurements obtained from the HP5 platform with those displayed by the Philips IntelliVue MP70 reference monitor. The simulator was connected separately to each system during the experiments. Simultaneous connection was avoided because respiration measurements rely on thoracic impedance estimation through high-frequency current injection, which could have introduced interference between the two monitoring devices.

Two types of validation were performed. First, the acquired raw physiological signals, namely ECG, PPG and respiration waveforms, were visually inspected to assess signal quality, waveform consistency and overall acquisition stability. Second, HR, RR and SpO₂ value were evaluated and compared with either the simulator configuration or the measurements displayed by the reference monitor, depending on the parameter under investigation. Temperature measurement reliability was not tested.

7.2.2 Signal acquisition

The raw signal was assessed qualitatively. The raw data from the reference monitor was not acquired. All data from the monitor is sent to a central server which stores the data. However to have those data sent to the server, it is required to have a patient file in the hospital. This was not done as it was requiring administrative procedure. However the waveform of the acquired signal can be compared to the display of the

reference multi-parameter monitor.

The vital sign simulator was configured with the settings presented in table 7.2. Each parameter defines a characteristic of the ECG, PPG or respiration signal. The *ProSim 8* provides pre-defined simulation, each one correspond to a patient type with specific pathology or condition, such as normal, tachycardia, bradycardia, heart attack, etc. For the signal acquisition test, the simulation parameters present in table 7.2 correspond to the normal and healthy patient pre-defined simulation.

Signal	Parameter	Value	Description
ECG	Waveform	NSR ^a (Adult)	Type of ECG waveform
	HR	60 bpm	Simulated cardiac frequency
	Axis	Intermediate	Intermediate electrical heart axis
	ST deviation	OFF	No ST-segment deviation applied
	Amplitude	0.65 mV	ECG peak amplitude
	Artifact type	OFF	Type of signal artefact added
Respiration	Waveform	Normal	Respiration waveform type
	Rate	20 rpm	Respiration frequency
	Ratio	1:1	Temporal ratio I:E inspiration/expiration
	Amplitude	1 Ω	Impedance variation amplitude
	Baseline	500 Ω	Baseline thoracic impedance
	Lead	LA	Respiration measurement lead
PPG	SpO ₂ value	97 %	Simulated oxygen saturation
	Pulse amplitude	3 %	Amplitude AC/DC ratio
	Transmission	Medium finger	Finger optical transmission model
	HR	60 bpm	Simulated pulse frequency
	Respiration	OFF	Respiratory modulation option
	Ambient light	Sunlight	Simulated ambient lighting condition
	Type	Nellcor	Pulse oximeter sensor emulated type/manufacturer

^a Normal sinus rhythm

Table 7.2: Configuration parameters used for the signal acquisition validation

Figure 7.5 shows the acquired signal from the modified HP5 platform. The signal values were obtained using the logging feature present in the OpenView application. The waveform can be compared with the waveform obtained with the *Philips* reference monitor, shown in figure 7.6. By visual assessment, the ECG and respiration waveform are really similar. The obtained ECG waveform is similar to the one obtained with the reference device: the QRS complex, P and T wave are well defined on both systems. The amplitude of the P and T wave seem also similar compared to the reference monitor. The time scale of the respiration waveform is different than for ECG and PPG signals on the reference monitor, it is the reason why it appears more stretched on the HP5 acquisition.

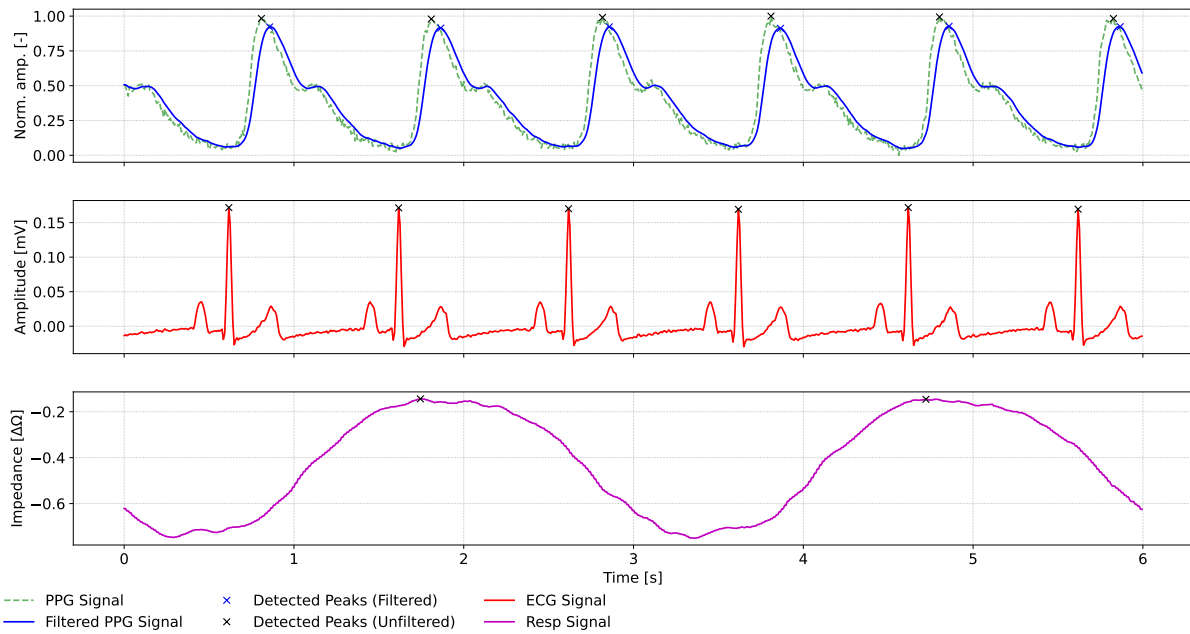


Figure 7.5: ECG, PPG and respiration signal acquisition from HP5, with additional signal processing for PPG waveform

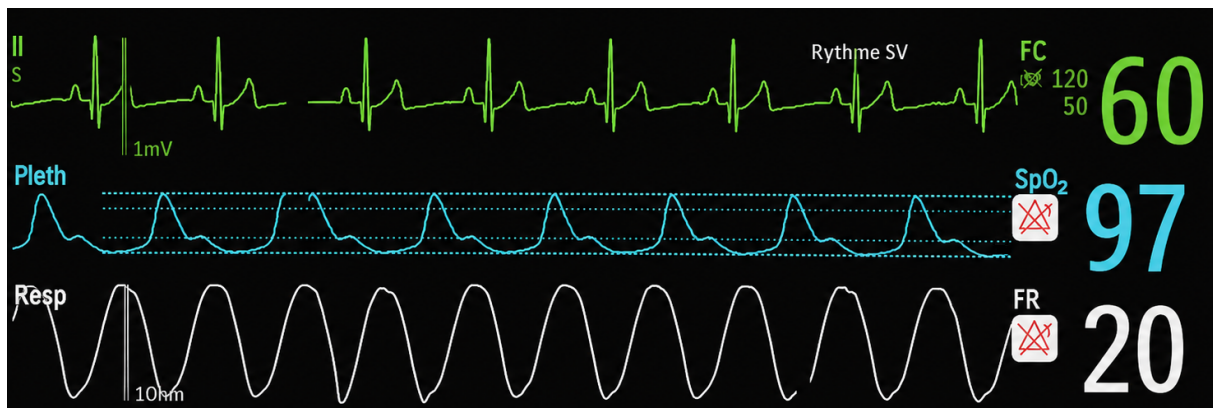


Figure 7.6: Picture of the monitor screen, taken with smartphone and background reflection removed with AI tool, original in figure 3

To obtain a similar looking PPG, some additional signal processing steps are done on the raw signal obtained from the HP5 :

- Remove the DC component, this will ensure that the signal has a positive and negative side.
- Take the inverse of the signal. The original signal is the light intensity trough the finger. The volume of blood present in peripheral tissue is the inverse : the more blood volume is present, the smaller the light intensity will be. In the HealthyPi interface, the signal must be interpreted as light intensity, which is less pertinent from a medical point of view.
- Add a low pass filter, the raw signal obtained with the simulator is noisy, as better shown by figure 7.9, especially when compared to the waveform obtained by the

reference monitor. A exponential moving filter with $\alpha = 0.3$, applied twice, was used. Both filtered and unfiltered waveform are displayed in figure 7.5 and 7.9.

The obtained waveforms are globally similar, with one noticeable difference during the diastolic phase : the diastolic peaks appears significantly more pronounced in the signal acquired with the HealthyPi. In the HealthyPi waveform, this peak reaches approximately half of the pulsatile amplitude, whereas in the reference monitor waveform it is closer one quarter of the pulsatile range.

By extracting characteristics from the acquired signals, it is possible to have quantitative measurements and assess if acquired signals are precise enough to be exploitable. Table 7.3 presents the extracted characteristics. Peaks are detected using the peak algorithm from the *scipy* Python library. For the ECG and PPG, HR is extracted for each pair of peak. Using equation 5.5, the amplitude of each peak in volt is also recovered. Using the same principle and equation 5.6, RR and impedance variation amplitude is recovered for the respiration signal. Table 7.3 contains the mean and standard deviation of all the extracted rate and amplitude. For ECG and PPG signals, 19 peaks are detected and 6 for the respiration signal during the 19.32 seconds recording.

Signal	Characteristic	Value $\mu \pm \sigma$
ECG	HR from ECG peaks	60.28 ± 0.95 bpm
	ECG amplitude	0.20 ± 0.0018 mV
PPG	HR from filtered PPG peaks	60.31 ± 0.87 bpm
	HR from unfiltered PPG peaks	60.35 ± 1.25 bpm
Respiration	RR from peaks	19.99 ± 0.16 rpm
	Impedance variation amplitude	$0.60 \pm 0.01\Omega$

Table 7.3: Extracted characteristics of the acquired ECG, PPG and respiration signals

From this table, it can be concluded that acquired signals are exploitable to extract temporal information. The computed HR and RR are exactly, if converted to an integer, the one simulated by the vital sign simulator. The filtering of the PPG signal seems to make the computation of the HR more precise and reliable : the mean HR value is closer to 60 BPM and the deviation is smaller. However the amplitude scaling of both ECG and respiration signal doesn't match the simulated amplitude. The ECG mean amplitude is 0.2 mV, which is around 3 times smaller than the simulated amplitude of 0.65 mV. For the respiration signal, the simulated impedance variation is of 1Ω and the observed one with the system is 0.6Ω . The vertical amplitude of the signal acquired by the reference monitor is much more accurate, as seen on figure 7.6, highlighted by the scale of 1mV and 1Ω .

7.2.3 Vital metrics

Heart rate

HR measurements, extracted from the PPG signal¹, were compared with the true HR simulated by the vital sign monitor. HR value between 60 and 180 bpm were simulated, by step of 10 HR beat. The HR given by the HealthyPi system is not stable : even with a

¹As the ECG was not expected to work, the HR value extracted from the ECG signal were not tested.

PPG signal with constant HR, the HR value oscillates around a certain mean value. The reference monitor doesn't have that behaviour and stabilize at the correct HR value. To have quantitative measurements of the deviation, a 1 minute video of the OpenView window was taken. HR values were recorded every 2 seconds, this give 30 HR values per simulated HR value. Results are presented in figure 7.7, containing on the x-axis the true HR value and the y axis the measured value. The dashed line represents the ideal results. Figure 7.8 display the mean and standard deviation of the error between the measured HR value and true HR value for each simulated HR value.

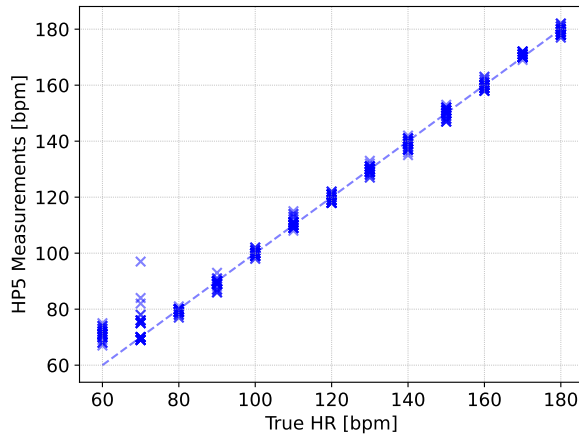


Figure 7.7: Scatter plot of measured HR in function of true HR

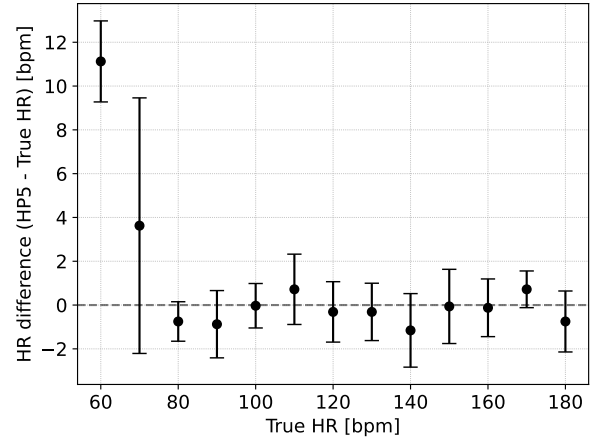


Figure 7.8: Mean and standard deviation of error for different true HR

The predicted HR values are unreliable for HRs below 80 bpm. The values estimated by the algorithm are systematically higher than the true HR. At 60 bpm, the error is particularly large, with the mean estimated value exceeding the true value by more than 10 bpm. At 70 bpm, the error becomes smaller, with a mean overestimation of approximately 4 bpm. However, the standard deviation at this HR is significantly larger than for the other measurements. This increased deviation is mainly explained by the presence of an outlier close to 100 bpm. For HR values of 80 bpm and above, the mean error remains below 1.5 bpm, with a maximum standard deviation of 1.69 bpm. No clear trend in either the bias or the standard deviation can be observed in this range. Several reasons have been identified to explain the poorer performance at lower HRs, ordered from the most impactful to the least impactful:

1. The peak detection algorithm uses a fixed minimum distance of 40 samples between detected peaks. At a sampling frequency of 125 Hz, this corresponds to 0.32 s. For HRs below approximately 80 bpm, the delay between the systolic and diastolic peaks can exceed this threshold. As illustrated in figure 7.9, the vertical dashed lines represent this 0.32 s exclusion interval after a detected peak, the diastolic peak may therefore fall outside this interval and be incorrectly detected as an independent cardiac peak, leading to an overestimation of the HR. Reducing this minimum distance is not possible, otherwise measurements close to 180 bpm would become unreliable. At 180 bpm, one cardiac cycle lasts approximately 0.33 seconds, corresponding to about 42 samples.
2. The HR estimation is performed on a buffer containing 250 samples, corresponding to 2 seconds of recording. At 60 bpm, only one complete cardiac cycle is present

in the buffer. Since the algorithm computes the median HR from all detected peaks, incorrectly detected peaks have a higher probability of influencing the final estimation.

3. The PPG waveform contains a significant amount of noise, especially when tested with the vital sign simulator. This noise can slightly alter the detected peak positions and therefore reduces the accuracy of the HR estimation. When using the identical peak detection algorithm on the filtered PPG signal, the diastolic peak is contained within the 0.32 seconds interval and is not detected as a peak, as shown in figure 7.9. This suggest that performing peak detection on filtered PPG signal could improve the reliability of the HR estimation, this was also supported by table 7.3.

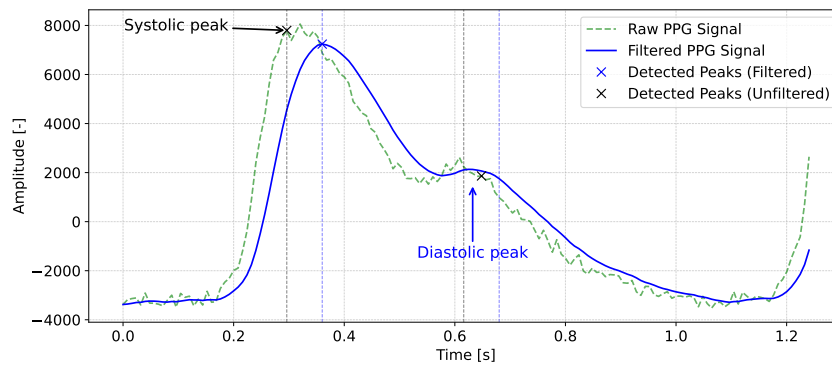


Figure 7.9: Detected PPG peaks with identical algorithm used on the HP5

This highlights that a more robust peak detection algorithm is needed. Moreover, results suggest that using the algorithm on the raw unprocessed signal is not optimal; adding low-pass and high-pass filtering would facilitate peak detection and improve the robustness of the algorithm.

Oxygen saturation level

For the assessment of the SpO_2 estimation, only four values were tested and compared with the reference monitor. Figure 7.10 presents the SpO_2 measured by the HP5 and reference monitor in function of the simulated SpO_2 value, while figure 7.11 shows the difference between both methods. The obtained results are poor, with the SpO_2 value being consistently overestimated. The first three measurements at lower SpO_2 values appear to follow an approximately proportional trend, although the last measurement does not follow this behaviour.

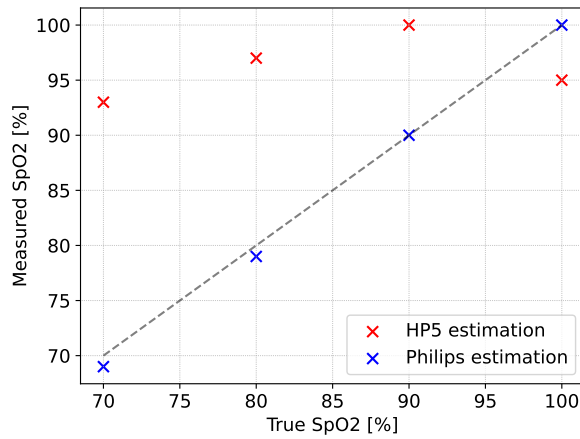


Figure 7.10: SpO₂ measurement with reference monitor and HP5

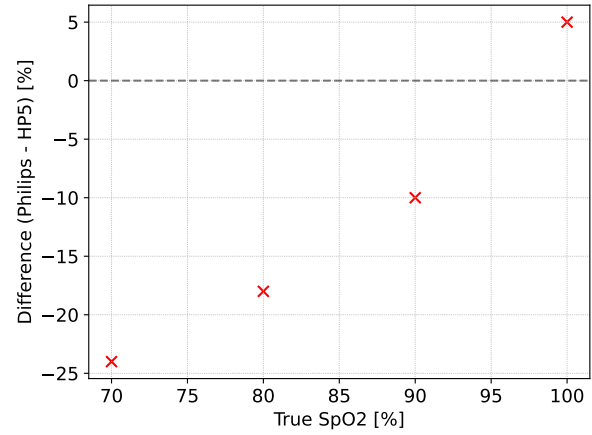


Figure 7.11: SpO₂ difference between HP5 and Philips monitor SpO₂ estimation

Several factors can explain these disappointing results. First, there is no guarantee regarding the quality or validity of the calibration curve used for the SpO₂ estimation of the HP5 system. In addition, the personnel of the *Saint-Luc clinic* pointed out a possible incompatibility between the finger simulation implemented by the **ProSim 8** and the probe used with the HP5. No exact reference for the probe could be identified; only the marking “Nellcor” and a serial number were present on the probe cable. The serial number did not provide additional information. Since the probe is marked as Nellcor, corresponding to the configured probe type used during the experiments (see Table 7.2), the configuration should theoretically be correct. However, there is insufficient information regarding the characteristics of the optical components used in the probe, such as the LED wavelengths and the photodiode properties. There could also be an implementation error, but no further investigation was done at this point of the work.

SpO₂ values were also tested with human finger on 2 different subjects. The observed value were at $\pm 2\%$ from the values of the reference monitor. However, since the readings are always between 94% and 100%, this error of $\pm 2\%$ is highly probable and not representative of the real performances of the SpO₂ algorithm.

Respiration rate

For the RR, the value displayed by the HealthyPi becomes stable after a short settling period following a change of RR on the vital signs simulator. Only the stabilized value was recorded. The results shown in Figure 7.12 indicate that the RR values provided by the HP are consistently overestimated. This explains why the RR value given by the system in first test seemed too high (see table 5.3). The first three data points follow an approximately linear trend, with the measured RR being systematically 1.65 times higher than the true RR value. The blue points correspond to the HP5 measurements scaled by a factor of 1.65^{-1} . After applying this correction, the first three points align closely with the true RR values. This correction does not apply to the last two points, however, because the *Data module* implements a hard threshold that saturates RR values above 60 rpm.

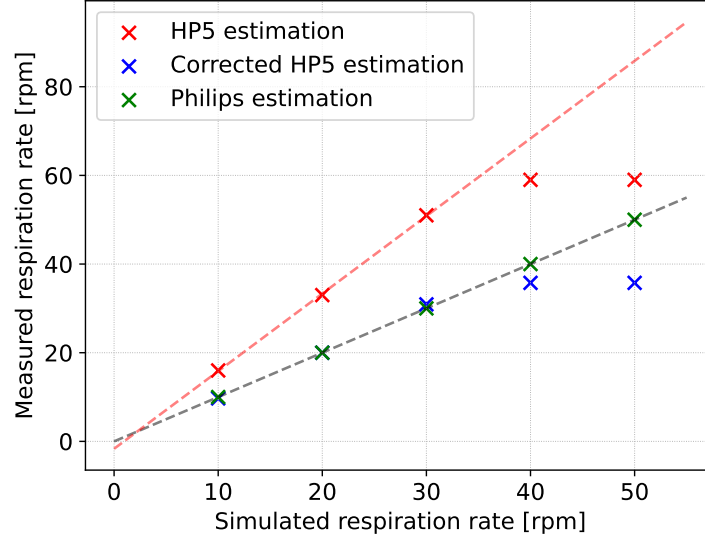


Figure 7.12: RR scatter of HP5 and Philips monitor estimation, with additional points for corrected calibration factor on HP5

The RR algorithm includes a calibration factor denoted as N_{cal} . Time estimation within the algorithm relies on the call frequency f_{cr} of the RR detection function. The calibration factor corresponds to the number of times the function is called during a 60-second interval, such that $N_{cal} = 60f_{cr}$. The RR is then estimated as:

$$RR = \frac{N_{cal}}{N_{cycle}} \quad (7.1)$$

where N_{cycle} represents the number of function calls occurring during one complete respiration cycle. In the original implementation, the calibration factor was set to 2500. However, the obtained results suggest that this value is approximately 1.65 times too high and should instead be closer to 1500. By inspecting the *Data module* responsible for calling the respiration algorithm, it can be observed that the algorithm is executed every fifth sample. Since the global system sampling rate is 125 Hz, the effective call frequency becomes $f_{cr} = 25$ Hz. This leads to $N_{cal} = 60 \cdot 25 = 1500$, which is consistent with the error observed experimentally. Although the global sampling rate of the system was modified from approximately 142 Hz to 125 Hz, this change alone does not correspond to the observed scaling factor of 1.65. Consequently, no clear explanation for the incorrect calibration of the original algorithm could be identified.

7.3 Maturity assessment of HealthyPi 5

7.3.1 Openness of HealthyPi 5

In order to evaluate the openness of project, the Open-o-meter framework, as defined earlier in section 2.1.3 was used. For each criteria, a score and its justification are detailed in table 7.4. The HealthyPi project demonstrates a relatively high level of product openness, with an Open-o-meter score of 3 out of 5 for product openness. This is notably achieved through the publication of hardware, software, and documentation under appropriate open-source licences, as well as the release of editable design files.

However, some limitations remain regarding reproducibility, particularly due to incomplete bill of materials information and the lack of assembly instructions. For components absent from the Bill of Materials (BoM), alternative ECG cables and temperature sensors can easily be identified. The case of the SpO₂ probe is more problematic as no precise reference is provided, although the purchased kit appears to use a *Nellcor*-type probe, but this conclusion cannot be drawn without buying the kit.

Process openness is more limited, with a score of 1.5 out of 3. While the project uses GitHub and provides access to version control and issue tracking, collaborative aspects such as contribution guidelines and active community management appear weak. The low activity of maintainers on the public issue tracker also suggests reduced long-term community engagement and project sustainability. In particular, a pull request² containing several modifications proposed in Chapter 5 was submitted during this work but was, to this day, not reviewed by original *Protocentral* contributors.

Criterion	Score	Justification
Product openness		
Use of relevant OS-H/OSS licences	1	Hardware is released under the <i>CERN-OHL-P v2</i> licence, software under the <i>MIT</i> licence, and documentation under the <i>Creative Commons Share-Alike 4.0 International</i> licence.
Design files publicly available	0.5	PCB design files are publicly available, however enclosure design files included in the complete HealthyPi kit are not provided.
BoM	0.5	A BoM for PCB components can be generated from the schematic, however no detailed references are provided for the ECG cable, temperature sensor, and SpO ₂ probe.
Assembly instructions available	0	No assembly instructions are provided. However, as the project mainly consists of a PCB design, assembly instructions may not be strictly necessary.
Files released in editable/original format	1	PCB design files are released in their original editable <i>Eagle</i> formats (.brd and .sch).
Process openness		
Version control system used	1	The project is hosted on GitHub and later development iterations are available through commits, although early development stages appear to have been conducted internally.
Contribution guide available	0	Documentation explains how to build firmware and flash the board, but no contribution guidelines or pull-request workflow are provided.
Issue tracking system available	0.5	GitHub Issues are used to report problems and suggest improvements. However, many issues remain unanswered, suggesting limited maintainer activity.
Total Open-o-meter score	4.5 / 8	Moderate openness with strong publication of technical resources, but limited collaborative and community-oriented processes.

Table 7.4: Open-o-meter evaluation of the HealthyPi project

7.3.2 Technical maturity

Table 7.5 presents an estimated OTRL level for each physiological parameter/signal monitoring implemented. It should be noted that reaching OTRL level 5 requires the characterisation and validation of manufacturing processes, which was outside the scope of this work. In practice, this mainly concerns PCB manufacturing and component assembly processes, which are typically handled by third-party PCB manufacturers and

²Available here : https://github.com/Protocentral/protocentral_healthypi5_zephyr/pull/60

assembly services. Nevertheless, compared to the previous HealthyPi 4 version and the previous work of Louis Breels, where only temperature acquisition was reported to operate reliably, this table highlights that a significant improvement in the functionality of the different sensing modalities has been achieved. It should also be noted that the HealthyPi 6 version is expected to be released soon and may address some of the limitations identified during this work.

Parameter acquisition	OTRL level	Justification
ECG	4	The ECG acquisition chain is functional and allows the observation of characteristic ECG features such as the R peak. However, further characterisation regarding accuracy, reliability, and the impact of electrode placement is still required.
PPG	4	The PPG acquisition is fully functional and provides stable signals. However multiple improvements are possible such as better robustness against motion artefacts and better ambient light cancellation.
Respiration waveform	4	Respiration signal acquisition is functional, although the reliability and calibration of the signal amplitude should still be improved.
HR	3	Heart-rate extraction from the PPG signal is implemented and operational, but the processing algorithms still require improvements and validation to be considered fairly reliable.
RR	4	RR estimation is functional, but additional algorithm adjustment and validation steps are required to ensure robustness and reliability.
SpO ₂	3	The theoretical principle and processing approach are consistent with methods found in the literature. However, the current implementation remains unreliable and requires significant improvements and additional investigations.
Temperature	4	Temperature acquisition is functional, but additional characterisation and validation are still necessary.

Table 7.5: Estimated OTRL level for each physiological parameter acquisition implemented on HealthyPi

7.3.3 Documentation maturity

According to the ODRL framework presented in table 2.2, the HP5 project only achieves ODRL level 1. The documentation is publicly released under the *Creative Commons Share-Alike 4.0 International* licence, satisfying the requirement of documentation publication under an open-source licence. However, the documentation process cannot be considered collaborative, as most of the documentation is hosted directly on the project website and cannot easily be modified or extended by external contributors. The currently available documentation contains a hardware overview, practical information regarding sensor connections, operating modes, and basic instructions to build and flash the firmware. However, the software architecture remains poorly documented. In particular, no detailed explanation is provided regarding the role and interaction of the different software threads, forcing contributors to perform significant reverse engineering before being able to modify or extend the firmware, as illustrated in this work.

Chapter 8

Discussion

8.1 Return of experience of HP5

This work provides a hands-on return of experience on the HP5 platform, from the perspective of an external contributor attempting to bring the system to a functional state. Overall, the HP5 represents a meaningful improvement over its predecessor: whereas the reproduction of HealthyPi 4 done in Louis Breels' master thesis [27] suffered from hardware-level errors on the PCB itself, the HP5 hardware proved to be functional. However, the software layer revealed several significant issues: critical bugs in the communication and the AFE4400 driver and configuration. This suggests the firmware was published without sufficient validation. After the fixes implemented in this work, ECG, respiration and PPG acquisition are functional, and HR and respiration rate estimation would be exploitable with algorithm adjustments. SpO₂ estimation however remains unreliable and requires further investigation. Beyond signal acquisition, significant work regarding validation, reliability assessment, risk management and regulatory compliance would still be required before any clinical deployment. The lack of software architectural documentation also remains a significant obstacle for potential new contributors.

8.2 Reduced cost

Smartphone integration enables a cost reduction of approximately 30% on the HP5 platform, depending on the communication channel chosen, as detailed in Chapter 6. More radically, the smartphone-based applications reviewed in Chapter 3 require almost no additional hardware at all, and are free to use as long as the application itself is freely available, making them the most cost-effective repurposing approach. The HP5 basic kit is available for 257€, which is already significantly lower than a medical-grade multi-parameter monitor: the *Philips IntelliVue MP70* used as reference in Chapter 7 is sold between 1750 and 4500\$ on the second-hand market¹. This cost difference is significant and shows the potential of open-source platforms to reduce hardware costs. However, this comparison must be put into perspective: in its current state, the HP5 performance is not comparable to medical and certified reference device, and significant development work would be needed to reach an equivalent level of reliability and

¹Offers available at <https://www.med-geraete.de/en/PHILIPS-IntelliVue-MP-70-Patient-monitor/1004767> and <https://sakomed.com/products/patient-monitors/philips-intellivue-mp70-patient-monitor/>

functionality. Furthermore, the cost of certification, required before an eventual clinical deployment, is not taken into account in this analysis and could alter the economic picture.

8.3 Mitigation to supply chain disruption

The contribution of smartphone repurposing to less rely on classical supply chain depends strongly on the type of application considered. For the smartphone-based applications reviewed in Chapter 3, since these rely on sensors already embedded in the device, almost no additional hardware is required. As long as the user has a smartphone meeting requirements for the use case, there is no dependency on supply chain. However, for a platform such as HP5, the situation is different. The system relies on specialised AFEs circuits, for which no direct equivalent exists. This was already experienced in Louis Breels' work [27], where component substitution on the HealthyPi 4 proved problematic. Having access to the design files does not help if some components are unavailable. In the event of a major supply chain disruption affecting specialised medical electronics or ICs in general, an open-source platform would be just as vulnerable as a commercial one.

8.4 Adaptation to local needs

Whether open-source platforms such as HP5 can be meaningfully adapted to local needs remains to be fully demonstrated, but this work provides encouraging evidence. The original project did not support USB connection to a smartphone nor UART communication; both are now functional. This illustrates that the openness of the firmware enable external contributors to add or adapt features according to their specific needs. This would not be possible with a commercial device. However, the current lack of documentation limits this potential and would need to be addressed to allow easier modification of the project.

8.5 Perspectives

Several perspectives emerge from this work:

- **HP5 enhancement:** investigate inaccurate SpO₂ readings, implement an alarm system and add blood pressure monitoring, required in general requirements of intermediate multi-parameter monitor. The upcoming HealthyPi 6 release should also be monitored closely.
- **Slimmed-down implementation:** design a minimal version of the PCB implementing only one communication channel as proposed in Chapter 6 to evaluate the achievable cost reduction in practice.
- **Validation of smartphone-based applications:** further validate the use cases reviewed in Chapter 3 on larger and more diverse populations, across multiple smartphone models and environmental conditions, in order to assess their reliability and generalizability. This could also be supported by a systematic literature review or a meta-analysis.

Chapter 9

Conclusion

This work investigated to what extent smartphone repurposing can serve as a viable platform for a more equitable healthcare, across three key objectives: cost reduction, mitigation of supply chain disruption, and adaptation to local needs.

A literature review of existing smartphone-based medical applications demonstrated that clinically relevant performance can be achieved for a range of diagnostic use cases — heart rate monitoring, SpO₂ estimation, jaundice detection, Parkinson’s disease and hearing loss screening - relying on sensors already embedded in the smartphone. These applications represent the most cost-effective repurposing approach, requiring no or little additional hardware investment. However, a recurring limitation is the lack of robustness across different smartphone models, which currently limits generalisability and clinical deployment.

In parallel, the HealthyPi 5 open-source multi-parameter monitor was selected as a case study for smartphone integration. After resolving several critical software bugs related to communication and signal acquisition configuration, ECG, respiration and PPG acquisition were brought to a functional state, and the smartphone was successfully integrated as a display, communication and power supply platform via USB and UART. A design space exploration showed that smartphone integration enables a cost reduction of approximately 30% on the platform, depending on the communication channel retained.

The results also highlight the current limitations of open-source medical platform HealthyPi: SpO₂ estimation remains unreliable, the path to certification is still long, and the lack of software architectural documentation represents a significant barrier for external contributors. Supply chain resilience, in the case of platforms relying on specialised analog frontend integrated circuits, is not improved by the open-source nature of the design.

Overall, smartphone repurposing shows genuine promise as a lever for more equitable healthcare access, but realising this potential requires sustained engineering effort, rigorous validation, and a clearer regulatory framework for open-source MDs.

Appendix

A.1 Use of AI tools

AI-based language assistance tools were used during this work and the writing of this document to improve grammar, sentence structure, readability, and overall writing clarity. These tools were also used for limited literature summarization, formatting assistance, and refinement of technical explanations originally produced by the author. The AI tools used included ChatGPT and Claude. GitHub Copilot was additionally used for code explanation and code-writing assistance related to Dart programming for the OpenView application. All experimental work, technical analyses, interpretations, and conclusions were conducted by the author.

A.2 Code

A.2.1 Communication

```
1 #include <stdio.h>
2 #include <fcntl.h>
3 #include <termios.h>
4 #include <asm/termbits.h> // for BOTHER and struct termios2
5 #include <sys/ioctl.h>
6 #include <stdlib.h>
7
8 int main(int argc, char *argv[]) {
9     if (argc != 3) {
10         printf("Usage: %s /dev/ttySAC0 <baudrate>\n", argv[0]);
11         return 1;
12     }
13
14     int fd = open(argv[1], O_RDWR | O_NOCTTY);
15     if (fd < 0) {
16         perror("open");
17         return 1;
18     }
19
20     struct termios2 tio;
21     ioctl(fd, TCGETS2, &tio);
22
23     // Output baud
24     tio.c_cflag &= ~CBAUD;
25     tio.c_cflag |= BOTHER;
26     tio.c_ospeed = atoi(argv[2]);
27 }
```

```

28 // Input baud - uses IBSHIFT offset in c_cflag
29 tio.c_cflag &= ~(CBAUD << IBSHIFT); // clear input baud
30 tio.c_cflag |= (BOTHER << IBSHIFT); // set input to BOTHER
31 tio.c_ispeed = atoi(argv[2]);
32
33 if (ioctl(fd, TCSETS2, &tio) < 0) {
34     perror("ioctl TCSETS2");
35     return 1;
36 }
37
38 printf("Baudrate set to %s\n", argv[2]);
39 return 0;
40 }

```

Listing 1: C program to modify the baudrate of the UART interface using ioctl system call

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.signal import butter, filtfilt
4
5 fs = 16000
6 T = 4
7 t = np.linspace(0,T, fs*T*10)
8
9 baseline_imp = 500
10 rpm = 30
11 resp_imp = np.sin(2 * np.pi * rpm/60 * t)
12 thorax_imp = baseline_imp + resp_imp # thorax impedance, which is the
    sum of baseline and respiratory component
13
14 #injection of high freq current
15 f_mod = 8000 #modulation frequency in Hz
16 mod_signal = np.sign(np.sin(2 * np.pi * f_mod * t))
17 injected_current = mod_signal * 10e-6 #10 uA
18
19 #ecg signal simulation
20 file = "20rpm_60bpm_ppg\ecg.csv"
21 ecg_raw = np.loadtxt(file, skiprows=1)
22 fs_ecg = 125
23 t_ecg = np.arange(0, (len(ecg_raw))/fs_ecg, 1/fs_ecg)
24 ecg_signal = np.interp(t, t_ecg, ecg_raw) #interpolate ECG signal to
    match the time vector of the respiration signal
25 #normalise from 0 to 1mV
26 ecg_signal = (ecg_signal - np.min(ecg_signal)) / (np.max(ecg_signal) -
    np.min(ecg_signal)) * 2e-3
27
28 #Measured voltage : content of electrical signal of heart + impedance
    modulated by respiration
29 measured_v = thorax_imp * injected_current + ecg_signal #measurements
    done with electrode
30
31 #Respiration signal extraction
32 gain = 1/10e-6 #gain of the system to convert voltage back to impedance
33 demodulated = measured_v * mod_signal * gain
34 cutoff = 5 #cutoff frequency in Hz
35 b, a = butter(4, cutoff/(0.5*fs), btype='low')
36 resp_signal = filtfilt(b, a, demodulated)
37

```

```

38 #ECG signal extraction
39 cutoff_ecg = 40 #cutoff frequency for ECG in Hz
40 b_ecg, a_ecg = butter(4, cutoff_ecg/(0.5*fs*10), btype='low')
41 ecg_extracted = filtfilt(b_ecg, a_ecg, measured_v)*10e3 #some gain
42
43 plt.plot(t, thorax_imp, color='orange', label='Thorax Impedance')
44 plt.plot(t, resp_signal, color='blue', label='Respiration Signal')
45 plt.plot(t, measured_v, color='green', label='Measured Voltage')
46 plt.plot(t, ecg_signal, color='red', label='ECG Signal')
47 plt.plot(t, ecg_extracted, color='purple', label='Extracted ECG Signal')
48 plt.xlabel('Time (s)')
49 plt.ylabel('Thorax Impedance (Ohms)')
50 plt.legend()
51 plt.show()
52
53 #Fourrier analysis of measured voltage
54 plt.rcParams.update({'font.size': 13})
55
56 from scipy.fft import fft, fftfreq
57 N = len(measured_v)
58 window = np.hanning(N)
59 yf = fft(measured_v * window)/np.sum(window)
60 ecg_f = fft(ecg_signal * window)/np.sum(window)
61 xf = fftfreq(N, 1/(fs*10))
62 plt.plot(xf[:N//2], 20*np.log10(np.abs(yf[:N//2])), color='blue', label=
    '$V$ FFT')
63 plt.plot(xf[:N//2], 20*np.log10(np.abs(ecg_f[:N//2])), color='green',
    label='$V_{ECG}$ FFT', linewidth=0.85)
64 plt.xlabel('Frequency (Hz)')
65 plt.ylabel('Magnitute (dB)')
66 plt.grid(which='both', linestyle='--', alpha=0.5)
67 plt.legend()
68 plt.annotate(
69     '',
70     xy=(70, -200),
71     xytext=(0.5, -200),
72     arrowprops=dict(arrowstyle='<->', color='black', lw=2),
73 )
74 plt.text(0.6, -190, 'ECG bandwidth', color='black')
75 plt.xscale('log')
76 plt.savefig('graphs/frequency_spectrum.pdf', bbox_inches='tight')
77 plt.show()

```

Listing 2: Python code for simulation of impedance and ECG measurements, without noise

A.3 Logs from Google Pixel 6 UART

```

1 [ 157.785344][ T1078] [15:53:21.327437][dhd][wlan]dhd_plat_l1ss_ctrl:
   Control L1ss RC side 1 ret:0
2 [ 157.890901][      C0] hrtimer: interrupt took 331665 ns
3 [ 158.167139][ T295] exynos-decon[0]: update decon bts for mode: 1080
   x2400x90(0:252747)(bts fps:90 mode:90 op:0)
4 [ 158.957574][ T1078] [15:53:22.499662][dhd][wlan]dhd_plat_l1ss_ctrl:
   Control L1ss RC side 0 ret:0

```

```

5 [ 159.053381][ T295] exynos-decon[0]: update decon bts for mode: 1080
  x2400x90(0:252747)(bts fps:90 mode:90 op:0)
6 [ 159.611171][ T295] exynos-decon[0]: update decon bts for mode: 1080
  x2400x60(0:168498)(bts fps:60 mode:60 op:0)
7 [ 159.888565][ T1954] alsa: alsa-aoc cmd [output] id 0x00c9, size 10,
  cntr 84
8 [ 159.891729][ T1954] cs35l41 spi13.0: main amp event 8
9 [ 159.894128][ T1954] cs35l41 spi13.1: main amp event 8
10 [ 159.896360][ T1954] cs35l41 spi13.0: cs35l41_hibernate event 0x8
  hibernate state 0
11 [ 159.896645][ T1954] cs35l41 spi13.1: cs35l41_hibernate event 0x8
  hibernate state 0
12 [ 159.896942][ T1954] alsa: 5 bd[ 159.900547][ T1954] cs35l41.[p[
  159.901200][ [ 159.901965][ T1954] alsa: be_shutdown: dai TDM_0_TX
  id 0xc0000007
13 [ 160.716187][ T1078] logbuffer_pcie1: [ 175] OC Initial Status check:
  0x5FC(0xf0)
14 [ 160.716187][ T1078]
15 [ 160.717126][ T1078] exynos-pcie-rc 14520000.pcie: AFC cal mode set to
  restart
16 [ 160.749692][ T1078] logbuffer_pcie1: [ 176] D state: 0, LTSSM: 0
17 [ 160.750614][ T1078] logbuffer_pcie1: [ 177] L0(0x11)
18 [ 160.755936][ T1078] pcieh 0001:01:00.0: enabling device (0000 ->
  0002)
19 [ 160.756687][ T1078] [15:53:24.298780][dhd][wlan]dhd_plat_l1ss_ctrl:
  Control L1ss RC side 1 ret:0
20 [ 162.234712][ T1078] [15:53:25.776802][dhd][wlan]dhd_plat_l1ss_ctrl:
  Control L1ss RC side 0 ret:0
21 [ 163.051651][ T1078] logbuffer_pcie1: [ 178] OC Initial Status check:
  0x5FC(0xf0)
22 [ 163.051651][ T1078]
23 [ 163.052114][ T1078] exynos-pcie-rc 14520000.pcie: AFC cal mode set to
  restart

```

Listing 3: Logs from Google Pixel 6 through UART port

```

1 RAMDUMP_MSG.txt:
2   reset message: KP: Watchdog detected hard LOCKUP on cpu 0
3   UUID: e529b6c9-0dce-ba4b-b0c9-e2944ab01adb
4   last kernel version: 6.1.145-android14-11-g09784c5ec50a-ab14810637
5   aosp kernel version: 6.1.145-android14-11-gfa1d6308d1fe-ab14691759
6   build: google/oriole/oriole:16/CP1A.260305.018.A1/14887622:user/
  release-keys
7   RST_STAT: 0x80 - SYSTEM_SWRESET_SYSTEM
8   GSA_RESET_STATUS: 0x10 - GSA_INTERMEDIATE_RESET
9   Reboot reason: 0xbaba - Kernel PANIC
10  Reboot mode: 0x0 - Normal Boot

```

Listing 4: Logs obtained with UART baudrate of 9600, watchdog detect a hard lockup on CPU 0, the Google Pixel 6 keeps restarting indefinitely

A.4 Screenshot of OpenView application

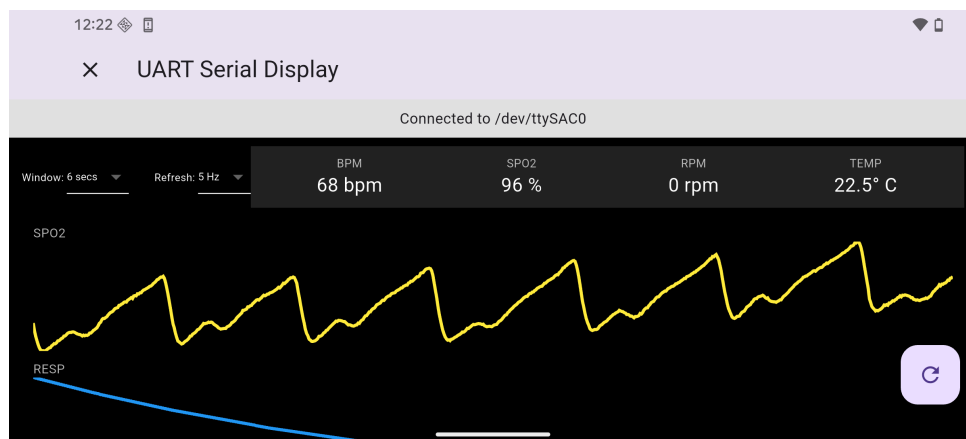


Figure 1: Screenshot of OpenView application, reading data from the UART port of the Google Pixel 6



Figure 2: Screenshot of OpenView application, with final application and firmware while tested on healthy subject

A.5 Picture of Philips IntelliVue MP70



Figure 3: Raw picture of the monitor screen, background was removed for better visualisation

A.6 Reproducible preliminary test for MAX30001 communication

At first, as the ECG only contained noise, basic test to assess the communication between the MAX30001 and RP2040 was done. By randomly touching the LA and RA probe, the signals displayed by Openview in figure 4 are obtained both on smartphone and computer. The x-axis is time and the y-axis is the amplitude of the different signal¹. When touching the probe, it is observed that the respiration trace is lower than it is not touched. The ECG has a quick decrease when fingers enter in contact (orange arrow) and a quick increase when fingers leave (purple arrow). This suggest that the communication between the MAX30001 AFE and the main controller is functional, as well as the Bluetooth connection between the board and the smartphone, and the USB connection between the board and computer.

¹No clear unit is usable for the y-axis, as no values or scale are provided

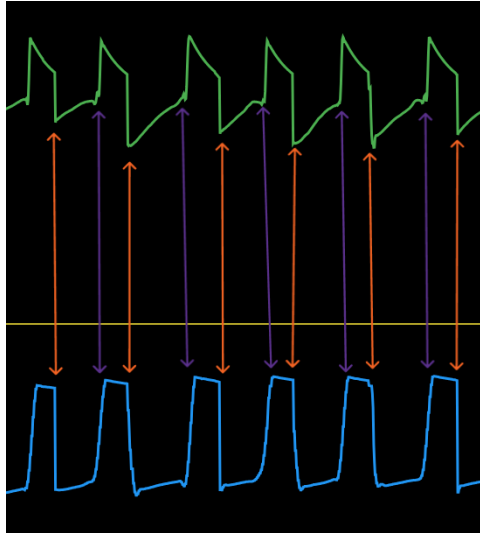


Figure 4: ECG (green) and respiration (blue) signals when randomly touching LA and RA probe. Orange arrows correspond to the timing when fingers come into contact with the probe and purples arrows when contact ends.

A.7 Firmware flash procedure

1. Connect the RP2040 USB port to the computer with a classic USB cable. Note that on the HealthyPi 5 board, two USB ports are present : one for the ESP32 and one for the RP2040. To flash the RP2040, the one connected to the RP2040 should be used, this is clearly marked on the board.
2. Hold the RP2040 boot button, located near the SD card slot and marked as "RP_BOOT" on the PCB.
3. While holding the RP2040 boot button, turn on the device by either switching the ON/OFF switch present on the board or connect the device with USB cable (in this case step 1 should have been ignored). The device should appear as a USB mass storage on the computer, named as *RPI-RP2*.
4. Copy the .UF2 file to the *RPI-RP2*, the HealthyPi will reboot and use the new firmware.

A.8 Price study information

What	Price information origin	When	Price (\$)
PCB Manufacturing	JLCPCB (medical option)	November 2025	40.00
SpO ₂ Sensor	AliExpress NL	May 2026	19.54
ECG Sensor	SparkFun	May 2026	8.75
Battery	Louis Breels Thesis	–	11.60
Screen	Amazon Belgium	May 2026	16.81
Electric Components	LCSC	November 2025	37.68

Table 1: Estimated hardware costs and suppliers

Note : Electronic components were extracted from the bill of material present in the Eagle files of the mainboard project. Prices originally listed in euros were converted to US dollars using an exchange rate of 1.16 USD/EUR.

Reference	Manufacturer	Type	All integrated	Price [€]	Availability	Interface
MAX30101	Maxim Integrated	PPG	Yes	13.73	Production	I2C
MAX30102	Maxim Integrated	PPG	Yes	13.91	Production	I2C
MAX86916	Maxim Integrated	PPG	Yes	18.90	Recommended for new design	I2C
MAXM86161	Analog Devices	PPG	Yes	8.96	Production/end of life ?	I2C
ADPD188GG	Analog Devices	PPG	Yes	12.55	Recommended for new design	I2C/SPI
AFE4462	Texas Instruments	PPG	No	9.94	New product	I2C/SPI
AFE49130	Texas Instruments	PPG and ECG	No	13.71	Active	I2C
AFE4950	Texas Instruments	PPG and ECG	No	15.56	Active	I2C/SPI
AFE4960P	Texas Instruments	PPG and ECG	No	16.98	Active	I2C/SPI
AFE4960	Texas Instruments	ECG	No	18.71	Active	I2C/SPI

Table 2: Examples of I2C-compatible AFE alternatives considered for communication between AFEs and smartphone using I2C bus. All integrated indicate if the IC contains all necessary components for PPG acquisition : a photodiode and LED to emit light. ECG AFE is necessary not "all integrated" since it requires external electrodes.

A.9 Requirements for multi-parameter monitor



Patient monitor multiparametric, advanced	
Published on: 19 November 2020 - Version: 3	
Version 2: 18/06/2014 (modification of the initial version)	
Version 1: 13/06/2012 (initial version)	
Technical Specification	Minimum requirements
General technical requirements	Continuous display on screen of patient ECG, respiration rate and heart rate, invasive and non-invasive blood pressure, body temperature, SpO₂ and CO₂ mainstream or side stream. Dynamic digital display that can show all active parameters. Unwanted parameters can be deselected from display. Operator can set audio-visual alarm levels for low or high levels of each parameter independently. Operates from mains voltage or from internal rechargeable battery. ECG patient connectors that are sterilizable and reusable are preferred. Hard copy printout of traces will not be required. • Multichannel (up to 12 leads) ECG measurement and selectable display; an extra option for simple five-lead connection preferred. • Heart rate measurement range to be at least 30–250 bpm, with accuracy better than ± 5 bpm and minimum gradation 1 bpm. • SpO₂ measurement range at least 70–99%, with accuracy better than $\pm 3\%$ and minimum gradation 1%. • Blood pressure monitoring range at least 30–270 mmHg, minimum gradation 1 mmHg. Cuff sizes neonatal/paediatric and adult. User selectable measurement intervals. • Internal pump for cuff inflation for non-invasive blood pressure measurement, with over pressure protection. • Temperature probe to be reusable, external skin contact type, but consumable to protect between patients or disinfection method explained. • Temperature range at least 30–40 °C, minimum gradation 0.1 °C. • Respiration rate measurement range at least 0–100 bpm, minimum gradation 1 bpm. • CO₂ monitoring capabilities. • Invasive blood pressure (IBP) monitoring capabilities. Automatic and programmable memory. Storage of continuous monitoring data. Trace signal velocity of at least 25 mm/s. LCD or TFT screen with: • analogue shape signals and numerical values visualization; • settable limits for the measured variables; • not less than 10° wide. Design must enable use in demanding environments, e.g. shock, vibration and free fall tests as per tests. Protections of all the functions against defibrillator discharges and electrosurgical units. Pace-maker detection. Data management functions (preferable). Capable of operating continuously in ambient temperature of 10–40 °C and relative humidity of 15–85% (90% preferable). Enclosure to have ingress protection level IPX1 or better.
Displayed parameters	Trend display of each parameter.
User adjustable settings	Alarm override and temporary silence facility to be included. Audio-visual alarms required: high and low levels for each parameter (operator variable settings), sensor/wire/probe disconnected, low battery.
Alarms	Alarm override and temporary silence facility to be included. Audio-visual alarms required: high and low levels for each parameter (operator variable settings), sensor/wire/probe disconnected, low battery, cuff leak, cuff disconnect, hose leak, inflation/deflation errors, failure to take successful reading, low battery notice. Power failure.
Consumables	ECG electrodes (if applicable): 100 sets.
Accessories	All the cables, sensors and connectors needed for full monitor functionality are to be included in the bid. Lead ECG cable: 2 per equipment. Lead ECG cable (if option offered): 2 per equipment. Sets of ECG connection electrodes (if reusable type): 5 sets. Tubes electrode gel (if required): 5 tubes. Reusable SpO₂ probes adult: 2 probes. Reusable SpO₂ probes paediatric use: 2 probes. Blood pressure – non-invasive: 3 paediatric reusable cuffs; 3 adult reusable cuffs. Blood pressure – invasive: 1 sensor for each channel offered. External skin temperature probes: 2 probes. If CO₂ mainstream technology: tube adapter: 3 per equipment; sensor: 3 per equipment. If CO₂ side stream technology: sample lines: 100 lines; water traps: 10 per equipment. Battery: 1 set.
Spare parts	1-year spare parts kit as per preventive maintenance programme including but not exclusively, sets of spare fuses (if non-resettable fuses used) and battery.
Power supply, voltage, frequency and plug vary across the countries	Operated by line electrical power supply with internal replaceable rechargeable battery backup allows operation for at least 1 hour in the event of power failure. Operates from AC power electric line: 100–240 V/50–60 Hz. Main power cable and plug adapted for various countries. Mains power cable length ≥ 2.5 m. Protections against over-voltage and over-current line conditions. Protection against defibrillator discharges and electrosurgical units. Automatic switch between battery and mains powered modes, when recharging or in mains power failure. The display shall show which power source is in use. Compliance with electrical standards and regulations.
Documentation requirements (English language mandatory)	Set: user and maintenance manuals, hard and soft copies, in English (mandatory) and other languages (preferable). Certificate of calibration and inspection. Troubleshooting, calibration and routine maintenance. List of all spare parts and accessories, with part numbers and contact details for parts supply. Document with contact details of manufacturer, supplier and local service agent.
Primary packaging label	Name and/or trademark of the manufacturer. Electrical power input requirements (voltage, frequency and socket type) and safety use and storage (keep away from oil, grease and petroleum-based or flammable products as well as smoking or open flames). Model or product reference. Information for particular storage conditions (temperature, pressure, light, humidity).
Standards, for the manufacturer	Certified quality management system for medical devices (e.g. ISO 13485). General quality management (e.g. ISO 9001). Application of risk management to medical devices (e.g. ISO 14971).
Regulatory approval/certification	Free sales certificate (FSC) or certificate for exportation of medical device provided by the authority in manufacturing country. Proof of regulatory compliance, as appropriate, per the product's risk classification (e.g. Food and Drug Administration [FDA] and/or Conformité Européenne [CE]).
Standards, for product performance	Compliance to the following international standards or to regional or national equivalent (including the technical tests for safety and performance from accredited laboratory or third party) for: IEC 60601-1 Medical electrical equipment – Part 1: General requirements for basic safety and essential performance. IEC 60601-1-1 Medical electrical equipment – Part 1-1: General requirements for safety – Collateral standard: Safety requirements for medical electrical systems. IEC 60601-1-2 Medical electrical equipment – Part 1-2: General requirements for basic safety and essential performance – Collateral standard: Electromagnetic compatibility – Requirements and tests. IEC 60601-1-8 General requirements for basic safety and essential performance – Collateral standard: General requirements, tests and guidance for alarm systems in medical electrical equipment and medical electrical systems). IEC 80601-2-49 Medical electrical equipment – Part 2-49: Particular requirements for the basic safety and essential performance of multifunction patient monitoring equipment. IEC 80601-2-30 Particular requirements for the basic safety and essential performance of automated non-invasive sphygmomanometer. IEC 60601-2-34 Particular requirements for the basic safety and essential performance of invasive blood pressure monitoring equipment). ISO 80601-2-55 Particular requirements for the basic safety and essential performance of respiratory gas monitors). ISO 80601-2-61 Medical electrical equipment – Part 2-61: Particular requirements for basic safety and essential performance of pulse oximeter equipment. IEC 60601-2-27 Particular requirements for the basic safety and essential performance of electrocardiographic monitoring equipment. Preferable if tested for: IEC 62366: Medical devices – Part 1: Application of usability engineering to medical devices. IEC 60068-1:2013 Environmental testing – Part 1: General and guidance. IEC 60068-2-31 Environmental testing – Part 2-31: Tests: Rough handling shocks, primarily for equipment-type specimens.
Warranty	5 years with regards efficiency and quality of the product (software upgrades included). Availability of accessories and spare parts for at least 5 years.

Figure 5: Complete general technical requirements for advanced multi-parameter monitor.

A.10 Description of SpO₂ and HR PPG-based algorithm

The different steps performed by the algorithm for HR and SpO₂ values, based on IR and red channel of PPG, are the followings :

1. Compute the mean/DC components of infrared and red channels.

2. The DC value is checked. If the value is below a certain threshold (1000 in this case), it means that the probe is physically disconnected or the contact with the connector is very poor. The algorithm stops here in this case.
3. Check that the DC values are not close to the saturation value of the ADC. If it is the case, the SpO₂ sensors is either pressed too hard or too much light is present, resulting in a saturated ADC input. The algorithm stops here in this case.
4. Same step is performed than step 2, but with a more strict threshold : if the DC value is below 10 000, a warning is triggered. Unlike previous step, if the DC value is too low, only a warning is triggered and the algorithm will continue.
5. Normalise the infrared channel by removing the DC component and invert the signal : peaks become valleys.
6. Detect location of the peaks, they will be use for heart rate estimation and estimation of AC components amplitude. If the number of valid peak is strictly below 2, the algorithm stops.
7. Compute heart rate based on the duration of each detected peak.
8. Compute AC and DC components for each peak.
9. Estimate SpO₂ values based on the AC/DC pairs, using the same R ratio explained in section 3.2.1.
10. Compute a confidence score that will be between 0 and 100%. It will penalize the following events : low perfusion index (AC/DC ratio for each color channel), few number of peaks and low AC components. The signal is considered valid if SpO₂ and HR computations have succeeded, confidence is above 40% and perfusion index is above 0.3%.

Bibliography

- [1] *Constitution of the World Health Organization*, en. Accessed: Apr. 12, 2026. [Online]. Available: <https://www.who.int/about/governance/constitution>.
- [2] D. Piaggio, R. Castaldo, M. Cinelli, S. Cinelli, A. Maccaro, and L. Pecchia, “A framework for designing medical devices resilient to low-resource settings,” en, *Globalization and Health*, vol. 17, no. 1, p. 64, Jun. 2021, ISSN: 1744-8603. DOI: 10.1186/s12992-021-00718-z. Accessed: Apr. 12, 2026. [Online]. Available: <https://doi.org/10.1186/s12992-021-00718-z>.
- [3] I. H. Marks, H. Thomas, M. Bakhet, and E. Fitzgerald, “Medical equipment donation in low-resource settings: A review of the literature and guidelines for surgery and anaesthesia in low-income and middle-income countries,” en, *BMJ Global Health*, vol. 4, no. 5, Sep. 2019, ISSN: 2059-7908. DOI: 10.1136/bmjgh-2019-001785. Accessed: Apr. 12, 2026. [Online]. Available: <https://gh.bmj.com/content/4/5/e001785>.
- [4] S. Bhaskar et al., “At the Epicenter of COVID-19—the Tragic Failure of the Global Supply Chain for Medical Supplies,” English, *Frontiers in Public Health*, vol. 8, Nov. 2020, ISSN: 2296-2565. DOI: 10.3389/fpubh.2020.562882. Accessed: Apr. 13, 2026. [Online]. Available: <https://www.frontiersin.org/journals/public-health/articles/10.3389/fpubh.2020.562882/full>.
- [5] N. Adika, *Smartphone and Social Media Usage 2025*, en-US, Jul. 2025. Accessed: Apr. 12, 2026. [Online]. Available: <https://www.geopoll.com/blog/smartphone-and-social-media-usage-2025/>.
- [6] *The Open Source Definition*, en-US, Jul. 2006. Accessed: May 20, 2026. [Online]. Available: <https://opensource.org/osd>.
- [7] J. Bonvoisin, R. Mies, J.-F. Boujut, and R. Stark, “What is the “Source” of Open Source Hardware?” en, *Journal of Open Hardware*, vol. 1, no. 1, Sep. 2017, ISSN: 2514-1708. DOI: 10.5334/joh.7. Accessed: May 26, 2026. [Online]. Available: <https://ojs.lib.uwo.ca/index.php/openhardware/article/view/17830>.
- [8] R. Farré, D. Gozal, V.-N. Nguyen, J. M. Pearce, and A. T. Dinh-Xuan, “Open-Source Hardware May Address the Shortage in Medical Devices for Patients with Low-Income and Chronic Respiratory Diseases in Low-Resource Countries,” en, *Journal of Personalized Medicine*, vol. 12, no. 9, p. 1498, Sep. 2022, ISSN: 2075-4426. DOI: 10.3390/jpm12091498. Accessed: May 20, 2026. [Online]. Available: <https://www.mdpi.com/2075-4426/12/9/1498>.
- [9] *Open Source Hardware Definition*, en. Accessed: May 20, 2026. [Online]. Available: <https://oshwa.org/resources/open-source-hardware-definition/>.

- [10] *Welcome | FabLabs*, en. Accessed: May 26, 2026. [Online]. Available: <https://www.fablabs.io/>.
- [11] U. Maquet, "Design and optimization of an open-source, stepper motor-driven syringe pump with potential for implementation in low-resource environments," eng, 2025. Accessed: May 21, 2026. [Online]. Available: <https://hdl.handle.net/2078.2/46245>.
- [12] R. Antoniou, J. Bonvoisin, P.-Y. Hsing, E. Dekoninck, and D. Defazio, "Defining success in open source hardware development projects: A survey of practitioners," en, *Design Science*, vol. 8, e8, Jan. 2022, ISSN: 2053-4701. DOI: 10.1017/dsj.2021.30. Accessed: May 21, 2026. [Online]. Available: <https://www.cambridge.org/core/journals/design-science/article/defining-success-in-open-source-hardware-development-projects-a-survey-of-practitioners/6DD6C830713A518971CA8AAC1458D3A0#sec26>.
- [13] J. Bonvoisin and R. Mies, "Measuring Openness in Open Source Hardware with the Open-o-Meter: 6th CIRPe Global Web Conference," *Procedia CIRP*, vol. 78, pp. 388–393, Nov. 2018, ISSN: 2212-8271. DOI: 10.1016/j.procir.2018.08.306. Accessed: May 21, 2026. [Online]. Available: <https://www.scopus.com/pages/publications/85059879300>.
- [14] *L'échelle TRL*, fr. Accessed: May 21, 2026. [Online]. Available: <https://www.horizon-europe.gouv.fr/l-echelle-trl-38806>.
- [15] R. Mies, M. Häuer, and M. Hassan, "Introducing readiness scales for effective reuse of open source hardware," *Procedia CIRP*, 32nd CIRP Design Conference (CIRP Design 2022) - Design in a changing world, vol. 109, pp. 635–640, Jan. 2022, ISSN: 2212-8271. DOI: 10.1016/j.procir.2022.05.306. Accessed: May 21, 2026. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2212827122007557>.
- [16] MassimoP, *MDR - Article 2 - Definitions*, en-GB, Jul. 2019. Accessed: May 23, 2026. [Online]. Available: <https://www.medical-device-regulation.eu/2019/07/10/mdr-article-2-definitions/>.
- [17] *Medical devices: Managing the mismatch: An outcome of the priority medical devices project*. Accessed: May 24, 2026. [Online]. Available: <https://iris.who.int/items/23abe7fb-30e0-42ee-9cd2-2d6ebc429285>.
- [18] *Medical device classification*. Accessed: May 24, 2026. [Online]. Available: <https://webgate.ec.europa.eu/udi-helpdesk/en/other-relevant-information/medical-device-classification.html>.
- [19] *ISO 13485 : Son importance et ses exigences*, fr. Accessed: May 26, 2026. [Online]. Available: <https://safetyculture.com/fr/themes/iso-13485>.
- [20] *Recognition of EN ISO 14971 as a harmonized standard in support of the European Medical Device Regulations*. Accessed: May 26, 2026. [Online]. Available: <https://committee.iso.org/sites/tc210/home/news/content-left-area/news-and-updates/recognition-of-en-iso-14971-as-a.html>.
- [21] *What is the ISO (International Organization for Standardization)?* en. Accessed: May 26, 2026. [Online]. Available: <https://www.techtarget.com/searchdatacenter/definition/ISO>.

- [22] *ISO 14971:2019*, fr. Accessed: May 26, 2026. [Online]. Available: <https://www.iso.org/fr/standard/72704.html>.
- [23] *Who we are*, en. Accessed: May 26, 2026. [Online]. Available: <https://iec.ch/who-we-are>.
- [24] R. Sriram, *Medical Device Design: The Essential, Step-by-Step Guide*, en-US, Mar. 2021. Accessed: May 26, 2026. [Online]. Available: <https://www.kolabtree.com/blog/medical-device-design-the-essential-step-by-step-guide/>.
- [25] A. Ahluwalia, C. De Maria, and A. Díaz Lantada, Eds., *Engineering Open-Source Medical Devices: A Reliable Approach for Safe, Sustainable and Accessible Healthcare*, en. Cham: Springer International Publishing, 2022, ISBN: 978-3-030-79362-3 978-3-030-79363-0. DOI: 10.1007/978-3-030-79363-0. Accessed: May 27, 2026. [Online]. Available: <https://link.springer.com/10.1007/978-3-030-79363-0>.
- [26] P. D. C. Johner, *Open-source software as a medical device?* en-US, Dec. 2025. Accessed: May 27, 2026. [Online]. Available: <https://blog.johner-institute.com/regulatory-affairs/open-source-software-as-a-medical-device/>.
- [27] L. Breels, "Study of critical medical devices towards an open-source implementation," eng, 2023. Accessed: May 19, 2026. [Online]. Available: <https://hdl.handle.net/2078.2/32358>.
- [28] J. M. Pearce, "Distributed Manufacturing of Open Source Medical Hardware for Pandemics," en, *Journal of Manufacturing and Materials Processing*, vol. 4, no. 2, p. 49, Jun. 2020, ISSN: 2504-4494. DOI: 10.3390/jmmp4020049. Accessed: May 27, 2026. [Online]. Available: <https://www.mdpi.com/2504-4494/4/2/49>.
- [29] *Heart arrhythmia - Symptoms and causes*, en. Accessed: Apr. 7, 2026. [Online]. Available: <https://www.mayoclinic.org/diseases-conditions/heart-arrhythmia/symptoms-causes/syc-20350668>.
- [30] *Heart rhythm problems (arrhythmia)*, en, Section: conditions, Feb. 2018. Accessed: Apr. 7, 2026. [Online]. Available: <https://www.nhs.uk/conditions/arrhythmia/>.
- [31] H. J. Hardon, Y. N. V. Kerkhof, B. Bartelds, J. A. E. Kammeraad, and A. W. V. Deutekom, "The Validation and Accuracy of Wearable Heart Rate Trackers in Children With Heart Disease: Prospective Cohort Study," EN, *JMIR Formative Research*, vol. 9, no. 1, e70835, Sep. 2025. DOI: 10.2196/70835. Accessed: Apr. 7, 2026. [Online]. Available: <https://formative.jmir.org/2025/1/e70835>.
- [32] J. Allen, "Photoplethysmography and its application in clinical physiological measurement," eng, *Physiological Measurement*, vol. 28, no. 3, R1–39, Mar. 2007, ISSN: 0967-3334. DOI: 10.1088/0967-3334/28/3/R01.
- [33] G. Quer, P. Gouda, M. Galarnyk, E. J. Topol, and S. R. Steinhubl, "Inter- and intraindividual variability in daily resting heart rate and its associations with age, sex, sleep, BMI, and time of year: Retrospective, longitudinal cohort study of 92,457 adults," *PLoS ONE*, vol. 15, no. 2, e0227709, Feb. 2020, ISSN: 1932-6203. DOI: 10.1371/journal.pone.0227709. Accessed: Apr. 7, 2026. [Online]. Available: <https://pmc.ncbi.nlm.nih.gov/articles/PMC7001906/>.
- [34] *What Is Seismocardiography (SCG)?* en. Accessed: May 26, 2026. [Online]. Available: <https://heartscan.app/about-scg>.

- [35] *Understanding CSI | Hands-on Wireless Sensing with Wi-Fi: A Tutorial*, en. Accessed: Apr. 7, 2026. [Online]. Available: <https://tns.thss.tsinghua.edu.cn/wst/docs/pre/localhost/wst/docs/pre/>.
- [36] P. Kocheta, N. S. Bhatia, and K. Obraczka, "Pulse-Fi: A Low-Cost System for Accurate Heart Rate Monitoring Using Wi-Fi Channel State Information," in *2025 21st International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*, ISSN: 2325-2944, Jun. 2025, pp. 226–230. DOI: 10.1109/DCOSS-IoT65416.2025.00037. Accessed: Apr. 7, 2026. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/11096342>.
- [37] M. J. Gregoski et al., "Development and Validation of a Smartphone Heart Rate Acquisition Application for Health Promotion and Wellness Telehealth Applications," en, *International Journal of Telemedicine and Applications*, vol. 2012, no. 1, p. 696 324, 2012, _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1155/2012/696324>, ISSN: 1687-6423. DOI: 10.1155/2012/696324. Accessed: Apr. 7, 2026. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1155/2012/696324>.
- [38] T. Coppetti et al., "Accuracy of smartphone apps for heart rate measurement," *European Journal of Preventive Cardiology*, vol. 24, no. 12, pp. 1287–1293, Aug. 2017, ISSN: 2047-4873. DOI: 10.1177/2047487317702044. Accessed: Apr. 6, 2026. [Online]. Available: <https://doi.org/10.1177/2047487317702044>.
- [39] W. Pipitprapat, *Full article: The validation of smartphone applications for heart rate measurement*. Accessed: Apr. 7, 2026. [Online]. Available: <https://www.tandfonline.com/doi/full/10.1080/07853890.2018.1531144#d1e675>.
- [40] A. Hossein et al., "Smartphone-Derived Seismocardiography: Robust Approach for Accurate Cardiac Energy Assessment in Patients with Various Cardiovascular Conditions," *Sensors (Basel, Switzerland)*, vol. 24, no. 7, p. 2139, Mar. 2024, ISSN: 1424-8220. DOI: 10.3390/s24072139. Accessed: Apr. 7, 2026. [Online]. Available: <https://pmc.ncbi.nlm.nih.gov/articles/PMC11014030/>.
- [41] J. A. Lipponen, S. Rana, and M. P. Tarvainen, "Validation of Smartphone-Based Seismocardiography for Heart Rate Variability Analysis in the Kubios Application," en,
- [42] *Guidelines for SpO2 Measurement | Analog Devices*. Accessed: Apr. 16, 2026. [Online]. Available: <https://www.analog.com/en/resources/technical-articles/guidelines-for-spo2-measurement--maxim-integrated.html>.
- [43] M. L. Vold, U. Aasebø, T. Wilsgaard, and H. Melbye, "Low oxygen saturation and mortality in an adult cohort: The Tromsø study," *BMC Pulmonary Medicine*, vol. 15, p. 9, Feb. 2015, ISSN: 1471-2466. DOI: 10.1186/s12890-015-0003-5. Accessed: Apr. 16, 2026. [Online]. Available: <https://pmc.ncbi.nlm.nih.gov/articles/PMC4342789/>.
- [44] N. Bui et al., "PhO2: Smartphone based Blood Oxygen Level Measurement Systems using Near-IR and RED Wave-guided Light," in *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '17, New York, NY, USA: Association for Computing Machinery, Nov. 2017, pp. 1–14, ISBN: 978-1-4503-5459-2. DOI: 10.1145/3131672.3131696. Accessed: Apr. 16, 2026. [Online]. Available: <https://dl.acm.org/doi/10.1145/3131672.3131696>.

- [45] F. Lamonaca, D. L. Carnì, D. Grimaldi, A. Nastro, M. Riccio, and V. Spagnolo, "Blood oxygen saturation measurement by smartphone camera," in *2015 IEEE International Symposium on Medical Measurements and Applications (MeMeA) Proceedings*, May 2015, pp. 359–364. DOI: 10.1109/MeMeA.2015.7145228. Accessed: Apr. 16, 2026. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/7145228>.
- [46] *Beer–Lambert law*, en. Accessed: Apr. 16, 2026. [Online]. Available: https://www.wikiwand.com/en/Beer%E2%80%93Lambert_law.
- [47] M. Nitzan, A. Romem, and R. Koppel, "Pulse oximetry: Fundamentals and technology update," *Medical Devices (Auckland, N.Z.)*, vol. 7, pp. 231–239, Jul. 2014, ISSN: 1179-1470. DOI: 10.2147/MDER.S47319. Accessed: Apr. 16, 2026. [Online]. Available: <https://pmc.ncbi.nlm.nih.gov/articles/PMC4099100/>.
- [48] D. L. Carnì, D. Grimaldi, A. Nastro, V. Spagnuolo, and F. Lamonaca, "Blood oxygenation measurement by smartphone," *IEEE Instrumentation & Measurement Magazine*, vol. 20, no. 3, pp. 43–49, Jun. 2017, ISSN: 1941-0123. DOI: 10.1109/MIM.2017.7951692. Accessed: Apr. 16, 2026. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/7951692>.
- [49] Z. Sun, Q. He, Y. Li, W. Wang, and R. K. Wang, "Robust non-contact peripheral oxygenation saturation measurement using smartphone-enabled imaging photoplethysmography," *EN, Biomedical Optics Express*, vol. 12, no. 3, pp. 1746–1760, Mar. 2021, ISSN: 2156-7085. DOI: 10.1364/BOE.419268. Accessed: Apr. 18, 2026. [Online]. Available: <https://opg.optica.org/boe/abstract.cfm?uri=boe-12-3-1746>.
- [50] X. Ding, D. Nassehi, and E. C. Larson, "Measuring Oxygen Saturation With Smartphone Cameras Using Convolutional Neural Networks," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 6, pp. 2603–2610, Nov. 2019, ISSN: 2168-2208. DOI: 10.1109/JBHI.2018.2887209. Accessed: Apr. 16, 2026. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8579580>.
- [51] K. A. Reddy, B. George, N. M. Mohan, and V. J. Kumar, "A Novel Calibration-Free Method of Measurement of Oxygen Saturation in Arterial Blood," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 5, pp. 1699–1705, May 2009, ISSN: 1557-9662. DOI: 10.1109/TIM.2009.2012934. Accessed: Apr. 16, 2026. [Online]. Available: <https://ieeexplore.ieee.org/document/4781608>.
- [52] L. de Greef et al., "Bilicam: Using mobile phones to monitor newborn jaundice," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '14, New York, NY, USA: Association for Computing Machinery, Sep. 2014, pp. 331–342, ISBN: 978-1-4503-2968-2. DOI: 10.1145/2632048.2632076. Accessed: Mar. 29, 2026. [Online]. Available: <https://dl.acm.org/doi/10.1145/2632048.2632076>.
- [53] A. J. H. Ngeow et al., "Development and Validation of a Smartphone Application for Neonatal Jaundice Screening," *JAMA Network Open*, vol. 7, no. 12, e2450260, Dec. 2024, ISSN: 2574-3805. DOI: 10.1001/jamanetworkopen.2024.50260. Accessed: Mar. 29, 2026. [Online]. Available: <https://doi.org/10.1001/jamanetworkopen.2024.50260>.

- [54] G.-T. Shen, Y.-H. Liu, J.-Y. Lin, K.-C. Hsu, and H.-H. Chou, "Development and validation of neonatal jaundice detection using smartphone application," *Computers in Biology and Medicine*, vol. 196, p. 110 845, Sep. 2025, ISSN: 0010-4825. DOI: 10.1016/j.combiomed.2025.110845. Accessed: Mar. 29, 2026. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0010482525011965>.
- [55] P. Padidar et al., "Detection of Neonatal Jaundice by Using an Android OS-Based Smartphone Application," en, *Iranian Journal of Pediatrics*, Research Article 29, Sep. 2018, Issue: 2 Journal Abbreviation: Inn J Pediatr Publication Title: Iranian Journal of Pediatrics Volume: 29, e84397. DOI: 10.5812/ijp.84397. Accessed: Mar. 29, 2026. [Online]. Available: <https://brieflands.com/journals/ijp/articles/84397#abstract>.
- [56] *Bilirubin Analyzers Equipment | Review, Compare, Get Quotes (RFQ) & Prices*, en. Accessed: Apr. 5, 2026. [Online]. Available: <https://www.news-medical.net/Clinical-and-Diagnostics/Bilirubin-Analyzers>.
- [57] J. A. Taylor et al., "Use of a Smartphone App to Assess Neonatal Jaundice," *Pediatrics*, vol. 140, no. 3, e20170312, Sep. 2017, ISSN: 0031-4005. DOI: 10.1542/peds.2017-0312. Accessed: Mar. 29, 2026. [Online]. Available: <https://doi.org/10.1542/peds.2017-0312>.
- [58] A. Mariakakis, M. A. Banks, L. Phillipi, L. Yu, J. Taylor, and S. N. Patel, "BiliScreen: Smartphone-Based Scleral Jaundice Monitoring for Liver and Pancreatic Disorders," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 2, 20:1–20:26, Jun. 2017. DOI: 10.1145/3090085. Accessed: Mar. 29, 2026. [Online]. Available: <https://dl.acm.org/doi/10.1145/3090085>.
- [59] J. Prajapati, D. Das, V. Udutalapally, R. Mahapatra, and P. N. Wasnik, "jScan: Smartphone-Assisted Bilirubin Quantification and Jaundice Screening," *IEEE Sensors Journal*, vol. 23, no. 21, pp. 26 654–26 661, Nov. 2023, ISSN: 1558-1748. DOI: 10.1109/JSEN.2023.3315452. Accessed: Mar. 29, 2026. [Online]. Available: <https://ieeexplore.ieee.org/document/10258028>.
- [60] D. Giavarina, "Understanding Bland Altman analysis," eng, *Biochemia Medica*, vol. 25, no. 2, pp. 141–151, 2015, ISSN: 1330-0962. DOI: 10.11613/BM.2015.015.
- [61] *Sclera*, en, Apr. 2026. Accessed: Apr. 6, 2026. [Online]. Available: <https://dictionary.cambridge.org/dictionary/english/sclera>.
- [62] B. R. Bloem, M. S. Okun, and C. Klein, "Parkinson's disease," English, *The Lancet*, vol. 397, no. 10291, pp. 2284–2303, Jun. 2021, ISSN: 0140-6736, 1474-547X. DOI: 10.1016/S0140-6736(21)00218-X. Accessed: Feb. 24, 2026. [Online]. Available: [https://www.thelancet.com/journals/lancet/article/PIIS0140-6736\(21\)00218-X/fulltext](https://www.thelancet.com/journals/lancet/article/PIIS0140-6736(21)00218-X/fulltext).
- [63] S. Sveinbjornsdottir, "The clinical symptoms of Parkinson's disease," en, *Journal of Neurochemistry*, vol. 139, no. S1, pp. 318–324, 2016, _eprint: <https://onlinelibrary.wiley.com/doi/abs/10.1111/jnc.13691>. ISSN: 1471-4159. DOI: 10.1111/jnc.13691. Accessed: Feb. 24, 2026. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/jnc.13691>.

- [64] A. Tsanas, M. A. Little, P. E. McSharry, and L. O. Ramig, "Accurate Telemonitoring of Parkinson's Disease Progression by Noninvasive Speech Tests," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 4, pp. 884–893, Apr. 2010, ISSN: 1558-2531. DOI: 10.1109/TBME.2009.2036000. Accessed: Mar. 1, 2026. [Online]. Available: <https://ieeexplore.ieee.org/document/5339170/>.
- [65] A. Schrag, Y. Ben-Shlomo, and N. Quinn, "How valid is the clinical diagnosis of Parkinson's disease in the community?" en, *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 73, no. 5, pp. 529–534, Nov. 2002, ISSN: 0022-3050, 1468-330X. DOI: 10.1136/jnnp.73.5.529. Accessed: Mar. 1, 2026. [Online]. Available: <https://jnnp.bmj.com/content/73/5/529>.
- [66] "Unified Parkinson's Disease Rating Scale (UPDRS) and The Movement-Disorder Society Sponsored-unified Parkinson's Disease Rating Scale (MDS-UPDRS)," en-US, pp. 307–309, Jan. 2010. DOI: 10.1016/B978-0-12-374105-9.00091-5. Accessed: Mar. 1, 2026. [Online]. Available: <https://www.sciencedirect.com/science/chapter/referencework/abs/pii/B9780123741059000915>.
- [67] A. K. Ho, R. Iannsek, C. Marigliani, J. L. Bradshaw, and S. Gates, "Speech Impairment in a Large Sample of Patients with Parkinson's Disease," en, *Behavioural Neurology*, vol. 11, no. 3, p. 327 643, 1999, _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1155/1999/327643>. ISSN: 1875-8584. DOI: 10.1155/1999/327643. Accessed: Mar. 12, 2026. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1155/1999/327643>.
- [68] J. R. D. BC-ANCDs PhD, *Motor Speech Disorders: Substrates, Differential Diagnosis, and Management*, en. Elsevier Health Sciences, Sep. 2012, ISBN: 978-0-323-07200-7.
- [69] B. Harel, M. Cannizzaro, and P. J. Snyder, "Variability in fundamental frequency during speech in prodromal and incipient Parkinson's disease: A longitudinal case study," *Brain and Cognition*, vol. 56, no. 1, pp. 24–29, Oct. 2004, ISSN: 0278-2626. DOI: 10.1016/j.bandc.2004.05.002. Accessed: Mar. 12, 2026. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0278262604001393>.
- [70] J. Ruz et al., "Smartphone Allows Capture of Speech Abnormalities Associated With High Risk of Developing Parkinson's Disease," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 8, pp. 1495–1507, Aug. 2018, ISSN: 1558-0210. DOI: 10.1109/TNSRE.2018.2851787. Accessed: Mar. 23, 2026. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8400578>.
- [71] A. D'Agostino, *The Explanation You Need on Binary Classification Metrics*, en-US, Aug. 2022. Accessed: Mar. 24, 2026. [Online]. Available: <https://towardsdatascience.com/the-explanation-you-need-on-binary-classification-metrics-321d280b590f/>.
- [72] *Benchmarking performance with sensitivity, specificity, precision and recall*, en, Feb. 2024. Accessed: Mar. 24, 2026. [Online]. Available: <https://epi2me.nanoporetech.com/benchmarking-performance/>.
- [73] *Classification: ROC et AUC | Machine Learning*, fr-x-mtfrom-en. Accessed: Mar. 25, 2026. [Online]. Available: <https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc?hl=fr>.

- [74] O. Orulluoğlu, *Area Under the Curve (AUC): A Robust Performance Measure of Classification Models*, en, Aug. 2023. Accessed: Mar. 25, 2026. [Online]. Available: <https://medium.com/@bayramorkunor/area-under-the-curve-auc-a-robust-performance-measure-of-classification-models-cbfc3549d8c6>.
- [75] C. Laganas et al., "Parkinson's Disease Detection Based on Running Speech Data From Phone Calls," *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 5, pp. 1573–1584, May 2022, ISSN: 1558-2531. DOI: 10.1109/TBME.2021.3116935. Accessed: Mar. 24, 2026. [Online]. Available: <https://ieeexplore.ieee.org/document/9556632/>.
- [76] S. Singh and W. Xu, "Robust Detection of Parkinson's Disease Using Harvested Smartphone Voice Data: A Telemedicine Approach," *EN, Telemedicine and e-Health*, vol. 26, no. 3, pp. 327–334, Mar. 2020, ISSN: 1530-5627. DOI: 10.1089/tmj.2018.0271. Accessed: Mar. 24, 2026. [Online]. Available: <https://doi.org/10.1089/tmj.2018.0271>.
- [77] Y. N. Zhang, "Can a Smartphone Diagnose Parkinson Disease? A Deep Neural Network Method and Telediagnosis System Implementation," en, *Parkinson's Disease*, vol. 2017, no. 1, p. 6 209 703, 2017, _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1155/2017/6209703>. ISSN: 2042-0080. DOI: 10.1155/2017/6209703. Accessed: Mar. 24, 2026. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1155/2017/6209703>.
- [78] M. A. Motin, N. D. Pah, S. Raghav, and D. K. Kumar, "Parkinson's Disease Detection Using Smartphone Recorded Phonemes in Real World Conditions," *IEEE Access*, vol. 10, pp. 97 600–97 609, 2022, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3203973. Accessed: Mar. 24, 2026. [Online]. Available: <https://ieeexplore.ieee.org/document/9887934/>.
- [79] M. Madruga, Y. Campos-Roca, and C. J. Pérez, "Addressing smartphone mismatch in Parkinson's disease detection aid systems based on speech," *Biomedical Signal Processing and Control*, vol. 80, p. 104 281, Feb. 2023, ISSN: 1746-8094. DOI: 10.1016/j.bspc.2022.104281. Accessed: Mar. 25, 2026. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1746809422007352>.
- [80] M. Gibson, C. Plahl, P. Zhan, and G. Cook, "Multi-condition Deep Neural Network Training," en, in *Konferenz Elektronische Sprachsignalverarbeitung*, TUDpress, Dresden, Mar. 2018, pp. 77–84, ISBN: 978-3-95908-128-3. Accessed: Mar. 26, 2026. [Online]. Available: <https://www.essv.de/paper.php?id=392>.
- [81] K. De Koster, *Gait*, en. Accessed: Mar. 29, 2026. [Online]. Available: <https://www.physio-pedia.com/Gait>.
- [82] P. R. Lan Kun-chan, "Gait Monitoring for Early Neurological Disorder Detection Using Sensors in a Smartphone: Validation and a Case Study of Parkinsonism - Paweeya Raknim, Kun-chan Lan, 2016," en, *Telemedicine and e-Health*, Jan. 2016. Accessed: Mar. 29, 2026. [Online]. Available: <https://journals.sagepub.com/doi/full/10.1089/tmj.2015.0005>.

- [83] S. Arora, V. Venkataraman, S. Donohue, K. M. Biglan, E. R. Dorsey, and M. A. Little, "High accuracy discrimination of Parkinson's disease participants from healthy controls using smartphones," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, ISSN: 2379-190X, May 2014, pp. 3641–3644. DOI: 10.1109/ICASSP.2014.6854280. Accessed: Mar. 29, 2026. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/6854280>.
- [84] H. Abujrida, E. Agu, and K. Pahlavan, "Smartphone-based gait assessment to infer Parkinson's disease severity using crowdsourced data," in *2017 IEEE Healthcare Innovations and Point of Care Technologies (HI-POCT)*, Nov. 2017, pp. 208–211. DOI: 10.1109/HIC.2017.8227621. Accessed: Mar. 29, 2026. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8227621>.
- [85] S. K. Sieberts et al., "Crowdsourcing digital health measures to predict Parkinson's disease severity: The Parkinson's Disease Digital Biomarker DREAM Challenge," en, *npj Digital Medicine*, vol. 4, no. 1, p. 53, Mar. 2021, ISSN: 2398-6352. DOI: 10.1038/s41746-021-00414-7. Accessed: Mar. 29, 2026. [Online]. Available: <https://www.nature.com/articles/s41746-021-00414-7>.
- [86] T. He, J. Chen, X. Xu, G. Fortino, and W. Wang, "Early Detection of Parkinson's Disease Using Deep NeuroEnhanceNet With Smartphone Walking Recordings," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 32, pp. 3603–3614, 2024, ISSN: 1558-0210. DOI: 10.1109/TNSRE.2024.3462392. Accessed: Mar. 29, 2026. [Online]. Available: <https://ieeexplore.ieee.org/document/10681495>.
- [87] W.-S. Lim et al., "Smartphone-derived multidomain features including voice, finger-tapping movement and gait aid early identification of Parkinson's disease," en, *npj Parkinson's Disease*, vol. 11, no. 1, p. 111, May 2025, ISSN: 2373-8057. DOI: 10.1038/s41531-025-00953-w. Accessed: Mar. 29, 2026. [Online]. Available: <https://www.nature.com/articles/s41531-025-00953-w>.
- [88] *World report on hearing*, en. Accessed: May 27, 2026. [Online]. Available: <https://www.who.int/publications/i/item/9789240020481>.
- [89] G. Livingston et al., "Dementia prevention, intervention, and care: 2024 report of the Lancet standing Commission," eng, *Lancet*, vol. 404, no. 10452, pp. 572–628, Aug. 2024, ISSN: 1474-547X. DOI: 10.1016/S0140-6736(24)01296-0.
- [90] B. J. Lawrence, D. M. P. Jayakody, R. J. Bennett, R. H. Eikelboom, N. Gasson, and P. L. Friedland, "Hearing Loss and Depression in Older Adults: A Systematic Review and Meta-analysis," *The Gerontologist*, vol. 60, no. 3, e137–e154, Apr. 2020, ISSN: 0016-9013. DOI: 10.1093/geront/gnz009. Accessed: May 27, 2026. [Online]. Available: <https://doi.org/10.1093/geront/gnz009>.
- [91] C.-H. Chen et al., "Diagnostic Accuracy of Smartphone-Based Audiometry for Hearing Loss Detection: Meta-analysis," *JMIR mHealth and uHealth*, vol. 9, no. 9, e28378, Sep. 2021, ISSN: 2291-5222. DOI: 10.2196/28378. Accessed: Mar. 20, 2026. [Online]. Available: <https://pmc.ncbi.nlm.nih.gov/articles/PMC8477297/>.

- [92] I. M. M. Melo et al., "Accuracy of smartphone-based hearing screening tests: A systematic review," en, *CoDAS*, vol. 34, e20200380, 2022, ISSN: 2317-1782. DOI: <https://doi.org/10.1590/2317-1782/20212020380>. Accessed: May 27, 2026. [Online]. Available: <https://www.scielo.br/j/codas/a/F4rrJTZSp3dZNdSzznCyg5B/?lang=en>.
- [93] M. Masalski, L. Kipiński, T. Grysiński, and T. Kręcicki, "Hearing Tests on Mobile Devices: Evaluation of the Reference Sound Level by Means of Biological Calibration," EN, *Journal of Medical Internet Research*, vol. 18, no. 5, e4987, May 2016. DOI: 10.2196/jmir.4987. Accessed: May 27, 2026. [Online]. Available: <https://www.jmir.org/2016/5/e130>.
- [94] P. Limkitisupasin, N. Utoomprurkporn, P. Potipimpanon, S. Tangwongchai, and D.-e. Bamiou, "Attitudes toward smartphone-based self-hearing screening among mild cognitive impairment: A mixed method study," *Clinical eHealth*, vol. 8, pp. 314–322, Dec. 2025, ISSN: 2588-9141. DOI: 10.1016/j.ceh.2025.12.001. Accessed: May 27, 2026. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2588914125000310>.
- [95] MEDEVIS. Accessed: May 19, 2026. [Online]. Available: https://medevis.who-healthtechnologies.org/devices/C0V_050.
- [96] MEDEVIS. Accessed: May 19, 2026. [Online]. Available: <https://medevis.who-healthtechnologies.org/about>.
- [97] C. Bloer, "The role of single-use ECG leads in reducing healthcare-associated infections," eng, *British Journal of Nursing*, vol. 30, no. 11, pp. 628–633, Jun. 2021, ISSN: 2052-2819. DOI: 10.12968/bjon.2021.30.11.628.
- [98] R. J. Martis, U. R. Acharya, and H. Adeli, "Current methods in electrocardiogram characterization," eng, *Computers in Biology and Medicine*, vol. 48, pp. 133–149, May 2014, ISSN: 1879-0534. DOI: 10.1016/j.combiomed.2014.02.012.
- [99] C. Massaroni, A. Nicolò, D. Lo Presti, M. Sacchetti, S. Silvestri, and E. Schena, "Contact-Based Methods for Measuring Respiratory Rate," eng, *Sensors*, vol. 19, no. 4, p. 908, Feb. 2019, ISSN: 1424-8220. DOI: 10.3390/s19040908.
- [100] C. Scavée, *Rappels des principes fondamentaux en électrocardiographie*, fr, Section: Cardiologie, May 2018. Accessed: May 17, 2026. [Online]. Available: <https://www.louvainmedical.be/article/592>.
- [101] *Multiphysiological Parameter Patient Monitoring | Analog Devices*. Accessed: May 17, 2026. [Online]. Available: <https://www.analog.com/en/resources/technical-articles/multiphysiological-parameter-patient-monitoring.html>.
- [102] "(PDF) Cardiac arrhythmia detection using deep learning," en, *ResearchGate*, DOI: 10.1016/j.procs.2017.11.238. Accessed: May 26, 2026. [Online]. Available: https://www.researchgate.net/publication/321812830_Cardiac_arrhythmia_detection_using_deep_learning.
- [103] J. Pan and W. J. Tompkins, "A Real-Time QRS Detection Algorithm," *IEEE Transactions on Biomedical Engineering*, vol. BME-32, no. 3, pp. 230–236, Mar. 1985, ISSN: 1558-2531. DOI: 10.1109/TBME.1985.325532. Accessed: May 26, 2026. [Online]. Available: <https://ieeexplore.ieee.org/document/4122029>.

- [104] M. Z. Suboh, R. Jaafar, N. A. Nayan, N. H. Harun, and M. S. F. Mohamad, "Analysis on Four Derivative Waveforms of Photoplethysmogram (PPG) for Fiducial Point Detection," English, *Frontiers in Public Health*, vol. 10, Jun. 2022, ISSN: 2296-2565. DOI: 10.3389/fpubh.2022.920946. Accessed: May 26, 2026. [Online]. Available: <https://www.frontiersin.org/journals/public-health/articles/10.3389/fpubh.2022.920946/full>.
- [105] R. Parkes, "Rate of respiration: The forgotten vital sign," eng, *Emergency nurse: the journal of the RCN Accident and Emergency Nursing Association*, vol. 19, no. 2, 12–17, quiz 18, May 2011, ISSN: 1354-5752. DOI: 10.7748/en2011.05.19.2.12.c8504.
- [106] S. Rolfe, "The importance of respiratory rate monitoring," eng, *British Journal of Nursing*, vol. 28, no. 8, pp. 504–508, Apr. 2019, ISSN: 2052-2819. DOI: 10.12968/bjon.2019.28.8.504.
- [107] *Medical Definition of Heart rate*, en. Accessed: May 17, 2026. [Online]. Available: https://www.rxlist.com/heart_rate/definition.htm.
- [108] H. J. Eur, "Heart rate variability. Standards of measurement, physiological interpretation, and clinical use. Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology," eng, *European Heart Journal*, vol. 17, no. 3, pp. 354–381, Mar. 1996, ISSN: 0195-668X.
- [109] *Taux de saturation en oxygène o2 : Définition, mesure et normes*, fr, Feb. 2021. Accessed: May 11, 2026. [Online]. Available: <https://www.passeportsante.net/fr/Maux/examens-medicaux-operations/Fiche.aspx?doc=taux-saturation-oxygene-o2>.
- [110] W. Chen, "Thermometry and interpretation of body temperature," eng, *Biomedical Engineering Letters*, vol. 9, no. 1, pp. 3–17, Feb. 2019, ISSN: 2093-985X. DOI: 10.1007/s13534-019-00102-2.
- [111] *La tension artérielle*, fr-FR, Aug. 2016. Accessed: May 17, 2026. [Online]. Available: <https://fedecardio.org/je-m-informe/la-tension-arterielle/>.
- [112] X. Quan et al., "Advances in Non-Invasive Blood Pressure Monitoring," eng, *Sensors*, vol. 21, no. 13, p. 4273, Jun. 2021, ISSN: 1424-8220. DOI: 10.3390/s21134273.
- [113] M. S. Siobal, "Monitoring Exhaled Carbon Dioxide," eng, *Respiratory Care*, vol. 61, no. 10, pp. 1397–1416, Oct. 2016, ISSN: 1943-3654. DOI: 10.4187/respcare.04919.
- [114] B. S. Kodali, "Capnography Outside the Operating Rooms," en-US, *Anesthesiology*, vol. 118, no. 1, p. 192, Jan. 2013, ISSN: 1528-1175. DOI: 10.1097/ALN.0b013e318278c8b6. Accessed: May 19, 2026. [Online]. Available: https://journals.lww.com/anesthesiology/fulltext/2013/01000/capnography_outside_the_operating_rooms.31.aspx.
- [115] D. Hess, "Capnometry and Capnography: Technical Aspects, Physiologic Aspects, and Clinical Applications," EN, *Respiratory Care*, vol. 35, no. 6, pp. 557–576, Jun. 1990, ISSN: 0020-1324. DOI: 10.1177/194336549003500606. Accessed: May 19, 2026. [Online]. Available: <https://doi.org/10.1177/194336549003500606>.
- [116] *HealthyPi 5*, en. Accessed: May 15, 2026. [Online]. Available: <https://www.crowdsupply.com/protocentral/healthypi-5>.
- [117] S. Prototyping, *Smart Prototyping*, en. Accessed: May 22, 2026. [Online]. Available: <https://www.smart-prototyping.com/Qwiic.html>.

- [118] *Zephyr bus (zbus) — Zephyr Project Documentation*. Accessed: May 21, 2026. [Online]. Available: <https://docs.zephyrproject.org/latest/services/zbus/index.html>.
- [119] bleuio, *Bluetooth Low Energy : Indication and Notification*, en, Oct. 2021. Accessed: May 20, 2026. [Online]. Available: <https://bleuio.medium.com/bluetooth-low-energy-indication-and-notification-b4a005f7d51b>.
- [120] *HCI UART — Zephyr Project Documentation*. Accessed: May 20, 2026. [Online]. Available: https://docs.zephyrproject.org/latest/samples/bluetooth/hci_uart/README.html#bluetooth_hci_uart.
- [121] *Flutter - Build apps for any screen*, en. Accessed: May 20, 2026. [Online]. Available: <https://flutter.dev/>.
- [122] *USB 2.0 Specification | USB-IF*. Accessed: Apr. 22, 2026. [Online]. Available: <https://www.usb.org/document-library/usb-20-specification>.
- [123] Toshiba, *TCR2EF series, TCR2EE series*. Accessed: May 26, 2026. [Online]. Available: <https://www.mouser.lu/datasheet/3/105/1/12B8A68EB9C8CC971154500D2A8FC79D798.pdf>.
- [124] *Hooking up the Sensors - Protocentral Healthypi 5*. Accessed: Apr. 22, 2026. [Online]. Available: https://healthypi.protocentral.com/06-Hooking_up_the_Sensors/.
- [125] T. Instrument, *AFE4400 Integrated Analog Front-End for Heart Rate Monitors and Low-Cost Pulse Oximeters datasheet (Rev. H)*. Accessed: May 22, 2026. [Online]. Available: <https://www.ti.com/lit/ds/symlink/afe4400.pdf>.
- [126] *Two's complement*, en. Accessed: Apr. 30, 2026. [Online]. Available: https://www.wikiwand.com/en/Two's_complement.
- [127] K. Klein-Wengel, *Quick and simple digital filters for smoothing and DC removal*, en, Dec. 2022. Accessed: Apr. 30, 2026. [Online]. Available: <https://kleinembedded.com/quick-and-simple-digital-filters-for-smoothing-and-dc-removal/>.
- [128] A. A. R. Kamal, J. B. Harness, G. Irving, and A. J. Mearns, "Skin photoplethysmography — a review," *Computer Methods and Programs in Biomedicine*, vol. 28, no. 4, pp. 257–269, Apr. 1989, ISSN: 0169-2607. DOI: 10.1016/0169-2607(89)90159-4. Accessed: Apr. 30, 2026. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0169260789901594>.
- [129] T. Instrument, *Respiration Rate Measurement Using Impedance Pneumography*. Accessed: May 26, 2026. [Online]. Available: <https://www.ti.com/lit/an/sbaa181/sbaa181.pdf?ts=1779720223579>.
- [130] L. G. Tereshchenko and M. E. Josephson, "Frequency Content and Characteristics of Ventricular Conduction," *Journal of electrocardiology*, vol. 48, no. 6, pp. 933–937, 2015, ISSN: 0022-0736. DOI: 10.1016/j.jelectrocard.2015.08.034. Accessed: May 21, 2026. [Online]. Available: <https://pmc.ncbi.nlm.nih.gov/articles/PMC4624499/>.
- [131] C. S. r. l. Allisiardi, *EMI filters: Benefits of application in the medical field*, en, Nov. 2020. Accessed: May 10, 2026. [Online]. Available: <https://www.linkedin.com/pulse/articolo-2-applicazioni-medicali-problematiche-emi-leakage-fuduli/?trackingId=rFDies+QTci2YjGHnBRNGQ==>.

- [132] A. Devices, *MAX30001-Ultra-Low-Power, Single-Channel Integrated Biopotential (ECG, R-to-R, and Pace Detection) and Bioimpedance (BioZ) AFE*. Accessed: May 21, 2026. [Online]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/max30001.pdf>.
- [133] *ECG and SpO2 issues · Issue #13 · Protocentral/protocentral_healthypi5_zephyr*, en. Accessed: May 10, 2026. [Online]. Available: https://github.com/Protocentral/protocentral_healthypi5_zephyr/issues/13.
- [134] *Issues with SPO2 / HR Monitoring on HealthyPi 5 Boards · Issue #50 · Protocentral/protocentral_healthypi5_zephyr*, en. Accessed: May 10, 2026. [Online]. Available: https://github.com/Protocentral/protocentral_healthypi5_zephyr/issues/50.
- [135] *Hooking up the Sensors*, en-US, May 2026. Accessed: May 21, 2026. [Online]. Available: <https://protocentral.com/product/healthypi-5-vital-signs-monitoring-hat-kit/docs/hooking-up-the-sensors/>.
- [136] *3 Lead ECG Placement: Correct Positions with Color Diagram*, en, Nov. 2025. Accessed: May 21, 2026. [Online]. Available: <https://med-linket-corp.com/blogs/news/3-lead-ecg-placement-step-by-step>.
- [137] M. Cadogan, *ECG Lead positioning*, en-US, May 2021. Accessed: May 21, 2026. [Online]. Available: <https://litfl.com/ecg-lead-positioning/>.
- [138] *Introduction to SPI Interface | Analog Devices*. Accessed: May 17, 2026. [Online]. Available: <https://www.analog.com/en/resources/analog-dialogue/articles/introduction-to-spi-interface.html>.
- [139] *I2C Primer: What is I2C? (Part 1) | Analog Devices*. Accessed: May 17, 2026. [Online]. Available: <https://www.analog.com/en/resources/technical-articles/i2c-primer-what-is-i2c-part-1.html>.
- [140] *UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter | Analog Devices*. Accessed: May 17, 2026. [Online]. Available: <https://www.analog.com/en/resources/analog-dialogue/articles/uart-a-hardware-communication-protocol.html>.
- [141] *Serial debugging - postmarketOS Wiki*. Accessed: Apr. 20, 2026. [Online]. Available: https://wiki.postmarketos.org/wiki/Serial_debugging.
- [142] *Flutter_libserialport | Flutter package*, en-us. Accessed: Apr. 20, 2026. [Online]. Available: https://pub.dev/packages/flutter_libserialport.
- [143] *Usb_serial | Flutter package*, en-us. Accessed: Apr. 20, 2026. [Online]. Available: https://pub.dev/packages/usb_serial.
- [144] *Abstract Methods and Classes (The Java™ Tutorials > Learning the Java Language > Interfaces and Inheritance)*. Accessed: May 2, 2026. [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/IandI/abstract.html>.
- [145] *Project Overview*, en-US. Accessed: May 17, 2026. [Online]. Available: <https://www.yoctoproject.org/about/project-overview/>.
- [146] *USB Type-C*, fr, Aug. 2014. Accessed: Apr. 26, 2026. [Online]. Available: https://www.wikiwand.com/fr/USB_Type-C.

- [147] *JST SH Cable Explained: Real-World Use Cases for 1.0mm Pitch Wiring in Electronics Projects*. Accessed: May 25, 2026. [Online]. Available: <https://www.aliexpress.com/s/wiki-ssr/article/jst-sh-cable>.
- [148] *Contact us via LiveChat!* en. Accessed: Apr. 26, 2026. [Online]. Available: https://secure.livechatinc.com/customer/action/open_chat?license_id=16484265&group=0&embedded=1&widget_version=3&unique_groups=0&organization_id=9212e363-271b-470a-9203-d9aa5dda4840&use_parent_storage=1&x-region=us-south1.
- [149] *Linux and the Devicetree — The Linux Kernel documentation*. Accessed: Apr. 26, 2026. [Online]. Available: <https://docs.kernel.org/devicetree/usage-model.html>.
- [150] *Cat (Unix)*, en. Accessed: Apr. 26, 2026. [Online]. Available: [https://www.wikiwand.com/en/Cat_\(Unix\)](https://www.wikiwand.com/en/Cat_(Unix)).
- [151] C. Willis, *Watchdog Strategies Within Real time operating systems | RTOS Tutorial*, en. Accessed: Apr. 26, 2026. [Online]. Available: <https://www.highintegritysystems.com/rtos/what-is-an-rtos/rtos-tutorials/how-to-use-watchdog-strategies/>.
- [152] *Bluetooth LE & Bluetooth -- ESP-FAQ latest documentation*. Accessed: May 22, 2026. [Online]. Available: <https://docs.espressif.com/projects/esp-faq/en/latest/software-framework/bt/ble.html>.
- [153] *Bluetooth PHY - How It Works And How To Leverage It – Punch Through*, en-US, Section: Bluetooth Low Energy (BLE), Dec. 2019. Accessed: May 22, 2026. [Online]. Available: <https://punchthrough.com/crash-course-in-2m-bluetooth-low-energy-phy/>.
- [154] *Setting serial baud rate on ESP-IDF does nothing | Hacker News*. Accessed: May 22, 2026. [Online]. Available: <https://news.ycombinator.com/item?id=44997023>.
- [155] K. Dissanayake, D. Kularatna-Abeywardana, N. Patel, and N. Kularatna, "A review of smartphone energy management: Power distribution, battery technologies, fast charging, thermal management and challenges," *Energy Conversion and Management: X*, vol. 30, p. 101735, May 2026, ISSN: 2590-1745. DOI: 10.1016/j.ecmx.2026.101735. Accessed: May 30, 2026. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2590174526002187>.
- [156] PHILIPS, *IntelliVue MP60/MP70 Patient Monitor*. Accessed: May 12, 2026. [Online]. Available: https://media.supplychain.nhs.uk/media/documents/n0889200/marketing/31990_n0889200.pdf.
- [157] FLUKE, *Fluke ProSim 8 User manual*. Accessed: May 12, 2026. [Online]. Available: https://www.flukebiomedical.com/sites/default/files/resources/ProSim8_umeng0300.pdf.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN

École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/epl