

Design and prototyping of an ECG simulator for medical device testing in low-resource settings

Master Thesis

Anna Graell Collado

Design and prototyping of an ECG simulator for medical device testing in low-resource settings

by

Anna Graell Collado

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on 29th September, 2023 at 10:45h.

Student number: 5211271
Project duration: September 1, 2022 – September 29, 2023
Thesis committee: Prof. dr. J. Dankelman, TU Delft, supervisor
Dr. ir. A. J. Knulst, Green Pastures Hospital, supervisor
Dr. ir. A. Loeve, TU Delft, supervisor
Prof. dr. ir. L. Abelmann, TU Delft, external member

Cover: Morning view from a room at Australian Camp, Dhampus, Nepal,
June 2023 (Anna Graell Collado)
Style: TU Delft Report Style, with modifications by Daan Zwaneveld

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Preface

This report summarises the work done during the last year. During this time, the idea of an ECG simulator that can be built, maintained and used in LRs has gone from theory to practice. The process culminated with a visit to Nepal, which allowed testing the Prototype in a real scenario, as well as discovering an amazing country and culture. Throughout the whole process, very positive and encouraging feedback about the Prototype was received by the many collaborators contacted. These included biomedical engineers and technicians based in Nepal and Sub-Saharan Africa, as well as other professionals with experience in medical device maintenance in LMICs. This work could not have been a success without all of you. I have been passionate about this project since the first day and I am incredibly grateful for all your help. I have experienced the mismatch between needs and market options, but I also got to know about innovations and projects driven by biomedical engineering and healthcare professionals that aim to change this. All this made me even more passionate about Biomedical Engineering, which I am convinced should be about providing solutions to real needs. I hope you enjoy these pages as much as I did.

Thanks for being here.

*Anna Graell Collado
Delft, September 2023*

Abstract

Ensuring reliable and safe performance of medical devices in healthcare institutions is crucial for the wellbeing of patients. For this, physiological simulators may be used, which provide reference signals to compare against. The usage of physiological simulators, such as ECG simulators, as medical device testers is widely extended in high-income countries. However, their high cost and need for frequent calibration by the manufacturer make their availability and usage in low-resource settings (LRs) unfeasible. Remote healthcare facilities, low- and middle-income countries (LMICs), and hospitals in general would highly benefit from new designs of simulators that can be produced, repaired and maintained locally. Therefore, we present the design of an affordable, portable Arduino UNO-based prototype ECG simulator with the novel feature of visual indicators for self-calibration checks. The ECG simulator was built with affordable and broadly available components and tools, retrieved from an educational electronics workshop. The current prototype can be used to test 3-lead ECG meters. It reproduces a lead II waveform extracted from MIT-BIH database, with 1.3 mV peak-to-peak and 0.1 mV negative offset. Heart rate simulation ranges from 35 to 130 bpm, in steps of 5 bpm. Signal data points defining a desired waveform and heart rate can be uploaded into the simulator via the user-friendly and open-source Arduino software. The simulator could also be modified to generate 10- or 12-leads by upscaling the electronics and adding more storage space. Analog signal generation is based on a pulse-width modulation signal, which is smoothed by a low-pass filter and decreased in amplitude with operational amplifiers, which also add a negative voltage offset. Visual indicators enable simulated frequency and amplitude checks without the need of external tools, while amplitude can be calibrated manually. Additionally, rechargeable batteries facilitate on-site testing of medical equipment. Evaluation of the prototype included performance and functionality tests. HR and amplitude accuracy of the simulated signal were quantified and compared to those from a commercial simulator. Lastly, the performance of the simulator in a real scenario was tested at the Green Pastures Hospital of Pokhara, Nepal. Functionality tests proved the correct detection of the simulated signal by 6 patient monitors of different models, manufacturers and purchase year.

Keywords: ECG, simulator, low-resource settings, affordable, Arduino, maintenance

Contents

Preface	i
Abstract	ii
Nomenclature	v
1 Introduction	1
1.1 What are physiological simulators?	1
1.2 Examples of physiological simulators	1
1.3 Medical device maintenance in Nepal and SSA	3
1.4 Thesis goal and structure	4
2 Design process	5
2.1 Design requirements	5
2.2 Concept solutions and concept selection	6
2.2.1 ECG simulation	6
2.2.2 NIBP simulation	8
2.2.3 SpO ₂ simulation	9
2.3 Focusing on ECG simulation	11
3 Design choices	13
4 Prototype development	17
4.1 First stage	17
4.1.1 Proof of concept	17
4.1.2 PWM + LPF design	18
4.1.3 Amplitude checks	19
4.1.4 Timing checks	22
4.2 Second stage	23
4.2.1 Attenuation and negative voltage offset design	23
4.2.2 Timing checks	24
4.3 Feedback and calibration system	26
4.3.1 Amplitude feedback	26
4.3.2 Amplitude adjustment	27
4.3.3 HR feedback	28
4.4 System overview	29
5 Final prototype	31
5.1 Software	31
5.1.1 Reference signal	31
5.1.2 Arduino code	32
5.2 Hardware	32
5.2.1 Signal generation	32
5.2.2 Amplitude attenuation and negative voltage offset	32
5.2.3 Signal inversion	32
5.2.4 Feedback	33
5.2.5 Calibration	34
5.3 Cost and other components	35
5.4 Power supply	35
6 Prototype evaluation	39
6.1 Testing on Green Pastures Hospital patient monitors	39

6.1.1	HR checks	39
6.1.2	Amplitude checks	39
6.1.3	Functionality tests	41
6.1.4	Feedback tests	43
6.1.5	Comparison to a healthy subject	43
6.2	Zurich Instrument's lock-in amplifier	43
6.3	Energy consumption	45
6.4	Design requirements assessment	45
7	Discussion	47
7.1	Performance	47
7.2	Cost and materials	49
7.3	Portability	49
7.4	Usability	49
7.5	Maintenance	50
7.6	Further improvements	50
7.7	Limitations of this work	51
8	Conclusions	53
	Acknowledgements	54
	References	55
A	Non-commercial simulators comparison	58
B	LMICs limitations	63
C	Arduino code	66
D	Arduinos comparison	74
E	ATmega programming	77
F	D/A conversion based on PWM and LPF	80
G	Low-pass filter design for D/A conversion and discussion	82
H	Attenuation and negative offset calculations	84
I	Simulator outputs: V+, GND and V-	86
J	Opamps characteristics and rationale	88
K	Table of costs	90
L	Simulator output on common oscilloscopes	91
M	Oscilloscope measurements: first stage timing checks	92
N	Monitor measurements: second stage timing checks	103
O	Patient monitors characteristics at Green Pastures Hospital	106
P	Monitor measurements: second stage amplitude checks	107
Q	Simulators accuracy and HR determination comparison	109
R	Permission to cite collaborators	114

Nomenclature

Abbreviations

Abbreviation	Definition
LMICs	low- and middle-income countries
ECG	electrocardiographic
LRs	low-resource settings
HR	heart rate
NIBP	non-invasive blood pressure
SpO ₂	blood oxygen saturation
DUT	device under test
NGO	non-governmental organisation
WHO	World Health Organisation
NPR	Nepali Rupees
PWM	pulse-width modulation
LPF	low-pass filter
PCB	printed circuit board
KB	kilobytes

1

Introduction

1.1. What are physiological simulators?

Ensuring reliable and safe performance of medical devices in healthcare institutions is crucial for the wellbeing of patients. For this, physiological simulators may be used, which provide reference signals to compare against. Several medical device manufacturers specify the need of physiological simulators for performance checks [35, 8, 11], which are advised during preventive maintenance (at least every one or two years) and after repairs.

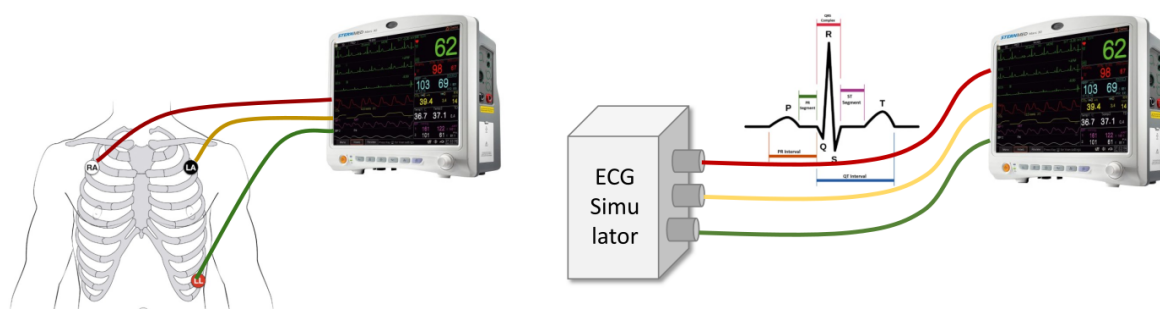


Figure 1.1: On the left, example of 3-leads ECG measurement by a patient monitor. On the right, schematic representation of ECG simulator and patient monitor connection. The simulator provides an ECG-like reference for performance tests.

1.2. Examples of physiological simulators

Commercial physiological simulators vary in cost and features, as shown by some examples in Fig. 1.2. Examples of basic models for single-parameter simulation presented here have prices around \$600-\$3,000. For multiparameter simulators, price increases with the number of signals and customizable options, ranging from \$2,000-\$13,000. The most expensive solution listed, ProSim 8 by Fluke Biomedical, is also the most complete one, including 50+ arrhythmia waveforms, customizable ECG amplitude, HR and NIBP [14]. Almost two times lower in price, the modular SimCube SC-5 + OxSim Flex simulator covers the same signal types but only offers 4 operation modes with fixed features [37, 36].

Besides their price, the complexity of use also increases with the number of features available. In the most simple models, all features can be accessed by means of buttons or switches, while more complex designs require navigation through digital menus and sub-menus.

Commercial simulators also require maintenance, recommended by manufacturers once per year [38, 15]. However, undisclosed and complex internal functioning, and the need for specialized tools and spare parts make repairs and calibration by the users difficult. Calibration services offered by manufacturers and distributors have generally a cost 5-10% of the purchase price. Only one of the manufacturers from Fig. 1.2, Pronk Technologies, provides details in their User's manual [38] about calibration and

spare parts, enabling calibration to be performed by the user to some extent. User and service manuals are also not always freely available, e.g., BC Biomedical charges \$114. However, when accessible, technical documentation is generally extensive, including accuracy and performance details.

Limitations of commercial simulator use in low-resource settings (LRs)

- High *price*
- Complex *use*, many functionalities
- Undisclosed *internal functioning*
- Limited *user manual* availability
- Specialized *tools and spare parts* required for *maintenance*

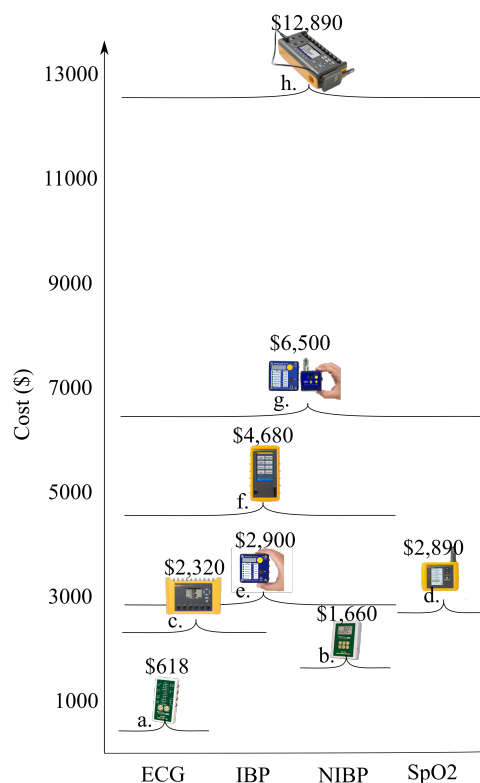


Figure 1.2: Commercial simulators comparison on cost and simulated signals (ECG, IBP, NIBP and SpO2). Models and manufacturers: a. PS-2005, BC Biomedical; b. NIBP-1010, BC Biomedical; c. ProSim 2, Fluke Biomedical; d. ProSim SPOT Light, Fluke Biomedical; e. SimCube SC-5, Pronk Technologies; f. ProSim 4, Fluke Biomedical; g. SimCube SC-5 + OxSim Flex Kit, Pronk Technologies; h. ProSim 8 + ProSim SpO2 test module, Fluke Biomedical.

The use of physiological simulators as medical device testers is well-established in high-income countries (HIC). However, the previously identified limitations of physiological simulators make their availability and use in low-resource settings (LRs) unfeasible. Besides commercial simulators, **online tutorials and commercial kits** to build your own ECG or NIBP simulator exist [23, 53, 4]. Although these are affordable and usually open-source solutions, they have not been validated for use as testers. No information on accuracy is provided, and they usually have output signals and connectors not suitable for medical devices.

Some viable alternatives to the use of physiological simulators exist. These include the comparison of performance between a device suspected to malfunction and one or two theoretically working properly. The use of reference signals not generated by a simulator is also common. An example is the use of auscultatory measurements as reference for blood pressure determination, which can only validate diastolic and systolic measurements but not dynamic ones. Similarly, manometer and expansion chambers can be used to test static pressure measurements by NIBP meters [35]) but not dynamic ones. A very common alternative to simulators is used for pulse oximetry checks, in which a 'healthy individual' is used as reference. This can check whether the electronics are working but not validate

the pulse oximeter readings. In ECG testing, a 'healthy individual' can be used to check cable integrity but not HR measurement or waveform reproduction. As seen, these alternatives either lack a fully characterised reference (for accuracy determination), are not able to reliably reproduce a reference signal (for repeatability measurements) or do not enable signal customization (for alarms or signal recognition tests). Thus, they can only partially check the device's functioning. This is because complete testing of a medical device includes performance testing (accuracy and repeatability), functional testing (confirm ability of the device to detect a signal or check that alarms go off, among others), and visual inspection (confirm physical integrity).

Research on the simulation of physiological signals is vast, with popular applications being medical training, mathematical modelling for diagnosis and treatment support, and validation of new measurement techniques. However, development of simulators is a less popular research topic. In previous work, a literature review on simulation of physiological signals was carried out, covering 120 articles [17]. Some examples of low-cost simulators purportedly developed for medical device testing were identified [41, 7, 32], although the majority had educational purposes. Many of the articles provided details for implementing the described systems (e.g., electric schematics, list of costs...), but they lacked comprehensive tests on simulator performance and functionality. From 73 articles that covered ECG simulation, for example, only five reported accuracy values. Additionally, many of the educational simulators had output signals and connectors not suitable for medical devices. Open-source resources (e.g., Arduino, Scilab, Python) were common among low-cost solutions, as well as references to "standard electronic components" use. However, only 4 articles made some reference to low- and middle-income countries (LMICs), and none of them studied the feasibility of local assembly and maintenance. Only imprecise terms such as "for developing countries" [28], "for third world countries" [1], "potential to be manufactured locally" [32] or "low cost to make it more accessible in developing countries" [50] were used. Simulators with higher performance implemented more complex electronic circuits, requiring specialized knowledge and components [41, 6, 2]. Another limitation of research simulators was the lack of information on how to calibrate them, if possible at all. See Appendix A for a comparison of non-commercial simulators.

Limitations of non-commercial ECG simulator use in LRs

- Not fully characterised *accuracy and performance*
- *Suitability for LMICs* not studied
- No complete information on how to perform *calibration*
- No visual *feedback*
- No *realistic* signal (educational simulators output in volts, or mathematically-defined signal)

1.3. Medical device maintenance in Nepal and SSA

A close collaboration with biomedical engineers in Nepal allowed for a deeper analysis of the situation of medical device maintenance and testing in this country. Thanks to the Nick Simon's Institute (NSI), an NGO that provides rural healthcare support in Nepal, the inventories of 5 rural hospitals (15-50 beds) and 2 bigger ones (GPH: 100 and UMHT: 169 beds) were accessed. These revealed patient monitors and ECG monitors as the most abundant devices in rural hospitals. In the two big hospitals, the most abundant devices were pulse oximeters, followed by patient monitors, NIBP meters, and ECG meter. These data is visualized in Fig. 2.9. Although the inventories are in many cases incomplete, they give an indication of the maintenance needs in the region. Regarding availability of patient simulators, 3 of them were identified. A ProSim 4 is owned by NSI, an old ECG simulator (SKX-2000) is owned by UMHT and a ProSim 8 is available at GPH. According to NSI, no other tester is available at the hospitals they give coverage to (166 rural hospitals). The rest of non-public hospitals in Nepal perform some medical device maintenance, according to the sources, but public hospitals do not. According to the latest censuses available, there are 125 public and 301 non-public hospitals in Nepal [9, 29]. When no calibrated reference is available, a 'healthy patient' or a 'working' unit are used as reference.

The limitations to medical equipment maintenance in Nepal are similar in Sub-Saharan Africa (SSA), according to three biomedical engineering workers with experience in the region. In fact, literature corroborates that the described situation is representative of LMICs [33, 44, 10, 56, 55]. Through literature

and conversations with biomedical engineering workers, limitations of medical device maintenance and simulator use in LMICs were identified (see Appendix B). At the highest level, the main limitations are low availability of calibration services and spare parts in the country and no management systems for health-care technology in local nor national level. This leads to a lack of BME training programmes and maintenance-dedicated budget. This, in turn, results in unattractive BME career prospects, lack of periodic maintenance programmes, lack of spare parts and absence of simulators at the hospitals... Unskilled personnel resulting from the lack of training limit periodic maintenance implementation, spare parts availability and simulators usage even further. Meanwhile, the high cost and complexity of commercial simulators are detrimental for their usage adoption in LMICs.

Main challenges in LMICs

- Low availability of *calibration services and spare parts*
- Unskilled *personnel*
- Lack of medical device *maintenance programmes and budget*

1.4. Thesis goal and structure

Commercial physiological simulators have been developed to be used in HICs. The acquisition and use of physiological simulators in LMICs and LRs is limited by their cost, complexity and low availability of maintenance services and spare parts. Alternative testing methods that do not make use of physiological simulators cannot fully test the functionality of medical devices. Meanwhile, online resources to 'build your own simulator' are generally not suitable for medical device testing and their accuracy is not reported. Some low-cost simulators have been published in research journals, but their suitability for LMICs has not been studied. Other aspects such as local production, maintenance and calibration have not been solved yet. For all these reasons, new designs of physiological simulators are required. This led to the goal of this master thesis: the design and development of a prototype ECG simulator for low-resource settings that can be produced, repaired and maintained locally. This will be referred to as the Prototype.

This thesis report is structured as follows. First, the **Design process** is described. This starts by formulating general design requirements (for a physiological simulator), and particular design requirements (for the simulation of ECG, NIBP and SpO₂ signals). Next, concept solutions for the simulation of ECG, NIBP and SpO₂ are listed, and Harris Profiles are used to select the most suitable ones. From this point onwards, the report focuses on a single physiological signal, namely ECG. At **Design choices**, the functions to be fulfilled by an ECG simulator for LRs and possible means to fulfil them are collected in a morphological chart. The means are discussed based on their affordability, usability and maintainability, and the most suitable ones are chosen. After this step, the general tools that will be employed for the development of a Prototype simulator are clear. However, during **Prototype development**, technical and experimental details allow the Prototype to evolve and get optimised as it is being built. After this, a **Final prototype** is reached, described in terms of software and hardware components. The performance and functionality of the final design are characterised at the beginning of **Prototype evaluation**. Based on this, an assessment of the design requirements by the final Prototype concludes the chapter. A **Discussion** on the performance achieved by the presented Prototype follows. In this section, limitations that were faced during this work are also mentioned. Finally, some guidelines on the further development of the Prototype are given, and conclusions from the whole process are drawn. At the end, a set of Appendices are attached. Among these, detailed descriptions of methods employed, technical specifications and raw measurement data.

2

Design process

2.1. Design requirements

The design of a simulator for testing medical devices was based on the situation and needs of **Medical device maintenance in Nepal and SSA**. First, general requirements for a simulator intended for use in LMICs and low-resource settings were listed (Fig. 2.1). Next, specific requirements for the simulation of ECG, NIBP and SpO₂ were listed, as well as some relevant regulations (see Fig. 2.2). These requirements tried to cover the basic functionality needed by simulators for ECG meter, NIBP meter, pulse oximeter or patient monitor testing.

Affordability	<ul style="list-style-type: none">• More affordable than commercial simulators (ProSim 8, 7071€; ProSim 2, 2260,35€; PS410, 1160€; SimCube 3390€). Around the price described in literature (Machado-Gamboa, 2018: 143€; Varela Guzman, 2021: 120€; Vergara Gil & González Vargas, 2021: 200€).• Materials should be locally available, possible to order or common components that can be reused from non-working devices. Daraz (https://www.daraz.com.np/) might be used as an indicator for local availability in Nepal. Aliexpress and component reuse might be considered for availability in SSA, electronic shops not common.
Usability	<ul style="list-style-type: none">• Easy to use without manual, intuitive
Maintainability	<ul style="list-style-type: none">• Spare parts should be locally available or possible to order• Calibration or alternatively performance testing of the relevant units should be possible in a basic electronics workshop (multimeter, oscilloscope, soldering station). Otherwise, there should be a way to indicate when further testing/calibration is required.
Functionality	<ul style="list-style-type: none">• Signal synchronisation: ECG, NIBP and SpO₂ simulation should be synchronised• Functionality: simulator can be used on patient monitors, ECG meters, NIBP meters and pulse oximeters to test their ability to measure ECG, NIBP, HR and/or SpO₂• Functionality: simulator should work with any patient monitor, ECG meter, NIBP meter or pulse oximeter• Updatability: possibility to add more signals to generate in the future• Portability: user should be able to test medical devices on site• Integrability: all signals integrated in one system• Performance: accuracy and repeatability of the system should be well defined• Working conditions: work in humid (Nepal: average 74 or 81 % depending on region, SSA: approx. 40-75% depending on region) and high temperature (Nepal: average mean 2-27 °C depending on region; SSA: mean outdoor 13-30 °C depending on region) conditions

Figure 2.1: General design requirements for a simulator intended for LMICs and low-resource settings.

ECG	<ul style="list-style-type: none"> • Signal waveform: should simulate one 'healthy patient' ECG waveform consisting of QRS complex and P and T waves • Signal amplitude: must be able to simulate signal components 0.5 – 1 mV in amplitude • Frequency of simulated signal: 60 bpm = 1 Hz • Relevant units to be calibrated are HR 	<p>Accuracy specifications: AMPLITUDE: $\pm 10\%$ of the nominal value of the output or $\pm 100 \mu\text{V}$, whichever is greater HR: $\pm 10\%$ or $\pm 5 \text{ 1/min}$ Based on IEC 60601-2-27</p>
NIBP	<ul style="list-style-type: none"> • Signal waveform: should simulate a 'healthy patient' blood pressure and HR reading • Simulated HR: 60 bpm = 1 Hz • Simulated blood pressure range: should simulate blood pressures around 90 mmHg – 120 mmHg (normal systolic) and 60 – 80 mmHg (normal diastolic) • Should be able to deal with inflation pressures 0 – 300 mmHg generated by NIBP meter • High signal repeatability is required • Relevant units to be calibrated are t.b.d. (static pressure, frequency and amplitude) 	<p>Accuracy specifications: BP DETERMINATION: $< 0.75 \text{ mmHg}$ or $< 0.5\%$ (following the 1/4th of DUT recommend.*) DYNAMIC REPEATABILITY: $\leq 2 \text{ mmHg}$ (BP Pump 2 from Fluke*) *no access to ISO 81060-5</p>
SpO ₂	<ul style="list-style-type: none"> • Signal waveform: should be able to simulate a 'healthy' human finger's red and infrared absorption spectra originated by photoemitters inside a pulse oximeter (including SpO₂ and HR information) • Signal amplitude: amplitude relationship red/infrared spectra defines the desired SpO₂ • SpO₂ simulation range: oxygen saturation level between 90 – 100 % SpO₂ • HR simulation range: 60 – 100 bpm • Relevant units to be calibrated are amplitude and frequency of simulated absorption spectra 	<p>Accuracy specifications: SpO₂ LEVEL: error $< 1\%$ SpO₂ over the range 70 - 100 % SaO₂ Based on ISO 80601-2-61:2011</p>

Figure 2.2: Specific design requirements for ECG, NIBP and SpO₂ simulators to be used in LMICs and low-resource settings.

2.2. Concept solutions and concept selection

For each signal type, concept solutions were gathered in a table, with possible calibration methods listed. Some of these are existing solutions (collected from literature or commercial products), while others are the result of a brainstorming process. Finally, Harris Profiles were used to compare the proposed solutions. Based on predefined criteria, the most suitable solution for the simulation of each parameter was chosen.

The evaluation criteria included: price, portability, simulated waveform updatability, availability of materials in LMICs, feasibility of calibration in LMICs, accuracy and repeatability of the simulated signal, independence from the manufacturer of the device under test (DUT) manufacturer and usability.

2.2.1. ECG simulation

Two main approaches exist for the simulation of an ECG signal that can be used to test ECG or patient monitors (see Fig. 2.3). These are analog-based and digital-based systems. The use of programmable digital controllers such as microcontrollers provides the system with higher flexibility in terms of signal updatability or customization. However, digital controllers require programming and digital circuits can become more complex to understand than simple analog ones. This also makes in-house repairs and testing of analog circuits easier, especially when using through-hole components, even though soldering stations are not broadly available in LMICs.

Both methods can achieve comparable accuracies, mainly defined by the timing accuracy of the oscillators used. Both analog components and digital controllers are scarce in LMICs but some models can be ordered online. Digital controllers allow for remote connection or online data storage, what can enable performance tracking for simulator accuracy checks.

All in all, both approaches are comparable in performance, but the increased flexibility provided by digital-based systems makes them more suitable for this application (see Fig. 2.4). Given the increased complexity usually associated with digital systems, efforts will be focused on making the circuit as simple to understand and build as possible. This has the goal of overcoming the identified limitations of limited specialized knowledge and limited tools and spare parts in LMICs.

ECG solutions

SOLUTIONS	EXPLANATION	CALIBRATION IDEAS
CRYSTAL-OSCILLATOR ANALOG CIRCUIT	Components of the ECG signal are generated by different passive components and added by an IC (integrated circuit)	<ul style="list-style-type: none"> Amplitude: variable resistor for 1mV reference signal calibration [Altman1981] --> needs reference to calibrate against --> 5V reference board? Frequency: high stability thanks to crystal oscillator Frequency: HR extracted from 60Hz power-line frequency [Janes1969] Voltage: voltage regulator Voltage: precision voltage reference Use front-end that simulates an ECG machine [Almeida2021]
uC DIGITAL CIRCUIT WITH D-to-A CONVERSION AND CONDITIONING CIRCUIT	Signals are digitally generated (stored or streamed), converted to analogue signal and conditioned by an electronic circuit	<ul style="list-style-type: none"> Monitor performance over time Waveform: oscilloscope compared to digital visualization Voltage: voltage regulator Voltage: precision voltage reference [Almeida2021] Digital oscilloscope Harmonics map Use front-end that simulates an ECG machine [Almeida2021]

Figure 2.3: Possible solutions for the generation of an ECG signal to be used for medical device testing.

ECG Harris Profile

	A) CRYSTAL-OSCILLATOR ANALOG CIRCUIT				B) uC DIGITAL CIRCUIT				
	--	-	+	++	--	-	+	++	
Price									Compared to the price of commercial simulators, they are both much cheaper. B) a bit more expensive and you need a laptop to programme it.
Portability									On site testing can be achieved by both, both can be hand held and have similar dimensions.
Updatability									uC in B) allows for re-programming, reproduced signals can also be changed via SD card or memory. Analog circuit in A) doesn't allow updating, only rewiring.
Availability of materials									Analog components more available than digital. Both can be ordered online.
Calibration/testing feasibility									Front-end for performance testing can be used in both. Performance tracking seems easier digitally (data storage/sending).
Accuracy and repeatability									Both can use highly stable crystal oscillators, voltage reference, freq. ref. B) can reproduce more realistic signals but that's not related to accuracy.
Manufacturer independence									ECG simulation only inputs a signal to the monitor, doesn't need to interact with it more than that, so both are manufacturer independent.
Usability									Usage of both is similar. Digital requires some programming.
TOTAL	= 7				= 8				miro

Figure 2.4: Evaluation of ECG simulation approaches for medical device testing in LMICs.

2.2.2. NIBP simulation

There are three approaches to simulate NIBP for testing of medical devices, presented in Fig. 2.5. The first one refers to the generation of a signal to be detected by the DUT's cuff, which resembles the way in which NIBP meters measure blood pressure in a patient. The second one consists on generating a pressure pulse directly into the hose of the device under test. The last one refers to the generation of an electrical signal to be input to the DUT.

While an electrical-only system has advantages in terms of stability of the generated signal (enhanced by no mechanical parts involved), its ability to generate signals to test the full functionality of an NIBP metre is limited. This cannot test leakage in the DUT's cuff or valve release. Moreover, its feasibility of implementation is not clear, as an electrical direct connection to the internal transducer is not commonly accessible in NIBP meters.

Given the variability and secrecy of measurement algorithms used by manufacturers, the introduction of a pressure signal directly at the DUT's hose appears more manufacturer-dependent and complex to operate than the generation of a signal to be read by the DUT's cuff. This becomes the most suitable method (Fig. 2.6).

Regarding calibration, static pressure calibration is well-established and based on known weights, while dynamic pressure calibration is not. Dynamic pressure checks could be based on monitoring the simulated signal in order to detect deviations over time.

NIBP solutions

SOLUTIONS	EXPLANATION	CALIBRATION IDEAS
SIGNAL READ BY DUT'S CUFF	Reproduction of the way a human arm generates oscillometric pulses on DUT.	<ul style="list-style-type: none"> • Static pressure: pressure gauge calibration with known weight • Frequency: crystal-controlled oscillator? • Dynamic pressure: analyse own output?
SIGNAL AT DUT'S HOSE	Simulator connected to DUT's hose, air pressure pulses input into the system. Reproduces oscillometric pulses created by arteries on DUT.	
ELECTRICAL SIGNAL	Electrical signal directly input to DUT. Reproduces electrical response of DUT's pressure transducer when measuring NIBP on human arm.	

Figure 2.5: Possible solutions to NIBP simulation for medical device testing.

NIBP Harris Profile

	A) SIGNAL READ BY DUT'S CUFF				B) SIGNAL AT DUT'S HOSE				C) ELECTRICAL SIGNAL				
	--	-	+	++	--	-	+	++	--	-	+	++	
Price													C) includes electronics+controller. A&B) need electronics + pressure pulse generators. A) also needs a structure to wrap cuff around, but this is cheap.
Portability													A&B) have moving, more delicate, probably bigger and heavier parts than C), less portable. C) can be hand-held.
Updatability													Programmable electronics can be used by the three, they allow database updatability.
Availability of materials													Electronics common in the three solutions have limited availability (can be ordered online). Pressure generator for A&B) can be ordered online. A) needs additional materials for structure, but these are easily available.
Calibration/testing feasibility													No established complete calibration method yet. For A&B) static pressure/repeatability tests are defined in standards, but not dynamic pressure. C) not mentioned in standards.
Repeatability													C) could be more repeatable, offer more fine control, because the others involve pressure transmission, mechanical parts...
Manufacturer independence													B&C) can account for manufacturer diff sampling frequencies mathematically ('the envelope'). A) should be independent of manufacturer because it would replicate the way blood flows.
Usability													It is not clear whether it is possible to use the electrical system in medical devices since access to the pressure transducer of DUTs is not commonly available.
TOTAL	= 4				= 2				= 3				miro

Figure 2.6: Evaluation of NIBP simulation solutions for medical device testing in LMICs.

2.2.3. SpO₂ simulation

Three approaches were defined for simulation of SpO₂ (see Fig. 2.8). The first one entails the optical emission of human blood absorption spectra and pulse. The second one consists of physically simulating this. The third uses optical modulation of the light emitted by a pulse oximeter.

Electro-optical modulation becomes the more expensive and difficult-to-implement method (see Fig. 2.7). Physically simulating the absorption properties of human tissue appears highly complex and is unprecedented, although it would provide a repeatable and manufacturer-independent reference. Optical emission is the standard method used by commercial simulators and has usability advantages (DUT can be used in the same way as during clinical measurements) over the other two. Compared to physical simulation, optical emission offers much more flexibility and updatability. This, together with its affordability and portability, makes it the most suitable solution. However, the fact that manufacturers differ in measurement wavelengths and patterns must be taken into account.

The development of a calibration device for SpO₂ simulators has not been solved by the industry yet [19]. Calibration can currently only be achieved through clinical trials, which are used to define the calibration

curves on which pulse oximeters based their measurements. Instead, a maintenance method could be based on monitoring the generated signal over time to be able to detect any performance decay.

SpO2 Harris Profile

	A) OPTICAL EMISSION				B) PHYSICAL SIMULATION				C) ELECTRO-OPTICAL MODULATION				
	--	-	+	++	--	-	+	++	--	-	+	++	
Price													The three solutions are cheaper than commercial simulators, but C) is more costly [Hornberger, 2002]
Portability													The three allow on-site testing, but C) needs a more bulky setup and a laptop as controller. B) could be as portable as A).
Updatability													To change simulated signal in B) you need different preparations and change them manually. A&C) are based on programmable controllers so updatability of signals is easy but needs programming or at least laptop.
Availability of materials													B) is potentially made with available materials, otherwise easily ordered online. A&C) can be ordered online.
Calibration/testing feasibility													Complex, no established method. A&C) would require physical reference or analysis of own output. B) would have to guarantee properties stability.
Accuracy and repeatability													Accuracy similar to commercial simulators is expected. B) could have higher repeatability, signal less dependant on working conditions.
Manufacturer independence													C) is intrinsically independent of manufacturer since it reuses the same light emitted by DUT. B) is in theory independent of manufacturer.
Usability													A) would be used in similar way as pulse oximetry (plug&play). B&C) would need some more setting up.
TOTAL	= 8				= 6				= 5				miro

Figure 2.7: Evaluation of SpO2 simulation solutions for medical device testing in LMICs.

SpO2 solutions

SOLUTIONS	EXPLANATION	CALIBRATION IDEAS
OPTICAL EMISSION	Optical generation of absorption spectra. Reproduces optical spectra generated by finger on pulse oximeter using external light source.	• Output monitoring
PHYSICAL SIMULATION	Physical element with well-characterised spectra interacts with DUT's probe. Reproduces way in which a human finger generates photoplethysmographic signal on a pulse oximeter.	• Well-characterised and stable spectra
ELECTRO-OPTICAL MODULATION	Electro-optical element intercepts and modulates light emitted by DUT before it reaches DUT sensor. Reproduces optical spectra generated by finger on pulse oximeter modulating DUT's emitted light.	• Output monitoring

Figure 2.8: Possible solutions to simulate SpO2 for medical device testing.

2.3. Focusing on ECG simulation

From here, this work focuses on ECG simulation, due to time limitations. ECG is one of the most common physiological signals used for patient monitoring, with ECG meters being listed as an essential medical technology in the "Package of Essential Noncommunicable (PEN) Disease Interventions for Primary Health Care in Low-Resource Settings" or "List of Priority Medical Devices for the First Level of Care for the Countries of the Americas" by WHO. To understand their usefulness in a real scenario, the inventories of 7 Nepali hospitals were consulted. All hospitals owned ECG meters, while 6/7 listed patient monitors and NIBP meters, and only 5/7 counted with pulse oximeters (see Fig. 2.9). According to these inventories, ECG would be the test signal needed to check most devices in rural hospitals, which currently do not own any simulators. Even though in the 2 big hospitals, pulse oximeters outnumbered the rest of devices, the development of a reference for pulse oximetry calibration is still an issue to be solved. As ISO80601-2-61 indicates, there is currently in the market no device capable of reproducing the complex interactions between light and human tissue taking place during pulse oximetry measurement [19]. Instead, SpO₂ simulators in the market can only be used to evaluate the ability of pulse oximeters to reproduce the calibration curves that the manufacturers designed into them. This is why they are correctly referred to as 'functional testers'. Instead of using a simulator, the connection of a pulse oximeter to a 'healthy individual' is mentioned by some manufacturers as a means to check that the electronics in the device are functioning, e.g., [34, 39]. Regarding NIBP meters, their integrity and accuracy can be tested without the need for simulators, as indicated by service manuals [35, 40]. In the case of ECG meters, while cables can be checked by connection to a human subject, alarms and HR determination can only be tested using simulators, as can be seen in user manuals [35, 8]. An ECG simulator can also be used to check automatic detection features, filters and waveform reproduction accuracy. Another singularity of ECG simulators is that they do not actively interact with the medical device under test. NIBP meters and SpO₂ simulators monitor the output of the DUT and adapt the signal generated accordingly, in terms of pressure or light patterns, respectively. Opposite to that, an ECG simulator does not need to monitor the response of an ECG meter. This allows this project to focus on other challenges related to LMIC suitability and accuracy characterisation.

Why focusing on ECG simulation?

- Due to the prevalence and use of ECG meters in hospitals
- Due to the impossibility of using a 'healthy individual' or other low-cost references to fully test ECG meters
- Due to the one-directional interaction between ECG simulator and DUT. This allows focusing on suitability for LMICs (affordability, maintainability, usability) and accuracy characterisation.

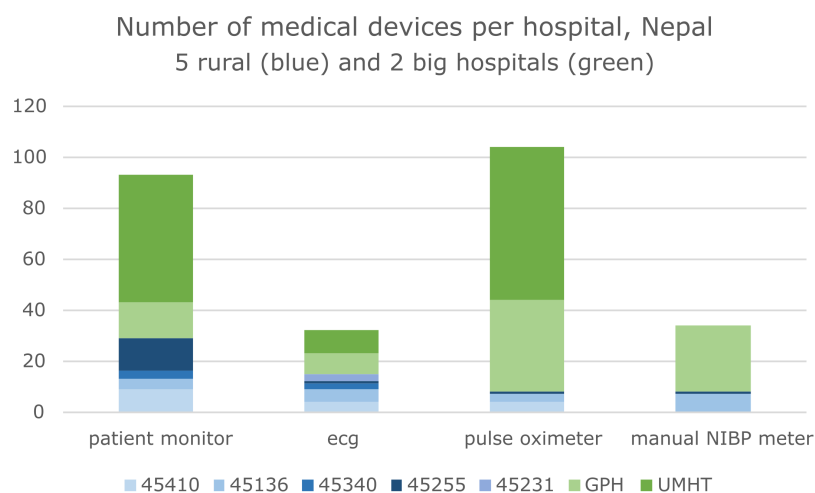


Figure 2.9: Amount of patient monitors, ECG monitors, SpO₂ and NIBP meters in 5 rural hospitals (15-50 beds, in blue) and 2 big hospitals (GPH: 100, UMHT: 169 beds, in green) according to available inventories. Data from NSI, GPH and UMHT.

3

Design choices

As a result of the **Design process**, an ECG simulator based on a programmable controller was chosen as the most suitable solution for an ECG simulator for medical device testing in LRs. This chapter explains the design choices made for each of the functions that the ECG simulator had to fulfil, shown in Fig. 3.1.

Morphological chart: programmable controller-based ECG simulator

FUNCTIONS	MEANS			
SWITCH ON/OFF AND NAVIGATE	Buttons/switches	Mouse/Keyboard + digital display	Plug & Play	
DISPLAY	Embedded screen (LCD)	External monitor	Indicators on cage	
SIMULATION CONTROLLER	Microcontroller chip	Microcontroller board (Arduino, Teensy...)	Single-board computer (Raspberry Pi)	Computer
SIGNAL GENERATION	Stored signal replay	Mathematical model	Electrical components (oscillator-based)	
DIGITAL-TO-ANALOG CONVERSION	DAC	PWM + LPF	Sound card	R2-R converter
SIGNAL CONDITIONING	Voltage divider	Opamp		
POWER	Battery	Powerline	USB to computer	
PERFORMANCE TESTING	Simulator compares output to reference and alerts performance deviation	Simulator monitors performance over time and detects deviations	The user compares output to target signal	
AMPLITUDE AND HR FEEDBACK	Visual indicators	Sound indicators		
REFERENCE	Voltage regulator	Frequency reference		
AMPLITUDE ADJUSTMENT	Trimmer resistor	Opamp		miro

Figure 3.1: Functions and means for a digitally controlled ECG simulator for LRs. Chosen solutions in light green.

The first two functions of the morphological chart define **how the user interacts with the ECG simulator**. The first one refers to switching ON/OFF the simulator and navigating through different simulation options or features. The simplest solution is a plug&play device, which switches on when connected to power and directly outputs a signal. This restricts the simulation, not allowing on-demand customization of the signal. Opposite to that, the use of a mouse/keyboard and digital display allows customization the most. However, this solution has the highest cost. A more balanced solution in terms of cost and versatility is offered by the use of buttons/switches. These are easily available and affordable components, with the possibility to be reused from old electronic devices. Moreover, they allow the selection of a range of features without requiring a digital display, only indicators (labels, colours...). For instance, a two-position switch with ON and OFF labels can work as an easy solution for turning the simulator ON/OFF. To give an indication of cost, a two-position switch is available at Daraz.com.np for 30 NPR.

The second function listed refers to the **visualization of the options/features offered by the simulator**. The use of an external monitor is the most expensive solution, although convenient if a computer is used as part of the simulator. However, the availability of computers is low in LMICs. Additionally, the use of a monitor would require the programming of a GUI, more complex than programming an LCD screen. An embedded screen like LCD is available for affordable prices online. For instance, LCD screens for 500 - 1000 NPR can be ordered at Daraz.com.np. Their performance is lower than a monitor but could be sufficient to display HR and amplitude options. However, a basic menu of HR and amplitude options can be achieved by the combination of indicators (e.g., labels or colours) next to buttons/switches. This is the cheapest solution, does not require programming and can be implemented with materials reused from other devices.

Next, the options for the **simulation controller** are described. Using a computer to control the simulator is the most expensive solution, not easily available, and reduces the portability of the system. Additionally, a GUI would be required, adding complexity to the system development. Using a microcontroller chip is the cheapest option (255 NPR in Nepal, aliexpress.com), but highest in complexity and less intuitive. One of the drawbacks of microcontroller use is the complexity of programming, usually in C language. This can be solved by pre-loading an Arduino bootloader, which makes the chip compatible with Arduino IDE. Arduino IDE is an interface that makes programming easy through an intuitive language, built-in functions, free tutorials... Given the popularity of Arduino, microcontroller chips with bootloader are available online. In Nepal, for instance, ATmega 328P microcontrollers pre-programmed with Arduino UNO bootloader are available at a local distributor (himalayansolution.com) for 450 NPR. However, this solution lacks the easy-accessible GPIO pins of an Arduino board, the USB and jack connectors, the in-built safety circuits such as voltage regulators, or the LEDs indicative of the state of the board. All these elements can be added to the microcontroller chip but that requires some technical skills and components. Microcontroller boards, such as Arduino, are more expensive solutions but benefit from the previously mentioned advantages, useful to reduce the external components and specialized knowledge needed to build an ECG simulator. Arduino is an open-source platform, supported by a huge online community with free resources, and increasingly available in LMICs. For instance, the Arduino UNO board is available at Daraz.com.np for 1460 NPR, while one of its competitors, Teensy, is not. One of the drawbacks of Arduino is the need for a computer for its programming, although this will not be needed anymore once the code is uploaded. Single-board computers have increased functionalities (HDMI, GPU, Ethernet) compared to microcontroller boards, but these are not required for a basic simulator. Moreover, a keyboard/mouse + digital display + Linux are needed to operate a single-board computer, increasing cost and complexity. In the case of the popular Raspberry Pi, this lacks the analog inputs that Arduino features, which allow the use of potentiometers or switches to control the simulation. It also lacks an analog-to-digital converter (ADC) and in-built storage space. All in all, an Arduino board is a better choice for embedded, low-power and low-cost applications such as an ECG simulator for LRs.

Regarding **signal generation**, three solutions were found. On one side, electrical components can be used to generate a signal that resembles an ECG waveform. In that case, signal generation will be based on oscillators, responsible for the generation of periodic signals with a specific frequency, the amplitude of which will be modified by other components such as inductors, resistors, and capacitors. Usually, separate circuits are used for the generation of the different waveform components (P-T waves, Q-S dips, R peak), and the signals are combined sequentially at the output. As one can imagine, this results in a complex circuit. Although frequency and amplitude can be modified, the waveform cannot

be updated, since that would require changing the electrical components. As a result, the generated signal has an ECG-like waveform but cannot reproduce realistic frequency-amplitude variations, which dynamically vary. An example of a simulator relying on oscillators and other electrical components for waveform generation was developed by [47]. On the other side, the signal can originate from a mathematical function that controls the output of a microcontroller (e.g., a PWM signal) or of a Digital-to-Analog Converter (DAC). Signal realism depends on the complexity of the modelling function. The simplest functions are not able to simulate realistic time-amplitude variations seen in ECG recordings, but they are less computationally demanding [16]. Another approach is based on modifying the microcontroller PWM output according to a signal stored in the simulator. This allows the reproduction of both patient data recordings and mathematical-based signals. This is the most versatile method, although it is limited by the storage capacity of the system. According to IEC standards, the use of both patient data and mathematical signals is required to test electrocardiographs with automated interpretation features [30]. For these reasons, signal replay was chosen as the most suitable method for signal generation.

Given that the only way to present a signal to an ECG or patient monitor is through its electrode cable, an analog signal must be generated. A DAC can be used for **digital-to-analog conversion**. Microcontrollers with built-in DACs exist, but these are less common than ADCs. Microcontroller boards with a DAC also exist, although the basic models of Arduino and Raspberry Pi do not include one. Alternatively, an external DAC chip can be bought, programmed and installed, but its availability in LMICs is limited. A workaround is found in some academic papers, which customize off-the-shelf soundcards or use those embedded in computers, for the simulation of ECG analog signals [12, 49, 21]. Disadvantages of DAC/soundcard include availability and chip programming, usually done in C language. Although these components are cheap, they are not as available as passive electrical components. An alternative to the use of a dedicated DAC is the combination of a PWM output (present in most microcontrollers) and a low-pass filter (LPF) built with resistors and capacitors. Dedicated DACs can achieve higher performance than PWM-based DACs. However, as explained in Appendix F, the performance of a PWM-based DAC can be enough for ECG simulation. This is mainly due to the low frequencies used in ECG. The mentioned advantages make PWM + LPF the solution that best balances cost, usability, availability and performance. Read Appendices F and G for more details on how this was implemented. Resistors in R2-R configuration can also be used to build a DAC, but its performance is dependent on resistor stability.

Signal conditioning is needed to achieve amplitude levels that simulate those of surface ECG signals, in the millivolts range. Additionally, depending on how the signal is generated, a negative voltage offset might be required, given that ECG recordings reach negative values. A voltage divider can be used to attenuate the signal, like in [24]. However, the use of operational amplifiers (opamps) allows attenuation and offset shifting towards negative values.

In terms of **power supply**, the use of batteries is the only solution that enables a stand-alone system. This increases portability, enabling on-site testing. However, batteries have an elevated cost, and rechargeable ones even more. For instance in Nepal, 4 rechargeable batteries have a cost of around 1128 NPR in Daraz.com.np.

The two last rows refer to **performance evaluation and adjustment**. These are the most challenging functions of the simulator, given the very small output amplitude (millivolts) and the associated noise. One approach could be the measurement of the simulator's output and its comparison to the reference signal. This would require extra hardware and manipulation of the output signal (e.g., amplification), which would add uncertainties to the measurement. A simpler approach is to let the user compare the output to the reference signal.

For **performance adjustment** the following functions must be fulfilled: indicate to the user the amplitude and frequency of the signal being simulated, provide a reference to compare the simulated signal to, and provide a means to precisely modify the generated output. The desire to keep complexity and the need for external tools to the minimum must be taken into account. Regarding amplitude/frequency indicators, these should respond directly to the output signal, without modifying it, and signal when the output differs from a reference. Solutions considered included sound and visual indicators. Sound indication would be useful to provide feedback on the generated HR, but not to indicate amplitude. Opposite to that, visual indicators can indicate both amplitude and HR. As a visual indicator of HR, an LED

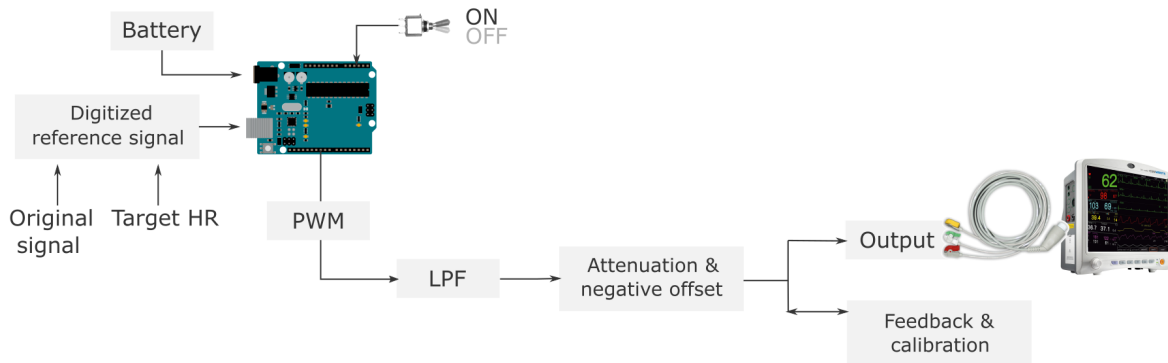


Figure 3.2: Conceptual design resulting from the means chosen to fulfil each function of the **Morphological chart**. Solid arrows indicate data/signal flow.

blinking at the same rate as the generated HR was implemented. In this case, the user is responsible for measuring the HR (counting blinks) and comparing it to the target value. However, given that LEDs respond to current and not in a linear way, the addition of opamps and reference voltages for amplitude indication was required (see Fig. 4.16). Luckily, the Arduino UNO board has in-built voltage regulators that provide outputs of 3.3 V and 5 V, which were used as reference voltages in the circuit.

Together with the visual feedback, a system to **adjust detected deviations** is required to perform calibration. Solutions chosen for other simulator functions (Arduino as controller, opamp for attenuation and voltage shift) condition the calibration options. The frequency accuracy of the simulator is defined by the Arduino's oscillator and cannot be adjusted. However, the high stability of oscillators was expected to guarantee no significant frequency deviations. In terms of amplitude adjustment, a trimmer resistor was used. This is a conventional solution used by commercial simulators and other devices that require precise adjustment. This component will be described in more detail in 4.3.2. **Amplitude adjustment**. Although an opamp can also be used to modify amplitude, it is an active component, more expensive and more complex to use. Opposite to that, trimmer resistors are cheap (12 NPR at Daraz.com.np) and can be tested and replaced easily if needed.

The combination of one solution per function describes the integrated conceptual design solution presented in Fig. 3.2.

4

Prototype development

The **Design choices** made in the previous chapter defined a system based on affordability, maintainability and usability. In the next step, the Prototype was developed. First, a PWM and low-pass filter system were designed and implemented for initial signal generation. This is referred to as the first stage. Then, signal conditioning was added, namely attenuation and negative offset. This defined the second stage. Afterwards, the feedback and calibration system was built. As the Prototype was being built, a deeper understanding of the technologies and methods used to fulfil the Prototype's functions was gained. With the tools available, tests were performed to characterise the system's performance and guide the next development steps. This allowed optimisation of some of the technologies and methods used. The aspects of the Prototype that evolved during its implementation are described in this chapter. The tests and technical details that guided this evolution are explained. The first stage is presented first, followed by the second stage and ending with the feedback and the calibration system.

4.1. First stage

4.1.1. Proof of concept

At the beginning of the Prototype development, a proof-of-concept was carried out to determine whether the suggested system based on Arduino UNO, PWM and LPF was capable of outputting an ECG-like signal in the millivolts range. For this, the in-built functions of Arduino were used to output a PWM signal controlled by the reference waveform. Arduino default settings were used, including 8-bit resolution and 980 PWM frequency. An ECG-like waveform extracted from [23] was used as reference (dashed blue line in Fig. 4.1). The reference datapoints were converted into 8-bit scale (values from 0 to 255) and uploaded into the Arduino to control the PWM output. Arduino code was written in Arduino IDE to update the PWM output every 1 millisecond (simulation rate) according to a new datapoint. When all points had been simulated, simulation started again from the first datapoint. A 1st order LPF with a cut-off frequency of 150 Hz was built, matching the ECG monitoring frequencies commonly used by patient monitors. To attenuate the generated signal from 5 Vpp down to 1 mVpp, a simple voltage divider built with resistors was used. This was chosen as the simplest method to attenuate a signal. Given the impossibility of measuring such a small signal with common oscilloscopes, a lock-in amplifier (Zurich Instrument's UHFLI 600 MHz) was used. The result is presented in Fig. 4.1). Although the signal presents visible noise, this test proved the possibility of generating an ECG-like waveform by means of PWM and LPF and downscaling it to a 1 mVpp signal. After the successful proof of concept, the next steps focused on improving the simulation accuracy and system performance.

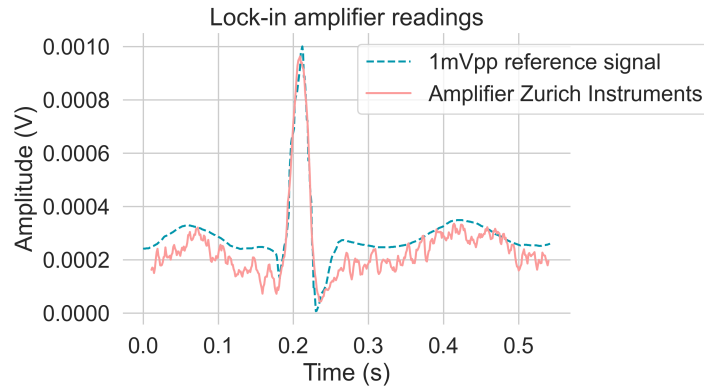


Figure 4.1: Comparison of reference (dashed line) and generated signal (solid line) when using a voltage divider. Signal was generated using PWM and LPF and attenuated down to 1mVpp by means of a voltage divider.

4.1.2. PWM + LPF design

The development of the Prototype was initiated by setting up a digital-to-analog converter based on Arduino's PWM output and a LPF. The previous step had proved the generation of an ECG-like analog signal with the potential to be interpreted by a patient monitor. Now, the goal was to improve the signal simulation obtained with the basic system consisting on default settings and a resistor divider. For this, a more realistic reference signal was chosen. ECG recording number 101 from MIT-BIH database was used (see Fig. 4.2). The MIT-BIH database was chosen due to its free availability and its usage as a reference signal in maintenance standards [31].

The design of PWM and LPF was further studied based on the characteristics of ECG monitoring signals recorded at the body surface. The frequencies of interest for ECG monitoring defined a 150 Hz cut-off frequency for the LPF. Given a specific cut-off frequency, the accuracy of the output signal could be improved by increasing the Arduino's default PWM frequency, although this decreased the PWM digital resolution. The best solution to trade-off was given by ATmega 9-bit fast PWM mode. Therefore, ATmega was programmed (following the steps explained in Appendix E) to increase the default PWM frequency to 31.25 kHz while using 9 bits of PWM resolution. With this, the system achieved a resolution of 0.01 V and a ripple of 0.04 V (according to LTSpice simulations), which adds up to a total uncertainty of 0.05 V.

Recordings from the MIT-BIH database have a step size of 0.1 mV. Thus, if the generated signal was attenuated down to the amplitude of the reference signal (1.3 mVpp), the present system would have a resolution of $2.5 \mu\text{V}$ and uncertainty of $13 \mu\text{V}$, which is approx. 7.70x higher than the smallest voltage difference to be reproduced. the designed system would achieve 7.70x better theoretical accuracy than the standard for ECG monitors (amplitude reproduction accuracies of $100 \mu\text{V}$ [18]). This would meet the general recommendation of a simulator being at least 4 times more accurate than the device under test [19].

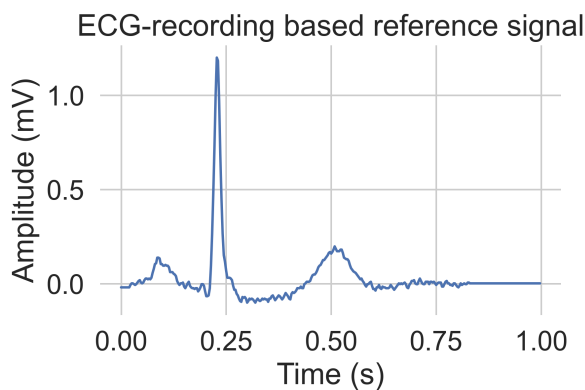


Figure 4.2: Reference signal based on ECG recording from MIT-BIH database.

Table 4.1: Amplitude of Figure 4.2 components

Component	Amplitude (mV)
Baseline	-0.02
P wave	0.14
Q dip	-0.07
R peak	1.20
S dip	-0.10
T wave	0.20

Next, a way to visualize the output of the designed system was required. Given that the PWM output of the Arduino ranges from 0 to 5 V, the signal could be visualized in a common oscilloscope. This, together with LT Spice simulations, helped optimise the PWM + LPF design. In Fig. 4.5, the study of the effect of different cut-off and PWM frequencies on the output signal is displayed. Figs. 4.5a-b show how given the cut-off frequency of 150 Hz, a PWM frequency of 980 Hz results in a lower ripple than that of 490 Hz. In the simulator's output, the reduction in ripple gives a better accuracy of the defined DC values, what is visible as a narrower baseline of Fig. 4.5b compared to Fig. 4.5a. These are the default PWM frequencies in Arduino. However, the improvement is substantial when the PWM frequency is improved further up, to 31.25 kHz (see Fig. 4.5d). Not only the baseline is more precisely defined (narrower), but also more accurate in the simulation of smaller waveform components (waves and dips). When a lower cut-off frequency is chosen, the system becomes slower, not being able to simulate the high frequencies that contain waves and dips, as Fig. 4.5c shows.

Once the system was optimised for ECG simulation, tests to characterise its amplitude and timing accuracy were performed. These are explained in the next subsection.

4.1.3. Amplitude checks

The Tablet Oscilloscope tBook mini TO1000 used enabled data storage and downloading. This access to the signal measurements facilitated amplitude and timing accuracy assessment of the signal simulated with the PWM + LPF system. The goal was to compare the signal recorded by the oscilloscope to a reference signal in which waveform components were precisely defined in time and amplitude. For this purpose, a reference waveform was mathematically defined using the equations by [45]. This is presented in Fig. 4.3. As detailed in Table 4.2, the reference waveform reaches a maximum voltage of 1.2 mV and a minimum of -0.13 mV, resulting in a 1.33 mVpp amplitude.

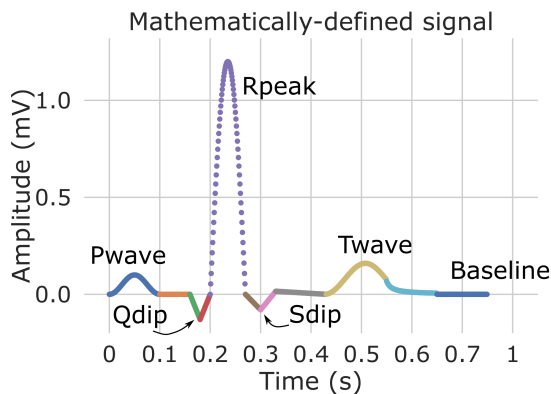


Table 4.2: Amplitude of Figure 4.3 components

Component	Amplitude (mV)
Baseline	0.00
P wave	0.10
Q dip	-0.13
R peak	1.20
S dip	-0.08
T wave	0.16

Figure 4.3: Mathematically-defined signal used for error measurement. Defined using equations from [45].

Original vs. generated signals ($f_{\text{PWM}} = 31.25 \text{ kHz}$, $f_c = 150 \text{ Hz}$)

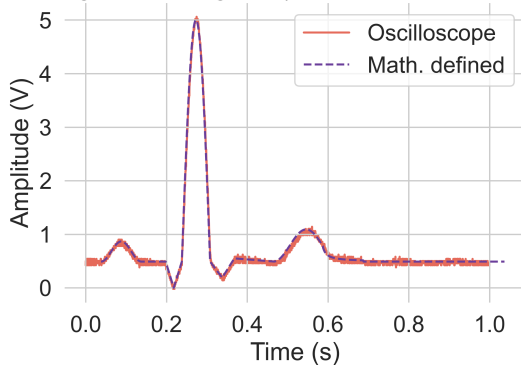


Figure 4.4: Amplitude accuracy assessment of simulation based on PWM + LPF. Reference mathematically-defined ECG waveform (dashed line) is compared to simulated signal recorded by oscilloscope (solid line).

The mathematically-defined waveform was uploaded onto the Arduino board and the output after the LPF was recorded using Tablet Oscilloscope tBook mini TO1000 (5kHz sampling rate, 1V/div, 8-bit resolution). A comparison of the reference and recorded waveforms is shown in Fig. 4.4.

The measurements are presented in Table 4.3 and Fig. 4.6. These were performed on one recorded waveform and the amplitude measurements of each component were averaged over the three surrounding datapoints.

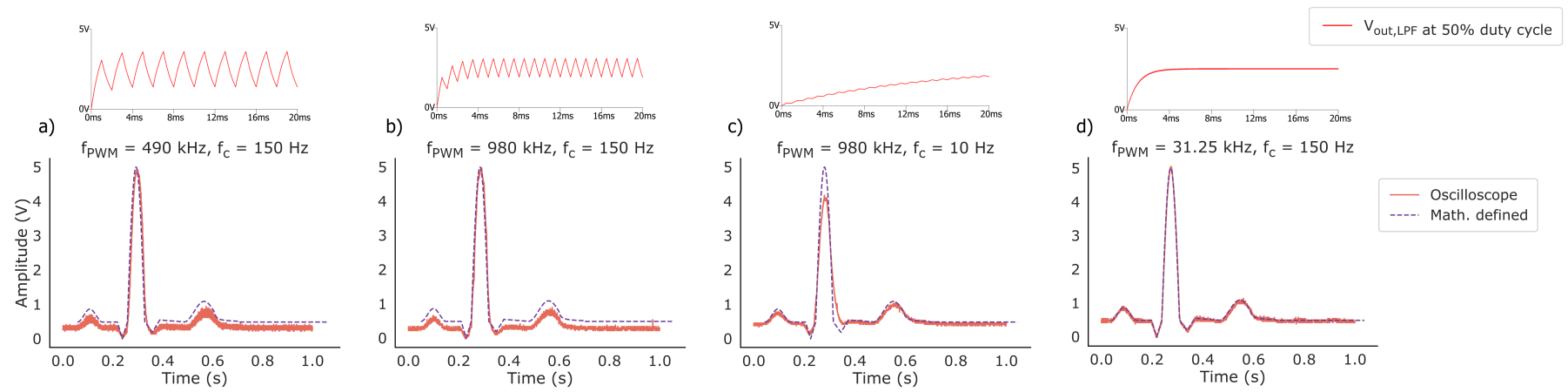


Figure 4.5: Effect of the cut-off and PWM frequencies on the output signal after the LPF. Using the same original data (blue), the output signal captured by an oscilloscope (red) varies depending on the cut-off and PWM frequencies. Above each plot, an LT Spice simulation of the output ripple during 2.5 V generation provides information on the resolution of the system under each condition.

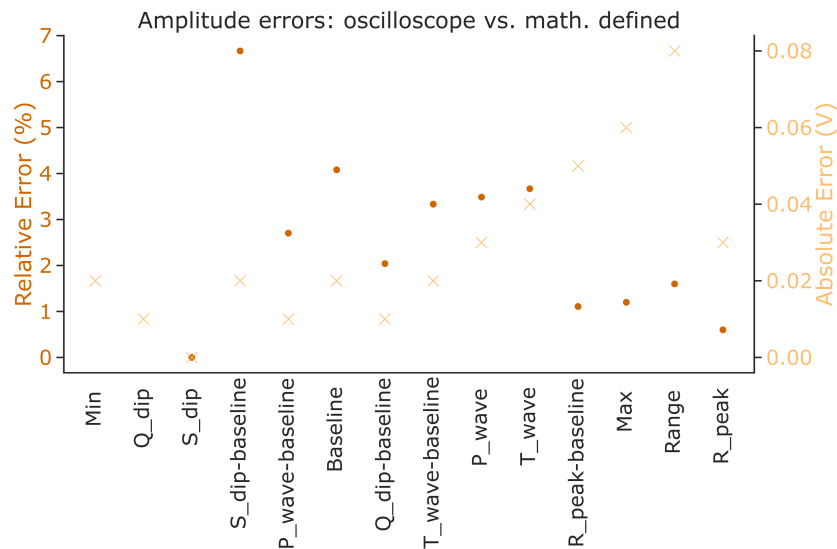


Figure 4.6: Amplitude error of simulated signal with respect to mathematically-defined signal, both shown in Fig. 4.4. Absolute amplitude error (V) is calculated as the absolute difference between simulated signal and reference signal. Relative error (%) is the absolute error divided by the reference value. Relative error for 'Min' and 'Q_dip' is not shown due to division by 0.

Table 4.3: Amplitude error of mathematically-defined waveform components and absolute values: reference signal in range 0-5V, measured signal at oscilloscope and calculated error (%) with respect to original value.

	Original 0-5V	Measured 0-5V	Absolute Error 0-5V	Relative Error 0-5V (%)
Min	0.0	-0.02	0.02	inf
Max	5.0	5.06	0.06	1.2
Range	5.0	5.08	0.08	1.6
Baseline	0.49	0.47	0.02	4.08
P_wave	0.86	0.83	0.03	3.49
Q_dip	0.0	0.01	0.01	inf
R_peak	5.0	5.03	0.03	0.6
S_dip	0.19	0.19	0.0	0.0
T_wave	1.09	1.05	0.04	3.67
P_wave-baseline	0.37	0.36	0.01	2.7
Q_dip-baseline	0.49	0.48	0.01	2.04
R_peak-baseline	4.51	4.56	0.05	1.11
S_dip-baseline	0.3	0.28	0.02	6.67
T_wave-baseline	0.6	0.58	0.02	3.33

After using a mathematically-defined waveform as reference for hardware design first and amplitude measurements after, this was replaced by an ECG recording from the MIT-BIH database available online, to check if the system would also perform as expected when simulating a real ECG recording. A comparison of the original ECG recording and the simulated one is presented in Fig. 4.7.

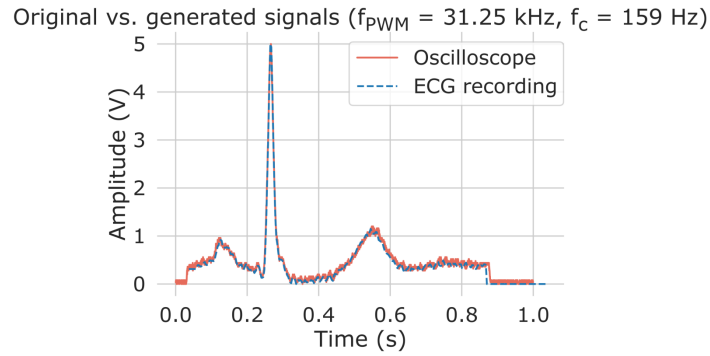


Figure 4.7: Comparison of simulated (solid line) and reference (dashed line) ECG waveform corresponding to patient recording number 101 from MIT-BIH database. The signal simulated using PWM and LPF was recorded using an oscilloscope.

4.1.4. Timing checks

Next, timing accuracy of the generated signal was analysed for heart rates from 35 to 130 bpm in steps of 5 bpm. In Python, the original ECG recording was either cropped or its baseline extended to represent the target HR. This waveform was then uploaded to the Arduino board. The PWM output after being filtered by the LPF was recorded for 8 seconds using Tablet Oscilloscope tBook mini TO1000 (5kHz sampling rate, 2V/div, 8 bits resolution).

Target R-R interval (in seconds) is inversely related to the target HR, and a factor of 60 is needed to harmonise the units:

$$\Delta t_{R-R_interval} [\text{s}] = \frac{1}{HR [\text{bpm}]} \cdot 60 \quad (4.1)$$

Oscilloscope data were analysed using Python. The **R-R interval** ($\Delta t_{R-R_interval}$), or time between consecutive R peaks, was measured and compared to the target value. Raw data and other figures can be found in Appendix M. The mean and standard deviation of the R-R interval errors extracted from oscilloscope data are presented in Fig. 4.8. **The mean error across all HRs is 1.56% of the target R-R interval.** The error measured always corresponded to a delay, what translates in a HR lower than the target one. This is measured by the HR determination error.

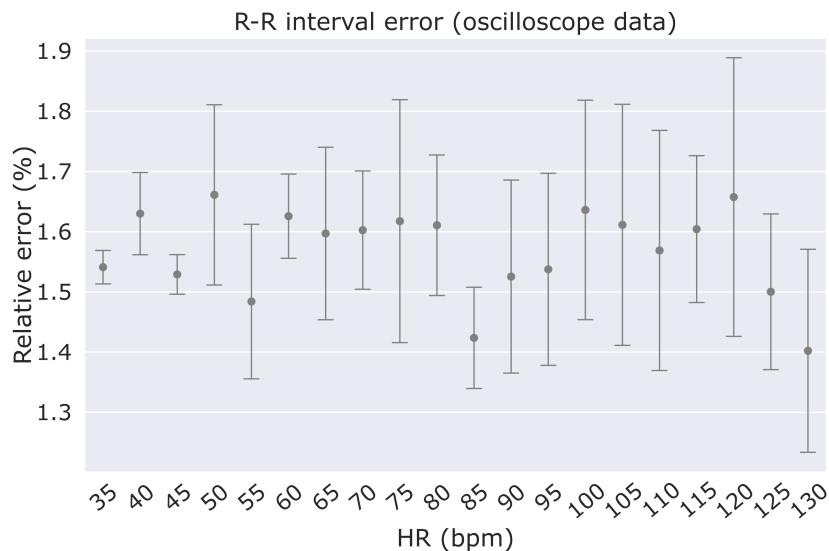


Figure 4.8: Relative error of time interval between two consecutive R peaks of generated signal recorded with oscilloscope compared to target. Dots represent mean relative error per HR and whiskers mark the standard deviation. Due to data recording time limit, the number of measurements included per HR goes from 4 at 35 bpm up to 17 at 130 bpm. Mean relative error across all HRs is 1.56%.

HR determination corresponds to the HR value that a patient monitor would display when reading the simulated signal. Patient monitors measure the time between consecutive R peaks and calculate the corresponding HR only in integer values. Given the behaviour observed, it is expected that patient monitors take an average over a few R peaks for HR determination, but the exact way differs among manufacturers, some of them not disclosing the method used (see Appendix Q).

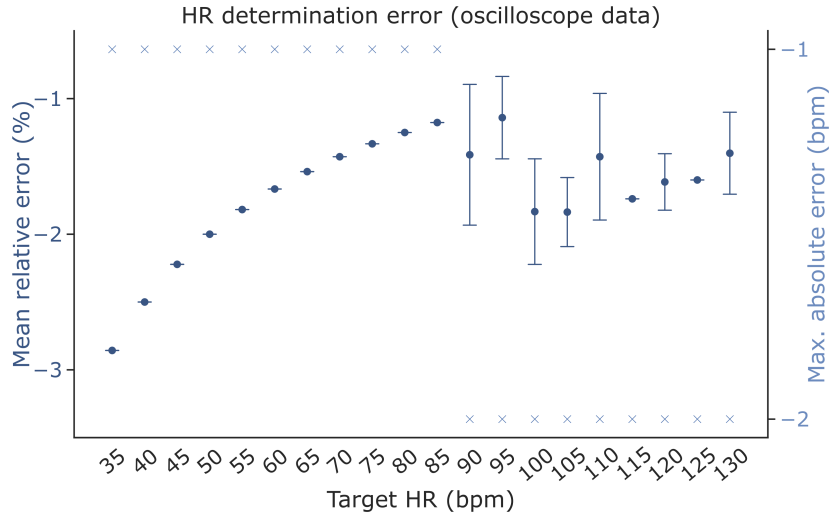


Figure 4.9: HR determination error measured on oscilloscope data of simulated HRs from 35 to 130 bpm in steps of 5 bpm. Mean relative error (circles) and standard deviation (whiskers) in dark blue. Maximum absolute error per HR in light blue. Target HR as reference.

In this work, HR determination error for each individual R-R interval was calculated (using Eq. 4.1) and rounded up to obtain an integer value. Absolute errors were calculated as the deviation of the HR determined by each R-R interval to the target value. Relative errors were obtained by calculating the deviation of the measured HR to the target HR and normalizing by that same reference.

In Fig. 4.9, the mean of the relative errors over each target HR is presented, as well as the maximum absolute deviation recorded for each target HR. The highest relative error (-2.86%) was recorded for the lowest HR (35 bpm). Regarding absolute errors, all HRs < 90 bpm show a maximum deviation of -1 bpm, while HRs \geq 90 bpm have a maximum deviation of -2 bpm. From Fig. 4.9 it is clear that for HR determination error, deviation from the mean increases for HRs \geq 90 bpm.

4.2. Second stage

4.2.1. Attenuation and negative voltage offset design

In order to achieve simulation of a realistic ECG signal such as the ECG recording extracted from the MIT-BIH database (Fig. 4.2), not only attenuation but also negative voltages had to be generated. This would adequate the signal for correct detection and interpretation by patient monitors (with common maximum display amplitudes of 5 mVpp).

To achieve attenuation and negative voltage offset with the least complexity and number of components, the use of a weighted difference amplifier was chosen. This configuration uses a single opamp to provide attenuation and offset. Besides the opamp, various resistors were required, precisely chosen to define the desired attenuation factor. The offset voltage was achieved by using the 5 V output from the Arduino board and attenuating it by means of resistors. The output of a weighted difference amplifier is described by equation 4.2. Detailed calculations are described in Appendix H. Opamp OP97 was chosen for this task, mainly due to its low input offset voltage and supply voltage suitable for battery operation. A more detailed description of the opamp suitability is presented in Appendix J.

$$V_{out} = -\frac{R2}{R1}V_{ref} + \left(1 + \frac{R2}{R1}\right) \frac{R4}{R3 + R4}V_{out_LPF} \quad (4.2)$$

The output signal characteristics made this stage the most challenging to design and test. From one side, the signal's low amplitude made it prone to be affected by noise, leakage currents, electrical component deviation, temperature and humidity... From the other side, the signal amplitude made it not visible in common oscilloscopes, which have sensitivities of a couple of millivolts (for examples check Appendix L). Even though patient monitors were designed to detect and display signals of this amplitude, downloading of the data was not available. Printing modules or digital access exists for some models, but these options were not available at Green Pastures Hospital (Pokhara, Nepal). In this situation, amplitude accuracy measurements were carried out on pictures taken from the patient monitors' screen and using ImageJ (Fiji) software. Inevitably, this method implied many inaccuracies that lowered measurement reliability. Later, access to a lock-in amplifier (UHFLI 600 MHz by Zurich Instruments) allowed better amplitude characterisation of the signal. All second-stage amplitude accuracy results are presented in [Prototype evaluation](#).

4.2.2. Timing checks

The first HR determination tests were carried out with a GE Healthcare patient monitor (M-NESTPR module) available at the MISIT lab in TU Delft, Netherlands. Since this patient monitor was out of calibration, the commercial simulator 214A (Dynatech Nevada, purchased in 1994 and lent by Reinier de Graaf Ziekenhuis, Delft) was used to check its HR reading accuracy. HRs from 30 to 130 bpm every 5 bpm were tested and resulted in an error of 0 bpm (see Appendix N). Therefore, GE Healthcare patient monitor was considered a good reference.

For HR determination checks, Frankie was connected to GE Healthcare patient monitor and set to simulate the same range of HRs. After waiting 1 min for settling, the HR displayed at the patient monitor screen was written down. After each measurement, Frankie was connected to an oscilloscope (Tablet Oscilloscope tBook mini TO1000, 5kHz sampling rate, 2V/div, 8 bits resolution) and 8 seconds of simulation were recorded. These data were analysed in Python. Synthetic data simulating the reference waveform delayed 0.0156 seconds per waveform was also generated for comparison. The delay of 0.0156 seconds was extracted from the timing analysis of the oscilloscope data (see Section 4.1.4 for more details). Comparison of the results can be seen in Fig. 4.10.

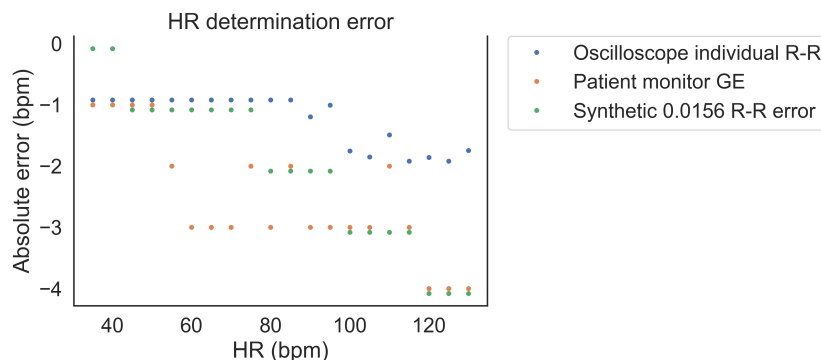


Figure 4.10: Comparison of HR determination absolute error (bpm) from oscilloscope data, GE patient monitor readings and synthetic data (original with 1.56% R-R interval error). Oscilloscope data corresponds to mean HR defined by individual R-R intervals. Blue/green dots have been moved 0.08 points upwards/downwards, respectively, for better visualization.

Timing improvements

The next step consisted on improving the timing accuracy. For this, the Arduino function used to keep track of the simulation time was improved. During the previous measurements, the `millis()` function was used, which has a theoretical accuracy of $\pm 0.5\%$. This function has some jitter, given by some Arduino internal processes that compensate small timing errors. For this reason, the function becomes accurate in the long term, but can occasionally deviate a couple of milliseconds until the error gets automatically corrected. Opposite to this, **the `micros()` function** has a higher accuracy and a resolution of $4\mu\text{s}$. In effect, after implementing the `micros()` function, HR determination maximum error reduced from 5.0% (3 bpm error for target HR=60 bpm) to 2.86% (1 bpm error for target HR=35 bpm). While with the `millis()` function, absolute errors were reaching 4 bpm, with the `micros()` function the biggest deviation from the target HR was 1 bpm. In terms of mean error, changing from `millis()` to `micros()` yields a decay

from 3.09 % to 0.98 %. The improvements can be seen in Fig. 4.11. For more details check Appendix N.2-N.3.

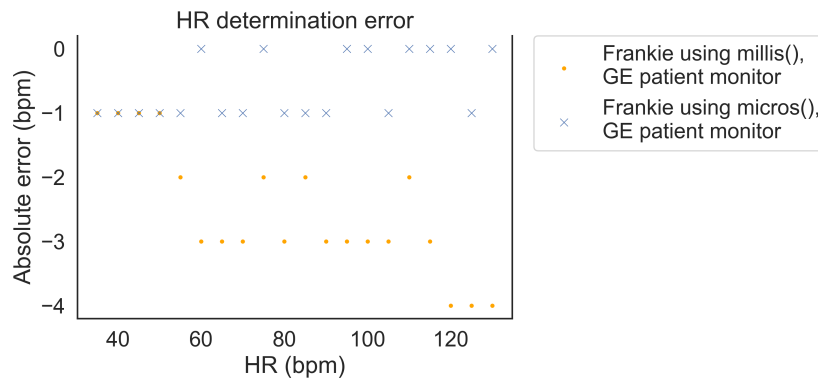


Figure 4.11: HR determination error comparison of millis() and micros() Arduino function. These are used to handle Frankie's timing events. HR values refer to GE Healthcare patient monitor readings.

To try to improve the timing accuracy further, the simulation rate was decreased a bit below the 2000 microseconds. The goal was to compensate for the timing error that was yielding to lower HRs than expected. This timing error was probably a result of the micros() function, which still has some error.

Tuning of the **simulation rate** was carried out at the Green Pastures Hospital of Pokhara, Nepal. There, the simulator was connected to 6 different patient monitors currently under use at the hospital. Patient monitors differed in manufacturer, year of construction and ECG cables. A summary of their characteristics can be found in Appendix O. Tests were performed at an average temperature of 32 °C and 47 % humidity. The procedure is explained next.

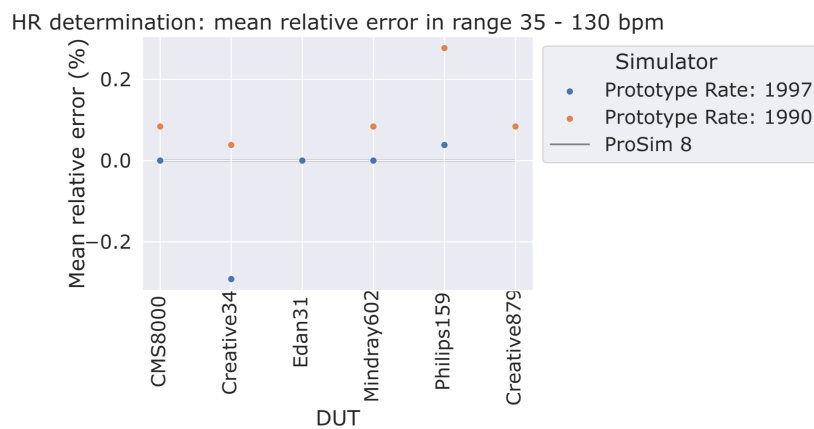


Figure 4.12: Mean relative error of HRs detected by 6 different patient monitors versus reference HR. For each patient monitor, 16 measurements were taken, corresponding to HRs from 35 to 130 bpm every 5 bpm. Relative deviation of on-screen HR from target value from reference was calculated for each HR. Then, the average across 16 was computed. Frankie was used in two different configurations: with a simulation rate of 1990 and 1997 microseconds. ProSim 8 was also tested.

The simulator was connected to 6 different patient monitors and was set to simulate the reference waveform extracted from MIT-BIH database (Fig. 5.1), modified for HRs from 35 to 130 bpm every 5 bpm. The different HRs were tested consecutively on one patient monitor. After waiting at least for 30 seconds, HR displayed on the screen of the patient monitor was noted down. This process was repeated setting a simulation rate of 1990 and 1997 microseconds. All patient monitors were tested once per sampling rate and target HR combination except Creative34, which was tested twice due to the variability of the HR ± 1 bpm on screen. In that case, a mean of both measurements was taken. Another exception is Creative879, which was only tested for 1990. After these tests, Frankie was replaced by ProSim 8 and the same range of HRs were tested. Wave group was set to NSR (Adult),

axis to intermediate and ST deviation to off. Patient monitors were set to monitoring mode and 3-leads ECG when available. After 30 seconds, HR displayed on the monitor screen was written down. The mean of the relative error across all HRs for each patient monitor is displayed in Fig. 4.12. In 4 of the 5 cases in which 1997 simulation rate was tested, this showed an improved relative error compared to 1990.

4.3. Feedback and calibration system

4.3.1. Amplitude feedback

A feedback system was designed to indicate when the simulated signal is bigger than expected without the need of external equipment. This system also had to aid in the adjustment of any detected amplitude deviations back into accepted ranges. The designed solution uses operational amplifiers in comparator configuration to compare the simulator's output signal to a threshold. LEDs connected to the output of the comparators are used to visually indicate when the simulated signal is above the aforementioned feedback. The working principle is explained next.

The positive comparators implemented compare their input and threshold signals and swing their output voltage towards the positive (V_{cc}) or negative supply (V_{ee}) according to:

$$V_{in} < V_{th} \rightarrow V_{out} = -V_{ee}$$

$$V_{in} > V_{th} \rightarrow V_{out} = +V_{cc}$$

Therefore, by connecting an LED to the comparator's output, this will shine when the simulated signal is above the threshold voltage. On the contrary, the LED will turn off when the generated signal stays below the threshold voltage. A series resistor to limit current flow into the LEDs was used ($R_8=R_{10}=R_{12}=1\text{ k}\Omega$ in Fig. 5.2).

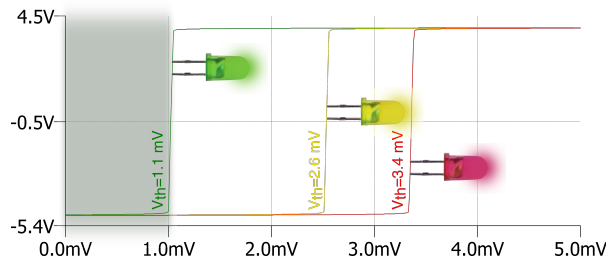


Figure 4.13: LTSpice simulation of positive comparator output for different inputs. Threshold voltages of 1.1 mV (green), 2.6 mV (yellow) and 3.4 mV (red).

Three different comparators were implemented, with 1.1 mV, 2.6 mV and 3.4 mV threshold signals. These thresholds were chosen based on the amplitudes that are correctly displayed by patient monitors. Usually, the default amplitude scale is used by healthcare workers. This displays signals up to 1 or 2 mV. Wider scales are available in most patient monitors, with the widest one displaying signals up to 5 mV in amplitude. A signal that exceeds the upper limit of the selected scale will appear saturated on screen (see Fig. 4.14).

During the simulator's design, efforts were made to keep the circuit simple and take the most advantage of Arduino's embedded functionalities. In this line, Arduino's 5 V output was used to provide the reference signals for the comparators. From this 5 V, the specific thresholds were obtained by means of resistor dividers. Thus, a threshold signal can be expressed as:

$$V_{th} = \frac{R_2}{R_2 + R_1} 5V \quad (4.3)$$

By making V_{th} equal to the different thresholds desired, the resistor values for each comparator were found. For the lowest threshold:

$$V_{th} = 1.1mV = \frac{R_2}{R_2 + R_1} 5V \rightarrow R_2 = 44, R_1 = 200k\Omega \quad (4.4)$$

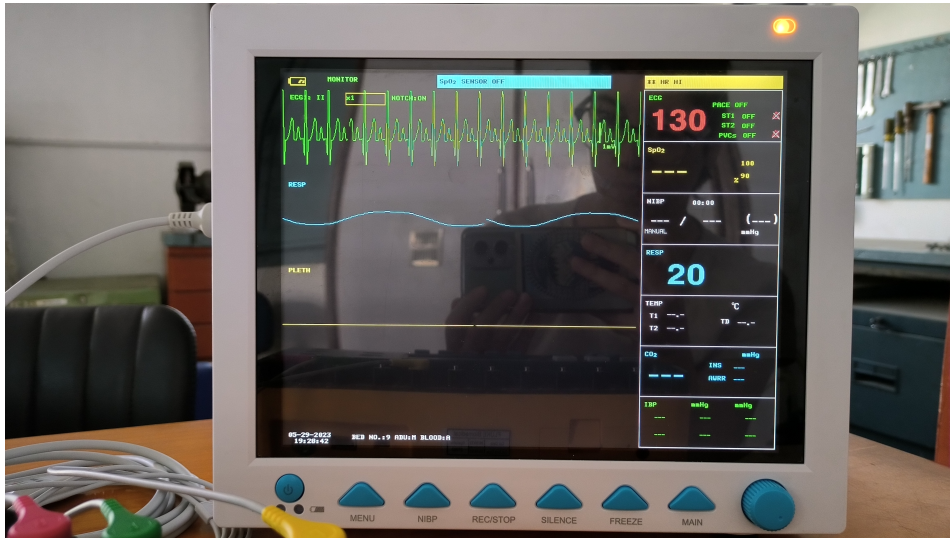


Figure 4.14: Patient monitor set on 1x scale displaying a 5 mV saturated signal on screen.

Similarly, for the medium threshold of $V_{th} = 2.6mV$:

$$2.6mV = \frac{R2}{R2 + R1} 5V \rightarrow R2 = 44, R1 = 84.2k\Omega \quad (4.5)$$

Finally, for a high threshold of $V_{th} = 3.4mV$:

$$3.4mV = \frac{R2}{R2 + R1} 5V \rightarrow R2 = 44, R1 = 64k\Omega \quad (4.6)$$

Theoretically, when the simulated signal is above the threshold plus the input voltage offset of the opamp ($75 \mu V$ for OP97), the LED connected to the opamp's output should shine. For this reason, low-input voltage offset OP97 opamps were chosen. LT Spice simulations were carried out to simulate the comparator's output behaviour (see Fig. 4.13). However, higher errors than expected were detected in practice. Quantifying the errors was challenging due to the difficulties of signal measurement. Given the limited availability of Zurich Instrument's lock-in amplifier, the designed feedback system was first designed based on theoretical values and then adjusted based on experimental behaviour of the LEDs. R4 was replaced by resistor values that defined theoretical output amplitudes (according to equation 4.2). The simulator was then switched on and the calibration mode was selected, checking whether the LEDs were on or off. In this way, the highest detected error was approx. 0.3 mV, but this was variable. This means that an LED was not shining when the output amplitude was set to coincide with its threshold, and only switched on when the output amplitude exceeded the threshold in 0.3 mV. Behaviour of the LEDs was found not always reliable, sometimes changing from on to off state (or vice versa) if cables were touched. These errors were probably attributable to leakage currents, noise and other non-idealities commonly present in breadboard-based systems.

4.3.2. Amplitude adjustment

Deviations of the simulator's output signal can be the result of resistance drift over time, component degradation, noise, temperature and humidity effects... Together with the amplitude feedback, a trimmer resistor was implemented to aid in the adjustment of amplitude deviations back into accepted ranges. For this, R4 in Fig. 5.2 was replaced by a trimmer resistor (see Fig. 4.15).

Trimmers are variable resistors that can be precisely adjusted by turning a screw, reason why they are commonly used for calibration purposes. Resistance between one of the external pins and the central one is adjusted by turning the screw, what moves the wiper, modifying the resistive path length that current needs to go through. In the simulator, a 1k trimmer resistor was employed. By means of a common screwdriver (~ 1.4 mm flat tip) the resistance of the trimmer resistor can be adjusted between a level close to 0Ω and $1 K\Omega$. Turning the trimmer's screw clockwise reduces the amplitude of the

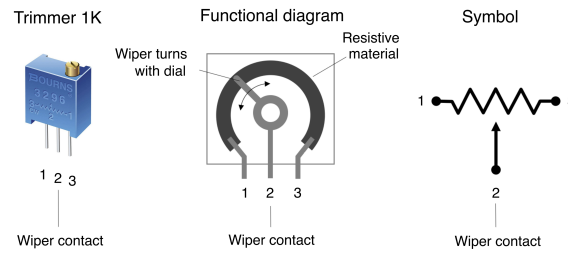


Figure 4.15: Trimmer resistor appearance, functional diagram and electric symbol. The path that the current has to cover between the external pins and the wiper contact defines the resistance between the external pins and the central one. By turning the trimmer screw, the wiper moves, achieving precise resistance adjustment.

a) Forward Current & Forward Voltage b) Luminous Intensity & Forward Current

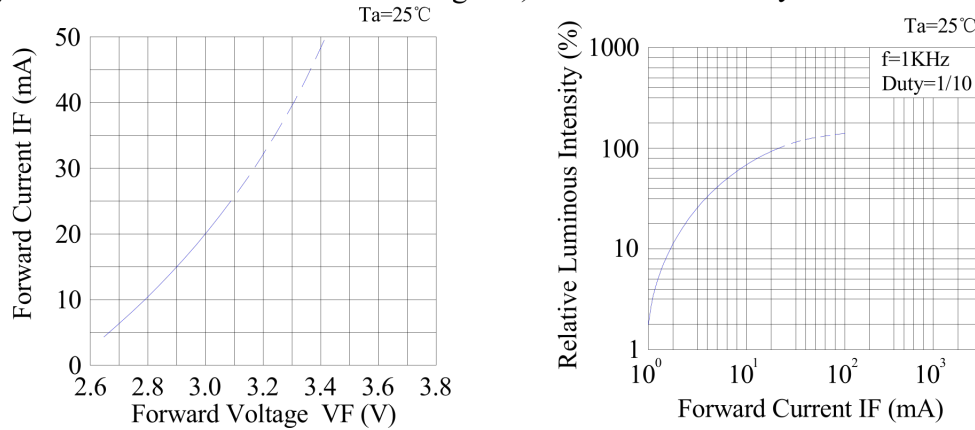


Figure 4.16: Characteristic curves of 5mm blue LED (504BC2E-B4-2BE) with minimum forward voltage 2.6 V. From: [22].

output signal, while turning it anti-clockwise increases it. This defines theoretical output amplitudes between 0 and 10 mVpp. Results on the use of the amplitude feedback and adjustment system are presented in [Feedback tests](#).

4.3.3. HR feedback

HR feedback to indicate frequency of the generated R peaks was designed. Given that amplitude was not of importance in this case, the signal generated after the LPF could be used to provide feedback on. This gave access to a signal high in amplitude, able to produce a response on a common LED without the need of amplifying or modifying it in any other way. Additional hardware required included an opamp in voltage follower configuration (Op3 in Fig. 5.2) and a resistor in series for current limiting. The voltage follower was required to create a high-input impedance after the LPF, i.e., to maximize the LPF output voltage transfer to the next stage. This resulted into an affordable solution, with standard 5mm LEDs having prices below 0.20€ in Europe and around 2-3 NPRs in Nepal.

The described HR feedback system was based on the light-current response of LEDs (Fig. 4.16). Above a threshold voltage that depends on the LED model, the current through an LED increases following an exponential curve that varies with the LED model (Fig. 4.16a). A blue LED with a threshold of 2.6 V was used [22]. The generated light intensity is directly proportional to the current flowing through the LED (Fig. 4.16b). This makes the blue LED respond to the ECG signal by shining every time the signal reaches the R peak and switching off the rest of the time. This is because R peaks reach 5 V and decay below 1 V at every heart beat. More details on LM358 and why it was chosen can be found in Appendix J.

By counting the number of LED blinks in one minute, the generated HR could be estimated. Although this method has a limited accuracy to precisely determine the HR simulated, it is very reliable to know whether a signal is being simulated or to distinguish between calibration or simulation mode. Moreover, if the simulation was limited to three different HRs representative of a low, healthy and high level, the

visual feedback would be highly reliable to distinguish between them. HR measurements based on LED blinking performed by a single user showed a maximum absolute deviation of 5 bpm from the target HR. For this application, a LM358 opamp could be used, which is cheap and common in Arduino applications. Compared to the other stages of the circuit, which require high precision due to millivolt-range input amplitudes, this opamp could have standard performance. This is because its input signal ranges from 0-5 V, and the bandwidth of interest is easily achievable with standard opamps.

4.4. System overview

The Prototype development process explained above resulted in the system presented in Fig. 4.17. This details the specific tools and methods implemented to fulfil the functions identified during the **Design process**.

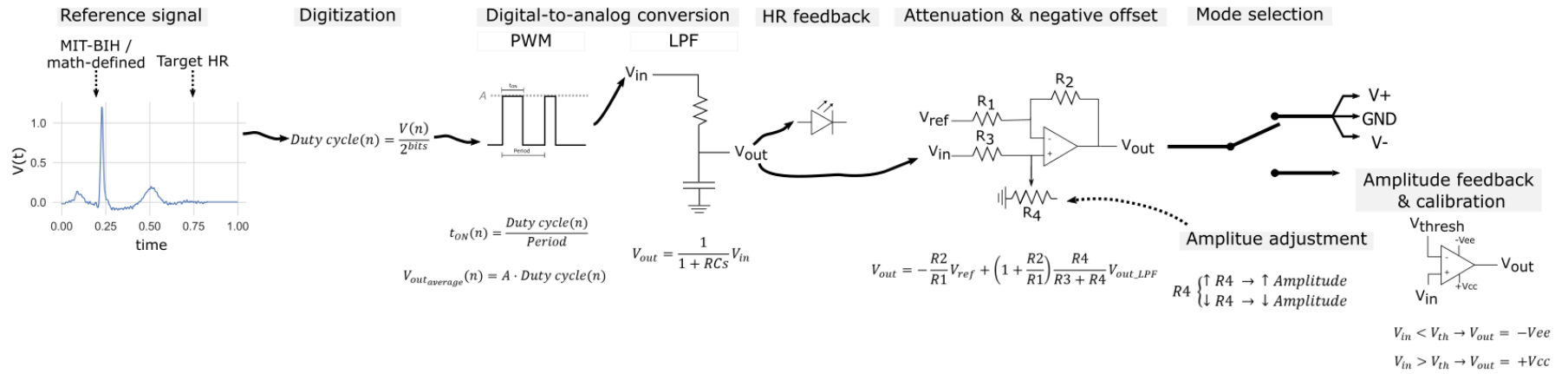


Figure 4.17: Schematic diagram of the technical solutions implemented. Gray boxes highlight the fulfilled functions. Solid arrows indicate signal flow.

5

Final prototype

The **Design process** defined the requirements that an ECG simulator for LRs had to fulfil, and selected the most suitable methods to solve them. Next, the chapter on **Design choices** further specified the tools used to implement given methods. However, it was not until diving into the **Prototype development**, that the implemented solutions could be fine-tuned and optimised, thanks to the experimental insights gained during hands-on work. As a result of this evolution, a **Final prototype** was reached, which is presented in this chapter. The Prototype description is divided into **Software** and **Hardware**.

5.1. Software

5.1.1. Reference signal

A Python code that processes an ECG recording from **MIT-BIH Arrhythmia Database** (recording 101) was written. This database is free and available online [26]. The original MIT-BIH data was recorded with a 360 Hz sampling rate and 11-bit resolution over a 10 mV range, defining a step size of 2.7 msec in time and 0.1 mV in amplitude.

The ECG recording 101 presented in Fig. 5.1 was used as a 'healthy' ECG waveform. First, the developed Python code extracts the first waveform from the dataset. Then, the waveform is extended or cropped to simulate different heart rates. Next, the datapoints are scaled to 9-bits resolution (following Eq. 5.1) and resampled to 2 msec/sample. After these steps, each datapoint from the ECG waveform is digitally defined by an integer between 0 - 511. These values are stored in the Arduino code responsible for repeating that waveform over time, explained in the next section.

Before the use of an ECG recording as a realistic reference signal, the simulator was also tested with a synthetic signal extracted from [23] and a mathematically-defined signal based on the equations described in [45]. This exemplifies how the reference signal can be easily changed by uploading it into the Arduino board.

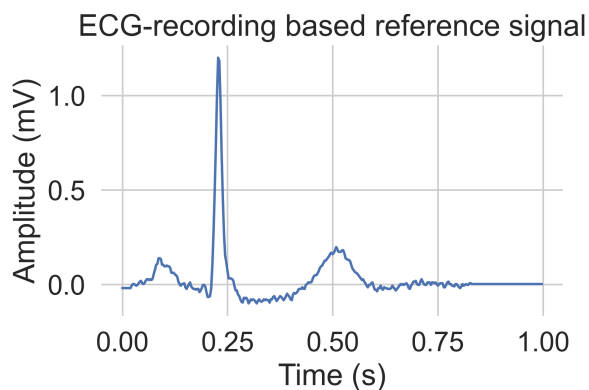


Figure 5.1: ECG recording number 101 from MIT-BIH database used by the simulator as reference signal.

5.1.2. Arduino code

An Arduino code was written to generate a PWM signal with 9-bit resolution, 31.25 kHz frequency and varying duty cycle. The code can be consulted in Appendix C. The duty cycle of the generated PWM signal is updated every 1997 microseconds with a new datapoint from the waveform defined in the previous step. Thereby, the n-th duty cycle is defined by:

$$Duty\ cycle(n) = \frac{V(n) + |V_{min}|}{V_{pp}} (2^{bits} - 1) = \frac{V(n) + |V_{min}|}{V_{pp}} 511 \quad (5.1)$$

where $|V_{min}|$ is the lowest voltage of the MIT-BIH signal in absolute value, and V_{pp} is the peak-to-peak voltage. Once all the datapoints have been simulated, simulation starts again from the first one. More information on the simulation rate can be found in 4.2.2.

5.2. Hardware

5.2.1. Signal generation

Signal generation is controlled by an Arduino UNO, which outputs a PWM signal of duty cycle that varies according to the reference signal. The PWM signal is then smoothed by an RC filter, achieving the generation of a DC output that reproduces the reference ECG waveform. In a second stage, the generated signal is attenuated to the millivolts range and provided with a negative offset. The hardware involved in these steps is described next.

Arduino UNO

Arduino UNO features an 8-bit ATmega328P microcontroller, a 16MHz ceramic resonator used as system clock, 6 PWM outputs and 32KB of flash memory. This board can be powered from the USB connector (5 V), the in-built DC power jack (7-12 V) or the Vin pin (7-12V). Other useful features for this work are the 5 V and 3.3 V output voltages available by the on-board voltage regulators.

Arduino was chosen due to its simplicity of programming, open-source characteristics and vast online community. Additionally, in-built features such as voltage regulators and safety circuits simplify the rest of the electrical circuit, which can focus on signal generation and feedback. Due to its popularity, Arduino UNO is increasingly available for purchase online and popular in educational institutions, also in LMICs. In Nepal, this board can be purchased online for 1,700 NPR at Daraz.com, the leading online marketplace.

PWM and low-pass filter

Digital-to-analog (D/A) conversion of the reference signal is achieved by modulating the duty cycle of an Arduino PWM output and smoothing out the generated signal by means of a low-pass filter (LPF). A LPF of 159 Hz cut-off frequency ($R=500\ \Omega$, $C=2\ \mu F$) was designed to meet the common ECG monitoring bandwidths of 0.05 - 150 Hz [20, 30]. The exact frequency of 159 Hz is defined by the electronic components available. With the cut-off frequency of the LPF defined by the application, ATmega was programmed to use a PWM resolution of 9 bits and increase the carrier frequency to 31.25kHz, in order to increase the resolution of the generated DC values. ATmega programming is explained in Appendix E. The use of ATmega's fast 9 bits PWM mode achieved a D/A conversion that met the requirements to simulate an ECG signal while keeping the system complexity and cost low.

Appendix F describes the working principle of this D/A conversion method. Appendix G explains the LPF design and discusses the alternatives to PWM and LPF for D/A conversion.

5.2.2. Amplitude attenuation and negative voltage offset

A weighted difference amplifier (Op1 in Fig. 5.2) was used to reduce the signal in amplitude and add a negative offset, to closely resemble the signal in Fig. 5.1. Resistors R2 (175 Ω) and R1 (3.3MEG Ω) are responsible for shifting the signal at the positive input terminal of Op2 (V_2+) -0.1mV. Resistors R4 (130 Ω) and R3 (500k Ω) reduce the signal with a gain of $\frac{26}{100000}$. For more detailed calculations refer to Appendix H. After this stage, the signal has an amplitude from -0.1 to 1.2 mV, or 1.3 mVpp.

5.2.3. Signal inversion

To be able to test an ECG or patient monitor, the simulator needs to provide at least three output signals. One of the signals reproduces the ECG reference waveform presented in Fig. 5.1. This is the result

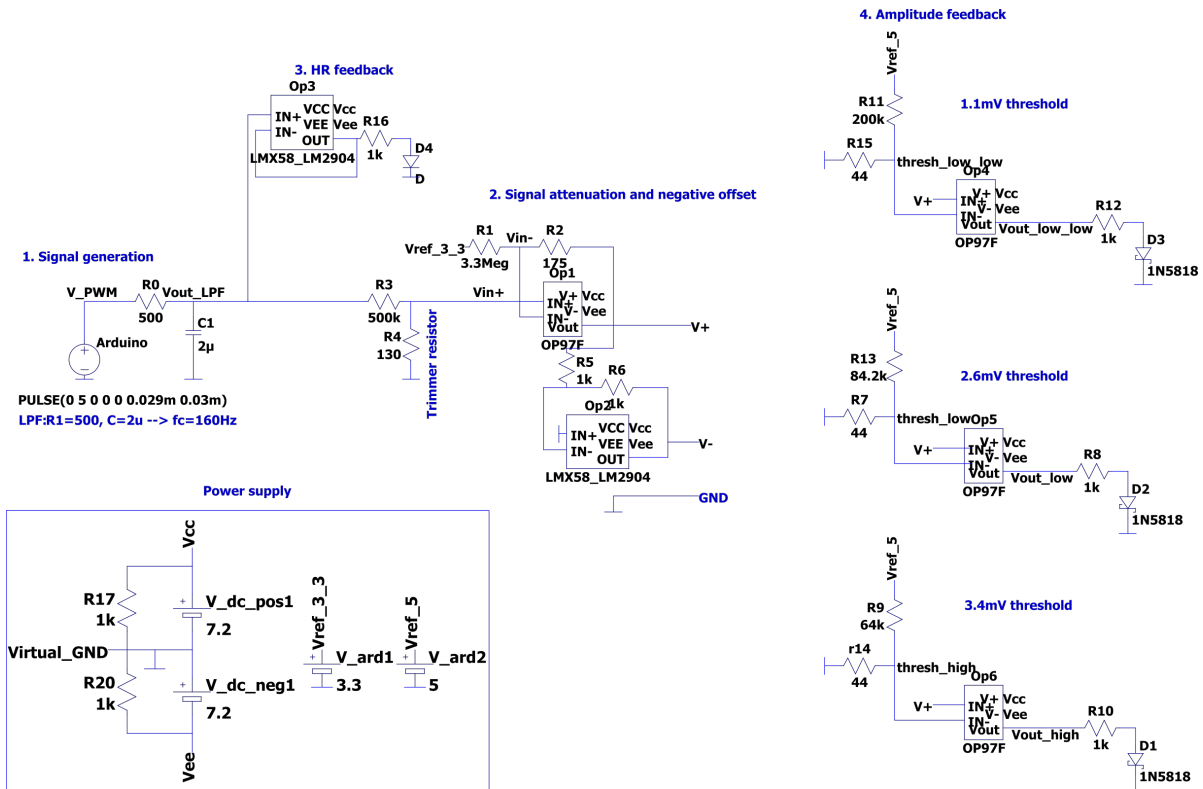


Figure 5.2: Electrical schematic of the prototype ECG simulator for LRs.

of the previous step and is denoted as V_{out1} in the electrical schematic. A second signal is obtained by inverting the first one (V_{out2}). This is achieved by means of an inverting amplifier (Op 2 in Fig. 5.2) with a gain of -1 ($R5=R6=1k\Omega$). This was chosen as an easy, affordable solution to generate a second signal. The third output corresponds to ground (GND), common to the whole circuit. These outputs can be connected to three leads of an ECG cable which is in turn connected to a monitor. The monitor then interprets the connection as a 3-lead ECG signal. In case of a cable with more than 3 leads, all of them can be tested by connecting them to the outputs, 3 connectors at a time. More details on the simulator outputs are provided in Appendix I.

5.2.4. Feedback

Amplitude feedback

A feedback system was designed to serve as an indicator of the output amplitude without the need for external tools. The designed system visually indicates when the simulated analog signal is bigger than some threshold values. Such deviation in amplitude can be the result of resistance drift over time, component degradation, noise, temperature and humidity effects... At the same time, this feedback system aids in amplitude calibration of the deviated signal. The challenge was to not modify the output signal and to use available components that could respond to very small amplitude variations. This was achieved by the use of operational amplifiers with low input offset voltage (for details see Appendix J).

Three operational amplifiers OP97 were used in a positive comparator configuration with threshold voltages of 1.1, 2.6 and 3.4 mV. Their output was connected to three LEDs, of green, yellow and red color, respectively. When the generated output is above a threshold voltage, the associated LEDs will shine. The theoretical accuracy with which OP97 can detect a deviation from the threshold values depends on its input offset voltage (max. $75 \mu V$ [3]).

Amplitude checks must be carried out during the simulation of a DC value. For this, a slide switch allows to change between the simulation of an ECG waveform and that of a square signal. Additionally, a rotary switch reused from old electronics equipment is employed to check one LED at a time. This reduces current consumption.

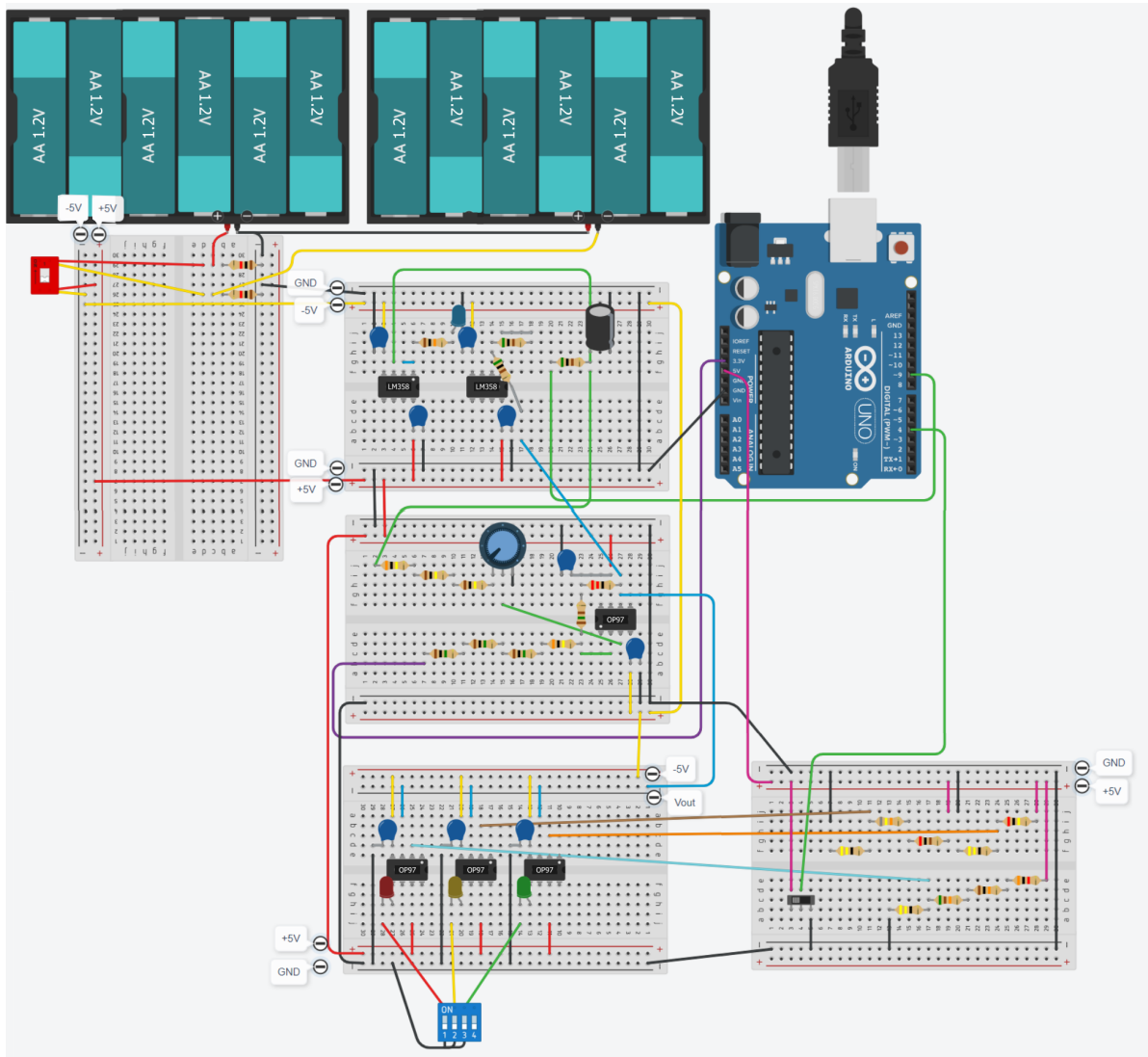


Figure 5.3: Wiring diagram of the prototype ECG simulator for LR.

HR feedback

A feedback system was designed to provide visual real-time information on the HR generated by the simulator, without the need of external equipment and without modifying the generated signal.

HR feedback is based on a blue LED that responds to the amplitude peaks of the generated signal after the LPF. By measuring the number of blue blinks per minute, the simulated HR can be approximated. Additional components required include an operational amplifier (LM358) in voltage follower configuration for impedance matching, and a series resistor for current limiting.

5.2.5. Calibration

Variations of the output amplitude can happen due to degradation of the electrical components, offsets or noise. When that happens, the proposed amplitude feedback system in combination with a trimmer resistor provides a method to adjust the output signal amplitude back to the designed ranges without the need of specialized external equipment. The only tool required is a screwdriver (~ 1.4 mm flat tip). With the current configuration, when the output exceeds 3.4 mV, the red LED will shine. By turning the trimmer's screw clockwise with the help of a small screwdriver, the amplitude of the output signal can be reduced. Once the amplitude is approximately below 3.4 mV, the red LED will go off. Checking the yellow light, once this goes off we know the signal is below 2.6 mV approximately. If the green LED is on, we know the amplitude is between 1.1 mV and 2.6 mV. Thus, the current configuration allows for tuning of the signal amplitude at a range suitable to be displayed in the default scale of patient

monitors (up to 3 mV). The red LED serves as an indicator of the direction in which the amplitude is being modified and also indicates when the signal is too high to be properly displayed at a default scale. The threshold levels can be easily modified by changing the associated resistors. Videos that show how to use the feedback system are provided in this [link](#).

5.3. Cost and other components

The Prototype was built on breadboards, and using through-hole components gathered from an educational electronics workshop. However, all the components are available at Daraz.com, except the opamps used, which can be purchased at himalayansolution.com (LM358) and Aliexpress.com (OP97). A list of components cost and purchase link is available at Appendix K. The total cost of the Prototype is around 5000 NPR.

Frankie output connectors consist of female banana jacks, which are commonly used as input pins in electronic equipment such as oscilloscopes or function generators. The ones used consist of two colored plastic parts around a metallic screw, and two fixing nuts. These were retrieved from an educational electronics workshop and are available for 0.4 €/u. at Tinytronics.nl (Netherlands) or 93 NPR/u. at Daraz.com (Nepal). The banana jacks were mounted inside-out so as to have the screw protruding to the outside. In this way, different typologies of ECG electrode cables could fit, as Fig. 5.5 shows.

The casing used is a standard carton box, of dimensions 20x25x3cm (see Fig. 5.4). Similar boxes are available at Daraz.com for 34 NPR. This is a light, affordable, easily available solution. The use of carton provides some ventilation to the electronics while being easy to customize with very basic tools, convenient for prototyping. Holes for the switches and connectors were performed using a cutter, and components were fixed using double-sided foam tape.

5.4. Power supply

Rechargeable AA batteries (Ni-MH, 1.2 V, 2450 mAh) were used to power the Arduino and the opamps. Arduino power requirements via the DC jack connector are 7-12 V and a maximum of 800 mA. Power supply for the opamps had to allow them to handle the Arduino's output signal (min 0 V, max 5 V) and provide negative output voltages at the last stage. To simplify the power circuit, twelve 1.2 V AA rechargeable batteries and a virtual ground were used to obtain +7.2 V and -7.2 V from them. Six 1.2 V AA batteries connected in series were used for the positive supply and six 1.2 V AA for the negative one. The Arduino board was powered by the positive supply, while the opamps were powered with the positive and negative supplies. In this way, the DC outputs of 5 V and 3.3 V from the Arduino were kept free to be used as reference voltages in the circuit. One 1 k Ω , 1 W, 1% tolerance resistor was used for power dissipation at each power supply circuit. This circuit is visible at the lower left side of the electrical schematic depicted in Fig. 5.2.

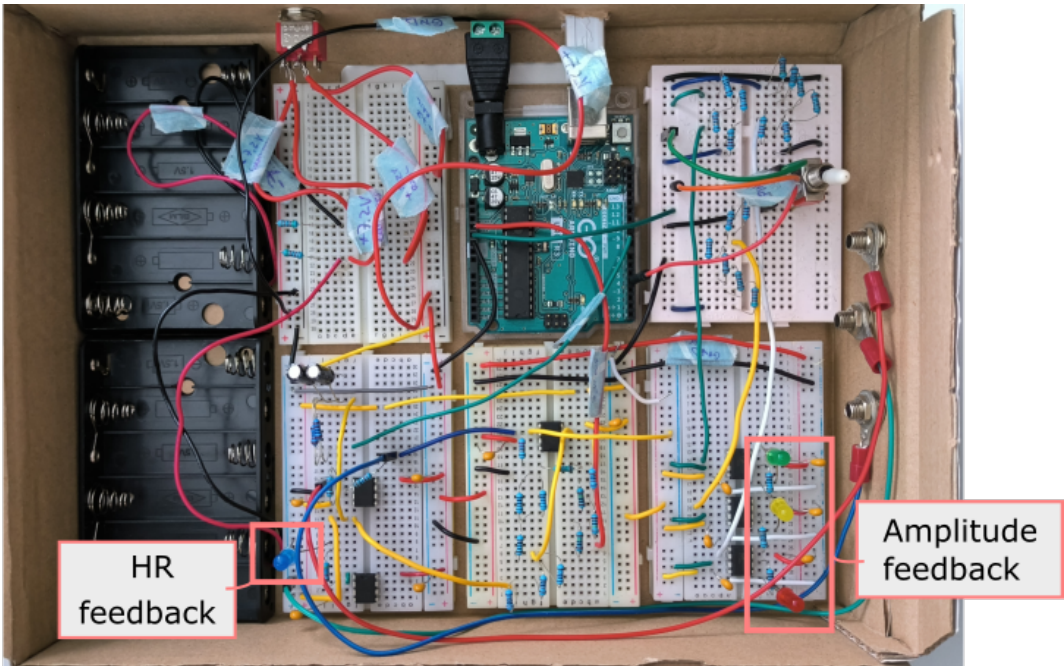


Figure 5.4: Images of Frankie's outer casing and inside electronics. The white cable is used to connect the Arduino to a laptop when the user wants to modify the simulated waveform.

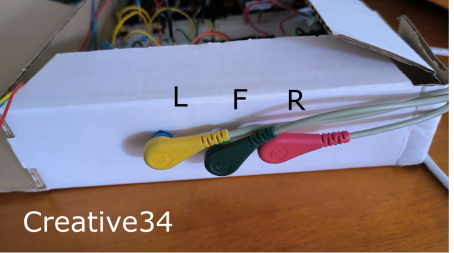


Figure 5.5: Connection of 6 different patient monitors to Frankie and one example of connection to ProSim 8.

6

Prototype evaluation

This chapter gathers the performed on the **Final prototype** to evaluate its accuracy and functionality. Tests to check that the Prototype performed as intended were carried out. These include basic functionality tests, testing of the feedback and amplitude adjustment system, and comparison to a healthy subject ECG. The rest of tests checked how well the Prototype worked. For this, HR, amplitude and waveform accuracy were studied.

This chapter presents the tests in the order in which they were performed. First, **Testing on Green Pastures Hospital patient monitors** is described. Then, tests using **Zurich Instrument's lock-in amplifier** at TU Delft are presented. Lastly, the Prototype's energy consumption is characterised. The information provided by the tests leads to a discussion on **Design requirements assessment**.

6.1. Testing on Green Pastures Hospital patient monitors

6.1.1. HR checks

HR simulation was tested on 5 different patient monitors at the Green Pastures Hospital (Pokhara, Nepal). The results can be seen in Fig. 6.1. With a simulation rate set to 1997, only 7 out of 21 HR measurements showed some deviation from the reference value. All deviations corresponded to a 1 bpm below the reference. The highest relative error was 1.42% and corresponded to 70 bpm (data can be found in Appendix N.1). In Fig. 6.2, ProSim 8 was tested using 6 patient monitors and accurately simulated all the HRs tested.

6.1.2. Amplitude checks

Amplitude checks were carried out on 6 patient monitors available for testing at the Green Pastures Hospital (Pokhara, Nepal). First, the simulator was connected to the patient monitor and set to simulate a target HR and amplitude. After a minute of stabilizing time, a picture of the monitor screen was taken. Next, the simulator was exchanged by ProSim 8, set to simulate the same target HR and amplitudes. ProSim's wave group was set to NSR (Adult), axis to intermediate and ST deviation to off. After a minute, a picture of the monitor's screen was taken. The absolute errors are visualized in Fig. 6.3. The IEC 60601-2-27:2011 [18] accepted accuracy for patient monitors waveform reproduction is also displayed (dashed line). Mean absolute error accross all measurements was $0.26 \text{ mV} \pm 0.17$. In terms of relative error, this was 13.5%. Raw data can be found in Appendix P.

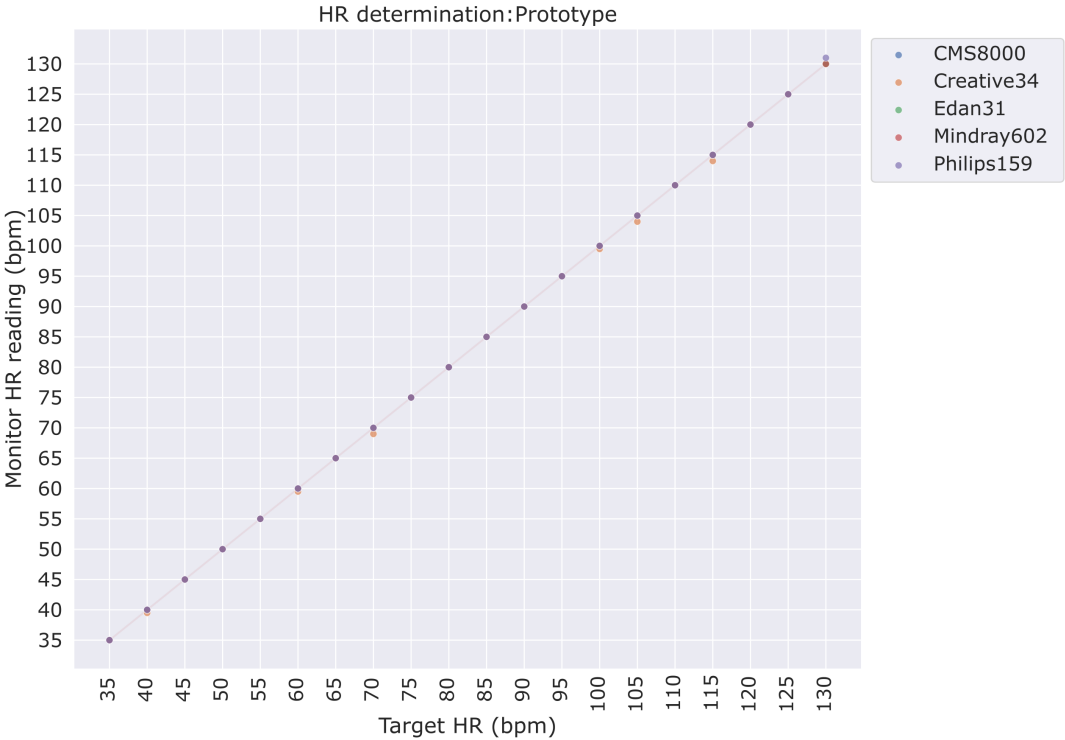


Figure 6.1: Comparison of monitor readings vs target HRs when using Prototype as simulator. Simulation rate was set to 1997 microseconds and tested on 5 patient monitors.

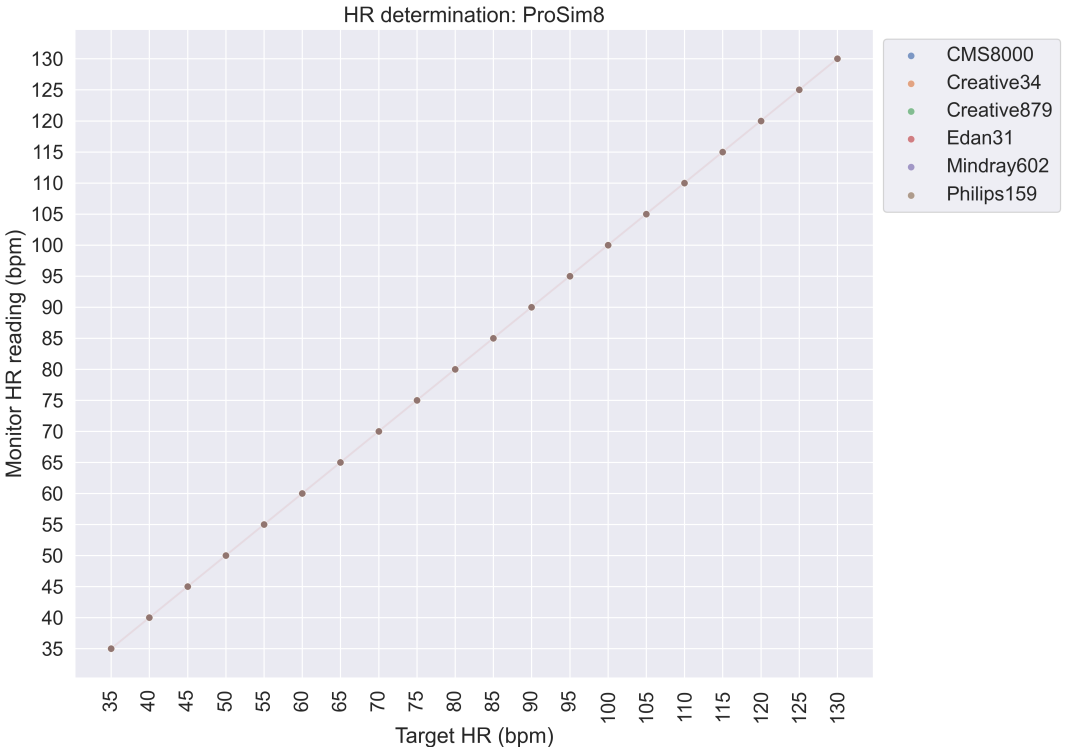


Figure 6.2: Comparison of monitor readings vs target HRs when using ProSim 8 as simulator. ProSim 8 was set to NSR (Adult) waveform, intermediate axis and ST deviation turned off. Tested on 6 patient monitors.

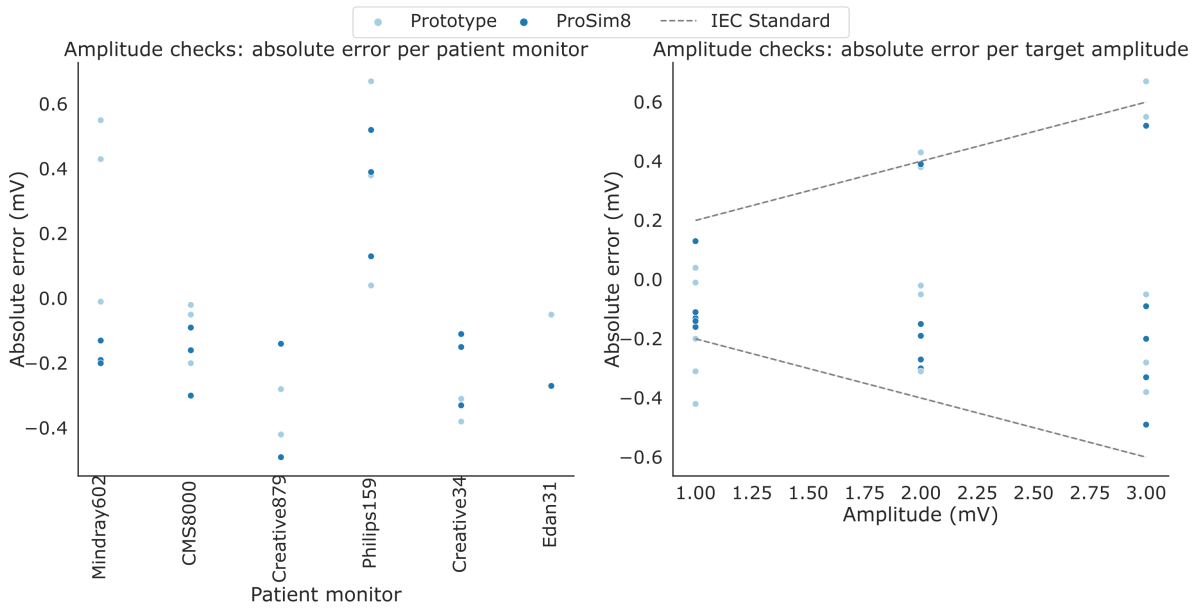


Figure 6.3: Accuracy of amplitude simulation with respect to target values. Four monitors were tested for 1, 2 and 3 mV simulation. Creative879 patient monitor was tested only for 1 and 3 mV. And Edan31 was tested for 2 mV only. All tests were performed at 60 bpm target HR except two at 130 bpm (Mindray602, 3mV, ProSim 8; Creative34, 2mV, ProSim 8). This differences in testing conditions resulted form limited availability of the patient monitors or image defects.

6.1.3. Functionality tests

Functionality tests consisted on checking the ability of the simulator to generate a signal that can be detected by patient monitors. Six different patient simulators currently under use at Green Pastures Hospital (Pokhara, Nepal) were connected to the simulator using their own electrode cable. The patient monitors differed in manufacturer, price range and purchase year, their details collected in Appendix O. Fig. 6.5 shows each patient monitor successfully displaying an ECG waveform and HR when connected to Frankie, as well as the equivalent connection to ProSim 8. Given that Frankie only has three outputs, in those cases in which the ECG cable had 5 electrodes, two were left unconnected.

The use of banana female jacks as connectors was successful in all cases, although some connections were more robust than others. Snap electrodes fit really well to the end of the screw, while alligator connectors could wrap correctly around it. However, if the cables were moved, this movement was transferred to the connection point, resulting in noise at the signal. The design of ProSim 8's pins provided a more robust connection thanks to a widening of the pin towards its end (see Fig. 6.4). This offers some resistance against the movement of both alligator and snap electrodes, keeping the contact point more stable. That design is inspired in the shape of ECG electrodes. ProSim 8's pins also have a hole through them which can be used to connect other kinds of cables, but this was not needed for the monitors tested.



Figure 6.4: ProSim 8 features specialized connectors the tip of which resembles ECG electrodes.

Prototype



ProSim 8



Figure 6.5: Connection tests: Prototype and ProSim 8 connected to 6 different patient monitors at Green Pastures Hospital (Pokhara, Nepal). Simulators were set to generate a 60 bpm, 2 mV ECG-like waveform in all cases except for Creative879 (60 bpm, 3 mV).

6.1.4. Feedback tests

Although the current feedback system design was not suitable for precise amplitude characterization of the output signal, its usage in combination with a trimmer resistor was tested to successfully bring an amplitude-deviated signal back into a range suitable for its visualization at a patient monitor screen.

This [link](#) contains recorded examples of feedback use and amplitude adjustment. One video presents the case of a signal that is 'too high' in amplitude, and how the feedback system is used to bring its levels back to the expected ones. The other video shows the case of a 'too low' amplitude and how the feedback system is used to increase it until it is visible at a patient monitor with 1x scale. The patient monitor used in both cases was CMS8000 (Contec Medical Systems).

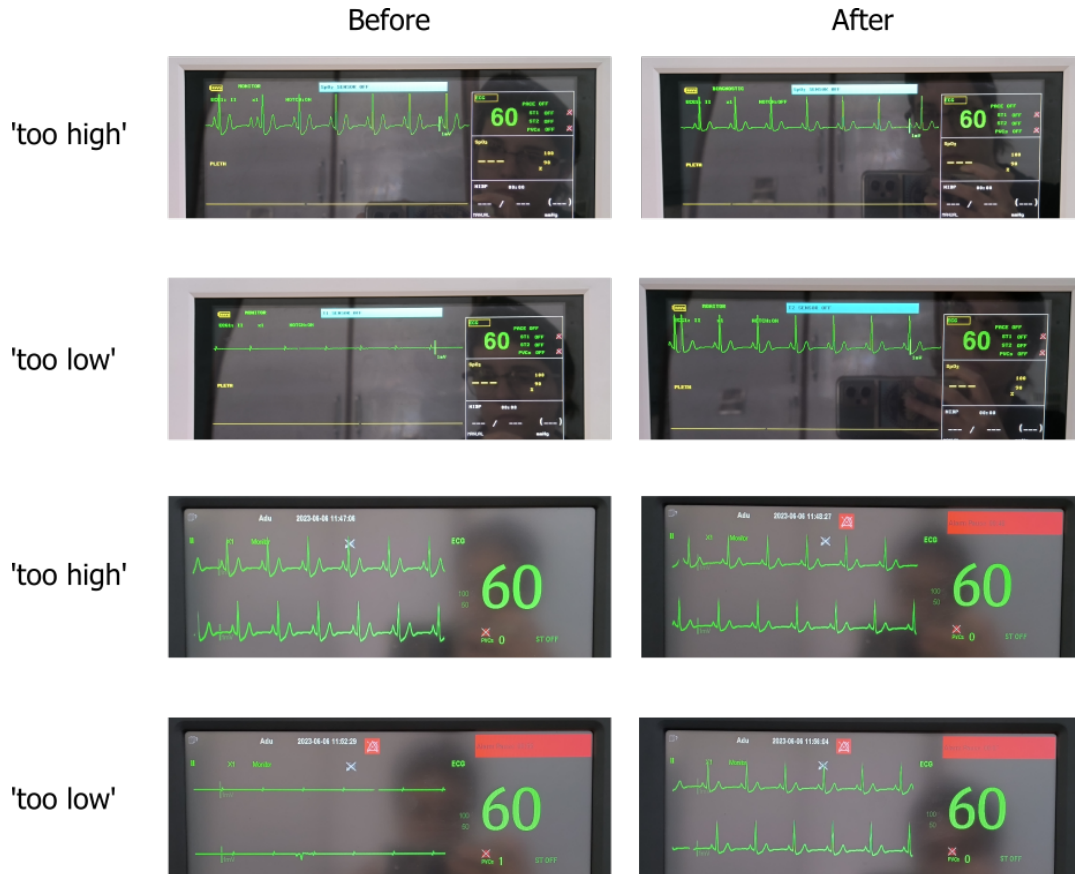


Figure 6.6: Examples of feedback use before and after adjusting amplitude deviations. Amplitudes that are 'too high' and 'too low' to be visualized correctly at a patient monitor (scale 1x) were brought back to acceptable levels by means of the feedback system designed, which does not require any external equipment besides a small screwdriver. Patient monitors used were CMS8000 (first two rows) and Mindray602 (last two).

6.1.5. Comparison to a healthy subject

A patient monitor (Philips159 in Appendix [O](#)) was connected to the Prototype first, a healthy subject next, and ProSim 8 later. Measurements were performed one right after the other, and the same three electrodes from the same cable were used. The patient monitor was set to 3-lead ECG detection in the three cases. The Prototype and ProSim 8 were set to a HR of 60 bpm, while the subject's detected HR was of 74 bpm. The results can be seen in Fig. [6.7](#).

6.2. Zurich Instrument's lock-in amplifier

Zurich Instrument's UHFLI 600 MHz lock-in amplifier was used to record the simulator's output signal. The lock-in amplifier was connected to outputs V+ and GND of the simulator. Measured data was downloaded and post-processed using Python. The results are presented in Fig. [6.8](#).

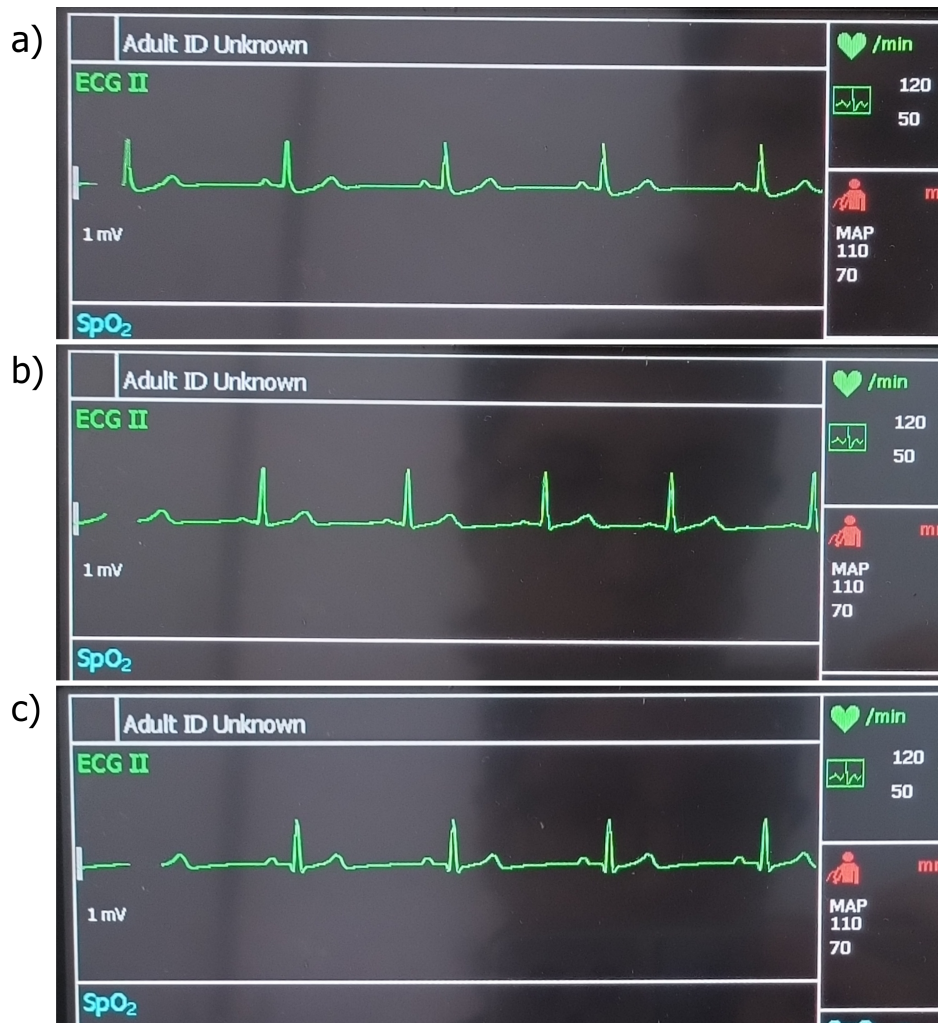


Figure 6.7: Qualitative comparison of Philips patient monitor readings when connected to a) Prototype ECG simulator for LRs set to 60 bpm; b) healthy subject; c) PorSim 8 commercial simulator set to 60 bpm.

Zurich Instrument's lock-in amplifier has some particularities that complicate its use and interpretation of the measurements. First, the scope only displays positive voltages. According to the measurements done, the device seems to add an offset to the input signal in order to bring it to fully positive values. To ease comparison, these measurement offsets were removed in a post-processing step using Python. As a result, the minimum value of the measured from Fig. 6.8 signals sits at 0 V. For the measurements, an input impedance of $1\text{M}\Omega$ was selected, corresponding to the oscilloscope's highest option. This maximizes signal transfer and is in the same order of magnitude as the minimum input impedance specified for patient monitors, of $10\text{M}\Omega$ [31].

Second, the lock-in amplifier has two modes of operation: scope and plotter, each of them with a wide range of options to be controlled from the LabOne software. The scope mode displays the raw detected signal, in this case sampled at 27 kSa/s . The raw data is shown with light purple colour in Fig. 6.8a). Given the visible noise in the signal, this was post-processed in Python for better characterisation of its amplitude. A Butterworth 1st order low-pass filter with a 150 Hz cut-off frequency was implemented using `scipy.signal` Python module. The cut-off frequency was chosen to simulate the one implemented by ECG monitors in monitoring mode. The filtered signal is shown in dark purple in Fig. 6.8a), with an amplitude of 1.70 mVpp . This represents a positive error of 0.3% (or $+0.4\text{ mV}$) with respect to the reference signal (of 1.30 mVpp). This measurement was performed on a single waveform.

The oscilloscope's plotter mode was also used. This offers various features to digitally filter the raw detected signal. A digital 1st order low-pass filter with cut-off frequency of 150 Hz was selected. The order of the filter was kept low to distort the signal as least as possible. A sampling frequency of 500

Sa/s was selected, as this corresponds to the simulation frequency used by the simulator Prototype. The result in Fig. 6.8b reveals a waveform completely filtered from the noise present in the scope signal (Fig. 6.8a). However, the plotter mode seems to alter the amplitude of the signal, which reached 2.36 mVpp. This is 1.81x bigger than the reference signal (of 1.30 mVpp). Amplifying the reference signal by a factor of 1.81 shows that the measured waveform closely follows the reference one (green dashed line in Fig. 6.8b).

Although both measurements (using the scope and using the plotter) were carried out on the same day, on the same signal and one right after the other, the signal displayed by Zurich Instrument's plotter has an increased amplitude with respect to the one measured using the scope method. This is especially visible at the R peak. This could have to do with the filtering method implemented by the oscilloscope. The limited availability of Zurich Instrument's oscilloscope and its complex functioning made its use for amplitude accuracy assessment not possible with the time available. However, the results presented in Fig. 6.8b allowed to confirm the high waveform accuracy of the simulator.

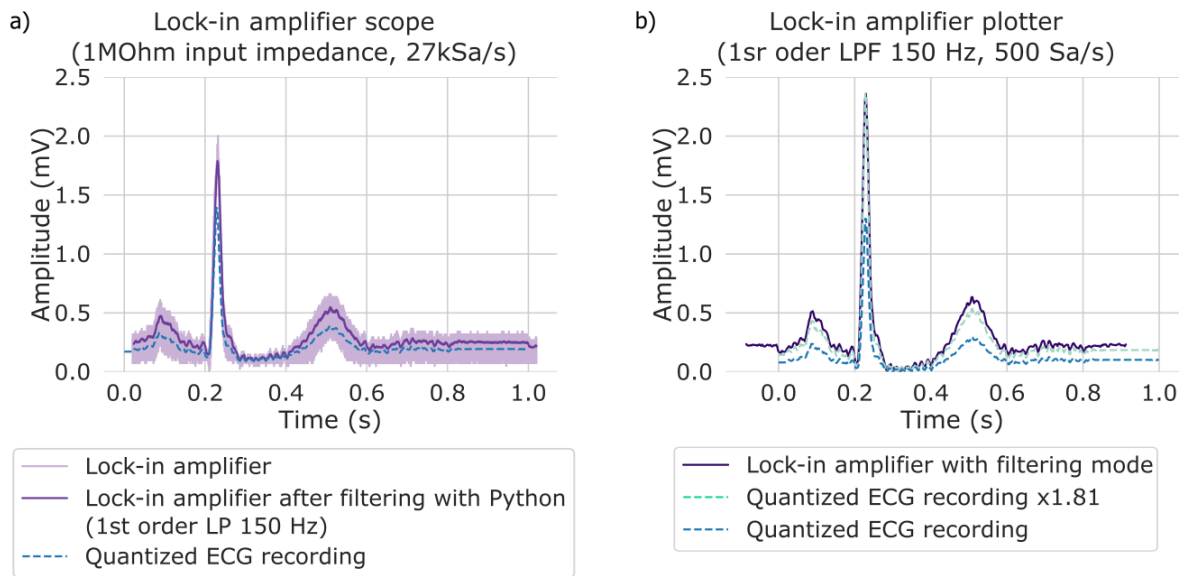


Figure 6.8: Zurich Instrument's oscilloscope readings of generated signal compared to reference. a) Generated signal as recorded in scope mode and 27 kSa/s (light purple). In dark purple, same signal after post-processing using low-pass filter (1st order, $f_c = 150$ Hz, Butterworth) in Python. b) Signal measured using plotter mode (dark purple) and applying low-pass filter ($f_c = 150$ Hz, 500 Sa/s). Dashed green line shows reference signal amplified 1.81x. Input impedance was set to $1\text{M}\Omega$ in both cases. Measured offset was removed in Python to bring signals to 0 V level.

6.3. Energy consumption

Measurements on the final version of the Prototype revealed a total of 58 mA drawn from the positive supply and 12 mA from the negative one. This corresponds to the schematic presented in Fig. 5.2 and simulation settings of ECG waveform and 60 bpm of heart rate. According to the measurements, the six 2450 mAh batteries of the positive supply will last for 42 h and 14 min approximately. The negative supply will last for around 204 h and 9 min. A current of 0.2 mA at the power dissipation resistors ($1\text{k}\Omega$) was measured, corresponding to a power dissipation of $40\ \mu\text{W}$. Although this could be perfectly handled by $1/4$ W resistors, these were found to get slightly warm during continuous use. In order to avoid any heating of the circuit and taking into account the high temperature conditions in Nepal, resistors of 1 W power rating were chosen. Current drawn from the Arduino 3.3 V and 5 V pins was $1\ \mu\text{A}$ and $158\ \mu\text{A}$, respectively. This is well below the maximum current of 1 A that the 5 V output can provide, and the 150 mA limit of the 3.3 V output.

6.4. Design requirements assessment

Ref.		Requirement	Discussion	Future improvements
GENERAL				
G.A.1.	AFFORDABILITY	More affordable than commercial simulators (ProSim 8, 7071€; ProSim 2, 2260,35€; PS410, 1160€; SimCube 3390€). Around the price described in literature (Machado-Gamboa, 2018: 143€; Varela Guzman, 2021: 120€; Vergara Gil & González Vargas, 2021: 200€).	The only manufacturer that can be purchased in Nepal is Fluke Biomedical. In 2021, ProSim 8 was purchased for 1,407,867 NPRs by Green Pastures Hospital. With a cost around 5,000 NPRs, Frankie's parts and components cost represents a 0.36% of ProSim's.	
G.A.2.		Materials should be locally available, possible to order or common components that can be reused from non-working devices. Daraz (https://www.daraz.com.np/) might be used as an indicator for local availability in Nepal. Aliexpress and component reuse might be considered for availability in SSA, electronic shops not common.	All components required to build Frankie can be ordered online at Daraz.com , except the operational amplifiers used, which can be purchased at himalayansolution.com (LM358) and Aliexpress.com (OP97).	
G.U.1.	USABILITY	Easy to use without manual, intuitive	The current version of the prototype requires uploading a new waveform to the Arduino when the simulated HR wants to be modified. ON/OFF switch is intuitive to use. Feedback system needs some training to be used.	Use of a switch to select from different HRs stored in Arduino memory was successfully tested, but this solution requires data storage extension before it becomes fully functional.
G.U.2.		Open source	Arduino open-source platform. ECG recording used as reference signal extracted from free online MIT-BIH database cited by IEC standards.	
G.M.1.	MAINTAINABILITY	Spare parts should be locally available or possible to order	See G.A.2.	
G.M.2.		Calibration or alternatively performance testing of the relevant units should be possible in a basic electronics workshop (multimeter, oscilloscope, soldering station). Otherwise, there should be a way to indicate when further testing/calibration is required.	Amplitude and HR testing can be achieved without external tools. Amplitude adjustment only requires a small screwdriver.	Amplitude feedback system needs to be characterised further in order to improve its accuracy, which could reach 75 uV.
G.F.1.	FUNCTIONALITY	Functionality: simulator can be used on patient monitors, ECG meters, NIBP meters and pulse oximeters to test their ability to measure ECG, NIBP, HR and/or SpO2	Proven to work with patient monitors from 5 different manufacturers (Mindray, Contect Medical Systems, Creative Medical, Edan, Philips).	
G.F.2.		Manufacturer independence: simulator should work with any patient monitor, ECG meter, NIBP meter or pulse oximeter	See G.F.1.	
G.F.3.		Updatibility: possibility to add more signals to generate in the future	Signal to be simulated can be uploaded to Arduino (up to 32kB).	Data storage space can be increased by means of external memory.
G.F.4.		Portability: user should be able to test medical devices on site	The use of rechargeable batteries makes the simulator totally portable.	Only needs computer to upload signal once
G.F.5.		Performance: accuracy and repeatability of the system should be well defined	Accuracy tests were carried out but repeatability tests were not.	When using a printed board, accuracy is expected to increase.
G.F.6.		Working conditions: work in humid (Nepal: average 74 or 81 % depending on region, SSA: approx. 40-75% depending on region) and high temperature (Nepal: average mean 2-27 °C depending on region; SSA: mean outdoor 13-30 °C depending on region) conditions	Proven to work in Pokhara, Nepal, at an average temperature of 32 °C and 47 % humidity.	
G.F.7.		Signal synchronisation: ECG, NIBP and SpO2 simulation should be synchronised	Not applicable	
G.F.8.		Integrability: all signals integrated in one system	Not applicable	
ECG SPECIFIC				
S.F.1.		Signal waveform: should simulate one 'healthy patient' ECG waveform consisting of QRS complex and P and T waves	Proven to be able to generate mathematically-defined waveforms and one ECG recording (#101 from MIT-BIH database).	
S.F.2.		Signal amplitude: must be able to simulate signal components 0.5 – 1 mV in amplitude	Proven to output a signal with an average amplitude of 1.13 mVpp +0.04 as displayed by the 6 patient monitors tested.	
S.F.3.		Frequency of simulated signal: 60 bpm = 1 Hz	Proven to simulate target HRs from 35 - 130 bpm in steps of 5 bpm.	Lowest HR that can be simulated now is 35 bpm due to storage space of Arduino (32 kB). This can be increased (see G.F.3.) or reference signal can be modified to require less storage space.
S.F.4.		Relevant units to be calibrated are HR	Simulated HR can be checked but not calibrated, it's given by Arduino UNO's clock source (crystal resonator, +/-5% accuracy) and cannot be modified. Amplitude can be adjusted to the levels: <2 mV, 2-3 mV and <4 mV. Adjustment is based on the feedback system, which has a precision around 0.2 mV.	Timing accuracy is given by Arduino's embedded oscillator by default, but an external one (with higher accuracy and maybe possibility of calibration) can be installed, although microcontroller programming is required. Feedback system can be further tested and accuracy can probably be increased by using other opamps and offset adjustment. Component availability in Nepal should be kept in mind.
Legend:	Met			
	Partially met			
	Not met			

7

Discussion

This chapter discusses the results achieved by the Prototype and the process that led to them. First, the **Performance** of the Prototype is discussed based on the experimental results. Next, **Cost and materials**, Protability, **Usability** and **Maintenance** of the final Prototype are discussed. Later, **Further improvements** required for the Prototype to be used as a medical device tester in LRs are discussed. The chapter finishes by listing the **Limitations of this work**.

7.1. Performance

The use of Arduino's PWM and a LPF for D/A conversion was proven to have resolution and accuracy suited for the generation of an analog 'healthy' ECG waveform. Using a PWM frequency of 31.25 kHz and a LPF with 150 Hz cut-off, a theoretical DAC uncertainty of 0.04 V in the range 0-5 V was achieved. Maximum error experimentally measured increased to 0.08V. This is a DAC error 2.38x smaller than the lowest amplitude of interest (S dip of 0.19 V).

In terms of **amplitude accuracy**, the IEC 60601-2-27:2011 specifies that an electrocardiograph should have a signal reproduction error $\leq \pm 20\%$ of the reference or $\pm 100 \mu\text{V}$, whichever is greater [18]. Given that no specific requirement for ECG testers/calibrators was found, this work follows the general guideline mentioned at an other medical device standard: ISO 80601-2-61. ISO 80601-2-61 mentions that, in general, "a calibrator should be at least four to ten times more accurate" than the DUT. Since the DUT for the Prototype is a patient monitor, an accuracy that is four times better than the one specified for patient monitors should be met to be a good reference.

Regarding the experimental tests on patient monitors, the Prototype did not meet the standard in 4 cases (see Fig. 6.3). Although ProSim 8 met the IEC standard in all cases, it did not reach the recommended 4x better accuracy than the DUT. This can be explained by the fact that the patient monitors add their own error to the displayed waveform. Furthermore, the measurement method that had to be used given the limitations also added inaccuracies to the amplitude measurements. Thus, instead of using these measurements to quantify simulator error, they can be used to compare the behaviour of the Prototype against ProSim 8. Looking at Fig. 6.3 we conclude that **although the Prototype tends to show a higher error, it follows the same behaviour as ProSim 8** when tested in different patient monitors and for different amplitudes. In other words, in the raw data from Fig. P.1, the dots representing ProSim 8 are always surrounded by dots from the Prototype. These would indicate that the hardware and software solutions used by different manufacturers interpret de amplitude information of ProSim 8 and the Prototype in the same way.

The amplitude measurements performed with Zurich Instrument's lock-in amplifier are not representative of the Prototype's mean accuracy, given that only one waveform of one target amplitude (1.30 mVpp) was analysed. This responded to time and equipment access constraints. The analysed waveform showed an **absolute amplitude error of 0.4 mV (or 30.8%)**. This result appears higher than the mean amplitude error obtained from patient monitors (of 13.5 %).

While the error measured at patient monitors met IEC 60601-2-27, none of the measurements met the accuracy recommendation for testing equipment. All in all, the experimental amplitude errors seem to indicate that **the current version of the Prototype is not recommended for use as amplitude reference**. Ways to improve this are suggested in [Further improvements](#).

According to its manual, ProSim 8 has an amplitude accuracy of $\pm(2\% \text{ of setting} + 0.05 \text{ mV})$. This would correspond to an accuracy of 5.8% when set to simulate a 1.3 mVpp ECG waveform. According to this, ProSim would meet the recommendation of being 4x more accurate than ECG monitors. Based on the experimental measurements, **the Prototype would have an accuracy between 5.3x and 2.3x worse than ProSim 8**.

The Prototype's measured amplitude error is hypothesized to result from resistor tolerances (1%), errors introduced by the use of breadboards and measurement unidealities. Despite this amplitude error, the cases in which a simulator is used to check amplitude reproduction accuracy of a patient monitor during maintenance at a hospital are rare. In the experiences had at a Dutch hospital, no amplitude accuracy checks were included in the preventive maintenance tests. The only time when amplitude was checked was during printing of ECG records, which display a 1 mV calibration mark.

Moving on to **HR**, the best results with a PWM frequency of 31.25 kHz were achieved when setting the simulation rate (or duty cycle update rate) to 1997 microseconds/sample. Meanwhile, the digitized reference waveforms were resampled to 500 Hz. Given that ECG recording frequencies are typically below 360 Hz, these can be correctly simulated using the developed Prototype. With these characteristics, **highest HR determination error was 1.42%. The biggest absolute deviation with respect to target HR was 1 bpm**. Although ProSim 8 showed no HR error on the experimental results, its manual specifies an accuracy of $\pm 1\%$ of setting. According to this, the Prototype's HR error would be higher than that of ProSim 8, but of the same order of magnitude. Regarding the standards, IEC 60601-2-27:2011 states that electrocardiographs should have a HR detection accuracy of $\pm 10\%$ or ± 5 1/min, whichever is greater. Assuming that an ECG simulator should have an accuracy 4x better than the DUT, **the Prototype meets the required accuracy to be used as a reference for HR testing**.

The HR accuracy of the Prototype should be mostly determined by Arduino UNO's clock source accuracy (of $\pm 0.5\%$ [27]). However, when testing HR determination on patient monitors, each monitor adds their own error to the measurement. For most patient monitors used, HR determination accuracy is ± 1 bpm or $\pm 1\%$, whichever is greater (Appendix Q). The combination of these two uncertainties gives an error of 1.5%, what would be in line with the HR determination error measured (of 1.42%).

Timing accuracy of the first stage of the Prototype was quantified on oscilloscope measurements. This gave a mean timing error of 1.56%. However, the measurements were performed before changing the simulation rate to 1997 microseconds/sample. Since the change was motivated by an improvement on HR determination by patient monitors, it is expected that an improved timing accuracy was achieved. Anyway, further tests on the final version of the design should be carried out to confirm and characterise the timing accuracy improvement.

Finally, in terms of waveform simulation, **the Prototype showed a higher similarity to the healthy patient ECG than ProSim 8**. This enables the possibility to test the ability of patient monitors to detect real ECG signals. ProSim's waveform is clearly mathematically defined, showing steeper Q and S dips. Opposite to that, the Prototype has a **more realistic waveform**.

While all commercial simulators mentioned reproduce mathematically-based signals, non-commercial ones use either patient data, mathematically-based signals, or both (Appendix A). The presented simulator was developed to simulate both patient data from MIT-BIH database and mathematical waveforms defined by equations from [45]. Moreover, any waveform can be uploaded into the Arduino's memory for simulation, which can store up to 32 KB. With a sampling frequency set to 500Hz, this enables storage of one waveform representing a $HR \geq 35$ bpm.

	Amplitude accuracy		HR accuracy (35 to 130 bpm)	
	Prototype	According to patient monitors display (1-3 mV range)	13.5% mean	≤1.42%
According to single-waveform Zurich Instr.		30.8% ¹		
ProSim 8	±(2%+0.05mV) or ≤7 % ²		1%	
ProSim 4	≤5%		1%	
PS-2005	≤2%		0.5%	
Narvaez 2017	<5%		<5%	
VergaraGil 2021	-		3.3% max 1.8% mean	
Vaismann 2000	1%		1%	
Vasilevich 2004	-		0.2%	
Wijaya 2021	-			7 bpm max

¹ at 1.3 mV, ² in 1-3 mV range

Figure 7.1: Accuracy comparison between Prototype, commercial and non-commercial simulators with available data. Data extracted from: [14, 13, 5, 28, 50, 46, 48, 51].

7.2. Cost and materials

The proposed ECG simulator has a cost of 2.3% of the most complete physiological simulators in the Nepali market, ProSim 8. However, a more fair comparison would be to a model like PS-2005 (BC Biomedical). This is a simple ECG simulator with a fixed waveform and four different HRs (30, 60, 120 and 240 bpm), besides sine, square and triangle signals. Taking into account its price in the European market, Frankie would have a cost 5.3% of PS-2005, although this is currently not sold by Nepali providers. All the utilised materials and tools were acquired from an educational electronics workshop and their equivalents in the Nepali market have been documented here.

7.3. Portability

All the commercial simulators reviewed during this work are portable and battery-operated. However, ProSim 8 is considerably bigger (30.22 cm x 14.48 cm x 8.64 cm) and heavier (1.81 kg) than the rest, making it not comfortable for prolonged hand-held operation. Some models work with rechargeable batteries (ProSim), while others work with 9V cells (PS-2005). Most research simulators require a connection to the power line, while DIY simulators tend to be more battery-operated.

7.4. Usability

In terms of usability, efforts mostly focused on making the electronics intuitive. The goal was to design a system easy to understand by a technician, so that it could be easily assembled but also repaired, maintained and customized if required. On the one hand, this was achieved by using common components (resistors, capacitors, LEDs), and keeping the number of elements low.

On the other hand, the use of switches and output connectors made the Prototype easy to operate and connect to a patient monitor. The use of switches makes the system easier to operate than ProSim 8, which features endless menus and submenus. Obviously, this simplification is achieved thanks to a limited number of options offered by the Prototype.

7.5. Maintenance

Allowing the user to perform maintenance on the Prototype without the need for specialized knowledge or equipment was one of the main goals. For this, all the schematics and components are documented and freely accessible online. This represents a huge improvement compared to all commercial simulators, which keep their internal functioning undisclosed. As a result, any repairs and maintenance must be performed by certified technicians and require specialised equipment, not available everywhere in the world.

Regarding the non-commercial simulators, although their schematics can be usually accessed, they lack information on maintenance and calibration. From the non-commercial simulators analysed in Appendix A, none provided calibration details. In fact, from the 76 articles on ECG simulation analysed in a previous literature review [17], only four gave vague information on calibration. Some simulators provide a visualisation of the signal, which could be interpreted as useful to detect an accuracy decay of the output signal. Most common solutions employed show the generated HR by means of an LCD or use a software interface to define the signal to be simulated. However, these solutions are not a representation of the actual output of the generator, instead, these are digitally generated. The developed Prototype aimed to provide a real visualisation of the generated signal. For this, a feedback system that directly responds to the output signal was designed. This is important to be able to check the simulator as a whole, including the electrical components that create the final analog signal.

The developed feedback system is an integrated solution for amplitude and HR visualisation. In combination with the trimmer resistor, the amplitude can be adjusted. The feedback LEDs indicate amplitude levels approximately between 1 and 2.5 mV, between 2.5 and 3.5 mV and >3.5 mV. The system proved useful to help bring the signal back to levels visible in a patient monitor display using 1x default scale (i.e., in a range 1 - 2.5 mV). Although the theoretical accuracy of the amplitude feedback system should be around 75 μV , higher errors and sometimes unreliable behaviour were detected. These are expected to improve when replacing the use of breadboards by printed circuit boards. Although some functional tests were performed, further qualitative characterisation of the feedback system is required. Moreover, the need for the opamps used at the feedback system to have lower long-term voltage offset drift than those used for signal generation was identified. For these reasons, the amplitude feedback and adjustment system need further development.

On the other side, the HR feedback helps to know the real-time state of the simulator without any external tool. The HR LED helps distinguish ON/OFF and simulation/calibration modes, as well as the generated HR. Compared to this, when commercial simulators are switched on, there is no indicator of what the actual output is.

7.6. Further improvements

To improve the overall performance of the Prototype, breadboards should be replaced by a printed printed circuit board or a soldering board. The use of breadboards adds a lot of uncertainty to the system, probably responsible for a big part of the measured errors (especially in the feedback system). Their replacement is expected to increase the reliability and performance of the Prototype. Given the goal of making the system easy to assemble and maintain with low technical skills, the use of printed circuit boards with through-holes to manually solder the big-sized components is recommended. This is an affordable solution available for ordering online. For example, five 10x10 cm printed circuits can be ordered from the Chinese manufacturer PCBway for 5 USD. The PCB design can be carried out using the open-source KiCad software (www.kicad.org).

To further improve the performance of the Prototype, hardware changes can be implemented. For amplitude accuracy improvement, resistors with better tolerances could be considered. Circuit simulations taking into account component deviations should be carried out to establish the impact of the tolerances used (of 1% in resistors). If their replacement is required to reach better accuracy levels, components that are easily available and affordable should be chosen. Timing accuracy can be improved by the use of an external oscillator. However, this would increase complexity, as C programming of the Arduino's microcontroller would be required. Given the accuracy achieved, this is not deemed necessary.

In general, more tests to characterise the Prototype's accuracy are required. Timing and amplitude measurements on the raw signal using Zurich Instrument's should be performed. These should cover

multiple waveforms and different target amplitudes and HRs in order to obtain more relevant accuracy data. Additionally, the particularities of the measurement method and features of Zurich Instrument's amplifier should be studied further in order to optimise its use for raw amplitude measurements. Next to that, the feedback system should also be tested further, in order to quantify its accuracy.

The addition of new functionalities would make the Prototype able to test more functions of a patient monitor. However, the increase in system and calibration complexity or cost should be taken into account when assessing their implementation. For instance, the simulation of more leads would ease testing of 10- and 12-lead ECG detection. With the current design, 10- or 12-lead simulation would require upscaling of the electronics. If the reproduction of recorded ECG derivations is desired, an opamp per lead with specific gain would be required, as well as increased memory to store each ECG lead. This method would be useful to measure the accuracy of each channel of a patient monitor. In this case, leads characterised in amplitude and time, such as the ones found in MIT-BIH database, would be required as reference. Otherwise, the current system can be extended using non-/inverting amplifiers with some gain that results in slightly different copies of the generated signal, so that the potential differences measured by the patient monitor are not 0. These signals will all reproduce the same reference waveform but have different amplitudes.

To test respiratory rate detection, resistance changes can be generated by a pin of Arduino's microcontroller. This is the method followed by HTM Workshop ECG simulator kit [53] to simulate respiration, and is slightly described in the video [54]. However, they warn that this functionality does not work consistently on a breadboard and requires a printed circuit board instead.

The proven simulation of realistic waveforms opens a door to the simulation of pathological ECGs and other characteristic waveforms that can be used to test automatic detection options in patient monitors. With the current design (150 Hz cut-off frequency), simulation of arrhythmias should be possible. As part of future developments, the performance of the Prototype when simulating other waveforms such as arrhythmias should be tested.

Future work is also required to improve usability. Especially, the feedback system use should become more intuitive. HR selection could also be improved, for instance by storing two different HRs on the Arduino memory and using a switch or potentiometer to select them. In a final design, the casing should be made of plastic or another material more resistant to fluids, humidity, bumps and easier to clean. In order to use broadly available materials, plastic boxes could be used. In Nepal, these are available at Daraz.com.np for around 100 NPR.

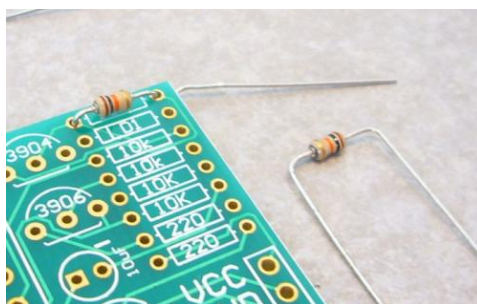


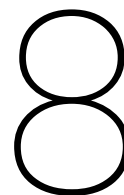
Figure 7.2: Through-hole printed circuit boards are suggested to replace breadboards. This solution should improve output noise. Compared to printed circuits with surface-mounted components, through-holes make the circuit more intuitive, enable reusing components, make component testing and repair easier and only require a soldering station.

7.7. Limitations of this work

- The availability and access to equipment with enough sensitivity and possibility to download the digital data were limited. This, together with the amplitude characteristics of the generated signal resulted in a limited number of measurements and the use of measuring methods that added uncertainty to the results.
- Access to some patient monitors at the Green Pastures Hospital (Pokhara, Nepal) was limited, since these were being used daily. Moreover, the stay in Nepal was limited to three weeks. These

reasons limited the number of tests on the patient monitors.

- The use of breadboards limits the performance of the system, as these suffer from current leakage and other unidealities. Breadboards were employed during the development of this Prototype because they allow easy testing and circuit modification. However, these should not be used in a more final version of the simulator. Recommendations on replacing them for a more accurate system are given in [Further improvements](#).



Conclusions

The developed Prototype generates an ECG-like signal by means of an Arduino-based PWM, a LPF (using RC components), and an attenuation/offset circuit (opamps and RC components), which is correctly detected by patient monitors. Patient monitors interpret the signal correctly when 3 electrodes are connected and provide a HR measurement based on the R-peaks frequency. Additionally, the Prototype has an inbuilt amplitude feedback and adjustment system that helps to identify amplitude deviations and adjust them. Although the amplitude feedback needs further work to make it reliable, the system was proven useful to bring amplitude deviations back to levels visible at a patient monitor's display. Additionally, visual feedback on the generated frequency is provided, what helps identify the state of the device (ON/OFF, simulation/calibration mode) and the simulated HR.

The number of features offered by the Prototype is limited compared to the more complete patient simulators in the market (e.g., ProSim 8). However, the Prototype focuses on providing the minimum features required to test a patient monitor. This allows simplification of the simulator as a whole, from the electronics to its usability. With a HR accuracy of 1.42% and maximum deviation of ± 1 bpm (in the range 35 - 120 bpm), the Prototype is suitable for testing high and low HR alarms, as well as HR determination of patient monitors. Meanwhile, accuracy improvements are required to be able to use the simulator as an amplitude reference. A HR ≥ 35 bpm and different waveforms can be simulated. For this, a signal with the desired waveform and HR (in a scale from 0 to 255 and sampling frequency of 500 Hz) must be defined in the Arduino code. A mathematically-defined and a real ECG waveform were successfully generated.

The Prototype's design followed the usability, maintainability and affordability requirements for a simulator for low-resource settings. All requirements were met except for the provision of accurate amplitude feedback, the reliability of which should be further improved, and the complete characterisation of the system performance, which needs further testing to be achieved. All the details required to comprehend and implement the Prototype are described in this work and are freely accessible online.

In spite of the required further work and the possibility for improvement, this master thesis has proven the potential of the Prototype to be a locally built and maintained ECG simulator for medical device testing in LRs.

Acknowledgements

To you, for reading this. To my supervisors, Jenny Dankelman, Arjan Knulst and Arjo Loeve, for offering me this project and guiding me through the journey. To Delft University Fund, FAST, which helped support my trip to Nepal. To Stackoverflow, for all the shared wisdom. To all the collaborators in these pages and those who I forgot to mention. To all the friends I made in Delft, who shared this journey with me. To my BME friends, for being amazing people to learn from. And especially to Edwin, for all the coffee. To the MISIT lab, especially Yannick, Eva, Thomas and Defne, for sharing laughs and some tears. And for going through the same ups and downs as me, with a few months of difference. To the people in 3mE, especially Jacques and Ruben, whose help was so valuable. To everyone I met in Nepal, especially Ramesh and Priya, for their enthusiasm in my project. To my group of friends from Lleida, who are not sure what I am working on after these years abroad. For the next 24 years of friendship. To Alba, Júlia, and my friends from Barcelona. And to Mariona, for agreeing to everything I say over the phone. And for the mental support.

Finally, to my family, for keeping me alive all these years and believing in me.

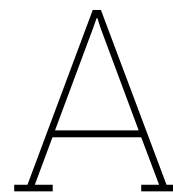
Look grandmas, I did it!

References

- [1] S Abrar et al. "Design and implementation of an embedded system for transmitting human ECG and web server for emergency services and remote health monitoring: A low cost ECG signal simulator and its transmitter, to send and store data in electronic databases, in remote location, to be accessed by authorized personnel when needed". In: *2012 International Conference on Open Source Systems and Technologies, ICOSST 2012*. 2012, pp. 50–55. ISBN: 9781467330978. DOI: [10.1109/ICOSST.2012.6472828](https://doi.org/10.1109/ICOSST.2012.6472828).
- [2] D Almeida, J Costa, and A Lourenco. "ECG simulator with configurable skin-electrode impedance and artifacts emulation". In: *Biomedical Physics and Engineering Express* 7.6 (2021). ISSN: 20571976. DOI: [10.1088/2057-1976/ac2b4e](https://doi.org/10.1088/2057-1976/ac2b4e).
- [3] Analog Devices. *Datasheet: OP97. Low Power, High Precision Operational Amplifier*. Tech. rep. 2009, p. 3.
- [4] L. Andrew. *ECG and Blood Pressure Simulator Project*. 2017. URL: <https://alowminumfoil.wordpress.com/2017/01/08/ecg-and-blood-pressure-simulator-project/>.
- [5] BC Biomedical. *Market price of PS-2005 5-leads ECG simulator*. Accessed on: 23/06/2023. URL: <https://www.bcgroupestore.com/>.
- [6] M J Burke and M Nasor. "An accurate programmable ECG simulator". English. In: *Journal of Medical Engineering and Technology* 25.3 (2001), pp. 97–102. ISSN: 03091902. DOI: [10.1080/03091900110051640](https://doi.org/10.1080/03091900110051640).
- [7] C Caner, M Engin, and E Z Engin. "The programmable ECG simulator". In: *Journal of Medical Systems* 32.4 (2008), pp. 355–359. ISSN: 01485598. DOI: [10.1007/s10916-008-9140-1](https://doi.org/10.1007/s10916-008-9140-1).
- [8] Ltd. Contec Medical Systems Co. *Service manual: CMS8000 patient monitor*. 2007.
- [9] Health Management Information System database. *Number of public hospitals in Nepal*. 2020. URL: dohs.gov.np.
- [10] Joseph F. Dyro. "Donation of medical device technologies". In: *Clinical Engineering Handbook*. Academic Press, 2004. Chap. 43, pp. 155–158. ISBN: 978-0-12-226570-9. DOI: [10.1016/B978-012226570-9/50047-8](https://doi.org/10.1016/B978-012226570-9/50047-8).
- [11] Edan. *Service manual: iM8 Series*. 2007.
- [12] J.-C. Edelmann et al. "An ECG simulator with a novel ECG profile for physiological signals". English. In: *Journal of Medical Engineering and Technology* 42.7 (2018), pp. 501–509. ISSN: 03091902. DOI: [10.1080/03091902.2019.1576788](https://doi.org/10.1080/03091902.2019.1576788).
- [13] Fluke Biomedical. *Technical Data_{ProSim4}*. Tech. rep. 2013.
- [14] Fluke Biomedical. *Technical Specifications: ProSim 8*. Tech. rep. 2011.
- [15] Fluke Biomedical. *Users Manual: ProSim 8*. 2011.
- [16] A K Gerasimov and Z N Pedonova. "Development of Hardware and Software for Generating Test ECG Signals". In: *Biomedical Engineering* 55.5 (2022), pp. 315–318. ISSN: 00063398. DOI: [10.1007/s10527-022-10126-1](https://doi.org/10.1007/s10527-022-10126-1).
- [17] Anna Graell Collado. *Review of ECG, HR, SpO2, respiration, and NIBP signal simulation methods: a first step towards a medical device tester for LMICs*. 2023.
- [18] International Electrotechnical Commission. *IEC 60601-2-27 Part 2-27: Particular requirements for the basic safety and essential performance of electrocardiographic monitoring equipment*. 2011.
- [19] *INTERNATIONAL STANDARD ISO 80601-2-61 Part 2-61: Particular requirements for basic safety and essential performance of pulse oximeter equipment*. 2011.

- [20] V Kasik et al. "Design and implementation of monitor tester for validation of ECG signals". English. In: *12th International Conference on Biomedical Electronics and Devices, BIODEVICES 2019*. 2019, pp. 127–133. ISBN: 9789897583537. DOI: [10.5220/0007248001270133](https://doi.org/10.5220/0007248001270133).
- [21] J Koebele, M Heinke, and B Ismer. "Soundcard based multichannel live-ECG simulator for research, development and education". In: *Biomedizinische Technik* 57 (2012), pp. 331–332. ISSN: 00135585. DOI: [10.1515/bmt-2012-4024](https://doi.org/10.1515/bmt-2012-4024).
- [22] LuckyLight Electronics Co. Ltd. *Technical data sheet Blue LED (504BC2E-B4-2BE)*. Tech. rep. 2020.
- [23] Jim Lynch. *ECG Simulator - Instructables*. 2013. URL: <https://www.instructables.com/ECG-Simulator/>.
- [24] A E Martínez, E Rossi, and L Nicola Siri. "Microprocessor-based simulator of surface ECG signals". English. In: *Journal of Physics: Conference Series* 90.1 (2007). ISSN: 17426588. DOI: [10.1088/1742-6596/90/1/012030](https://doi.org/10.1088/1742-6596/90/1/012030).
- [25] Microchip Technology Inc. *Datasheet: ATmega48A/...* 2018.
- [26] G B Moody and R G Mark. "The impact of the MIT-BIH Arrhythmia Database". In: *IEEE Engineering in Medicine and Biology Magazine* 20.3 (2001), pp. 45–50. ISSN: 1937-4186 VO - 20. DOI: [10.1109/51.932724](https://doi.org/10.1109/51.932724).
- [27] Murata. *Datasheet: CSTCE16MOV53-R0, Arduino's ceramic resonator*. 2021.
- [28] J M N Narvaez and G O Avendaño. "Design and implementation of vital signs simulator for patient monitor". In: *Journal of Telecommunication, Electronic and Computer Engineering* 9.2-4 (2017), pp. 7–9. ISSN: 21801843.
- [29] Government of Nepal National Statistics Office. *Census of Private Hospitals*. 2013. URL: microdata.cbs.gov.np.
- [30] Nederlands Normalisatie-instituut. *NEN-EN-IEC 60601-2-25 Part 2-25: Particular requirements for the basic safety and essential performance of electrocardiographs*. 2015.
- [31] Nederlands Normalisatie-instituut. *NEN-EN-IEC 60601-2-47 Part 2-47: Particular requirements for the basic safety and essential performance of ambulatory electrocardiographic systems*. 2015.
- [32] A D Paul et al. "Design and development of microcontroller based ECG simulator". In: *5th Kuala Lumpur International Conference on Biomedical Engineering, BIOMED 2011, Held in Conjunction with the 8th Asian Pacific Conference on Medical and Biological Engineering, APCMBE 2011*. Vol. 35 IFMBE. 2011, pp. 292–295. ISBN: 9783642217289. DOI: [10.1007/978-3-642-21729-6_76](https://doi.org/10.1007/978-3-642-21729-6_76).
- [33] Lora Perry and Robert Malkin. "Effectiveness of medical equipment donations to improve health systems: How much medical equipment is broken in the developing world?" In: *Medical and Biological Engineering and Computing* 49.7 (2011). ISSN: 01400118. DOI: [10.1007/s11517-011-0786-3](https://doi.org/10.1007/s11517-011-0786-3).
- [34] Philips. *Service Manual: IntelliVue MP60/70 Patient Monitor*. 2016.
- [35] Philips. *Service Manual: SureSigns VM Series Patient Monitor*. 2011.
- [36] Pronk Technologies. *Technical Specifications: OxSim Flex* ©. Tech. rep. 2019.
- [37] Pronk Technologies. *Technical Specifications: SC-5 SimCube*. Tech. rep. 2019.
- [38] Pronk Technologies. *Users Manual: SimCube™ NIBP Simulator*. 2021.
- [39] Shenzhen Mindray Bio-Medical Electronics Co. Ltd. *Service Manual: T1 Patient Monitor*. 2016.
- [40] Shenzhen Mindray Bio-Medical Electronics Co. Ltd. *uMEC10/uMEC12/uMEC15 Patient Monitor Operator's Manual*. 2016.
- [41] G P Shorten and M J Burke. "A versatile temperature-stable ECG simulator". In: *Journal of Medical Engineering and Technology* 35.2 (2011), pp. 92–102. ISSN: 03091902. DOI: [10.3109/03091902.2010.539771](https://doi.org/10.3109/03091902.2010.539771).
- [42] The World Bank. *GDP deflator (base year varies by country) - Netherlands*. 2020. URL: <https://data.worldbank.org/indicator/NY.GDP.DEFL.ZS?end=2020&locations=NL&start=1994>.

- [43] Gavin Thompson. "Statistical literacy guide: How to adjust for inflation". In: *House of Commons Library (UK Parliament)* February (2009), pp. 1–6. URL: <http://www.parliament.uk/briefing-papers/sn04962.pdf>.
- [44] Teresa Angela Trunfio et al. "Medical Technologies Procurement, Management and Maintenance in Developing Countries: The Case of Health Challenges in Africa". In: *IFMBE Proceedings 80* (2021), pp. 793–804. ISSN: 14339277. DOI: [10.1007/978-3-030-64610-3_89](https://doi.org/10.1007/978-3-030-64610-3_89).
- [45] B N Usta, B Tepeyurt, and E Karakulak. "Simple Synthetic ECG Generation via PWM Output of Microcontroller". In: *5th International Symposium on Multidisciplinary Studies and Innovative Technologies, ISMSIT 2021*. Institute of Electrical and Electronics Engineers Inc., 2021, pp. 27–30. ISBN: 9781665449304. DOI: [10.1109/ISMSIT52890.2021.9604584](https://doi.org/10.1109/ISMSIT52890.2021.9604584).
- [46] M V Vaisman, D A Prilutskii, and S V Selishchev. "A programmable simulator of electrocardiographic signals". In: *Biomedical Engineering 34.2* (2000), pp. 98–102. ISSN: 00063398. DOI: [10.1007/BF02403682](https://doi.org/10.1007/BF02403682).
- [47] I Valais et al. "Design and Construction of a Prototype ECG Simulator". In: *EJST 9* (2014), pp. 11–18.
- [48] M. Yu Vasilevich and D. A. Prilutskii. "A generator for testing signals of electroencephalographs and electrocardiographs". In: *Biomedical Engineering 38.3* (2004). ISSN: 00063398. DOI: [10.1023/B:BIEN.0000042110.75882.22](https://doi.org/10.1023/B:BIEN.0000042110.75882.22).
- [49] A Vehkaoja and J Lekkala. "A simple bioelectrical signal simulator for measurement device testing". In: *19th IMEKO World Congress 2009*. Vol. 3. 2009, pp. 1429–1433. ISBN: 9781615675937.
- [50] J C Vergara Gil and A M Gonzalez-Vargas. "UAOSIM-ECG: An open-source 12-lead electrocardiography simulator". In: *2nd IEEE International Congress of Biomedical Engineering and Bioengineering, CI-IB and BI 2021*. Institute of Electrical and Electronics Engineers Inc., 2021, pp. 1–4. ISBN: 9781665408554. DOI: [10.1109/CI-IBBI54220.2021.9626054](https://doi.org/10.1109/CI-IBBI54220.2021.9626054).
- [51] N H Wijaya et al. "The design of electro cardiograph signal generator using IC 14521 and IC 14017". In: *Journal of Robotics and Control (JRC) 2.4* (2021), pp. 270–273. ISSN: 27155056. DOI: [10.18196/jrc.2490](https://doi.org/10.18196/jrc.2490).
- [52] Elliot Williams. *Make: AVR Programming*. Make: Community, 2014. ISBN: 9781449355784.
- [53] HTM Workshop. *ECG simulator kit*. URL: <https://htm-workshop.com/ecg-simulator-educational-resources/>.
- [54] HTM Workshop. *ECG simulator kit - Software Theory video*. URL: <https://www.youtube.com/watch?v=eKph8Kj9MUM&t=622s>.
- [55] World Health Organization (WHO). *Human Resources for Medical Devices: The role of biomedical engineers*. 2017. ISBN: 9789241565479.
- [56] World Health Organization (WHO). *Medical Devices: Managing the mismatch*. 2010. ISBN: 9789241564045.



Non-commercial simulators comparison

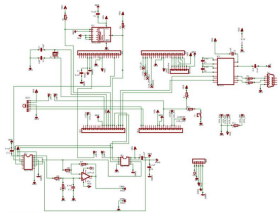
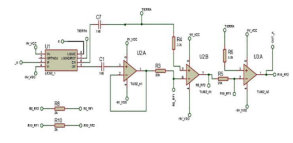
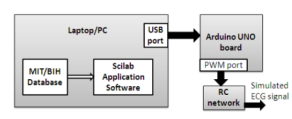
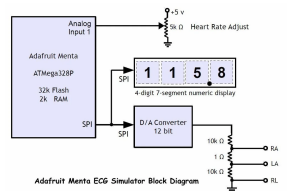
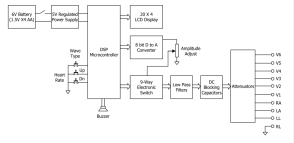
Table A.1: Comparison of non-commercial simulators that included cost details from 120 articles reviewed in [17]. All refer to academic articles except Lynch2013, a DIY project published at instructables.com. *Inflation-adjusted prices [43], base year is 2022 due to available data [42].

<i>Ref.</i>	<i>Cost</i>	<i>Test signal</i>	<i>Purpose</i>	<i>Components</i>	<i>Limitations</i>
Janes1969	<415.6 USD	HR	Medical device testing	Analog components	Not realistic, fixed waveform
Ughi2016	3% of commercial simulators in Indonesia	NIBP	Medical device testing	Arduino, electro-pneumatic system	3.8% error compared to reference simulator, mathematically-defined signal, repeatability tests
Machado2018	143 USD	SpO2	Educational	Raspberry Pi, electro-optical system	Manufacturer dependant, not portable, functional tester
Diaz2021	400 USD, more than 10x less than commercial simulators	ECG	Education	Raspberry Pi, external DAC and conditioning circuit	No schematics or testing data of ECG signal
Paul2011	11.35 USD vs. 200-1000 USD for commercial simulators in Bangladesh	ECG	Medical device testing	Microcontroller PWM, RC filter	No accuracy tests, one lead
Putnam1980	<134.4 USD*	ECG	Medical device testing	Analog components	Not realistic, fixed waveform
Usta2021	<15 USD	ECG	Medical device testing	Arduino PWM, RC filter, opamp, SD card	Output in the volts range, not portable
Vergara2021	<200 USD	ECG	Educational	Raspberry Pi, 8x DAC, conditioning circuit	Not portable
Yener2018	<50 USD	ECG	Educational	Arduino PWM, RC filter, opamp, SD card	Output in the volts range, not portable
Varela2021	120 USD	NIBP	Educational	Raspberry Pi, electro-pneumatic system	Bulky, not portable, mathematically-defined signal, repeatability tests
Mariappan2013	<44.77 USD*	ECG	Educational	Arduino, RC filter, laptop	Connected to laptop; price excluding cost of laptop
Lynch2013	53.64 USD*	ECG	DIY	Arduino, DAC, resistive voltage divider	Output in the volts range, no accuracy tests

Table A.2: Detailed comparison of non-commercial ECG low-cost simulators. All refer to academic articles except Lynch2013, a DIY project published at instructables.com; and Michalek2006, a master thesis available at University of Central Florida repository. *Inflation-adjusted prices [43], base year is 2022 due to available data [42].

Caner2008	VergaraGil2021	Mariappan2013	Lynch2013	Michalek2006	<i>Reference</i>
12 leads HR Amplitude Noise	12 leads Noise Arrhythmias		3 leads HR	Amplitude Frequency	<i>Simulated signals</i>
	Max. relative error: 3.3% (w.r.t. expected value), mean relative error: 1.8 % Uncertainty: - HR ≤ 110: ±1 - 110 < HR ≤ 200: ±2 - 200 < HR ≤ 250: ±3				<i>Reported performance</i>
Power line	Power line	6-20 V	Power line	9V rechargeable battery	<i>Power source</i>
Relatively inexpensive to produce	155.85 €2022	< 44.77* €2022	50.91* €2022		<i>Cost (HW)</i>
Equations used from	Yes: Python	Yes: SciLab	Yes: Python (for ECG)		<i>SW Open-source</i>

Burke2001	Yes: Raspberry Pi 3 B+ (not open-source) + all schematics available	Yes: Arduino UNO (open-source)	digitization), codes available Yes: Arduino Menta		<i>HW Open-source</i>
(standard commercially available electronic components)		Arduino UNO + RC circuit (4kohm, 10uF) + Computer	Arduino Menta + 12 bit ADC + Resistive voltage divider	MSP430 microcontroller + buttons + LCD + Flash Memory chip 66Mbit	<i>Hardware</i>
Microcontroller (specifically a Digital Signal Controller)	Microcontroller	(Arduino UNO is based on) ATmega328P with 2KB SRAM, 32KB FLASH, 1KB EEPROM + Computer required for computation, communication and visualization	(Menta features an) Arduino ATmega328P microprocessor with 32K of Flash memory and 2K of RAM		<i>Controller</i>
Synthetic (based on math equation from Burke2001 = Zhang2011)	Synthetic (based on math equation from Zhang2011)	Patient data (MIT-BIH Database = PhysioNet)	Bio-synthetic (patient data digitized and HR controlled via repetition interval time)	Patient data (MIT-BIH Database = PhysioNet) stored in updatable flash memory 66Mbit (limits amount)	<i>Signal origin</i>
LCD + keypad	Self-made graphic interface (Tkinter) + mouse + keyboard + monitor	Scilab software on laptop	potentiometers + LCD	push buttons + LCD display	<i>Interface</i>

	Stand-alone system (no PC needed)				
Testing ECG meters	Educational	Educational	DIY		<i>Purpose</i>
					<i>Calibration</i>
HR: 30, 60, 90, 120 bpm Noise: 0 - 20 dB in three steps Amplitude: in the volts range	HR: 50 - 250 bpm Amplitude: up to 5 mV	Amplitude: in the volts range	HR: 30 - 110.4 bpm Amplitude: in the volts range	Amplitude: 1.5V, 2.5V	<i>Signal range</i>
Only 4 HRs simulation “When building prototype circuit on printed circuit boards, extreme care had to be exercised in constructing the DSPIC adaptor card, particularly with regard to grounding and shielding”					<i>Mentioned limitations</i>
	 Figure 2. Analog conditioning circuit		 Adafruit Menta ECG Simulator Block Diagram		<i>Schematic/Diagram</i>

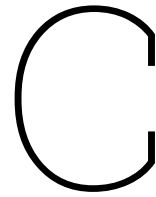
B

LMICs limitations

Table B.1: Summary on limitations of medical devices maintenance in Nepal and SSA. Information extracted from conversations with Dieuwertje (Biomechanical Designer with experience in Kenya), Jean-Luc Mpamiye (Biomedical Engineer Lead in Liberia), Anna Worm (Biomedical Engineer Advisor with experience in SSA), Ashish Chauhan (Biomedical Engineer in Nepal), Otto Ardeby (Biomedical Engineer consultant in Nepal) and Arjan Knulst (Biomedical Engineer Lead in Nepal). PM refers to preventive maintenance.

Statement	Explanation/Arguments	Limitations	Needs	Possible solutions
Most hospitals don't own simulators	<ul style="list-style-type: none"> - Equipment is expensive - In Nepal, there is no testing equipment besides few oxygen analyzers in hospitals supported by NSI 	Price Equipment availability	More affordable simulators More simulators	Cheaper than commercial simulators Equipment lending Donations
Budget for biomedical equipment maintenance is limited	<ul style="list-style-type: none"> - In SSA, generally funded by donors - In SSA, no dedicated maintenance budget - In GPH, part of each clinical department's budget is allocated to the biomedical department, based on their number of devices - In GPH, in case of no budget, foreign donors are searched 	No dedicated budget	Dedicated maintenance budget	Medical device management system
PM is not common in LMICs	<ul style="list-style-type: none"> - Hospitals with biomedical staff do PM but it's not common - Generally unskilled in biomedical engineering staff - In SSA, most countries have BME degree programmes and BMET training programmes, but not all 	Trained personnel	Personnel training	BME training programmes Educational simulators
Lack of trained personnel	<ul style="list-style-type: none"> - In SSA, low salaries and low career perspective 	Career prospects	Better salaries and career prospects	
Lack of spare parts	<ul style="list-style-type: none"> - In SSA, main problem is lack spare parts due to lack of funds and also lack of proactivity/skills of the personnel 	Trained personnel Proactivity of personnel Budget	Personnel training Dedicated maintenance budget	BME training programmes Educational simulators Medical device management system

<p>When preventive maintenance is done, it is very limited</p>	<ul style="list-style-type: none"> - In Tansen they do have the technical skills but lack spare parts or testing equipment - Technicians are busy repairing broken equipment - In GPH maintenance is based on testing and repairing, no parts replacement - In SSA, main problem is buying spare parts due to lack of funds and lack of proactivity/skills 	<p>Spare parts availability Testers availability</p>	<p>Personnel training Dedicated maintenance budget Locally available spare parts</p>	<p>BME training programmes Medical device management system Identification of local resources</p>
	<ul style="list-style-type: none"> - In Tansen they only simulate 60 bpm normal ECG and state no params can be modified (SKX-2000) 	<p>Testers features</p>	<p>Customizable tester features</p>	<p>Needed features and range identification</p>
<p>Not enough knowledge or resources to use available simulators</p>	<ul style="list-style-type: none"> - In SSA, they are often very complex and require regular expensive calibration in Europe - In SSA, schools and hospitals which have simulators come from donations and don't have budget for calibration or required training, end up not being used 	<p>Testers complexity Budget for calibration Trained personnel</p>	<p>Less complex testers Dedicated maintenance budget Personnel training</p>	<p>Main needed features identification Medical device management system BME training programmes</p>
<p>Simulators are frequently used when available</p>	<ul style="list-style-type: none"> - In Tansen they have an ECG simulator and they use it very often - NOT IN SSA, with no budget to calibrate and limited training, they are often not used 			
<p>No resources to maintain testing equipment</p>	<ul style="list-style-type: none"> - In SSA, equipment found in schools and hospitals coming from donations and no budget to calibrate it and limited training - Anna: They require regular expensive calibration in Europe 	<p>Budget for calibration Trained personnel Calibration availability</p>	<p>Dedicated maintenance budget Personnel training Locally available calibration</p>	<p>Medical device management system BME training programmes Local resources for calibration identification</p>
<p>Biomedical equipment testing is a fairly new practise in LMICs</p>	<ul style="list-style-type: none"> - Ashish, NSI - GPH hired first Biomedical Engineer in 2020 			



Arduino code

Below these lines, the code uploaded into the Arduino board is presented. First, some variables are defined. In simulation mode, the simulation rate is controlled by the variable `period = 1997` microseconds. In calibration mode, the variable `period2 = 500` milliseconds controls the time that the PWM is ON and OFF. Next, the code includes all the signals (based on waveform 101 from MIT-BIH database) with different HRs that were tested. For the simulation of a specific HR, the code line that contains the datapoints of that specific HR must be uncommented. The example below is set to simulate 60 bpm. By defining the signals as a `const` variable, this is stored in the 32 KB flash memory of Arduino, leaving the 2 KB of RAM free for other variables. After this, in the `setup()` function, Arduino's pin 4 is set as output, to be able to select simulation/calibration mode by means of a switch. Next, registers are programmed for 9-bit fast PWM (31.25 kHz) at pin 9. The `loop()` function is responsible for executing two different while statements depending on whether the switch at pin 4 selects calibration or simulation mode. When pin 4 is ON, simulation of a square signal is programmed by switching the PWM output ON and OFF every `period2 = 500` milliseconds. This defines a square waveform of 2 Hz. When pin 4 is OFF, the code sends a new datapoint from the stored waveform every `period = 1997` microseconds. The `loop()` function is responsible for keeping track of time by means of `micros()` and `millis()` functions.

```
#include <avr/io.h>

// Global variables available anywhere in the program
int i = 0;
int period = 1997; //in microseconds
int period2 = 500; //in milliseconds
unsigned long startMicros;
unsigned long startMillis;
unsigned long currentMicros;
unsigned long currentMillis;

// HR
//35
//const short y_data[] = {31, 30, 32, 31, 31, 31, 31, 31, 31, 31, 31, 32, 36, 39, 40, 39, 38, 36, 36, 38, 40,
43, 45, 46, 44, 40, 39, 41, 43, 47, 49, 49, 49, 48, 49, 50, 47, 50, 59, 63, 71, 79, 79, 84, 92, 93,
93, 89, 80, 78, 79, 76, 75, 77, 77, 76, 76, 76, 72, 64, 61, 63, 63, 60, 58, 58, 55, 49, 42, 39, 36,
37, 40, 39, 41, 40, 32, 29, 32, 35, 39, 43, 40, 34, 31, 30, 29, 29, 28, 28, 30, 30, 25, 23, 26, 30,
34, 38, 38, 36, 31, 22, 14, 12, 13, 14, 22, 45, 83, 131, 188, 253, 322, 398, 471, 511, 504, 455,
378, 299, 233, 177, 129, 99, 88, 79, 64, 52, 50, 51, 51, 50, 48, 45, 36, 27, 27, 26, 20, 16, 14, 8,
5, 1, 2, 9, 12, 9, 11, 11, 3, 0, 2, 6, 8, 6, 7, 8, 5, 3, 1, 0, 5, 9, 6, 4, 3, 3, 7, 8, 6, 8, 7, 6, 7,
3, 2, 7, 8, 9, 13, 15, 13, 8, 5, 6, 9, 11, 14, 14, 12, 10, 7, 7, 10, 14, 18, 20, 16, 13, 12, 7, 6,
13, 20, 21, 20, 20, 22, 24, 28, 30, 29, 29, 31, 33, 30, 28, 33, 38, 39, 41, 43, 43, 43, 43, 45, 50,
55, 57, 58, 57, 59, 62, 66, 70, 74, 76, 77, 80, 84, 87, 87, 88, 91, 94, 96, 99, 103, 104, 102, 101,
104, 107, 112, 116, 113, 107, 106, 106, 105, 107, 109, 109, 106, 99, 92, 91, 90, 89, 89, 88, 83, 77,
72, 69, 67, 64, 63, 62, 61, 58, 56, 54, 51, 48, 48, 49, 45, 42, 41, 36, 32, 35, 38, 38, 40, 42, 43,
37, 29, 28, 28, 24, 27, 31, 31, 34, 36, 29, 26, 31, 33, 31, 31, 31, 29, 29, 30, 29, 31, 35, 36, 32,
30, 27, 26, 29, 32, 34, 37, 39, 38, 38, 39, 38, 38, 40, 41, 39, 36, 31, 29, 33, 37, 39, 43, 44, 45,
46, 43, 38, 38, 38, 40, 46, 49, 47, 42, 39, 38, 37, 39, 43, 43, 43, 42, 38, 39, 44, 43, 41, 42, 42,
40, 37, 36, 40, 41, 40, 42, 42, 36, 31, 33, 37, 40, 44, 45, 41, 39, 39, 37, 36, 36, 34, 38, 41, 39,
```



```
29, 26, 28, 32, 33, 36, 39, 39, 38, 39, 39, 38, 39, 42, 40, 38, 33, 28, 31, 36, 38, 42, 45, 44, 46,
45, 40, 38, 38, 38, 43, 49, 49, 44, 40, 39, 37, 38, 41, 43, 43, 43, 40, 38, 43, 44, 41, 42, 43, 41,
38, 36, 38, 41, 41, 41, 42, 39, 33, 32, 36, 39, 42, 45, 43, 39, 40, 38, 36, 36, 35, 36, 40, 40, 37,
36, 33, 31, 32, 36, 39, 38, 36, 38, 40, 39, 38, 38, 40, 42, 41, 40, 40, 39, 40, 40, 39, 40, 40, 39,
40, 40, 39, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40,
40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40,
40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40,
37];
//65
//const short y_data[]={31, 30, 32, 31, 31, 32, 31, 31, 31, 31, 32, 36, 39, 40, 39, 38, 37, 36, 38, 40,
42, 45, 46, 44, 40, 40, 41, 43, 47, 49, 50, 49, 48, 49, 50, 47, 50, 59, 63, 71, 79, 79, 84, 93, 93,
93, 90, 81, 78, 80, 76, 75, 78, 77, 76, 76, 76, 72, 65, 61, 63, 63, 60, 59, 58, 56, 49, 43, 40, 37,
37, 40, 40, 40, 40, 33, 29, 32, 35, 39, 43, 40, 34, 32, 30, 29, 30, 29, 28, 30, 30, 25, 23, 26, 30,
34, 38, 38, 36, 32, 23, 14, 12, 13, 14, 21, 42, 80, 127, 183, 248, 316, 392, 466, 511, 508, 463, 388,
308, 240, 183, 134, 102, 89, 80, 66, 53, 50, 51, 51, 50, 49, 46, 38, 28, 27, 27, 20, 16, 14, 9, 5,
2, 1, 9, 12, 9, 11, 11, 4, 0, 1, 6, 8, 6, 7, 8, 6, 3, 1, 0, 4, 8, 6, 5, 3, 2, 6, 8, 6, 8, 8, 6, 7, 4,
1, 6, 9, 8, 12, 15, 13, 9, 5, 6, 9, 11, 14, 15, 13, 10, 7, 6, 10, 13, 17, 20, 17, 14, 12, 8, 5, 11,
20, 21, 20, 20, 21, 24, 27, 30, 30, 29, 31, 33, 31, 28, 32, 38, 39, 41, 43, 43, 43, 43, 45, 49, 54,
57, 58, 58, 59, 62, 65, 70, 74, 76, 77, 79, 84, 87, 88, 88, 91, 94, 96, 98, 102, 105, 103, 101, 103,
107, 111, 116, 114, 108, 106, 106, 105, 107, 109, 110, 108, 101, 94, 91, 91, 89, 89, 89, 85, 79, 73,
70, 68, 65, 63, 63, 61, 59, 57, 54, 52, 49, 48, 49, 47, 42, 41, 37, 32, 34, 38, 38, 39, 42, 43, 39,
31, 28, 29, 25, 26, 31, 31, 33, 36, 32, 26, 30, 33, 32, 31, 31, 30, 29, 30, 30, 34, 36, 33, 31,
28, 26, 28, 32, 33, 36, 39, 39, 38, 39, 39, 38, 39, 42, 40, 37, 32, 28, 32, 36, 38, 42, 45, 44, 46,
44, 39, 38, 38, 39, 44, 49, 48, 44, 40, 39, 37, 38, 42, 43, 43, 43, 40, 38, 43, 44, 41, 42, 43, 41,
38, 36, 38, 41, 41, 41, 42, 38, 32, 32, 36, 39, 43, 45, 42, 39, 40, 37, 36, 36, 35, 36, 40, 40, 37,
35, 33, 31, 32, 36, 39, 38, 36, 38, 40, 38, 38, 38, 40, 42, 41, 40, 40, 39, 40, 40, 39, 40, 40, 39,
40, 40, 39, 40, 39, 40, 40, 39, 40, 40, 39, 40, 40, 40, 40, 40, 39, 40, 40, 39, 40, 39, 40, 39, 40,
39, 40, 40, 39, 40, 40, 39, 40, 39, 39, 41, 37];
//70
//const short y_data[]={31, 31, 32, 31, 31, 31, 31, 31, 31, 31, 32, 36, 39, 39, 39, 38, 36, 36, 38, 40,
43, 45, 46, 44, 40, 40, 41, 43, 47, 49, 49, 49, 48, 49, 50, 47, 50, 59, 63, 71, 79, 79, 84, 93, 93,
93, 90, 81, 78, 79, 76, 75, 78, 77, 76, 76, 76, 72, 65, 62, 63, 63, 61, 58, 58, 56, 49, 43, 40, 37,
37, 40, 40, 41, 40, 33, 29, 32, 35, 39, 43, 41, 34, 31, 30, 29, 30, 29, 28, 30, 30, 25, 23, 26, 30,
34, 38, 38, 36, 32, 22, 14, 12, 13, 14, 21, 42, 80, 127, 184, 249, 317, 393, 467, 511, 507, 461, 386,
306, 239, 182, 133, 101, 88, 80, 65, 53, 50, 51, 51, 50, 49, 46, 38, 28, 27, 26, 20, 16, 14, 9, 5,
2, 1, 9, 12, 9, 11, 11, 4, 0, 2, 6, 8, 6, 7, 8, 6, 3, 1, 0, 4, 9, 6, 5, 3, 2, 7, 8, 6, 8, 7, 6, 7, 4,
1, 6, 9, 8, 12, 15, 13, 9, 5, 6, 9, 11, 14, 15, 12, 10, 7, 6, 10, 13, 17, 20, 17, 14, 12, 7, 5, 12,
20, 21, 20, 20, 21, 24, 28, 30, 30, 29, 31, 33, 30, 28, 32, 38, 39, 41, 43, 43, 43, 43, 45, 50, 54,
57, 58, 58, 59, 62, 65, 70, 74, 76, 77, 79, 84, 87, 88, 88, 91, 94, 96, 99, 102, 105, 103, 101, 103,
107, 111, 116, 114, 108, 106, 106, 105, 107, 109, 110, 108, 101, 93, 91, 91, 89, 89, 89, 85, 78, 73,
70, 68, 65, 63, 63, 61, 59, 56, 54, 52, 49, 48, 49, 47, 42, 41, 37, 32, 35, 38, 38, 39, 42, 43, 39,
30, 28, 29, 25, 26, 31, 31, 33, 36, 31, 26, 30, 33, 32, 31, 31, 30, 29, 30, 29, 30, 34, 36, 33, 30,
28, 26, 28, 32, 33, 36, 39, 39, 38, 39, 39, 38, 40, 41, 40, 37, 32, 28, 32, 36, 38, 42, 45, 44, 46,
44, 39, 38, 38, 39, 44, 49, 48, 44, 40, 39, 37, 38, 42, 43, 43, 43, 40, 38, 43, 44, 41, 42, 43, 41,
38, 36, 39, 41, 41, 41, 42, 38, 32, 32, 36, 39, 43, 45, 42, 39, 40, 37, 36, 36, 35, 36, 41, 40, 37,
35, 32, 31, 33, 36, 39, 38, 35, 39, 40, 38, 38, 38, 40, 42, 40, 39, 40, 39, 40, 40, 39, 40, 40, 39,
40, 39, 39, 41, 37];
//75
//const short y_data[]={31, 31, 32, 31, 31, 32, 31, 31, 32, 31, 32, 36, 39, 40, 40, 38, 37, 36, 38, 40,
42, 45, 46, 44, 40, 40, 41, 43, 47, 50, 50, 49, 48, 49, 50, 47, 50, 59, 63, 71, 79, 79, 84, 93, 94,
93, 90, 81, 78, 80, 76, 75, 78, 78, 76, 76, 76, 73, 65, 61, 64, 63, 61, 59, 58, 56, 49, 43, 40, 37,
37, 40, 40, 40, 40, 33, 29, 32, 35, 39, 43, 41, 34, 32, 30, 29, 30, 29, 28, 30, 31, 26, 23, 26, 30,
34, 38, 38, 36, 32, 23, 14, 12, 13, 14, 20, 41, 79, 125, 181, 246, 315, 390, 465, 510, 509, 465, 391,
310, 242, 185, 135, 102, 89, 81, 66, 53, 51, 51, 51, 49, 46, 38, 28, 27, 27, 21, 16, 14, 9, 5,
2, 1, 8, 12, 9, 11, 12, 5, 0, 1, 6, 8, 7, 7, 8, 6, 3, 1, 0, 4, 9, 6, 5, 4, 2, 6, 8, 6, 8, 8, 6, 7, 4,
1, 6, 9, 8, 12, 15, 13, 9, 5, 5, 9, 11, 14, 15, 13, 10, 7, 6, 9, 13, 17, 20, 17, 14, 12, 8, 5, 11,
19, 21, 20, 20, 21, 23, 27, 30, 30, 29, 31, 33, 31, 28, 32, 38, 39, 41, 43, 43, 43, 43, 45, 49, 54,
57, 58, 58, 59, 61, 65, 69, 74, 76, 77, 79, 83, 87, 88, 88, 91, 94, 96, 98, 102, 105, 104, 101, 103,
107, 111, 116, 115, 108, 106, 107, 106, 107, 109, 110, 108, 102, 94, 91, 91, 90, 89, 89, 85, 79, 74,
70, 68, 66, 64, 63, 62, 59, 57, 54, 52, 49, 48, 50, 47, 42, 41, 38, 33, 34, 38, 38, 39, 42, 43, 40,
31, 28, 29, 25, 25, 31, 31, 33, 36, 32, 26, 29, 33, 32, 31, 31, 30, 29, 30, 30, 30, 34, 36, 34, 31,
29, 26, 28, 32, 33, 36, 39, 39, 38, 39, 39, 38, 39, 42, 40, 38, 33, 28, 31, 36, 38, 42, 45, 44, 46,
45, 40, 38, 38, 38, 43, 49, 49, 44, 40, 39, 37, 38, 42, 43, 43, 43, 40, 38, 43, 44, 41, 42, 43, 41,
38, 36, 38, 41, 41, 41, 42, 39, 33, 32, 36, 39, 42, 45, 42, 40, 40, 38, 36, 36, 35, 36, 40, 40, 38,
35];
//80
//const short y_data[]={31, 31, 32, 31, 31, 32, 31, 31, 32, 31, 32, 36, 39, 40, 40, 38, 37, 36, 38, 40,
42, 45, 46, 44, 40, 40, 41, 43, 47, 49, 50, 49, 48, 49, 50, 47, 50, 59, 63, 71, 79, 79, 84, 93, 94,
93, 90, 81, 78, 80, 76, 75, 78, 78, 76, 76, 76, 73, 65, 61, 64, 63, 61, 59, 58, 56, 49, 43, 40, 37,
37, 40, 40, 40, 40, 33, 29, 32, 35, 39, 43, 41, 34, 32, 30, 29, 30, 29, 28, 30, 31, 26, 23, 26, 30,
34, 38, 38, 36, 32, 23, 14, 12, 13, 14, 20, 41, 79, 125, 181, 246, 315, 390, 465, 511, 509, 465, 391,
```

```
310, 242, 185, 135, 102, 89, 81, 66, 53, 51, 51, 51, 51, 49, 46, 38, 28, 27, 27, 21, 16, 14, 9, 5,
2, 1, 8, 12, 9, 11, 12, 5, 0, 1, 6, 8, 7, 7, 8, 6, 3, 1, 0, 4, 9, 6, 5, 4, 2, 6, 8, 6, 8, 8, 6, 7, 4,
1, 6, 9, 8, 12, 15, 13, 9, 5, 5, 9, 11, 14, 15, 13, 10, 7, 6, 9, 13, 17, 20, 17, 14, 12, 8, 5, 11,
19, 21, 20, 20, 21, 23, 27, 30, 30, 29, 31, 33, 31, 28, 32, 38, 39, 41, 43, 43, 43, 43, 45, 49, 54,
57, 58, 58, 59, 61, 65, 69, 74, 76, 77, 79, 83, 87, 88, 88, 91, 94, 96, 98, 102, 105, 104, 101, 103,
107, 111, 116, 115, 108, 106, 107, 106, 107, 109, 110, 108, 102, 94, 91, 91, 90, 89, 89, 85, 79, 74,
70, 68, 66, 64, 63, 62, 59, 57, 54, 52, 49, 48, 50, 47, 42, 41, 38, 33, 34, 38, 38, 39, 42, 43, 40,
31, 28, 29, 25, 25, 31, 31, 33, 36, 32, 26, 29, 33, 32, 31, 31, 30, 29, 30, 30, 30, 34, 36, 34, 31,
29, 26, 28, 32, 33, 36, 39, 39, 38, 39, 39, 38, 39, 42, 40, 38, 33, 28, 31, 36, 38, 42, 45, 44, 46,
45, 40, 38, 38, 38, 43, 49, 49, 44, 40, 39, 37, 38, 42, 43, 43, 43, 40, 38, 43, 44, 41, 41, 43, 42,
37};
//85
//const short y_data[]={31, 30, 32, 31, 31, 31, 31, 31, 31, 30, 32, 36, 39, 40, 39, 38, 36, 36, 38, 40,
42, 45, 46, 43, 40, 39, 40, 43, 47, 49, 49, 49, 47, 49, 49, 47, 51, 59, 63, 71, 79, 79, 84, 92, 93,
92, 89, 80, 78, 79, 75, 75, 77, 77, 76, 76, 75, 71, 64, 61, 63, 62, 60, 58, 57, 55, 48, 42, 39, 36,
37, 40, 39, 40, 39, 31, 29, 33, 35, 39, 43, 39, 33, 31, 30, 29, 29, 28, 28, 30, 30, 24, 23, 27, 30,
35, 38, 38, 36, 31, 20, 13, 12, 13, 15, 23, 48, 88, 137, 195, 261, 330, 406, 477, 511, 498, 445, 367,
289, 225, 170, 124, 96, 86, 77, 61, 51, 50, 51, 51, 50, 48, 44, 35, 27, 27, 25, 19, 16, 13, 8, 4, 1,
3, 10, 11, 9, 11, 10, 2, 0, 3, 7, 8, 6, 7, 8, 5, 3, 0, 0, 6, 8, 5, 4, 3, 3, 8, 7, 6, 8, 7, 6, 7, 2,
2, 8, 8, 9, 14, 15, 12, 7, 5, 7, 10, 12, 14, 14, 12, 9, 6, 7, 11, 14, 18, 19, 15, 13, 11, 6, 6, 15,
21, 21, 19, 20, 22, 25, 29, 30, 29, 29, 32, 32, 29, 28, 35, 39, 39, 42, 43, 42, 43, 46, 51, 55,
57, 57, 57, 59, 63, 66, 71, 75, 76, 77, 81, 85, 87, 87, 89, 92, 94, 96, 100, 103, 104, 101, 101,
104, 108, 113, 115, 110, 106, 106, 105, 105, 108, 109, 109, 104, 96, 91, 90, 89, 89, 89, 87, 81, 75,
71, 68, 66, 64, 62, 62, 60, 57, 55, 53, 50, 48, 49, 48, 44, 41, 39, 34, 33, 37, 38, 38, 40, 43, 41,
34, 28, 29, 27, 24, 29, 31, 31, 35, 34, 27, 27, 32, 32, 31, 31, 30, 29, 29, 30, 29, 32, 36, 34, 31,
29, 26, 27, 31, 33, 34, 38, 39, 38, 38, 39, 38, 38, 41, 40, 38, 34, 29, 30, 35, 37, 41, 44, 44, 45,
45, 40, 38, 36};
//90 bpm
//const short y_data[]={31, 31, 32, 31, 31, 31, 31, 31, 31, 31, 32, 36, 39, 40, 39, 38, 37, 36, 38, 40,
42, 45, 46, 44, 40, 39, 41, 43, 47, 49, 49, 49, 48, 49, 50, 47, 50, 59, 63, 71, 79, 79, 84, 92, 93,
92, 90, 80, 78, 79, 76, 75, 78, 77, 76, 76, 76, 72, 64, 61, 63, 63, 60, 58, 58, 55, 49, 42, 39, 37,
37, 40, 40, 40, 40, 32, 29, 32, 35, 39, 43, 40, 34, 31, 30, 29, 29, 28, 28, 30, 30, 25, 23, 26, 30,
34, 38, 38, 36, 32, 22, 14, 12, 13, 14, 22, 44, 83, 130, 187, 253, 321, 397, 470, 511, 504, 456, 380,
300, 234, 178, 130, 99, 88, 79, 64, 52, 50, 51, 51, 50, 48, 45, 37, 27, 27, 26, 20, 16, 14, 9, 5, 1,
2, 9, 12, 9, 11, 11, 4, 0, 2, 6, 8, 6, 7, 8, 6, 3, 1, 0, 5, 9, 6, 4, 3, 3, 7, 8, 6, 8, 7, 6, 7, 3,
2, 7, 8, 8, 13, 15, 13, 8, 5, 6, 9, 11, 14, 14, 12, 10, 7, 7, 10, 14, 18, 20, 16, 13, 12, 7, 5, 13,
20, 21, 20, 20, 22, 24, 28, 30, 29, 29, 31, 33, 30, 28, 33, 38, 39, 41, 43, 43, 43, 43, 45, 50, 55,
57, 58, 58, 59, 62, 66, 70, 74, 76, 77, 80, 84, 87, 88, 88, 91, 94, 96, 99, 103, 105, 102, 101, 104,
107, 112, 116, 113, 107, 106, 106, 105, 107, 109, 109, 107, 99, 93, 91, 90, 89, 89, 88, 83, 77, 72,
69, 67, 65, 63, 62, 61, 58, 56, 54, 51, 48, 48, 49, 46, 42, 41, 36, 32, 35, 38, 38, 40, 42, 43, 37,
29, 28, 28, 24, 27, 31, 31, 34, 36, 30, 26, 31, 33, 31, 31, 31, 29, 29, 30, 29, 31, 35, 36, 32, 30,
27, 26, 29, 32, 34, 37, 39, 39, 36};
//95
//const short y_data[]={31, 31, 31, 31, 31, 31, 31, 31, 31, 31, 32, 36, 39, 39, 39, 38, 36, 36, 38, 40,
43, 45, 46, 44, 40, 39, 41, 43, 47, 49, 49, 49, 48, 49, 50, 47, 50, 59, 63, 71, 79, 79, 84, 92, 93,
93, 89, 80, 78, 79, 76, 75, 77, 77, 76, 76, 76, 72, 64, 61, 63, 63, 60, 58, 58, 56, 49, 42, 39, 36,
37, 40, 39, 41, 40, 32, 29, 32, 35, 39, 43, 40, 34, 31, 30, 29, 29, 29, 28, 30, 30, 25, 23, 26, 30,
34, 38, 38, 36, 32, 22, 14, 12, 13, 14, 21, 44, 82, 130, 187, 252, 321, 396, 470, 511, 505, 457, 381,
301, 235, 179, 130, 100, 88, 79, 64, 52, 50, 51, 51, 50, 48, 45, 37, 28, 27, 26, 20, 16, 14, 9, 5,
1, 2, 9, 12, 9, 11, 11, 4, 0, 2, 6, 8, 6, 7, 8, 6, 3, 1, 0, 5, 9, 6, 4, 3, 3, 7, 8, 6, 8, 7, 6, 7, 4,
2, 7, 9, 8, 13, 15, 13, 9, 5, 6, 9, 11, 14, 14, 12, 10, 7, 7, 10, 14, 18, 20, 16, 13, 12, 7, 5, 13,
20, 21, 20, 20, 22, 24, 28, 30, 29, 29, 31, 33, 30, 28, 33, 38, 39, 41, 43, 43, 43, 43, 45, 50, 55,
57, 58, 58, 59, 62, 65, 70, 74, 76, 77, 80, 84, 87, 88, 88, 91, 94, 96, 99, 102, 104, 103, 101, 103,
107, 111, 116, 113, 107, 106, 106, 105, 107, 109, 109, 107, 100, 93, 91, 90, 89, 89, 88, 84, 77, 72,
69, 67, 65, 63, 62, 61, 59, 56, 54, 51, 48, 48, 49, 46, 42, 41, 36, 32, 35, 38, 38, 39, 42, 43, 37,
30, 28, 28, 25, 26, 31, 31, 33, 36, 30, 26, 31, 33, 31, 31, 31};
//100
//const short y_data[]={31, 31, 31, 31, 31, 31, 31, 31, 31, 31, 32, 36, 39, 40, 40, 38, 37, 36, 38, 40,
43, 45, 46, 44, 40, 40, 41, 43, 47, 50, 49, 49, 48, 49, 50, 47, 50, 59, 63, 71, 79, 79, 84, 93, 93,
93, 90, 81, 78, 80, 76, 75, 78, 78, 76, 76, 76, 73, 65, 62, 63, 63, 61, 59, 58, 56, 49, 43, 40, 37,
37, 40, 40, 41, 40, 33, 29, 32, 35, 39, 43, 41, 34, 32, 30, 29, 30, 29, 28, 30, 31, 26, 23, 26, 30,
34, 38, 38, 36, 32, 23, 14, 12, 13, 14, 20, 41, 79, 125, 181, 246, 315, 390, 465, 511, 509, 465, 391,
310, 242, 185, 135, 102, 89, 81, 66, 53, 51, 51, 51, 50, 49, 46, 38, 28, 27, 27, 21, 16, 14, 9, 5,
2, 1, 8, 12, 9, 11, 12, 5, 0, 1, 6, 8, 7, 7, 8, 6, 3, 1, 0, 4, 9, 6, 5, 4, 2, 6, 8, 6, 8, 8, 6, 7, 4,
1, 6, 9, 8, 12, 15, 13, 9, 5, 5, 9, 11, 14, 15, 13, 10, 8, 6, 9, 13, 17, 20, 17, 14, 12, 8, 5, 11,
19, 21, 20, 20, 21, 23, 27, 30, 30, 29, 31, 33, 31, 28, 32, 38, 39, 41, 43, 43, 43, 43, 45, 49, 54,
57, 58, 58, 59, 61, 65, 69, 74, 76, 77, 79, 83, 87, 88, 88, 91, 94, 96, 98, 102, 105, 104, 101, 103,
107, 111, 116, 115, 108, 106, 107, 105, 107, 109, 110, 108, 102, 94, 92, 91, 90, 89, 89, 85, 79, 74,
70, 68, 65, 64, 63, 62, 59, 57, 54, 52, 49, 48, 50, 47, 43, 41, 38, 33, 34, 38, 38, 38, 42, 44, 39,
33};
//105
```

```
//const short y_data[]={31, 30, 32, 31, 31, 32, 31, 31, 32, 30, 32, 36, 39, 40, 39, 38, 37, 36, 38, 40,
42, 45, 46, 44, 40, 39, 41, 43, 47, 49, 49, 49, 48, 49, 50, 47, 50, 59, 63, 71, 79, 79, 84, 92, 93,
93, 90, 81, 78, 80, 76, 75, 78, 77, 76, 76, 76, 72, 65, 61, 63, 63, 60, 59, 58, 56, 49, 43, 40, 37,
37, 40, 40, 40, 40, 33, 29, 32, 35, 39, 43, 40, 34, 31, 30, 29, 30, 29, 28, 30, 30, 25, 23, 26, 30,
34, 38, 38, 36, 32, 22, 14, 12, 13, 14, 21, 42, 80, 127, 184, 249, 317, 393, 467, 511, 507, 462, 386,
306, 239, 182, 133, 101, 88, 80, 65, 53, 50, 51, 51, 50, 49, 46, 38, 28, 27, 26, 20, 16, 14, 9, 5,
1, 1, 9, 12, 9, 11, 11, 4, 0, 1, 6, 8, 6, 7, 8, 6, 3, 1, 0, 4, 8, 6, 4, 3, 2, 6, 8, 6, 8, 7, 6, 7, 4,
1, 6, 9, 8, 12, 15, 13, 9, 5, 6, 9, 11, 14, 15, 12, 10, 7, 6, 10, 13, 17, 20, 17, 14, 12, 7, 5, 12,
20, 21, 20, 20, 21, 24, 28, 30, 30, 29, 31, 33, 30, 28, 32, 38, 39, 41, 43, 43, 43, 43, 45, 50, 54,
57, 58, 58, 59, 62, 65, 70, 74, 76, 77, 79, 84, 87, 88, 88, 91, 94, 96, 99, 102, 105, 103, 101, 103,
107, 111, 116, 114, 108, 106, 106, 105, 107, 109, 110, 108, 101, 94, 91, 91, 89, 89, 89, 84, 78, 73,
69, 68, 65, 63, 63, 61, 58, 57, 54, 51, 51, 43};
//110
//const short y_data[]={31, 26, 35, 30, 30, 33, 30, 31, 32, 30, 32, 36, 38, 40, 39, 38, 37, 36, 38, 40,
42, 45, 46, 43, 40, 39, 40, 43, 47, 49, 50, 49, 48, 49, 49, 47, 50, 58, 63, 71, 79, 79, 84, 92, 93,
92, 90, 81, 78, 80, 76, 75, 78, 77, 76, 76, 76, 72, 64, 61, 63, 63, 60, 59, 58, 55, 49, 42, 39, 37,
37, 40, 40, 40, 40, 32, 29, 33, 35, 39, 43, 40, 34, 31, 30, 29, 29, 28, 28, 30, 30, 25, 23, 26, 30,
34, 38, 38, 36, 32, 22, 14, 12, 13, 14, 21, 44, 82, 129, 186, 251, 320, 396, 469, 510, 505, 458, 382,
302, 236, 179, 131, 100, 88, 79, 64, 52, 50, 51, 51, 50, 48, 45, 37, 28, 27, 26, 20, 16, 14, 9, 5,
1, 2, 9, 12, 9, 11, 11, 4, 0, 2, 6, 8, 6, 7, 8, 6, 3, 1, 0, 5, 9, 6, 5, 3, 3, 7, 8, 6, 8, 7, 6, 7, 4,
2, 7, 8, 8, 13, 15, 13, 9, 5, 6, 9, 11, 14, 14, 12, 10, 7, 7, 10, 13, 18, 20, 16, 14, 12, 7, 5, 12,
20, 21, 19, 20, 21, 24, 28, 30, 29, 31, 33, 30, 28, 33, 38, 39, 41, 43, 43, 43, 43, 45, 50, 54,
57, 58, 57, 59, 62, 65, 70, 74, 76, 77, 79, 84, 88, 87, 89, 91, 94, 96, 98, 102, 105, 102, 101, 104,
106, 112, 116, 113, 108, 106, 106, 106, 106, 110, 110, 106, 101, 93, 89, 93, 88, 87, 95, 72};
//115
//const short y_data[]={31, 24, 37, 29, 30, 34, 29, 32, 33, 29, 33, 36, 38, 41, 39, 38, 38, 36, 38, 41,
42, 45, 47, 43, 40, 40, 40, 43, 47, 49, 50, 49, 48, 49, 49, 47, 50, 58, 63, 71, 79, 79, 84, 93, 94,
92, 90, 81, 78, 80, 76, 75, 78, 77, 76, 76, 76, 72, 65, 61, 64, 63, 60, 59, 58, 56, 49, 42, 40, 37,
37, 40, 40, 40, 40, 33, 28, 33, 35, 39, 43, 40, 34, 32, 30, 29, 30, 29, 28, 30, 30, 25, 23, 26, 30,
34, 38, 38, 36, 32, 22, 14, 12, 13, 14, 21, 42, 80, 128, 184, 249, 318, 393, 467, 510, 507, 461, 386,
306, 239, 182, 133, 101, 88, 80, 65, 53, 50, 51, 51, 50, 49, 46, 38, 28, 27, 26, 20, 16, 14, 9, 5,
2, 1, 9, 12, 9, 11, 11, 4, 0, 2, 6, 8, 6, 7, 8, 6, 3, 1, 0, 4, 9, 6, 5, 3, 2, 7, 8, 6, 8, 7, 6, 8, 4,
1, 7, 8, 8, 13, 15, 13, 9, 5, 6, 9, 11, 14, 14, 13, 10, 7, 7, 10, 13, 17, 20, 17, 14, 12, 8, 5, 11,
20, 21, 19, 20, 21, 24, 28, 30, 30, 29, 31, 33, 30, 27, 33, 38, 39, 41, 43, 43, 43, 43, 45, 50, 54,
57, 58, 57, 59, 61, 65, 70, 74, 76, 78, 79, 84, 88, 87, 89, 91, 93, 97, 98, 102, 106, 102, 101, 104,
105, 113, 116, 111, 111, 104, 103, 116, 87};
//120 - take pic
//const short y_data[]={31, 25, 37, 29, 30, 34, 29, 32, 33, 29, 33, 36, 38, 41, 39, 38, 38, 36, 38, 41,
42, 45, 47, 43, 41, 40, 40, 43, 47, 49, 50, 49, 48, 49, 50, 48, 51, 58, 64, 71, 79, 80, 84, 93, 94,
93, 90, 81, 78, 80, 76, 75, 78, 77, 76, 76, 76, 73, 65, 61, 64, 63, 61, 59, 58, 56, 50, 43, 40, 37,
37, 40, 40, 40, 41, 33, 29, 32, 35, 39, 44, 41, 34, 32, 30, 30, 30, 29, 28, 30, 31, 26, 23, 26, 30,
34, 39, 38, 36, 33, 23, 14, 13, 13, 14, 20, 41, 79, 125, 181, 246, 315, 390, 465, 511, 509, 465, 391,
310, 243, 185, 135, 102, 89, 81, 66, 53, 51, 51, 51, 51, 49, 46, 38, 28, 27, 27, 21, 16, 14, 9, 5,
2, 1, 8, 12, 9, 11, 12, 5, 0, 1, 6, 8, 7, 7, 9, 6, 3, 2, 0, 4, 9, 6, 5, 4, 2, 6, 8, 6, 8, 8, 6, 8, 4,
1, 6, 9, 8, 12, 15, 14, 9, 5, 6, 9, 11, 14, 15, 13, 11, 7, 7, 9, 13, 17, 20, 17, 14, 12, 8, 5, 11,
20, 22, 20, 20, 21, 23, 28, 30, 30, 29, 31, 33, 31, 27, 32, 38, 39, 41, 43, 43, 43, 43, 45, 50, 53,
57, 58, 57, 59, 61, 64, 70, 73, 76, 78, 78, 84, 88, 86, 90, 91, 91, 99, 97, 99, 115, 86};
//125
//const short y_data[]={31, 27, 35, 30, 30, 33, 30, 31, 32, 30, 32, 36, 38, 40, 39, 38, 37, 36, 38, 40,
42, 45, 46, 43, 40, 40, 40, 43, 47, 49, 50, 49, 48, 49, 50, 47, 51, 59, 63, 71, 79, 79, 84, 93, 93,
92, 90, 80, 78, 80, 76, 75, 78, 77, 76, 76, 76, 72, 64, 61, 63, 63, 60, 59, 58, 55, 49, 42, 39, 37,
37, 40, 40, 40, 40, 32, 29, 33, 35, 39, 43, 40, 34, 32, 30, 29, 30, 28, 28, 30, 30, 25, 23, 26, 30,
34, 39, 38, 36, 32, 22, 14, 13, 13, 14, 22, 45, 84, 131, 188, 254, 323, 398, 471, 510, 503, 455, 378,
299, 233, 177, 129, 99, 87, 79, 64, 52, 51, 51, 51, 50, 48, 45, 37, 27, 26, 20, 16, 14, 8, 5, 1,
2, 10, 12, 9, 11, 11, 4, 0, 2, 7, 8, 6, 7, 8, 6, 3, 1, 0, 5, 8, 6, 4, 3, 3, 7, 8, 6, 8, 7, 6, 7, 4,
2, 7, 9, 8, 13, 15, 12, 9, 5, 6, 10, 11, 14, 15, 12, 10, 7, 6, 10, 14, 17, 20, 16, 13, 12, 7, 5, 14,
20, 21, 20, 20, 22, 24, 28, 31, 29, 29, 32, 32, 30, 29, 33, 39, 40, 41, 44, 43, 42, 44, 45, 50, 56,
56, 58, 59, 57, 63, 66, 68, 77, 75, 75, 88, 70};
//130
//const short y_data[]={31, 28, 33, 30, 30, 32, 30, 31, 32, 30, 32, 36, 38, 40, 39, 38, 37, 36, 37, 40,
42, 45, 46, 43, 40, 39, 40, 43, 47, 49, 49, 49, 47, 49, 49, 47, 51, 59, 63, 71, 78, 79, 84, 92, 92,
92, 89, 80, 78, 79, 75, 75, 77, 77, 76, 76, 75, 71, 63, 61, 63, 62, 60, 58, 57, 55, 48, 42, 39, 36,
37, 40, 39, 40, 39, 31, 29, 33, 35, 39, 43, 39, 33, 31, 30, 29, 29, 28, 28, 30, 29, 24, 23, 27, 30,
35, 38, 37, 35, 30, 20, 13, 12, 13, 15, 24, 49, 89, 138, 196, 262, 331, 408, 478, 511, 497, 443, 364,
287, 223, 168, 122, 96, 86, 76, 61, 51, 50, 51, 51, 49, 48, 44, 34, 26, 27, 25, 18, 16, 13, 8, 4, 1,
3, 11, 11, 9, 11, 10, 2, 0, 3, 7, 8, 6, 7, 8, 5, 3, 0, 0, 6, 8, 5, 4, 3, 3, 8, 7, 6, 8, 6, 6, 7, 2,
3, 8, 8, 9, 14, 15, 12, 7, 4, 7, 10, 12, 14, 14, 11, 9, 6, 7, 11, 14, 19, 19, 15, 13, 10, 6, 7, 15,
21, 20, 19, 21, 22, 25, 29, 30, 29, 29, 32, 32, 28, 28, 35, 38, 39, 42, 42, 43, 43, 42, 47, 51, 54,
60, 51};
//135
```

```

//const short y_data[]={31, 30, 32, 31, 31, 32, 31, 31, 32, 30, 32, 36, 39, 40, 39, 38, 37, 36, 38, 40,
    42, 45, 46, 44, 40, 39, 41, 43, 47, 49, 49, 49, 48, 49, 50, 47, 50, 59, 63, 71, 79, 79, 84, 92, 93,
    92, 90, 80, 78, 79, 76, 75, 78, 77, 76, 76, 76, 72, 64, 61, 63, 63, 60, 58, 58, 55, 49, 42, 39, 37,
    37, 40, 39, 40, 40, 32, 29, 32, 35, 39, 43, 40, 34, 31, 30, 29, 29, 28, 28, 30, 30, 25, 23, 26, 30,
    34, 38, 38, 36, 32, 22, 14, 12, 13, 14, 22, 44, 83, 130, 187, 253, 321, 397, 470, 510, 504, 456, 380,
    300, 234, 178, 130, 99, 88, 79, 64, 52, 50, 51, 51, 50, 48, 45, 37, 28, 27, 26, 20, 16, 14, 9, 5, 1,
    2, 9, 12, 9, 11, 11, 4, 0, 2, 6, 8, 6, 7, 8, 6, 3, 1, 0, 5, 9, 6, 5, 3, 3, 7, 8, 6, 8, 7, 6, 7, 3,
    2, 7, 8, 8, 13, 15, 13, 8, 5, 6, 9, 11, 14, 14, 12, 10, 7, 7, 10, 14, 18, 20, 16, 13, 12, 7, 5, 13,
    20, 21, 19, 20, 22, 24, 28, 30, 29, 29, 31, 33, 30, 27, 34, 38, 39, 43, 40};

//150
//const short y_data[]={31, 33, 30, 32, 31, 31, 32, 31, 31, 31, 32, 36, 39, 39, 40, 39, 36, 36, 38, 40,
    43, 45, 46, 44, 40, 40, 41, 43, 47, 50, 49, 49, 48, 49, 50, 47, 50, 59, 63, 71, 79, 79, 84, 93, 93,
    93, 90, 81, 78, 80, 76, 75, 78, 78, 76, 76, 76, 73, 65, 62, 63, 63, 61, 59, 58, 56, 49, 43, 40, 37,
    37, 40, 40, 40, 40, 33, 29, 32, 35, 39, 43, 41, 34, 32, 30, 29, 30, 29, 28, 30, 31, 26, 23, 26, 30,
    34, 38, 38, 36, 32, 23, 14, 12, 13, 14, 20, 41, 79, 125, 181, 246, 315, 390, 465, 511, 509, 465, 391,
    310, 242, 185, 135, 102, 89, 81, 66, 53, 51, 51, 51, 50, 49, 46, 38, 28, 27, 27, 21, 16, 14, 9, 5,
    2, 1, 8, 12, 9, 10, 11, 5, 0, 1, 6, 8, 6, 7, 8, 6, 3, 1, 0, 4, 8, 6, 4, 3, 2, 6, 8, 6, 7, 8, 6, 7, 4,
    1, 6, 9, 8, 12, 15, 13, 9, 5, 5, 9, 11, 13, 15, 13, 10, 8, 6, 9, 14, 16, 20, 18, 11, 18};

// The setup function runs once when you press reset or power the board
void setup() {
    Serial.begin(9600);
    pinMode(4,INPUT); // Switch connected to pin 4

    // Reset both Timer/counters 1
    TCCR1A = 0;
    TCCR1B = 0;

    // Set UNO pin 9/PB1 and pin10/PB2 to output
    DDRB |= _BV(DDB1) | _BV(DDB2); // we set up both PB1 (pin9) and PB2 (pin10) // Pin 9 connected to
        simulation circuit

    // Set registers for toggling pins, timer resolution and timer prescaler
    // Mode 6 for 9-bit fast PWM -> 2^9=512 resolution
    // Prescaler set to 1 -> freq=clk/(2^9x1)=(16x10^6)/(512x1)=31.25kHz
    TCCR1A = _BV(COM1A1) | _BV(COM1B1) | _BV(WGM11);
    TCCR1B = _BV(WGM12) | _BV(CS10);

    // Get initial time
    startMicros = micros();
    startMillis = millis();
}

// The loop function runs over and over again forever
void loop() {
    // Generate square function to test amplitude simulation
    // when switch is ON
    if (digitalRead(4)== HIGH){
        const short y_data2[]={511,0};
        i=0;
        while(i < sizeof(y_data2)/sizeof(y_data2[0])){
            currentMillis = millis();
            if(currentMillis - startMillis >= period2){
                OCR1A = y_data2[i];
                startMillis = currentMillis;
                i++;
            }
        }
        i = 0;

    // Generate ECG waveform
    // when switch is OFF
    } else {
        i=0;
        while(i < sizeof(y_data)/sizeof(y_data[0])){
            currentMicros = micros();
            if(currentMicros - startMicros >= period){
                OCR1A = y_data[i];
                startMicros = currentMicros;
                i++;
            }
        }
    }
}

```



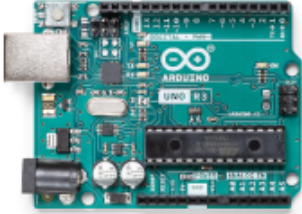
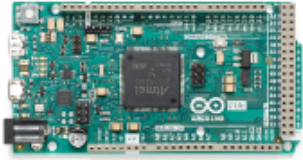


```
    }  
  }  
  i = 0;  
}  
  
}
```

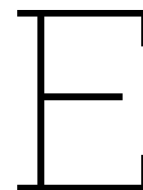
D

Arduinos comparison

Table D.1: Comparison of some Arduino models with potential to meet the computational requirements of ECG simulation.

	Arduino UNO R3	Arduino DUE	Arduino MEGA2560	Arduino MKR ZERO with SD
<i>Supply voltage</i>	5V USB, 7-12V (recomm.) in-built power jack or Vin	5V USB, 7-12V (recomm.) in-built power jack or Vin	5V USB, 7-12V (recomm.) in-built power jack or Vin	5V USB, 3.7V Li-Po battery or 3.3V Vin
<i>Operating voltage - defines max voltage of PWM</i>	5V operating voltage	3.3V operating voltage	3.3V operating voltage	3.3V operating voltage
<i>Output voltages</i>	5V 3.3V	5V 3.3V	5V 3.3V	5V 3.3V
<i>uC</i>	8-bit ATmega328P 16 MHz	32-bit AT91SAM3X8E 84 MHz	8-bit ATmega2560 16 MHz	32 bit SAMD21 48MHz
<i>Oscillator CLK</i> - Type - Freq	- Ceramic resonator - 16 MHz	- Crystal - 32 kHz used to generate 84MHz → this cannot be outputted in arduino	- Crystal - 16 MHz	- Crystal - 32MHz (RTC), 48MHz
<i>Oscillator theoretical accuracy</i>	±.5% for ceramic	assume +-30ppm for crystal	assume +-30ppm for crystal	assume +-30ppm for crystal
<i>ADC y/n, bits</i>	No	Yes 2x, 12 bit	No	Yes 1x, 10-bit
<i>PWM outputs and bits</i>	6 PWM outputs controlled by 3 Timers	12 PWM outputs can be set to 12/16 bit	15 PWM outputs can be set to 12/16 bit	12 PWM outputs can be set to 8/10 bit

	Timer 1 has 16-bits, Timer 2&3 have 8-bits			
<i>Memory</i>	2KB SRAM, 32KB FLASH, 1KB EEPROM	96KB SRAM, FLASH MEMORY 512 KB	FLASH MEMORY 256 KB of which 8 KB used by bootloader SRAM 8 KB EEPROM 4 KB	Flash Memory 256 KB Flash Memory for Bootloader 8 KB SRAM 32 KB EEPROM no
<i>Price at store.arduino.cc</i>	24€	42€	42€	26€
<i>Price at daraz.com.np</i>	1,700 NPR	6,250 NPR	6,105 NPR	Not found
<i>Image</i>				



ATmega programming

The Arduino function `analogWrite(pin, duty cycle)` provides an intuitive way to generate a PWM signal at the specified output pin and with the selected duty cycle. However, using this function severely limits the functionalities of ATmega328 and the resolution of the generated signal.

One of the limitations is in terms of PWM frequency. When using `analogWrite()`, this is fixed to 490 Hz for pins 3, 5, 6, 11 and 980 Hz for pins 9, 10. Opposite to this, ATmega can generate PWM frequencies from as low as 30.5 Hz to as high as 62.5 kHz. A second limitation is in terms of PWM resolution. While `analogWrite()` uses 8 bits by default, ATmega PWM resolution can be increased to 9, 10 and even 16 bits. These functionalities can still be accessed by programming ATmega328 directly using C code instead of Arduino pre-built functions. Finally, the `analogWrite()` function has a bigger code size and takes longer to execute than the equivalent C code required for setting up a PWM mode [52].

Given the previous reasons, and despite the fact that the code becomes less intuitive, `analogWrite()` was not used in this work. The alternative is to directly program the PWM options of the ATmega. For this, setting the bits of specific registers that control PWM generation is needed. Information needed for this is found in the datasheet of the microcontroller, and summarized in Table E.1. Although the code becomes way less intuitive, and setting it up requires a good understanding of the datasheet, once these lines of code are set, they will not require any further modification, since the desired PWM frequency and resolution should be fixed. The only register that needs frequent updates is the duty cycle, which changes for every data point of the ECG waveform. However, this is as simple as setting `OCR1A` to the desired value. For these reasons, the decline in intuitiveness of the code is expected to not impact the simulator use too much.

The steps and code required to program ATmega for 9-bit fast (31.25kHz) PWM output at pin 9 are explained next. Reasons for choosing these resolution and PWM frequency are explained more in depth in Appendix F.

PWM output is only available in 6 pins of Arduino UNO (pin 3, 5, 6, 9, 10, 11). In the code attached, both pin 9 and 10 are configured, although only the signal from pin 9 is effectively used by Frankie. Technical information required for PWM programming is extracted from ATmega328 datasheet [25] and presented in Figure XYZ. According to this, Pin 9 in the Arduino is connected to pin PB1 in ATmega328, and pin 10 is connected to PB2. First of all, these pins must be configured as outputs by setting `DDB1` and `DDB2` to 1. The line of code for that is:

```
DDRB |= \_BV(DDB1) | \_BV(DDB2);
```

From the datasheet we obtain that PB1 and PB2 are both controlled by register `OC1A` and that Timer/Counter 1 needs to be configured, using the control registers `TCCR1A` and `TCCR1B` (see Table E.1). A description of these registers is presented in Table E.2.

Table E.1: Arduino PWM ports and their associated ports, registers and timers in ATmega328. Data extracted from [25].

Arduino port	ATmega328	Register	Timer/Counterx	# of bits	Control registers
3	PD3	OC2B	Timer/Counter2	8	TCCR2A and TCCR2B
5	PD5	OC0B	Timer/Counter0	8	TCCR0A and TCCR0B
6	PD6	OC0A	Timer/Counter0	8	TCCR0A and TCCR0B
9	PB1	OC1A	Timer/Counter1	16	TCCR1A and TCCR1B
10	PB2	OC1B	Timer/Counter1	16	TCCR1A and TCCR1B
11	PB3	OC2A	Timer/Counter2	8	TCCR2A and TCCR2B

Table E.2: ATmega358P registers TCCR1A and TCCR1B description. Extracted from section 16.11. from datasheet [25].

Bit (0x80)	7	6	5	4	3	2	1	0	
	COM1A1	COM1A0	COM1B1	COM1B0	–	–	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

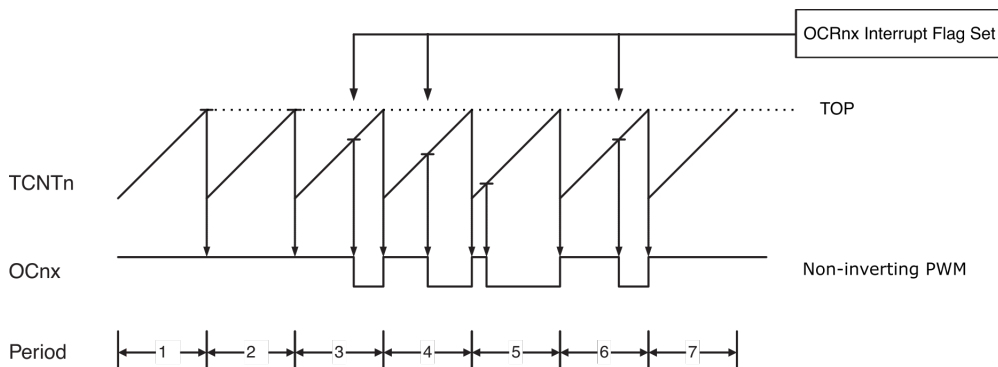
Bit (0x81)	7	6	5	4	3	2	1	0	
	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

In TCCR1A, the first four bits control the OC1A and OC1B register, which are assigned to PB1 and PB2 in ATmega and pins 9 and 10 in Arduino. To use non-inverting fast PWM mode, we must set OCR1A1:0 to 10 and OCR1B1:0 to 10. The last two bits of TCCR1A control WGM, or Waveform Generation Mode, which has two extra bits in register TCCR1B. For a 9-bit fast PWM signal, mode 6 of WGM must be configured. This is achieved by setting WGM12:0 to 110. In this case, the TOP value is 511. This means that the counter will increment until the counter value matches 511, after which the counter will be cleared and the next timer clock cycle will start (see Fig. E.1). PWM resolution in bits is defined by:

$$R_{FPWM} = \frac{\log(TOP + 1)}{\log(2)}$$

The last three bits of TCCR1B control CS12:0, which are used to select the clock prescaler divider (1, 8, 64, 256, or 1024). No prescaling is achieved by setting CS12:0 bits to 001. The PWM frequency for the output can be calculated as:

$$f_{OCnxPWM} = \frac{f_{clk_IO}}{(1 + TOP)} = \frac{16 \cdot 10^6}{(1 + 511)} = 31.25 \text{ kHz}$$

**Figure E.1:** Timing diagram of Fast PWM mode. Modified from [25]

Summarizing, the code to set up Timer 1 with the desired characteristics is:

```
TCCR1A = _BV(COM1A1) | _BV(COM1B1) | _BV(WGM11);  
TCCR1B = _BV(WGM12) | _BV(CS10);
```

After this, duty cycle is specified by setting OCR1A to the desired value (integer from 0 to 511). For instance, a duty cycle of 50% would correspond to 256, since $256/511=0.05$. This would read as:

```
OCR1A = 256; //duty cycle => 256/511 => 50\%
```

The only thing left to do is translate the ECG datapoints into integers from 0 - 511, with 0 representing the lowest voltage recorded and 511 the maximum. After the LPF, the generated PWM will get smoothed into a DC voltage with $amplitude = 5V \cdot duty\ cycle$.

F

D/A conversion based on PWM and LPF

In this work, a PWM signal and LPF are used to achieve Digital-to-Analog conversion of an ECG waveform. The Arduino PWM output is a square wave with 5V amplitude and variable duty cycle. By modifying the ratio of time that the PWM signal is ON versus OFF during one period (i.e., duty cycle), any DC values from 0V - 5V can be obtained after low-pass filtering the square signal. The resolution being fixed by the PWM frequency and the LPF architecture (LPF filter design is discussed in Appendix G).

The working principle of this D/A conversion method lies in the frequency components of the PWM signal. In the frequency domain the PWM signal is composed by a DC component of $amplitude = 5V \cdot duty_cycle$ and other components at odd-integer multiples of the PWM frequency. A LPF analogically filters signals above its cut-off frequency. Thus, ideally, by combining a PWM with a LPF, all the high frequency components can be filtered off, with only the DC component remaining. However, in real life, some design limitations need to be considered.

One of the limitations of PWM usage as DAC is the trade-off between DAC bandwidth and DAC resolution. This must be taken into account when defining the cut-off frequency of the LPF. With a cut-off frequency that is too low, the frequencies that can be generated by the DAC are also reduced. With a cut-off that is too high, accuracy of the generated DC voltage will be reduced by a phenomenon called ripple (see Fig. F.3). This happens when the cut-off frequency is too close to the harmonics and they cannot be effectively filtered, resulting in an output signal contaminated by higher frequency components. Luckily, ECG signals have low frequencies, up to just 150Hz for monitoring purposes. This means that we can set a cut-off very close to 150Hz without losing frequencies of interest.

Then, given a fixed cut-off frequency, the output ripple can be improved by increasing the PWM frequency. This is because the harmonics will lie further away from the cut-off frequency and thus be better filtered by the LPF. However, increasing PWM frequency comes at a cost of PWM digital resolution. Arduino UNO system clock runs at a fix frequency of 16MHz. For a desired PWM frequency of 31.25kHz, 512 clock counts per cycle of PWM will be available to specify the duty cycle. This corresponds to a PWM resolution of 9 bits. If a higher PWM frequency is desired, for instance 62 kHz, the clock counts per cycle lower to 258, i.e. 8 bits. In terms of the output DC voltage, this sets the difference between a resolution of 0.39% the maximum DC level (8 bits) or 0.20% (9 bits). ATmega 9-bit fast PWM mode offers the best compromise between cut-off and PWM frequency. In this mode, ATmega328 is programmed to increase the default PWM frequency to 31.25 kHz while using 9 bits of PWM resolution (see Appendix E for details). The system has a resolution of 0.01 V and a ripple of 0.04 V (according to LTSpice simulations), what sums up to a total uncertainty of 0.05 V. This is **resolution 3.8x than the lowest amplitude component in an ECG signal** scaled to 0-5 Vpp (taking signal in Fig. 4.3 as reference, where S dip = 0.19 V).

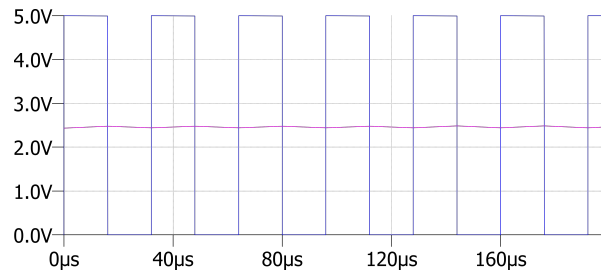


Figure F.1: LTSpice simulation of D/A conversion using PWM and LPF. PWM is a square signal of 5V amplitude, 31.25kHz frequency and in this example 50% duty cycle (in blue). A low pass filter with 159Hz cut-off frequency smooths out the high frequencies, outputting a DC voltage of $amplitude = 5V \cdot duty_cycle$ (in pink).

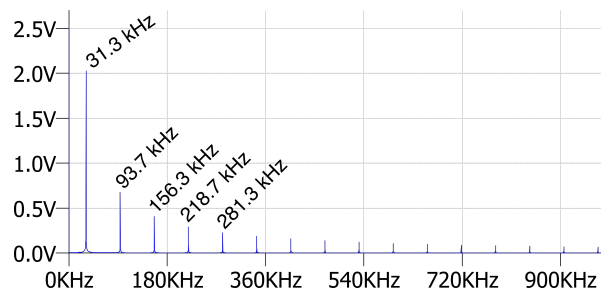


Figure F.2: Fast Fourier Transform of PWM signal of 31.25kHz carrier frequency, 5V max amplitude and 50% duty cycle. Notice DC component of amplitude 2.5V and frequency components at odd-integer harmonics of the PWM frequency. Data from LTSpice simulation.

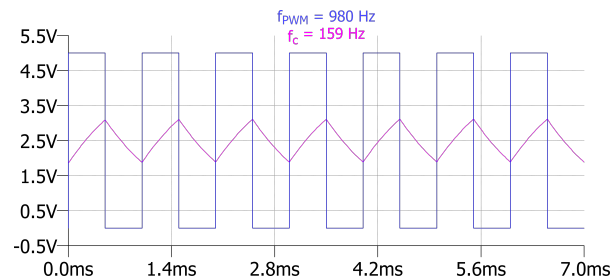


Figure F.3: LTSpice simulation of high ripple PWM + LPF combination. A PWM signal of 980 Hz carrier frequency, 50% duty cycle (in blue) is filtered by a low-pass with cut-off at 159 Hz (in pink). Notice the big ripple in the filtered signal, which reduces the resolution of the DC voltage that has an average value of 2.5 V.

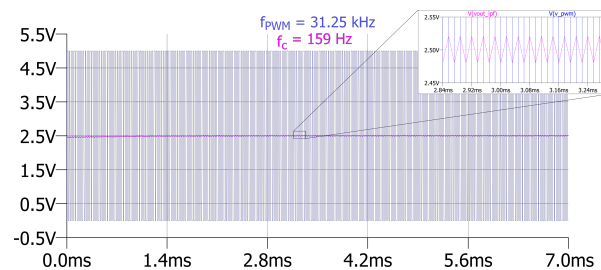


Figure F.4: LTSpice simulation of low ripple PWM + LPF combination. A PWM signal of 31.25 kHz carrier frequency, 50% duty cycle (in blue) is filtered by a low-pass with cut-off at 159 Hz (in pink). Notice the filtered signal has an average DC value of 2.5 V and improved ripple of 40 mV.

G

Low-pass filter design for D/A conversion and discussion

A first-order low-pass filter is designed to smooth the PWM square signal coming from Arduino UNO. The goal is to clean the high-frequency components present in the square wave and only let the DC components pass, while preserving the ECG frequencies of interest. Given that the square signal was set to 31.25 kHz and the ECG frequencies of interest for monitoring purposes are up to 150 Hz, a cut-off around 160 Hz is chosen. The transfer function of a 1st order low pass filter constructed from a resistor and a capacitor is presented in Eq. G.1. An exact frequency of 159 Hz was achieved by the components available.

$$\frac{V_{out_LPF}(s)}{V_{in}(s)} = \frac{1}{1 + RC \cdot s}, \text{ where } s = j\omega \quad (G.1)$$

And $\omega = 2\pi f$ is the angular frequency. The cut-off frequency is defined as the frequency at which the amplitude of the output signal drops by 3 dB. In an RC filter this is given by:

$$f_c = \frac{1}{2\pi RC} \quad (G.2)$$

Then, making Eq. G.2 equal to 160 Hz and considering the resistors and capacitors available, we obtain:

$$R \cdot C = \frac{1}{2\pi \cdot f_c} \rightarrow C = 2\mu F, R = 500\Omega \rightarrow f_c = 159Hz \approx 160Hz \quad (G.3)$$

To build the circuit, two 1 μF capacitors in series and two 250 Ω resistors were used. The filter simulation is presented in Fig. G.1. At frequencies above the cut-off frequency (159 Hz), the amplitude decays as -20 dB/dec. Regarding the phase, a shift of 45° before and after the cut-off frequency is visible Fig. G.1. These are characteristics of an RC low-pass filter.

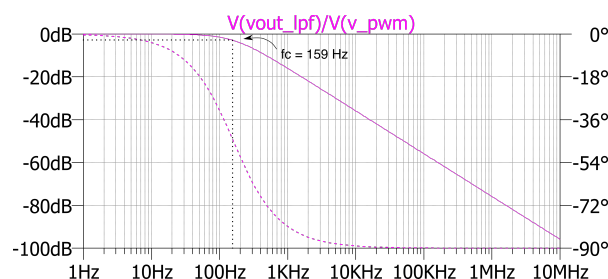


Figure G.1: Frequency response of first-order low-pass filter with 159 Hz cut off ($R=500\Omega$, $C=2\mu F$).

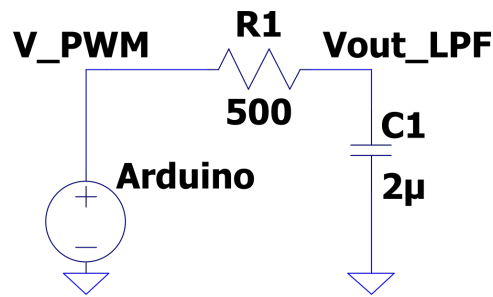


Figure G.2: First-order low-pass filter with 159 Hz cut-off frequency.

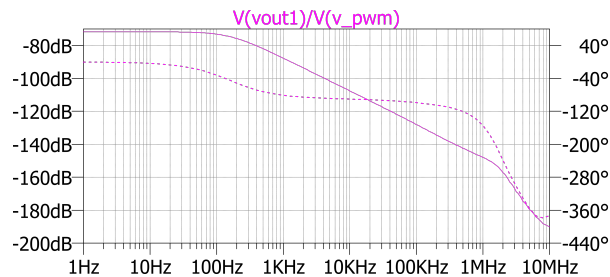
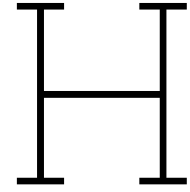


Figure G.3: Frequency response of the system. The effects of 1st order low-pass filter with cut-off frequency of 159 Hz and weighted differential amplifier with -72 dB of gain and 0.9 MHz closed-loop bandwidth are visible.

Different methods can be implemented for the filtering of a PWM signal for digital-to-analog conversion. A first order low-pass filter is the simplest method, provided that only two types of components are required (R, C) and these are affordable and easily available. Also, its design complexity is low and based on the broadly known Eq. G.1. Drawbacks of a first order filter are the slow decay of amplitude with frequency, of only -20 dB/dec. Higher order filters show a steeper curve after their cut-off frequency, increasing -20 dB/dec per filter order, what leads to less ripple. However, being able to increase the PWM frequency up to 31.25 kHz, what moves the square signal harmonics far away from the cut off frequency, the simplest first order filter proved sufficient for the current application. Figures F.3 and F.4 show the decrease in ripple when increasing the PWM frequency from the default value of 980 Hz to 31.25 kHz. Partially, this is due to the filtering performed by ECG monitors, which will usually use low pass filters (with cut-offs around 150 Hz for monitoring mode) and other filtering techniques.

Higher order filters are more complex and costly costs. With the use of a bigger number of components, thermal drift and component value variation become more problematic. Cost and board space consumption also increase. An alternative to passive filters are active filters, which make use of in-built opamps. Usually, the drawback of using active filters is the requirement of implementing opamps with a bandwidth 5x to 10x higher than the highest expected frequency. In the current application, with a PWM frequency of 31.25 kHz, suitable opamps would still be cheap to implement. However, introducing more opamps leads to higher power requirements, which are desired to be low for battery-operated applications. Moreover, in LMICs, the availability of simple passive components such as resistors and capacitors is higher than that of opamps, and their cost is lower. In terms of repairs, passive components are also easier to test and replace. Also their functioning and purpose inside a circuit are more intuitive, which is considered of importance in the current application.

Finally, a different alternative to the use of PWM and a filter for digital-to-analog conversion is the use of DACs, or digital-to-analog converters. These are chips that can have a cost of a couple of euros and exist in various resolutions (8-, 9-, 10-, 12-bits...). For the current application, DAC chips have the same disadvantages as the use of opamps. It is not common for Arduinos to have built-in DACs, although some new models like the recently released Arduino UNO R4 includes a 12-bit one. The Arduino UNO Rev3 used here does not include a DAC. Although DACs provide higher accuracy, especially at high frequencies, for the characteristics of this work, PWM + LPF is considered a more suitable solution.



Attenuation and negative offset calculations

In this appendix, calculations for the attenuation and voltage offset stage (Fig. 5.2) are explained. For this, a weighted difference amplifier is used. Its mathematical analysis is done by superposition. First, one voltage source is replaced by a short circuit and the voltage that the remaining source generates at the output is calculated. We repeat this with the opposite voltage source and finally combine both results to obtain the output voltage of the whole circuit. The calculations are provided next.

1. Weighted difference amplifier

When $V_{in}=0$, opamp 2 acts as an inverting amplifier, resulting in an output of:

$$V_{out,1} = -\frac{R2}{R1}V_{ref} \quad (H.1)$$

When $V_{ref}=0$, opamp 2 acts as a non-inverting amplifier. Thus,

$$V_{out,2} = \left(1 + \frac{R2}{R1}\right)V_+ \quad (H.2)$$

Given the voltage divider at the positive input, V_+ can be described as:

$$V_+ = \frac{R4}{R3 + R4}V_{out_LPF} \quad (H.3)$$

With the total output voltage given by $V_{out} = V_{out,1} + V_{out,2}$ and combining H.2 and H.3, we obtain:

$$V_{out} = -\frac{R2}{R1}V_{ref} + \left(1 + \frac{R2}{R1}\right)\frac{R4}{R3 + R4}V_{out_LPF} \quad (H.4)$$

2. Offset voltage and R2, R1 determination

In the above expression, the negative term corresponds to the negative offset of the output signal, while the positive term will be mostly defined by the gain we want to achieve. Starting with the negative offset, we want this to be equal to -0.1 mV. However, we must also correct for the input offset voltage of the OP97, which has a max value of 0.075 mV [3]. Therefore:

$$V_{off} = -0.1mV - 0.075mV = -\frac{R2}{R1}V_{ref} \quad (H.5)$$

The reference voltage used is the 3.3V output voltage available at Arduino UNO. Then, solving for R2 and R1 and selecting standard resistor values available:

$$\frac{R2}{R1} = \frac{0.175 \cdot 10^{-3}}{3.3} \approx \frac{172\Omega}{3.3MEG\Omega} \quad (H.6)$$

Notice that 172Ω are specified for R2 instead of 175Ω , given that this value is more easily obtained by combining two standard resistors of 150Ω and 22Ω .

3. Gain and R3, R4 determination

The PWM outputted by Arduino and smoothed by the low pass filter varies between 0 and 5V. The desired output signal should range between -0.1 and 1.2 mV. Thus, a gain of $\frac{1.3 \cdot 10^{-3}}{5}$ is required. The term in equation H.4 that multiplies the LPF output voltage should then be equal to the required gain:

$$\left(1 + \frac{R2}{R1}\right) \frac{R4}{R3 + R4} = \frac{1.3 \cdot 10^{-3}}{5} \quad (\text{H.7})$$

Replacing by the R1 and R2 values from H.6, the previous expression becomes:

$$\left(1 + \frac{172}{3.3 \cdot 10^6}\right) \frac{R4}{R3 + R4} = \frac{1.3 \cdot 10^{-3}}{5} \quad (\text{H.8})$$

The left side of equation H.8 can be simplified given that $\left(1 + \frac{172}{3.3 \cdot 10^6}\right) = (1 + 0.000052) \approx 1$. Rearranging the remaining terms and assuming $5 - 1.3 \cdot 10^{-3} \approx 5$, R4 and R3 are defined as:

$$\frac{R4}{R3} = \frac{1.3 \cdot 10^{-3}}{5 - 1.3 \cdot 10^{-3}} \approx \frac{130\Omega}{500k\Omega} \quad (\text{H.9})$$

4. Resistor values used

To define R1 to R4, combinations of standard value resistors in series were used. All resistors were metal film resistors with 1% tolerance and 1/4 W power rating. The used resistors are listed below.

R1: 3x 1MEG Ω + 300k Ω

R2: 150 Ω + 22 Ω

R3: 2x 100k Ω + 300k Ω

R4: 3x 10 Ω + 100 Ω

I

Simulator outputs: V+, GND and V-

The most common representation of an ECG trace corresponds to lead II, which measures the potential difference between an electrode placed on the patient's foot (labeled F, LL or colored green) and the electrode on their right arm (R, RA or red), following 1.1. Lead II is one of the three signals that form the three-leads ECG, the simplest ECG modality. 3-leads ECG is based on 3 electrodes from which Lead I, II and III are obtained as potential differences. Electrode positioning and an example of what the recorded signals look like are shown in Fig. 1.1. The design of the Prototype is focused on simulating lead II correctly. However, at least three electrode cables need to be connected for the patient monitors to start detecting an ECG trace. Otherwise, the lack of connection is detected as a cable error. For this reason, the Prototype must provide three outputs to which ECG electrode cables must be connected.

The Prototype's electrical circuit was designed with a focus on using common, broadly-available components and keeping the complexity low. Most part of the circuit is dedicated to generating a signal with the characteristics of the reference one. Once this is achieved, a simple solution to obtain a second signal is the use of an inverting amplifier with gain 1 that just inverts the generated signal. A third potential already available in the circuit is ground. These three signals become the outputs of the Prototype in the order shown in 1.2.

By connecting the ground signal to the foot electrode cable and the negative signal to the right-arm electrode cable, the positive waveform will be displayed at Lead II in the patient monitor. Using this configuration, Lead III will display the inverted signal and Lead I will display a signal 2x in amplitude. This was tested in GE Healthcare monitor, see Fig. 1.3. Equivalently, the positive signal can be connected to the foot cable and the ground signal to the right-arm cable.

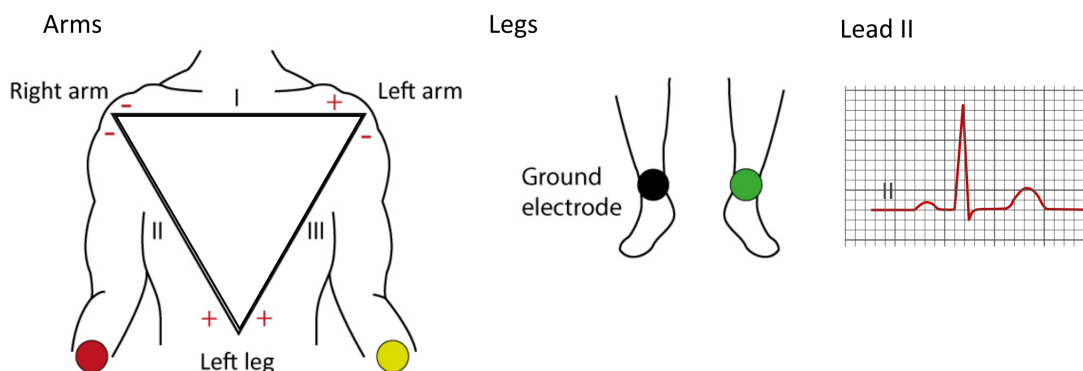


Figure 1.1: Three-leads ECG: electrodes positioning and example recordings of Lead I, II and III.

The mentioned configurations can test the integrity of electrode cables, since in case a cable is not working, the patient monitor will display a connection error. This was the case in one of the patient monitors tested, in which the Prototype effectively worked to detect a faulty electrode cable.

If the user wants to check the correct display of a positive signal on each lead channel, this can be achieved by changing the connection of the cables and taking into account the polarity of the Prototype's outputs and how each lead is calculated.

$$\text{Lead II} = F - R \text{ (IEC standard) or } \text{Lead II} = LL - RA \text{ (AHA standard)} \quad (1.1)$$

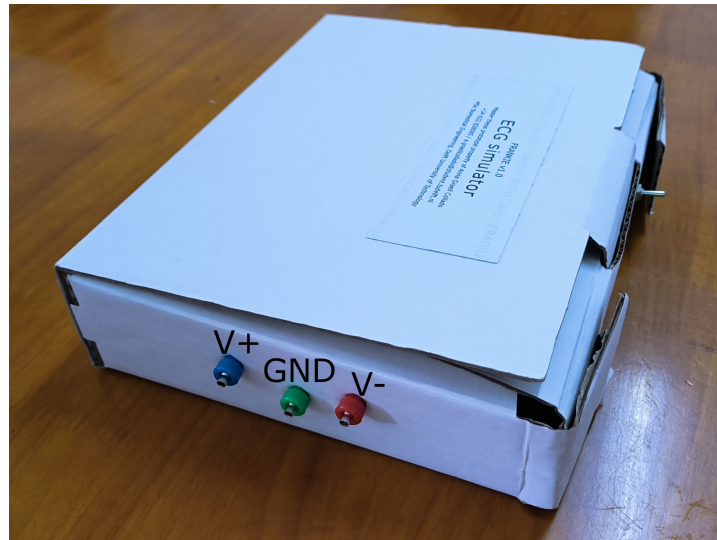


Figure I.2: Prototype connectors provide three different signals, from left to right: positive ECG waveform (V+), ground (GND) and negative ECG waveform (V-).

Lead I



Lead II



Lead III

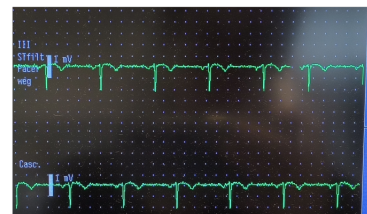
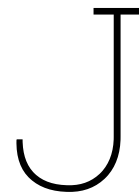


Figure I.3: Lead I, II and III detected by GE Healthcare patient monitor when the Prototype's outputs are connected to electrode cables as: GND to F; V- to R and V+ to L.



Opamps characteristics and rationale

OP97 (Mfr. No. OP97FPZ) was chosen due to:

- Input offset voltage <1 mV: 30 μV (typ.), 75 μV (max.) at 25 °C
- Supply voltage range suitable for battery-operated: $\pm 2.25\text{ V}$ to $\pm 20\text{ V}$
- Low supply current: 380 μA (typ.), 600 μA (max.)
- Output voltage swing to within 1 V of the rails

The closed-loop bandwidth of 0.9 MHz exceeds the demands of the application. The same applies to the slew rate, of 0.15 V/ μs (typ., $-40\text{ }^\circ\text{C} < T_A < +85\text{ }^\circ\text{C}$). The most demanding case happens at opamp 1, with an input signal of 1.3 mV and frequencies up to 159 Hz. The slew rate needed to handle that would be:

$$SR = 2 \cdot \pi \cdot 159 \cdot 1.3 \cdot 10^{-6} = 1.29\text{ V/s or } 0.00000129\text{ V}/\mu\text{s}$$

Which is widely covered by the slew rate of OP97.

The extended working temperature range ($-40\text{ }^\circ\text{C}$ to $+85\text{ }^\circ\text{C}$) and a temperature drift below 0.6 $\mu\text{V}/^\circ\text{C}$ are useful for its implementation in high temperature conditions such as in Nepal.

Although OP97 was chosen due to its low input voltage offset, below 75 μV at 25 °C, this is stated to increase up to 200 μV for extreme temperature conditions $-40\text{ }^\circ\text{C} < T_A < +85\text{ }^\circ\text{C}$, what could explain the higher amplitude determination error of the feedback system being higher than expected.

Another drawback of OP97 is the long-term offset voltage stability of 0.3 $\mu\text{V}/\text{month}$ indicated in the datasheet. Only when this reaches the same level as the smallest generated voltage feature (0.1 mV) it will become relevant. This will happen after $\frac{0.1\text{mV}}{0.003\text{mV}/\text{month}} = \frac{33\text{months}}{12\text{months}/\text{year}} = 2,75\text{ years}$. Given that the same opamp models are used for signal generation and the feedback, the same offset will be present in average in both stages, making it impossible for the current system to detect such a drift. To improve that, the opamps used at the feedback stage should have a lower long-term offset voltage stability than those used in the amplifying stage.

LM358-N (Mfr. No. LM358N/NOPB) was chosen due to:

- Supply voltage range: 3 - 26 V (single) or $\pm 1.5 - \pm 13\text{ V}$ (dual)
- Supply current: 0.5 mA (typ.); 1.2 mA (max.) at $V^+ = 5\text{ V}$

Its operating temperature is 0 - 70 °C. In this case the slew rate demands are higher given the amplitude of the input signal:

$$SR = 2 \cdot \pi \cdot 159 \cdot 5 = 5000\text{ V/s or } 0.005\text{ V}/\mu\text{s}$$

However, this is still exceeded 20x by the slew rate of 0.1 V/ μs of LM358N. The bandwidth of 1MHz also widely covers the 159Hz of the ECG signal. This opamp has an input offset voltage of 2 mV but that is not a problem given that it is used at the beginning of the circuit, when the signal has a range 0 - 5 V.

Both opamps were acquired with a through-hole configuration, allowing use in a breadboard. For decoupling the supply pins, 100-nF capacitors were placed as close as possible to the op-amp power supply pins. For each opamp, a capacitor was placed between V+ and ground, and another one between V- and ground.

K

Table of costs

Table K.1: Components cost in Nepal

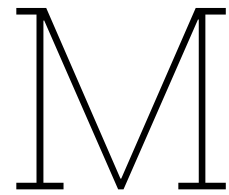
	Units	Price/u.	Total	Currency	Link
<i>Ardino UNO</i>	1	1460	1460	NPR	link
<i>Rechargeable batteries</i>	4	282	1128	NPR	link
<i>2S Protection board</i>	2	225	450	NPR	link
<i>Battery holder</i>	2	127	254	NPR	link
<i>Resistors box</i>	1	125	125	NPR	link
<i>Capacitors 0.1uF</i>	2	36	72	NPR	link
<i>OP97 opamp DIP-8</i>	4	74	296	NPR	link
<i>LM358 opamp DIP-8</i>	2	34	68	NPR	link
Banana jack female	3	93	279	NPR	link
<i>Breadboard small</i>	5	110	550	NPR	link
<i>Jumper wire</i>	1	138	138	NPR	link
<i>DC jack male 5.5mm</i>	1	34	34	NPR	link
<i>Carton box ~20x25x3cm</i>	1	24	24	NPR	link
<i>DPDT Switch</i>	2	30	60	NPR	link
<i>Trimmer resistor 1kΩ</i>	1	12	12	NPR	link
<i>Rotary switch</i>	1	39	39	NPR	link
TOTAL			4950	NPR	

L

Simulator output on common oscilloscopes



Figure L.1: ECG-like waveforms in the millivolt range detected by three common oscilloscopes. a) 2 mVpp ECG waveform generated by 214A commercial simulator as detected by Rigol DS2072A Oscilloscope (2.5 mV/div). b) 1 mVpp ECG waveform generated by the Prototype (PWM + LPF + voltage divider) as detected by Tablet Oscilloscope tBook mini TO1000 (10 mV/div). c) 5 mVpp ECG waveform generated by ProSim 8 as detected by Fluke 125B Industrial Scopemeter (5 mV/div).



Oscilloscope measurements: first stage timing checks

Here, oscilloscope data from first stage timing checks is presented. First, a table containing the R-R interval error from the oscilloscope data. Second, plots comparing the data recorded by the oscilloscope to the reference data. Relative error of HR determination and R-R interval are also included in the plots.

Table M.1: R-peaks timing error: comparison between original data (Original_time) and data measured with oscilloscope (Measured_time). Error calculations of oscilloscope data include absolute and relative time differences between consecutive R peaks, as well as absolute and relative HR determination error.

	Target _HR	Original _time	Measured _time	dt_abs _diff	HR_abs _diff	HR_ round _diff	dt_diff _percent	HR_diff _percent	HR_round _diff_ percent
0	35	0.2278	0.2498						
1	35	1.9417	1.9898	0.0261	-0.5253	-1.0	1.5235	-1.5006	-2.8571
2	35	3.6556	3.7308	0.0271	-0.5452	-1.0	1.5818	-1.5572	-2.8571
3	35	5.3694	5.4708	0.0261	-0.5253	-1.0	1.5235	-1.5006	-2.8571
4	35	7.0833	7.211	0.0263	-0.5293	-1.0	1.5352	-1.512	-2.8571
0	40	0.2278	0.2568						
1	40	1.7278	1.7806	0.0238	-0.6248	-1.0	1.5867	-1.5619	-2.5
2	40	3.2278	3.3062	0.0256	-0.6712	-1.0	1.7067	-1.678	-2.5
3	40	4.7278	4.8296	0.0234	-0.6144	-1.0	1.56	-1.536	-2.5
4	40	6.2278	6.3546	0.025	-0.6557	-1.0	1.6667	-1.6393	-2.5
0	45	0.2278	0.2592						
1	45	1.5611	1.6128	0.0203	-0.6738	-1.0	1.52	-1.4972	-2.2222
2	45	2.8944	2.9668	0.0207	-0.6869	-1.0	1.55	-1.5263	-2.2222
3	45	4.2278	4.3208	0.0207	-0.6869	-1.0	1.55	-1.5263	-2.2222
4	45	5.5611	5.6738	0.0197	-0.6541	-1.0	1.475	-1.4536	-2.2222
5	45	6.8944	7.0278	0.0207	-0.6869	-1.0	1.55	-1.5263	-2.2222

0	50	0.2278	0.2138						
1	50	1.4278	1.4328	0.019	-0.7793	-1.0	1.5833	-1.5587	-2.0
2	50	2.6278	2.6544	0.0216	-0.8841	-1.0	1.8	-1.7682	-2.0
3	50	3.8278	3.8728	0.0184	-0.7551	-1.0	1.5333	-1.5102	-2.0
4	50	5.0278	5.0926	0.0198	-0.8116	-1.0	1.65	-1.6232	-2.0
5	50	6.2278	6.3108	0.0182	-0.747	-1.0	1.5167	-1.494	-2.0
6	50	7.4278	7.5334	0.0226	-0.9243	-1.0	1.8833	-1.8485	-2.0
0	55	0.2278	0.2512						
1	55	1.3194	1.3598	0.0169	-0.8395	-1.0	1.5511	-1.5275	-1.8182
2	55	2.4111	2.467	0.0155	-0.7711	-1.0	1.4229	-1.4029	-1.8182
3	55	3.5028	3.5748	0.0161	-0.8004	-1.0	1.4779	-1.4563	-1.8182
4	55	4.5944	4.6848	0.0183	-0.9078	-1.0	1.6794	-1.6517	-1.8182
5	55	5.6861	5.7906	0.0141	-0.7025	-1.0	1.2947	-1.2781	-1.8182
6	55	6.7778	6.8984	0.0161	-0.8004	-1.0	1.4779	-1.4563	-1.8182
0	60	0.2278	0.26						
1	60	1.2278	1.276	0.016	-0.9449	-1.0	1.6	-1.5748	-1.6667
2	60	2.2278	2.2928	0.0168	-0.9913	-1.0	1.68	-1.6522	-1.6667
3	60	3.2278	3.3078	0.015	-0.8867	-1.0	1.5	-1.4778	-1.6667
4	60	4.2278	4.3248	0.017	-1.0029	-1.0	1.7	-1.6716	-1.6667
5	60	5.2278	5.3414	0.0166	-0.9797	-1.0	1.66	-1.6329	-1.6667
6	60	6.2278	6.3572	0.0158	-0.9333	-1.0	1.58	-1.5554	-1.6667
7	60	7.2278	7.3738	0.0166	-0.9797	-1.0	1.66	-1.6329	-1.6667
0	65	0.2278	0.265						
1	65	1.15	1.2014	0.0142	-0.9851	-1.0	1.5373	-1.5141	-1.5385
2	65	2.0722	2.1382	0.0146	-1.0124	-1.0	1.5807	-1.5561	-1.5385
3	65	2.9944	3.076	0.0156	-1.0807	-1.0	1.6892	-1.6611	-1.5385
4	65	3.9167	4.0116	0.0134	-0.9303	-1.0	1.4506	-1.4299	-1.5385
5	65	4.8389	4.9512	0.0174	-1.2033	-1.0	1.8843	-1.8495	-1.5385
6	65	5.7611	5.8868	0.0134	-0.9303	-1.0	1.4506	-1.4299	-1.5385
7	65	6.6833	6.8242	0.0152	-1.0534	-1.0	1.6458	-1.6191	-1.5385
8	65	7.6056	7.7606	0.0142	-0.9851	-1.0	1.5373	-1.5141	-1.5385
0	70	0.2278	0.2794						
1	70	1.0861	1.152	0.0143	-1.1429	-1.0	1.6621	-1.635	-1.4286
2	70	1.9444	2.0238	0.0135	-1.0798	-1.0	1.5689	-1.5447	-1.4286
3	70	2.8028	2.896	0.0139	-1.1114	-1.0	1.6155	-1.5898	-1.4286
4	70	3.6611	3.7666	0.0123	-0.9849	-1.0	1.4291	-1.409	-1.4286

5	70	4.5194	4.6388	0.0139	-1.1114	-1.0	1.6155	-1.5898	-1.4286
6	70	5.3778	5.5124	0.0153	-1.2216	-1.0	1.7786	-1.7476	-1.4286
7	70	6.2361	6.3838	0.0131	-1.0482	-1.0	1.5223	-1.4995	-1.4286
8	70	7.0944	7.2564	0.0143	-1.1429	-1.0	1.6621	-1.635	-1.4286
9	70	7.9528	8.1282	0.0135	-1.0798	-1.0	1.5689	-1.5447	-1.4286
0	75	0.2278	0.2108						
1	75	1.0278	1.0238	0.013	-1.1993	-1.0	1.625	-1.599	-1.3333
2	75	1.8278	1.8356	0.0118	-1.0902	-1.0	1.475	-1.4536	-1.3333
3	75	2.6278	2.6492	0.0136	-1.2537	-1.0	1.7	-1.6716	-1.3333
4	75	3.4278	3.4622	0.013	-1.1993	-1.0	1.625	-1.599	-1.3333
5	75	4.2278	4.2756	0.0134	-1.2356	-1.0	1.675	-1.6474	-1.3333
6	75	5.0278	5.0902	0.0146	-1.3442	-1.0	1.825	-1.7923	-1.3333
7	75	5.8278	5.9006	0.0104	-0.9625	-1.0	1.3	-1.2833	-1.3333
8	75	6.6278	6.7148	0.0142	-1.308	-1.0	1.775	-1.744	-1.3333
9	75	7.4278	7.5298	0.015	-1.3804	-1.0	1.875	-1.8405	-1.3333
10	75	8.2278	8.3402	0.0104	-0.9625	-1.0	1.3	-1.2833	-1.3333
0	80	0.2278	0.2514						
1	80	0.9778	1.0128	0.0114	-1.1978	-1.0	1.52	-1.4972	-1.25
2	80	1.7278	1.7756	0.0128	-1.3424	-1.0	1.7067	-1.678	-1.25
3	80	2.4778	2.5372	0.0116	-1.2185	-1.0	1.5467	-1.5231	-1.25
4	80	3.2278	3.2998	0.0126	-1.3218	-1.0	1.68	-1.6522	-1.25
5	80	3.9778	4.0616	0.0118	-1.2392	-1.0	1.5733	-1.549	-1.25
6	80	4.7278	4.8236	0.012	-1.2598	-1.0	1.6	-1.5748	-1.25
7	80	5.4778	5.5854	0.0118	-1.2392	-1.0	1.5733	-1.549	-1.25
8	80	6.2278	6.3478	0.0124	-1.3012	-1.0	1.6533	-1.6264	-1.25
9	80	6.9778	7.1116	0.0138	-1.4454	-1.0	1.84	-1.8068	-1.25
10	80	7.7278	7.8722	0.0106	-1.1149	-1.0	1.4133	-1.3936	-1.25
0	85	0.2278	0.2624						
1	85	0.9333	0.9772	0.0092	-1.0998	-1.0	1.3102	-1.2933	-1.1765
2	85	1.6389	1.6924	0.0096	-1.1468	-1.0	1.3669	-1.3485	-1.1765
3	85	2.3444	2.4088	0.0108	-1.2873	-1.0	1.537	-1.5137	-1.1765
4	85	3.05	3.124	0.0096	-1.1468	-1.0	1.3669	-1.3485	-1.1765
5	85	3.7556	3.8402	0.0106	-1.2639	-1.0	1.5087	-1.4862	-1.1765
6	85	4.4611	4.5552	0.0094	-1.1233	-1.0	1.3386	-1.3209	-1.1765
7	85	5.1667	5.2714	0.0106	-1.2639	-1.0	1.5087	-1.4862	-1.1765
8	85	5.8722	5.9866	0.0096	-1.1468	-1.0	1.3669	-1.3485	-1.1765

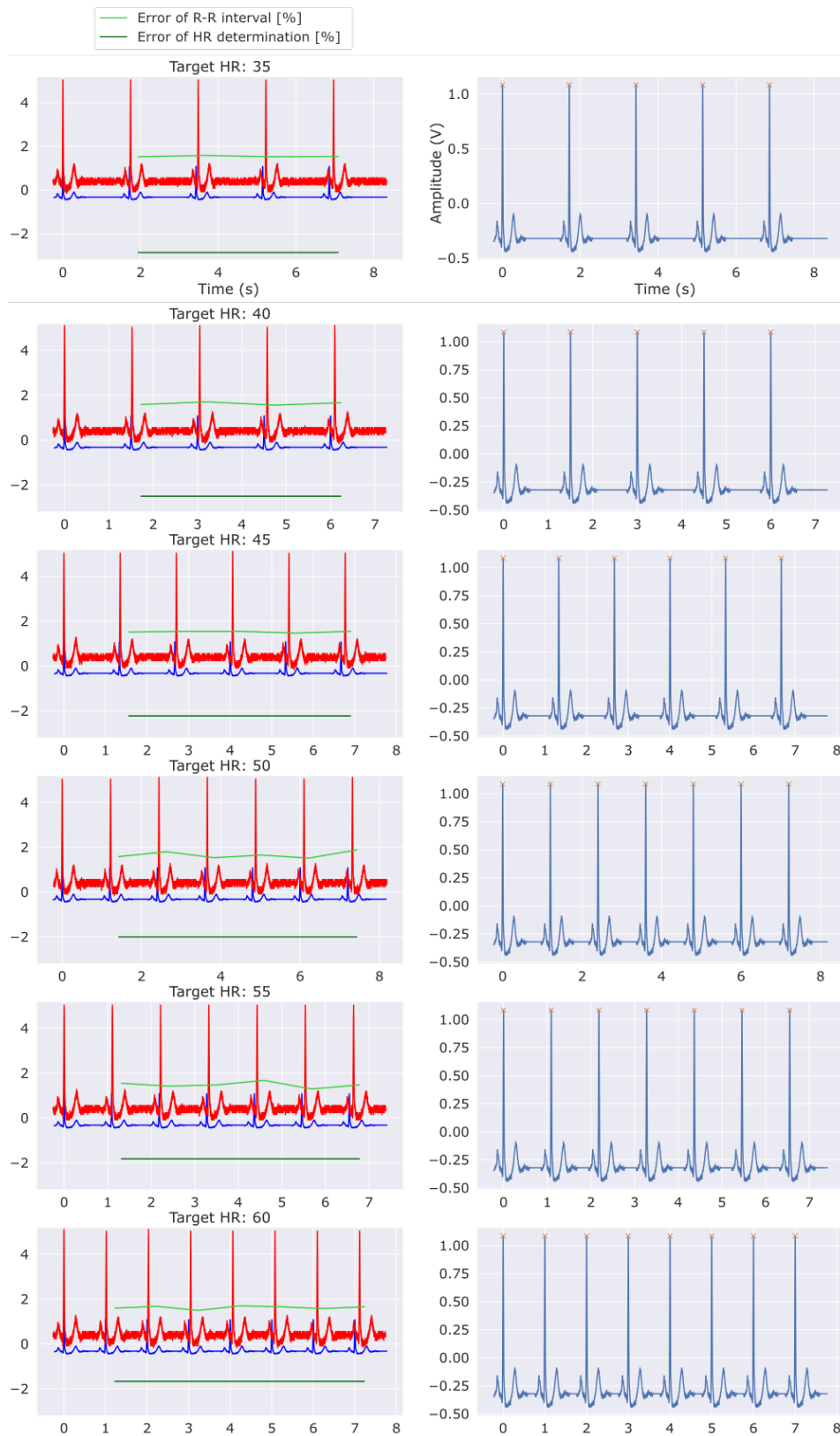
9	85	6.5778	6.7018	0.0096	-1.1468	-1.0	1.3669	-1.3485	-1.1765
10	85	7.2833	7.4178	0.0104	-1.2405	-1.0	1.4803	-1.4587	-1.1765
11	85	7.9889	8.134	0.0106	-1.2639	-1.0	1.5087	-1.4862	-1.1765
0	90	0.2278	0.2458						
1	90	0.8944	0.9216	0.0091	-1.2163	-1.0	1.37	-1.3515	-1.1111
2	90	1.5611	1.6	0.0117	-1.5566	-2.0	1.76	-1.7296	-2.2222
3	90	2.2278	2.2764	0.0097	-1.2951	-1.0	1.46	-1.439	-1.1111
4	90	2.8944	2.9522	0.0091	-1.2163	-1.0	1.37	-1.3515	-1.1111
5	90	3.5611	3.6304	0.0115	-1.5305	-2.0	1.73	-1.7006	-2.2222
6	90	4.2278	4.3068	0.0097	-1.2951	-1.0	1.46	-1.439	-1.1111
7	90	4.8944	4.9836	0.0101	-1.3475	-1.0	1.52	-1.4972	-1.1111
8	90	5.5611	5.662	0.0117	-1.5566	-2.0	1.76	-1.7296	-2.2222
9	90	6.2278	6.3374	0.0087	-1.1638	-1.0	1.31	-1.2931	-1.1111
10	90	6.8944	7.014	0.0099	-1.3213	-1.0	1.49	-1.4681	-1.1111
11	90	7.5611	7.691	0.0103	-1.3737	-1.0	1.55	-1.5263	-1.1111
0	95	0.2278	0.2268						
1	95	0.8583	0.8652	0.0078	-1.1692	-1.0	1.2441	-1.2288	-1.0526
2	95	1.4889	1.5058	0.01	-1.492	-1.0	1.593	-1.568	-1.0526
3	95	2.1194	2.1468	0.0104	-1.5504	-1.0	1.6564	-1.6294	-1.0526
4	95	2.75	2.7864	0.009	-1.3456	-1.0	1.4344	-1.4141	-1.0526
5	95	3.3806	3.4286	0.0116	-1.7253	-2.0	1.8467	-1.8132	-2.1053
6	95	4.0111	4.0676	0.0084	-1.2575	-1.0	1.3392	-1.3215	-1.0526
7	95	4.6417	4.7076	0.0094	-1.4042	-1.0	1.4978	-1.4757	-1.0526
8	95	5.2722	5.3484	0.0102	-1.5212	-1.0	1.6247	-1.5987	-1.0526
9	95	5.9028	5.988	0.009	-1.3456	-1.0	1.4344	-1.4141	-1.0526
10	95	6.5333	6.6286	0.01	-1.492	-1.0	1.593	-1.568	-1.0526
11	95	7.1639	7.269	0.0098	-1.4627	-1.0	1.5612	-1.5372	-1.0526
12	95	7.7944	7.9098	0.0102	-1.5212	-1.0	1.6247	-1.5987	-1.0526
0	100	0.2278	0.225						
1	100	0.8278	0.8342	0.0092	-1.5102	-2.0	1.5333	-1.5102	-2.0
2	100	1.4278	1.4458	0.0116	-1.8967	-2.0	1.9333	-1.8967	-2.0
3	100	2.0278	2.0552	0.0094	-1.5425	-2.0	1.5667	-1.5425	-2.0
4	100	2.6278	2.6646	0.0094	-1.5425	-2.0	1.5667	-1.5425	-2.0
5	100	3.2278	3.2738	0.0092	-1.5102	-2.0	1.5333	-1.5102	-2.0
6	100	3.8278	3.886	0.0122	-1.9928	-2.0	2.0333	-1.9928	-2.0
7	100	4.4278	4.4946	0.0086	-1.4131	-1.0	1.4333	-1.4131	-1.0

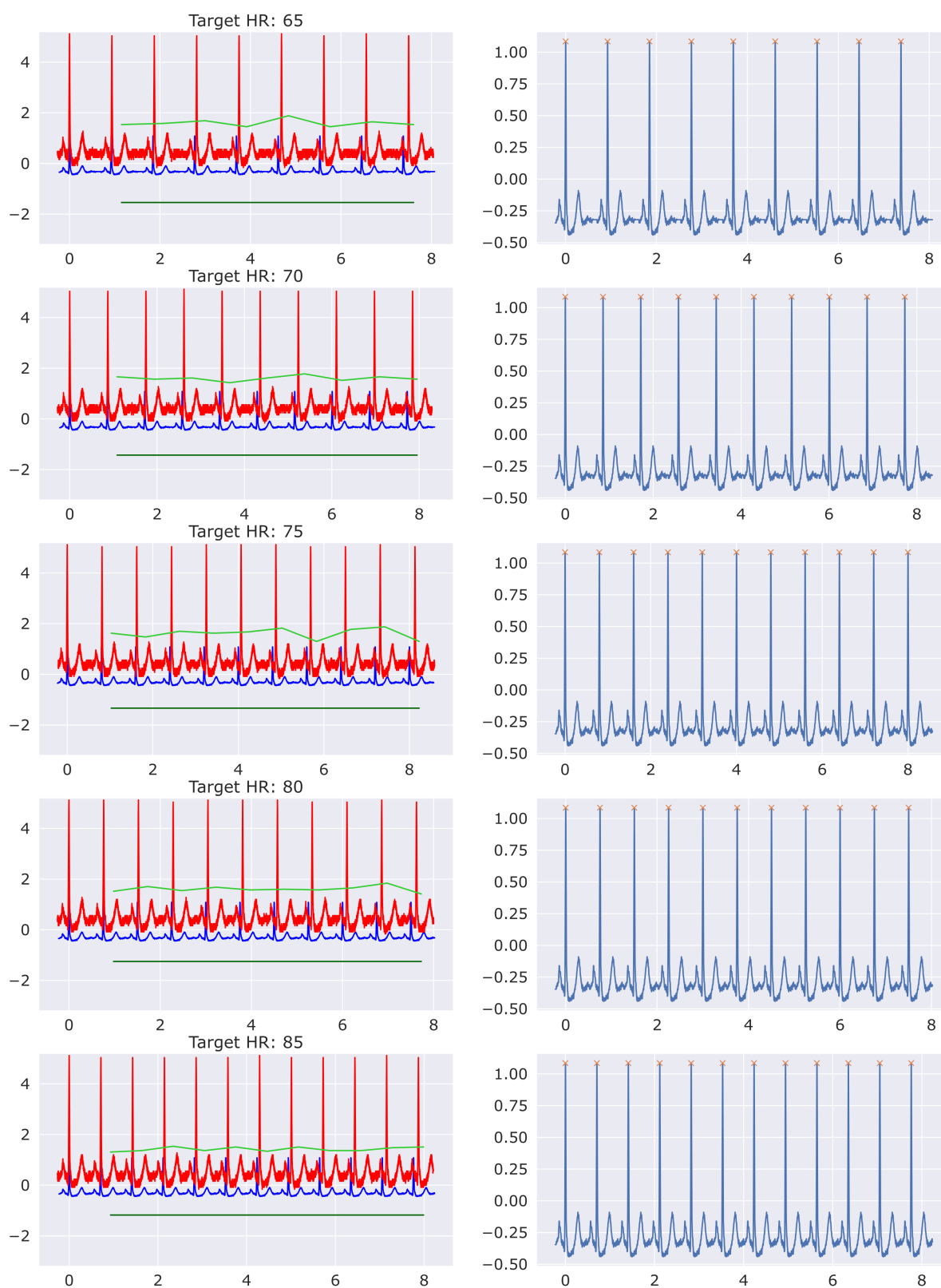
8	100	5.0278	5.1036	0.009	-1.4778	-1.0	1.5	-1.4778	-1.0
9	100	5.6278	5.7138	0.0102	-1.6716	-2.0	1.7	-1.6716	-2.0
10	100	6.2278	6.323	0.0092	-1.5102	-2.0	1.5333	-1.5102	-2.0
11	100	6.8278	6.9334	0.0104	-1.7038	-2.0	1.7333	-1.7038	-2.0
12	100	7.4278	7.5428	0.0094	-1.5425	-2.0	1.5667	-1.5425	-2.0
0	105	0.2278	0.2206						
1	105	0.8	0.8024	0.0096	-1.7261	-2.0	1.6738	-1.6462	-1.9048
2	105	1.3722	1.3834	0.0088	-1.5841	-2.0	1.534	-1.5108	-1.9048
3	105	1.9444	1.9652	0.0096	-1.7261	-2.0	1.6738	-1.6462	-1.9048
4	105	2.5167	2.5458	0.0084	-1.513	-2.0	1.4641	-1.443	-1.9048
5	105	3.0889	3.128	0.01	-1.797	-2.0	1.7437	-1.7138	-1.9048
6	105	3.6611	3.7082	0.008	-1.4418	-2.0	1.3942	-1.375	-1.9048
7	105	4.2333	4.2922	0.0118	-2.1146	-2.0	2.0583	-2.0167	-1.9048
8	105	4.8056	4.8724	0.008	-1.4418	-2.0	1.3942	-1.375	-1.9048
9	105	5.3778	5.455	0.0104	-1.8678	-2.0	1.8136	-1.7813	-1.9048
10	105	5.95	6.0344	0.0072	-1.299	-1.0	1.2544	-1.2388	-0.9524
11	105	6.5222	6.6162	0.0096	-1.7261	-2.0	1.6738	-1.6462	-1.9048
12	105	7.0944	7.1978	0.0094	-1.6907	-2.0	1.6388	-1.6124	-1.9048
13	105	7.6667	7.7794	0.0094	-1.6907	-2.0	1.6388	-1.6124	-1.9048
14	105	8.2389	8.3608	0.0092	-1.6552	-2.0	1.6039	-1.5786	-1.9048
0	110	0.2278	0.2286						
1	110	0.7722	0.7816	0.0086	-1.705	-2.0	1.5714	-1.5471	-1.8182
2	110	1.3167	1.3356	0.0096	-1.9008	-2.0	1.7551	-1.7248	-1.8182
3	110	1.8611	1.8886	0.0086	-1.705	-2.0	1.5714	-1.5471	-1.8182
4	110	2.4056	2.4406	0.0076	-1.5084	-1.0	1.3878	-1.3688	-0.9091
5	110	2.95	2.9934	0.0084	-1.6657	-1.0	1.5347	-1.5115	-0.9091
6	110	3.4944	3.5472	0.0094	-1.8617	-2.0	1.7184	-1.6893	-1.8182
7	110	4.0389	4.098	0.0064	-1.2716	-1.0	1.1673	-1.1539	-0.9091
8	110	4.5833	4.6526	0.0102	-2.018	-2.0	1.8653	-1.8311	-1.8182
9	110	5.1278	5.2064	0.0094	-1.8617	-2.0	1.7184	-1.6893	-1.8182
10	110	5.6722	5.7586	0.0078	-1.5478	-1.0	1.4245	-1.4045	-0.9091
11	110	6.2167	6.3116	0.0086	-1.705	-2.0	1.5714	-1.5471	-1.8182
12	110	6.7611	6.8644	0.0084	-1.6657	-1.0	1.5347	-1.5115	-0.9091
13	110	7.3056	7.416	0.0072	-1.4296	-1.0	1.3143	-1.2972	-0.9091
14	110	7.85	7.9704	0.01	-1.979	-2.0	1.8286	-1.7957	-1.8182
0	115	0.2278	0.2058						

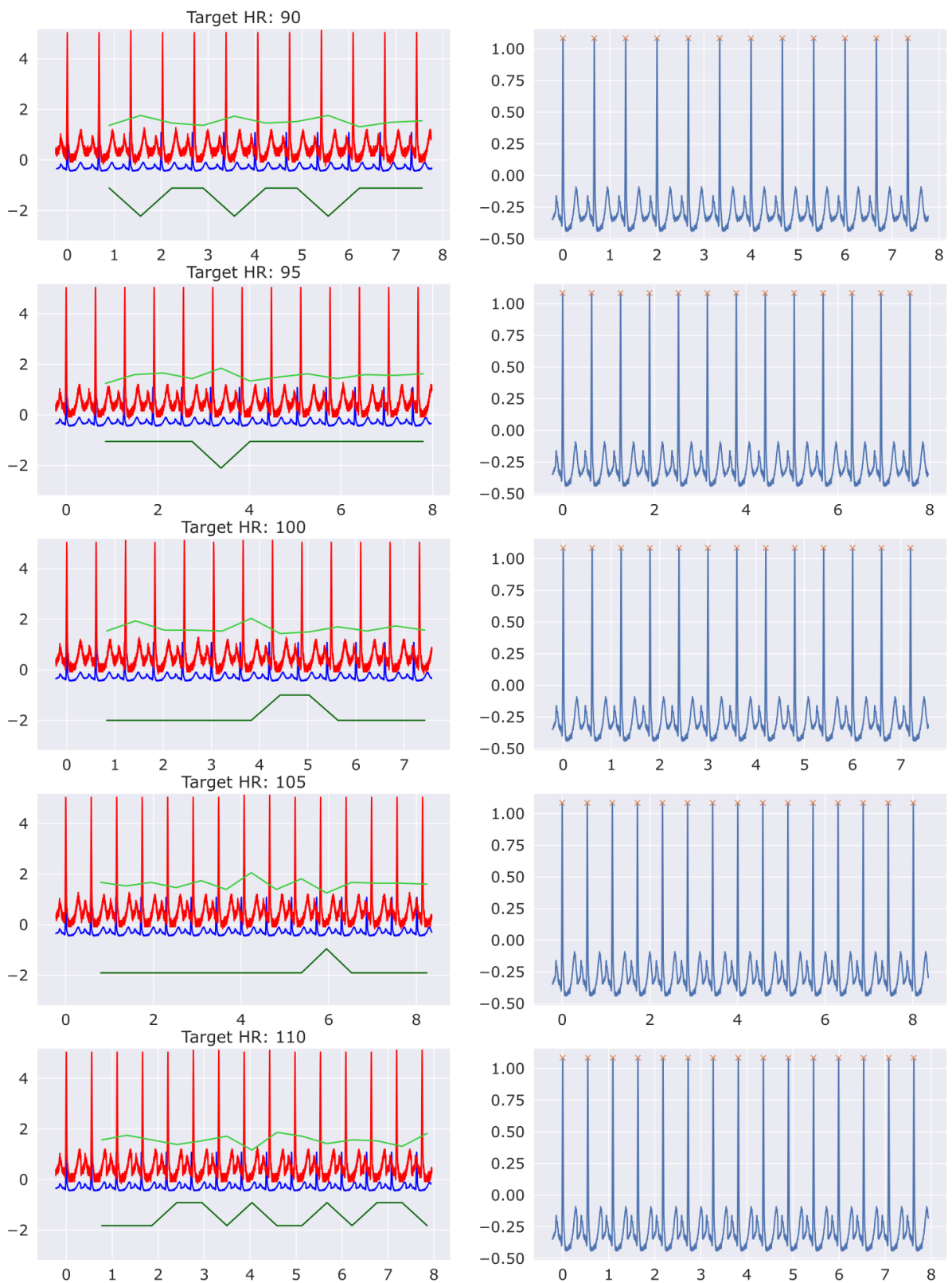
1	115	0.75	0.7362	0.0082	-1.7714	-2.0	1.566	-1.5418	-1.7391
2	115	1.2722	1.2668	0.0084	-1.8141	-2.0	1.6043	-1.5789	-1.7391
3	115	1.7944	1.7984	0.0094	-2.0268	-2.0	1.7957	-1.7641	-1.7391
4	115	2.3167	2.329	0.0084	-1.8141	-2.0	1.6043	-1.5789	-1.7391
5	115	2.8389	2.8586	0.0074	-1.6006	-2.0	1.4128	-1.3931	-1.7391
6	115	3.3611	3.389	0.0082	-1.7714	-2.0	1.566	-1.5418	-1.7391
7	115	3.8833	3.9198	0.0086	-1.8567	-2.0	1.6426	-1.616	-1.7391
8	115	4.4056	4.4504	0.0084	-1.8141	-2.0	1.6043	-1.5789	-1.7391
9	115	4.9278	4.9818	0.0092	-1.9843	-2.0	1.7574	-1.7271	-1.7391
10	115	5.45	5.5128	0.0088	-1.8993	-2.0	1.6809	-1.6531	-1.7391
11	115	5.9722	6.042	0.007	-1.5149	-2.0	1.3362	-1.3186	-1.7391
12	115	6.4944	6.5728	0.0086	-1.8567	-2.0	1.6426	-1.616	-1.7391
13	115	7.0167	7.1028	0.0078	-1.6861	-2.0	1.4894	-1.4675	-1.7391
14	115	7.5389	7.634	0.009	-1.9418	-2.0	1.7191	-1.6901	-1.7391
15	115	8.0611	8.1648	0.0086	-1.8567	-2.0	1.6426	-1.616	-1.7391
0	120	0.2278	0.1928						
1	120	0.7278	0.7	0.0072	-1.7035	-2.0	1.44	-1.4196	-1.6667
2	120	1.2278	1.2088	0.0088	-2.0755	-2.0	1.76	-1.7296	-1.6667
3	120	1.7278	1.717	0.0082	-1.9362	-2.0	1.64	-1.6135	-1.6667
4	120	2.2278	2.2254	0.0084	-1.9827	-2.0	1.68	-1.6522	-1.6667
5	120	2.7278	2.7324	0.007	-1.6568	-2.0	1.4	-1.3807	-1.6667
6	120	3.2278	3.243	0.0106	-2.4912	-2.0	2.12	-2.076	-1.6667
7	120	3.7278	3.7486	0.0056	-1.3291	-1.0	1.12	-1.1076	-0.8333
8	120	4.2278	4.257	0.0084	-1.9827	-2.0	1.68	-1.6522	-1.6667
9	120	4.7278	4.7656	0.0086	-2.0291	-2.0	1.72	-1.6909	-1.6667
10	120	5.2278	5.275	0.0094	-2.2144	-2.0	1.88	-1.8453	-1.6667
11	120	5.7278	5.7822	0.0072	-1.7035	-2.0	1.44	-1.4196	-1.6667
12	120	6.2278	6.2906	0.0084	-1.9827	-2.0	1.68	-1.6522	-1.6667
13	120	6.7278	6.7996	0.009	-2.1218	-2.0	1.8	-1.7682	-1.6667
14	120	7.2278	7.3084	0.0088	-2.0755	-2.0	1.76	-1.7296	-1.6667
15	120	7.7278	7.8162	0.0078	-1.8432	-2.0	1.56	-1.536	-1.6667
16	120	8.2278	8.3254	0.0092	-2.1681	-2.0	1.84	-1.8068	-1.6667
0	125	0.2278	0.2198						
1	125	0.7083	0.7068	0.0064	-1.6522	-2.0	1.341	-1.3233	-1.6
2	125	1.1889	1.1946	0.0072	-1.8543	-2.0	1.5075	-1.4851	-1.6
3	125	1.6694	1.6828	0.0076	-1.955	-2.0	1.5908	-1.5658	-1.6

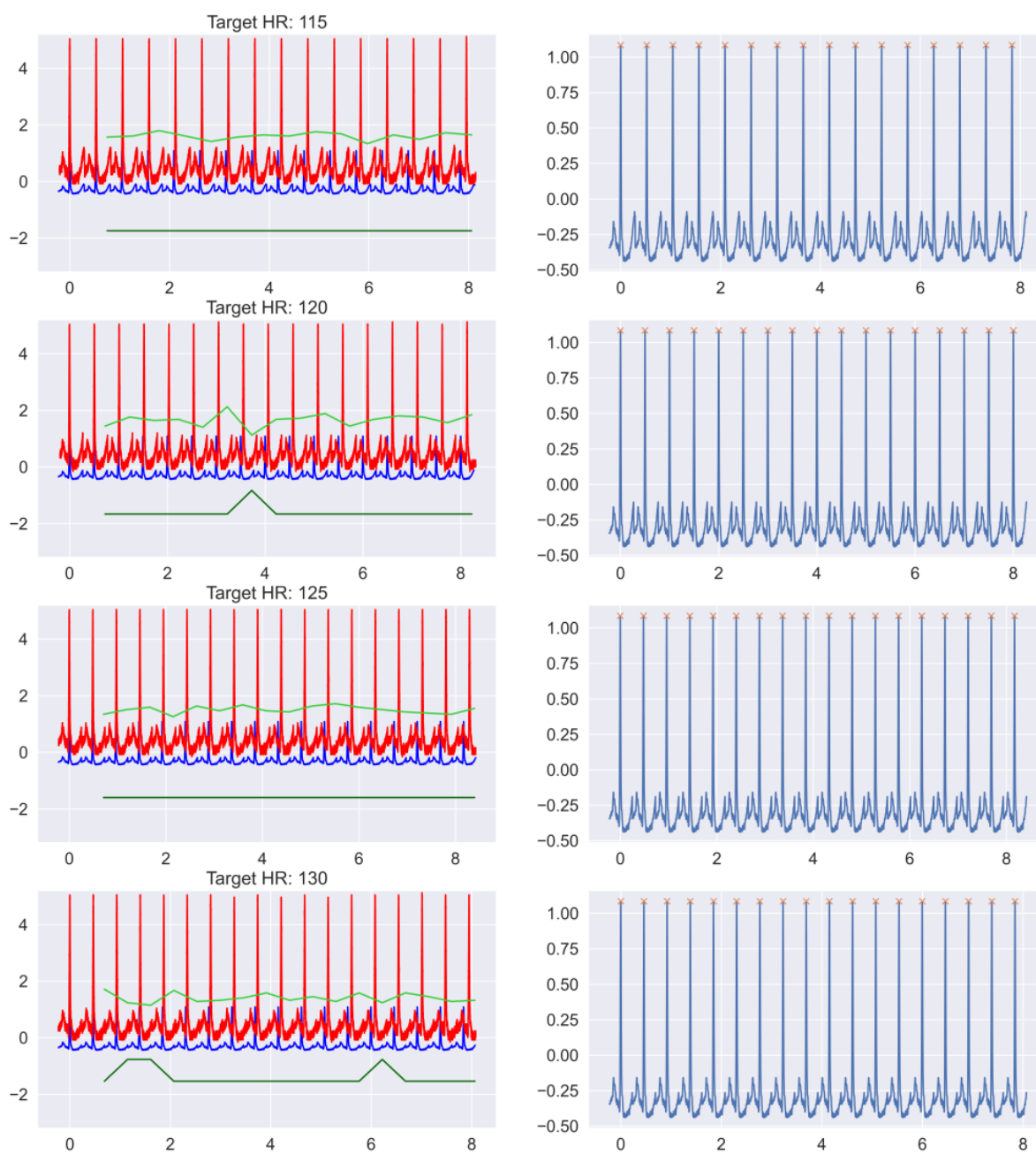
4	125	2.15	2.1694	0.006	-1.5509	-2.0	1.2578	-1.2422	-1.6
5	125	2.6306	2.6578	0.0078	-2.0054	-2.0	1.6324	-1.6062	-1.6
6	125	3.1111	3.1454	0.007	-1.8038	-2.0	1.4659	-1.4447	-1.6
7	125	3.5917	3.634	0.008	-2.0557	-2.0	1.674	-1.6464	-1.6
8	125	4.0722	4.1216	0.007	-1.8038	-2.0	1.4659	-1.4447	-1.6
9	125	4.5528	4.609	0.0068	-1.7533	-2.0	1.4243	-1.4043	-1.6
10	125	5.0333	5.0974	0.0078	-2.0054	-2.0	1.6324	-1.6062	-1.6
11	125	5.5139	5.5862	0.0082	-2.1059	-2.0	1.7156	-1.6867	-1.6
12	125	5.9944	6.0744	0.0076	-1.955	-2.0	1.5908	-1.5658	-1.6
13	125	6.475	6.5622	0.0072	-1.8543	-2.0	1.5075	-1.4851	-1.6
14	125	6.9556	7.0496	0.0068	-1.7533	-2.0	1.4243	-1.4043	-1.6
15	125	7.4361	7.5368	0.0066	-1.7028	-2.0	1.3827	-1.3638	-1.6
16	125	7.9167	8.0238	0.0064	-1.6522	-2.0	1.341	-1.3233	-1.6
17	125	8.3972	8.5118	0.0074	-1.9047	-2.0	1.5491	-1.5255	-1.6
0	130	0.2278	0.2266						
1	130	0.6889	0.6956	0.0079	-2.1887	-2.0	1.7108	-1.6821	-1.5385
2	130	1.15	1.1624	0.0057	-1.5858	-1.0	1.2337	-1.2187	-0.7692
3	130	1.6111	1.6288	0.0053	-1.4755	-1.0	1.147	-1.134	-0.7692
4	130	2.0722	2.0976	0.0077	-2.1341	-2.0	1.6675	-1.6401	-1.5385
5	130	2.5333	2.5646	0.0059	-1.6408	-2.0	1.2771	-1.261	-1.5385
6	130	2.9944	3.0318	0.0061	-1.6958	-2.0	1.3205	-1.3033	-1.5385
7	130	3.4556	3.4994	0.0065	-1.8057	-2.0	1.4072	-1.3877	-1.5385
8	130	3.9167	3.9678	0.0073	-2.0248	-2.0	1.5807	-1.5561	-1.5385
9	130	4.3778	4.435	0.0061	-1.6958	-2.0	1.3205	-1.3033	-1.5385
10	130	4.8389	4.9028	0.0067	-1.8605	-2.0	1.4506	-1.4299	-1.5385
11	130	5.3	5.3698	0.0059	-1.6408	-2.0	1.2771	-1.261	-1.5385
12	130	5.7611	5.8382	0.0073	-2.0248	-2.0	1.5807	-1.5561	-1.5385
13	130	6.2222	6.305	0.0057	-1.5858	-1.0	1.2337	-1.2187	-0.7692
14	130	6.6833	6.7734	0.0073	-2.0248	-2.0	1.5807	-1.5561	-1.5385
15	130	7.1444	7.2412	0.0067	-1.8605	-2.0	1.4506	-1.4299	-1.5385
16	130	7.6056	7.7082	0.0059	-1.6408	-2.0	1.2771	-1.261	-1.5385
17	130	8.0667	8.1754	0.0061	-1.6958	-2.0	1.3205	-1.3033	-1.5385

Figure M.1: Comparison of original (blue) and generated signal (red) of 0.5V scale recorded with oscilloscope. Both signals are aligned at the first R peak. Relative error of time difference between R peaks (light green) and HR determination (dark green) are also shown.









N

Monitor measurements: second stage timing checks

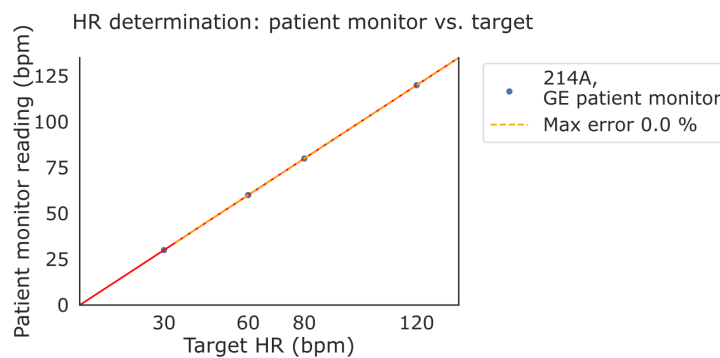


Figure N.1: HR determination checks using 214A simulator (Dynatech Nevada, 1994) and GE Healthcare patient monitor. Two measurements were carried out for each HR tested.

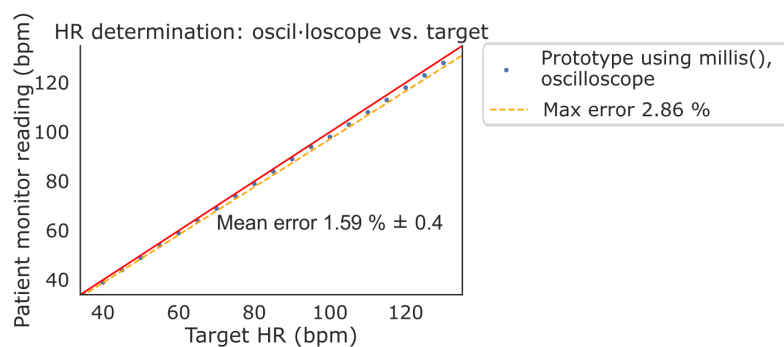


Figure N.2: HR determination checks of Prototype simulator programmed with millis() function. Measurements performed by oscilloscope data, processed with Python.

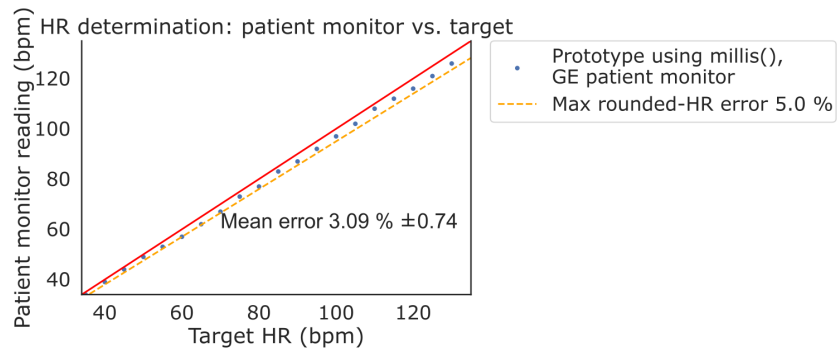


Figure N.3: HR determination checks using Prototype simulator programmed with millis() function. Measurements performed by GE Healthcare monitor.

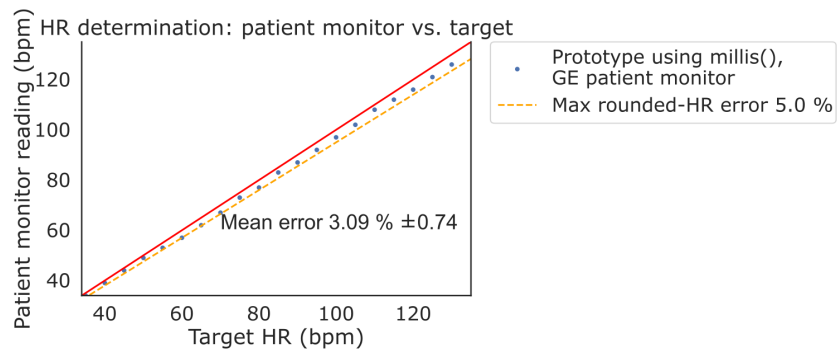


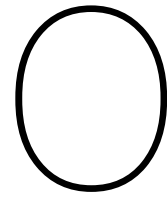
Figure N.4: HR determination checks of prototype simulator programmed with micros() function. Measurements performed by GE Healthcare monitor.

Table N.1: (Part 1) HR determination by 5 patient monitors at Green Pastures Hospital
 'Rate' refers to simulation rate, set to 1997 microseconds for all of them.

idx	Tester	DUT	Rate	35	40	45	50	55	60	65	70	75
2	Prototype	CMS8000	1997.0	35.0	40	45.0	50.0	55.0	60	65.0	70.0	75.0
3	Prototype	CMS8000	1997.0		40				60			
4	Prototype	Creative34	1997.0	35.0	39	45.0	50.0	55.0	59	65.0	69.0	75.0
5	Prototype	Creative34	1997.0	35.0	40	45.0	50.0	55.0	60	65.0	69.0	75.0
12	Prototype	Edan31	1997.0	35.0	40	45.0	50.0	55.0	60	65.0	70.0	75.0
19	Prototype	Mindray602	1997.0	35.0	40	45.0	50.0	55.0	60	65.0	70.0	75.0
25	Prototype	Philips159	1997.0		40				60			
26	Prototype	Philips159	1997.0	35.0	40	45.0	50.0	55.0	60	65.0	70.0	75.0

Table N.2: (Part 2) HR determination by 5 patient monitors at Green Pastures Hospital
 'Rate' refers to simulation rate, set to 1997 microseconds for all of them.

idx	80	85	90	95	100	105	110	115	120	125	130	135	Year con- struc.
2	80.0	85.0	90.0	95.0	100.0	105.0	110.0	115.0	120.0	125.0	130.0	135.0	2021.0
3							110.0		120.0				2021.0
4	80.0	85.0	90.0	95.0	100.0	104.0	110.0	114.0	120.0	125.0	130.0	135.0	2014.0
5	80.0	85.0	90.0	95.0	99.0	104.0	110.0	114.0	120.0	125.0	130.0	135.0	2014.0
12	80.0	85.0	90.0	95.0	100.0	105.0	110.0	115.0	120.0	125.0	130.0		2015.0
19	80.0	85.0	90.0	95.0	100.0	105.0	110.0	115.0	120.0	125.0	130.0	135.0	2020.0
25		85.0		95.0			110.0		120.0	125.0	131.0	135.0	2010.0
26	80.0	85.0	90.0	95.0	100.0	105.0	110.0	115.0	120.0	125.0	131.0	135.0	2010.0



Patient monitors characteristics at Green Pastures Hospital

Table O.1: Details of patient monitors tested in Green Pastures Hospital

DUT	Device model	Manufacturer according to type plate	Year of construction
CMS8000	CMS8000	Contec Medical Systems C0. Ltd	2021.0
Creative34	UP-7000	Creative Medical	2014.0
Creative879	UP-7000	Creative Medical	2022.0
Edan31	iM8	Edan	2015.0
Mindray602	uMEC12	Mindray	2020.0
Philips159	SureSigns VM4	Philips	2010.0

P

Monitor measurements: second stage amplitude checks

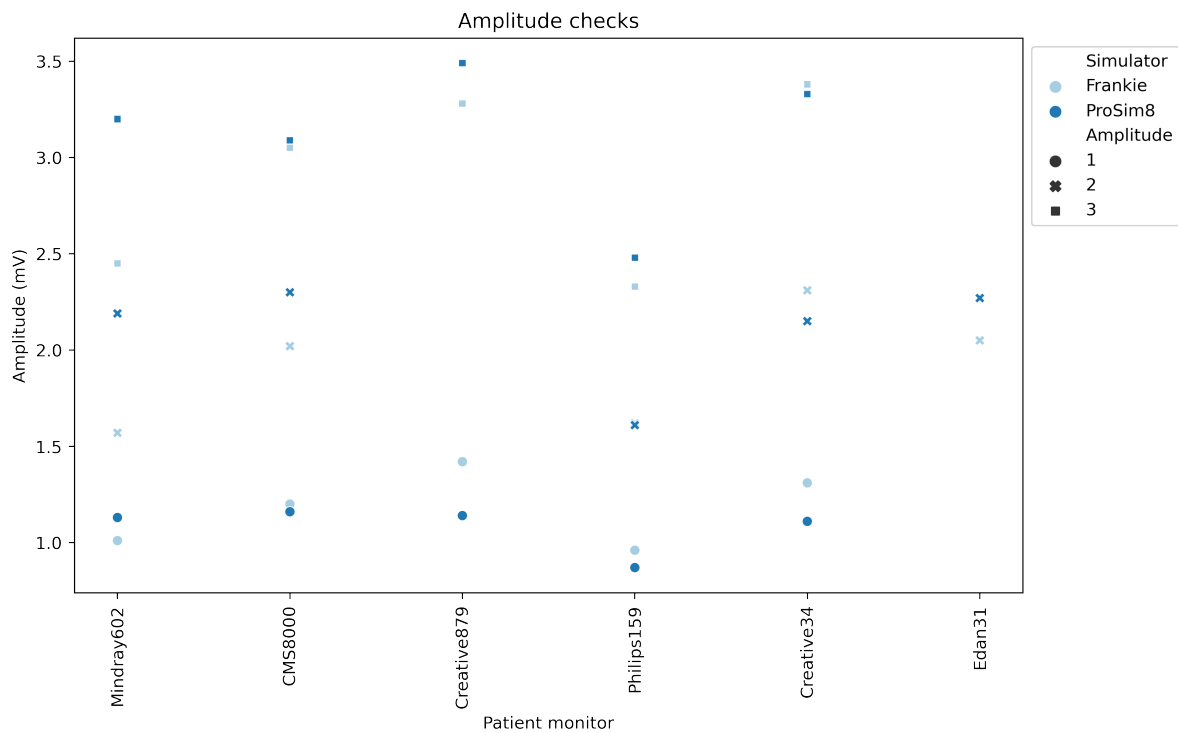
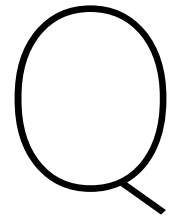


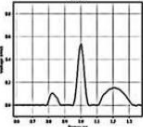
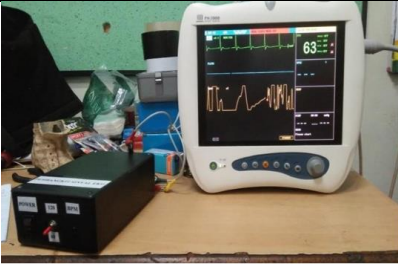
Figure P.1: Raw data of amplitude checks measured on 6 patient monitors in Green Pastures Hospital (Pokhara, Nepal). Pictures of the monitors' display were taken and digitally measured using Fiji software.

Patient monitor	HR	Amplitude	Measured amplitude mean	Measured amplitude std	Simulator
Mindray602	60	1	1.01	0.02	Frankie
Mindray602	60	2	1.57	0.05	Frankie
Mindray602	60	3	2.45	0.08	Frankie

Mindray602	60	1	1.13	0.02	ProSim8
Mindray602	60	2	2.19	0.03	ProSim8
Mindray602	130	3	3.2	0.04	ProSim8
CMS8000	60	1	1.2	0.03	Frankie
CMS8000	60	2	2.02	0.04	Frankie
CMS8000	60	3	3.05	0.02	Frankie
CMS8000	60	1	1.16	0.02	ProSim8
CMS8000	60	2	2.3	0.01	ProSim8
CMS8000	60	3	3.09	0.02	ProSim8
Creative879	60	1	1.42	0.02	Frankie
Creative879	60	3	3.28	0.05	Frankie
Creative879	60	1	1.14	0.02	ProSim8
Creative879	60	3	3.49	0.04	ProSim8
Philips159	60	1	0.96	0.02	Frankie
Philips159	60	3	2.33	0.01	Frankie
Philips159	60	2	1.62	0.01	Frankie
Philips159	60	2	1.61	0.01	ProSim8
Philips159	60	1	0.87	0.02	ProSim8
Philips159	60	3	2.48	0.01	ProSim8
Creative34	60	1	1.31	0.02	Frankie
Creative34	60	2	2.31	0.05	Frankie
Creative34	60	3	3.38	0.02	Frankie
Creative34	60	1	1.11	0.03	ProSim8
Creative34	130	2	2.15	0.01	ProSim8
Creative34	60	3	3.33	0.02	ProSim8
Edan31	60	2	2.05	0.04	Frankie
Edan31	60	2	2.27	0.02	ProSim8



Simulators accuracy and HR determination comparison

RESEARCH SIMULATORS		
<p>VergaraGil 2021</p>	<p>3.3% max rel error, although most of errors are below 2.5% mean relative error of 1.8 % uncertainty of ± 1 from 50 to 110 bpm, ± 2 from 120 to 200bpm, and ± 3 at 250bpm Uncertainty: HR < 110: +- 1 120 < HR <= 200: +-2 HR = 250: +-3</p>	<p>Lead II</p>  <p>Open-source Simulator</p>
	<p>We used a BeneHeart D6 vital signs monitor connected to the simulator. The experiments were carried out at a temperature between 22° C and 23° C and a relative humidity between 60% and 70%. Ten data captures, five upstream and five downstream, were taken. The measurement interval was ten beats and the measurement range started at 50 and ended at 250</p>	
<p>Vasilevich 2004</p>	<p>amplitude error <1.2 %; accuracy of freq adjustment ≥ 99.8 %</p>	
<p>Narvaez and Avedaño 2017</p>	<p><5% difference for HR and amplitude</p>	
<p>Vaisman 2000</p>	<p>± 1 % HR and amplitude accuracy</p>	
<p>Wijaya 2021</p>	<p>experimental uncertainty values of 63 - 127 bpm HR accuracy (for 60 and 120 bpm simulation)</p>	

The rest was FFT, correlation coefficient, mean square error, imprecise such as “meets standard” or no performance information at all.

COMMERCIAL SIMULATORS	
ProSim 8	Amplitude: +/- (2% of setting + 0.05mV) ECG rate accuracy: +/-1% of setting
ProSim 4	Amplitude: +/-5% of setting lead II ECG Rate accuracy: +/-1% of setting
213A	Amplitude accuracy (might be referring to square signal only, 2 Hz at 1 mV) is +/-5% on lead II
PS-2005 BC Biomedical	Amplitude accuracy: +/- 2% Frequency accuracy: +/- 0.5% of setting

To better interpret the recorded errors, the resolution of each patient monitor should be taken into account.

Mindray602	HR resolution	1 bpm
	Accuracy	± 1 bpm or $\pm 1\%$, whichever is greater
	Sensitivity	200uV (lead II)
	HR averaging method	In compliance with the requirements in Clause 201.7.9.2.9.101 b) 3) of IEC60601-2-27, the following method is used: If the last 3 consecutive RR intervals are greater than 1200 ms, the 4 most recent RR intervals are averaged to compute the HR. HR averaging method Otherwise, heart rate is computed by subtracting the maximum and minimum ones from the most recent 12 RR intervals and then averaging them. The HR value displayed on the monitor screen is updated every second
	Source	Manual
CMS8000	HR accuracy	± 1 bpm or $\pm 1\%$, whichever is greater
	HR averaging	Average value of latest 6 R-R intervals?
	Resolution	1 bpm
	Sensitivity	> 200 uV P-P
	Source	(not in the manual): https://www.hospitalstore.com/contec-medical-cms8000-12.1-inch-multipara-monitor/

Creative34	HR averaging	Averages the recent eight beats having RR intervals falling within the acceptable limits.
	HR accuracy	± 1 bpm or $\pm 1\%$, whichever is greater
	HR display range	15 bpm ~ 350 bpm
	Input signals range in amplitude	$\pm (0.5 \text{ mVp} \sim 5 \text{ mVp})$
	Sensitivity selection	$\times 1/2$, 5mm/mV tolerance: $\pm 5\%$ $\times 1$, 10mm/mV tolerance: $\pm 5\%$ $\times 2$, 20mm/mV tolerance: $\pm 5\%$
	Frequency response	Monitoring mode: 0.5 Hz ~ 40 Hz (+0.4 dB, -3.0 dB) Diagnostic mode: 0.05 Hz ~ 75 Hz (+0.4 dB, -3.0 dB)
	Source	Manual: https://bmet.ewh.org/bitstream/handle/20.500.12091/1637/Patient-Monitor-User-Manual.pdf?sequence=1&isAllowed=y
Creative879	Should be same as Creative 34	
Edan	Scale signal	1mVPP, accuracy is $\pm 5\%$
	Bandwidth	Diagnosis: 0.05Hz to 150Hz Monitor: 0.5Hz to 40Hz Surgery: 1Hz to 20Hz
	HR range	ADU: 15 bpm ~ 300 bpm PED/NEO: 15 bpm ~ 350 bpm
	Accuracy	$\pm 1\%$ or 1 bpm, whichever is greater
	Resolution	1 bpm
	Source	Not in manual: https://edanusa.com/wp-content/uploads/sites/12/2020/12/iM8-specs-sheet.pdf
	HR averaging method 1	Normally, heart rate is computed by excluding the minimum and maximum values from the 12 most recent RR intervals and averaging the residual 10 RR intervals.
	Method 2	If each of three consecutive RR intervals is greater than 1200ms, then the four most recent RR intervals are averaged to compute the HR.
Philips	HR range	15 – 300 bpm
	HR accuracy	+1 % or +- 5 bpm (whichever is greater)
	Bandwidth	Normal monitoring: 0.15 – 40 Hz
	ECG size (sensitivity)	2.0, 1.0, 0.5, 0.25 cm/mV or auto
	Input signal range	+ - 5 mV
	HR averaging method	Normally, HR is computed by averaging the 12 most recent RR intervals If 3 consecutive RR intervals is greater than 1200 ms (that is, rate less than 50 bpm, 80 for neonates), then the 4 most recent RR intervals are averaged to compute HR
	Source	Manual: https://az767150.vo.msecnd.net/pdf/Philips_SureSigns_VM4_Owners_Manual.pdf

GE Healthcare M-NESTPR	HR range	30 to 250 bpm
	HR accuracy	± 5 bpm or ± 5 %, whichever is greater
	Resolution	1 bpm
	Bandwidth	Monitoring: 0.5 to 30 Hz (-3 dB, with 50 Hz reject filter); 0.5 to 40 Hz (-3 dB, with 60 Hz reject filter) Diagnostic: 0.05 to 100 Hz
	HR averaging time	10s
	HR update Interval	5s
	Source	Manual: http://www.frankshospitalworkshop.com/equipment/documents/ecg/service_manuals/Datex-Ohmeda%20S-5%20M-Modules%20-%20Technical%20reference%20manual.pdf

By standard IEC 60601-2-27, HR averaging method should be disclosed.

R

Permission to cite collaborators

From: Otto Ardeby

Hello!

It would be an honor! 😊

Good luck with finalizing and presenting!

Otto

From: Anna:

> Hi again Otto,

> I am about to finish writing my master thesis and I would like to ask you for
> permission to mention your name regarding the information you shared with
me
> about the situation of medical device maintenance in Nepal.

> With all the information shared by you and others I made a table summarizing
> the main needs and possible solutions in low-resource settings. Your name
> would be added in the table caption mentioning you contributed to the
> information.

> Let me know what you think of it. The thesis will be uploaded at TU Delft
online
> repository and available there for free access.

> Sincerely,

> Anna

From: Anna Worm, 13 sep. 2023

Hi again Anna! How are you?? I hope everything is great. I am about to finish writing my thesis and I would like to ask you for permission to mention your name regarding the information you shared with me about the situation of medical device maintenance in SSA.

With all the information shared by you and others I made a table summarizing the main needs and possible solutions in LMICs. Your name would be added in the table caption mentioning you contributed to the information.

Let me know what you think of it! The thesis will be uploaded at TU Delft online repository and available there for free access. 13:17 ✓✓

Hi! yes sure no problem 13:25

From: Jean-Luc Mpamije, 13 sep. 2023

Hi again Jean-Luc! How are you? I hope everything is great. I am about to finish writing my thesis and I would like to ask you for permission to mention your name regarding the information you shared with me about the situation of medical device maintenance in Africa.

With all the information shared by you and others I made a table summarizing the main needs and possible solutions in low-resource setting. Your name would be added in the table caption mentioning you contributed to the information.

Let me know what you think of it! The thesis will be uploaded at TU Delft online repository and available there for free access. 13:20 ✓✓

Hi Anna!
Thank you for your work.
I appreciate you sharing the information. 13:23

Tu
Do you give me permission to mention your name in the report then? 😊

Yes, no problem. 10:48

From: Dieuwertje Drexhage

Anna Graell Collado • 1:18 PM

Hi again Dieuwertje! How are you?? I hope everything is great. I am about to finish writing my thesis and I would like to ask you for permission to mention your name regarding the information you shared with me about the situation of medical device maintenance in LMICs.

With all the information shared by you and others I made a table summarizing the main needs and possible solutions in LMICs. Your name would be added in the table caption mentioning you contributed to the information.

Let me know what you think of it! The thesis will be uploaded at TU Delft online repository and available there for free access.

TODAY

Dieuwertje Drexhage • 1:00 PM

Hi Anna, sorry totally forgot to react. Of course that is totally fine!

From: Anna

Yes, of course, you can see the new version with the change attached.

Thank you,

Anna

From: Ashish Chauhan

Can you re-write " In Nepal, there is no testing equipment besides few oxygen analyzers in hospitals supported by NSI " as " - In Nepal, there is almost no testing equipment besides few oxygen analyzers in government hospitals supported by NSI"

You can mention my name after the change.

Warm regards,
Ashish

From: Anna

Hi,

yes, of course. Please find it attached. The table gathers the input from you and the other collaborators together. Your name will be cited together with the rest of collaborators at the top of the table, mentioning that you contributed to its content (but does not specify what specifically you said).

Let me know what you think of it.

Sincerely,

Anna

From: Ashish Chauhan

Hello Anna,

I am fine with you mentioning my name but before that can you share the table that has my input so that i can reconfirm it.

Warm Regards,

Ashish

From: Anna

> Hello again Ashish,

> I am Anna Graell Collado, the TU Delft student working with Arjan Knulst who

> contacted you for details about hospital inventories and testing equipment at NSI.
> Arjan gave me your email to contact you.

> I am about to finish writing my master thesis and I would like to ask you for
> permission to mention your name regarding the information you shared with me.

> With all the information shared by you and others I made a table summarizing the
> main needs and possible solutions in low-resource settings. Your name would be
> added in the table caption mentioning you contributed to the information.

> Let me know what you think of it. The thesis will be uploaded at TU Delft online
> repository and available there for free access.
> Thank you in advance,

Anna
