UNIVERSITY OF CALIFORNIA,
IRVINE

Development for Embedded Fetal Monitoring System Using Electrocardiogram-Based Algorithm Assisted by Photoplethysmogram

THESIS

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Electrical and Computer Engineering

by

Andy Yu-Ting Fok

Thesis Committee:
Associate Professor Hung Cao, Chair
Assistant Professor Zhou Li
Professor A. Lee Swindlehurst

2023

DEDICATION

To family
for their constant and loving support

To friends
for the bits of joy and confidence

To school
for a stellar conclusion to such a large chapter of my life

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Professor Hung Cao, who provided me this excellent opportunity to work alongside him on such a wonderful project. His incredible insights never cease to amaze, and I am honored to have been a part of his research.

Those in the HERO Lab and Sensoriis who I have come to meet and know for the past year also deserve my thanks. In particular, thank you to Tim Etchells for overseeing the technical aspects of my work and to Tai Le for mentoring me throughout the research and program. I would also like to thank Daniel Jilani and Alex Huang for guiding me through how the Lullaby algorithm works, Xing Xia (Chris) for sharing his extensive knowledge of the ECG sensor with me, and Mohamed Benomar for his guidance on hardware integration. It was also a pleasure to meet Floranne Ellington, Jimmy Zhang, and Ramses Seferino Trigo Torres, who all shared the academic knowledge and motivation that I needed to keep moving forward.

I acknowledge Springer Nature for providing me permission to use a small portion from *Interfacing Bioelectronics and Biomedical Sensing*, Chapter 3.1.1 from "Electrocardiogram: Acquisition and Analysis for Biological Investigations and Health Monitoring".

ABSTRACT OF THE THESIS

Development for Embedded Fetal Monitoring System Using Electrocardiogram-Based Algorithm Assisted by Photoplethysmogram

by

Andy Yu-Ting Fok

Master of Science in Electrical and Computer Engineering

University of California, Irvine, 2023

Associate Professor Hung Cao, Chair

As the next step in mobile health monitoring, an embedded system for monitoring the health of pregnant women and their fetuses in the home setting is nearing the brink of possibility. The main obstacles currently in the way are heavy algorithms and real-time performance, which have been solved with lighter computation for fetal heart rate on the system's hardware (rather than via the cloud and through a server). This thesis will focus on the research done in optimizing the quality of electrocardiogram (ECG) sensor data so that the accompanying algorithm can return the most accurate fetal heart rate possible, which includes providing a powerful microcontroller for the algorithm to compute on and incorporating photoplethysmogram (PPG) data to help identify ECG peaks (which are vital to finding heart rate) that may otherwise have not been detected. Components for this system were selected to have a balanced tradeoff between sensor accuracy (to be able to capture ECG and PPG at reduced voltages due to abdominal placement) and power conservation (to minimally utilize the PPG and reduce energy during wireless transmissions). As part of ensuring proper connection between the components of the fetal system, outputted ECG simulator data from two different datasets and sensor-retrieved ECG data were compared using cross-correlation and showed 98% similarity between the two signals. Timing measurements for sensor-reading and algorithm functions were then taken by toggling a GPIO pin on the microcontroller and using an oscilloscope to read how long the pulse stayed high. These results caused several small modifications to the system in order to compensate for longer-than-expected function calls, with the most significant change being a decrease in the amount of data that can be processed by the algorithm. Simulator data being run through the algorithm confirmed a corresponding loss in accurate heart rate calculations (due to less detected peaks), but adequate accuracy in pinpointing the time indices of the signal's peaks. Simultaneous sensor readings were taken to finally examine how well PPG data would be able to help calculate heart rate in the event of unidentified or misidentified ECG peaks, and results showed impressive correlation. Should these PPG experiments be further investigated and the algorithm optimized to process more data as intended, this monitoring system can make great progress towards being able to provide accurate fetal data more consistently throughout the course of pregnancies.

CHAPTER 1 – INTRODUCTION

1.1. Motivation

With the advancements in wearable technology today, it has never been easier to measure one's health relatively accurately, quickly, and in real-time. Heart rate monitoring and calorie expenditure through physical activity, via smartwatches, are modern and popular examples of constantly developing features to aid in overall wellness. However, the market for more advanced health monitoring is still in development [1]. In particular, one major goal of the health community is being able to measure a fetus's heart rate consistently and around the clock, as it is vital towards identifying maternal and fetal well-being during the mother's pregnancy [2]. Currently, however, this type of monitoring is primarily done only viably at in-person clinical locations, which has brought up even more concerns recently for pregnant women in light of COVID infections that place them at greater risk [3]. To resolve this dilemma, a device capable of performing such monitoring should be: designed with mobility and convenience in mind, created with acceptable heart rate acquisition, labelled safe for consistent use, and developed for users and medical staff to use as a form of preliminary review.

1.2. Background and Related Work

A standard method for obtaining heart rate signals is called electrocardiogram, which measures electrical signals from heart activity that transmit throughout the entire body [4]. The signal is periodic in nature, with each period consisting of several important components as shown in **Figure 1** [4] such as the Q peak, R peak, T peak, and QRS interval or segment. The human fetal ECG shares a similar shape as the human adult ECG [5].

Typical amplitudes also appear to be in the single-digit millivolt range for adults [4] and in the tens-digit microvolt range for fetuses [6], though these are most likely after amplification on the corresponding systems' devices.

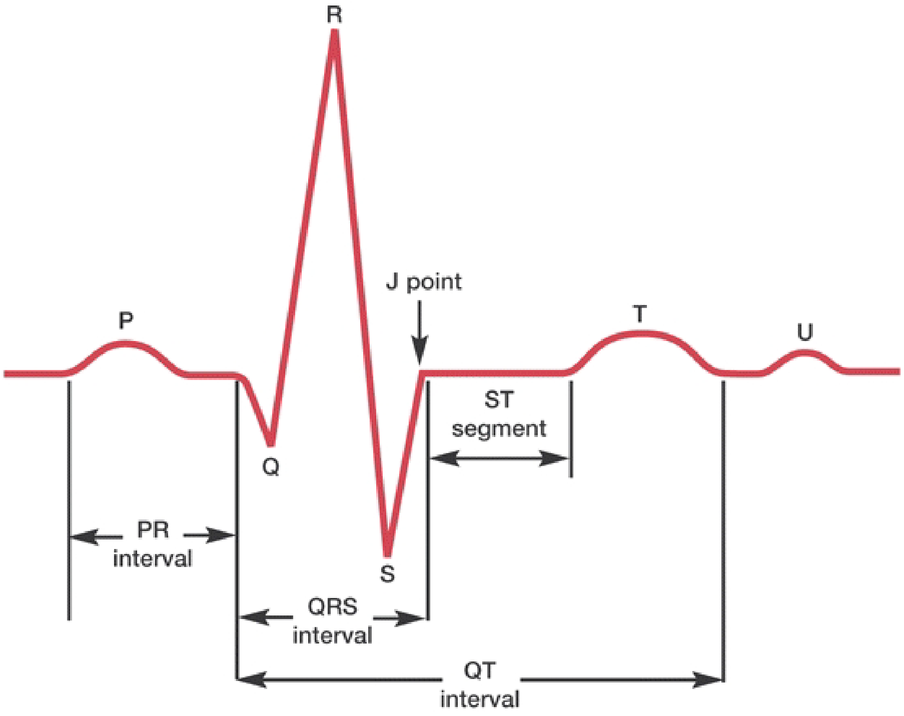


Figure 1 – Major Components of a Period from Typical ECG Signal

There are currently several systems on the market or in development that utilize fetal ECG to measure fetal heart rate. Most of them, however, do not provide the ability for real-time or mobile computation due to the complexity in algorithms such as the standard template subtraction [3]. One project focused on creating a lighter algorithm called Lullaby that would also theoretically fit on an embedded device and run in real-time (17.8 ms of computation time per function call, with each call occurring every second) [3]. Preprocessing uses peak finding algorithms to find the maternal QRS segments (mQRS's) and reconstruct a maternal signal that is used for template subtraction to obtain the residual fetal signal. Then the actual steps of the algorithm take place, which performs operations on a 3D array of time differences between pairs of peaks; using the outputted fetal period and reference time position, the remaining fetal peaks can be determined.

Improvements in other aspects of the ideal fetal monitoring system were made in other projects. One system addressed the mobile aspect by creating a wearable patch that simply captured and sent data over to a mobile app, which then utilized a cloud server to perform the main computations using an algorithm called Extended Kalman Filtering [2]. Follow up research was done to additionally incorporate real-time computation by replacing the algorithm with Lullaby and performing it on the patch device itself [1].

1.3. Purpose

Lullaby will serve as the primary algorithm used in this project to find fetal ECG peaks and calculate fetal heart rate. But, despite great strides towards feasible heart rate monitoring in an embedded system, the algorithm can only perform so well with the data given to it. On the sensing portion of such a system, a smaller and less-resourced ECG sensor will generally be more susceptible to noise, motion artifacts, etc. compared to a complete 12-lead ECG device [7]. So, the sensor may be the cause of issues with the algorithm in running optimally and correctly identifying QRS complexes.

One idea would involve incorporating other sensors altogether to provide additional data that ECG alone may not be able to capture. In particular, there may be merit in installing a photoplethysmography sensor, which measures the blood flow directly under skin, with the help of light shining on the skin to illuminate the blood vessels more. Much work has been done in the past several years to investigate the viability of using PPG sensing as a convenient method for heart rate monitoring able to assist or potentially replace ECG sensing, especially in constrained environments and embedded systems [8]. Fetal monitoring with such a sensor has also been proposed in previous research [9]. If the heart rates derived from ECG and PPG are similar (**Figure 2** [10] shows what a usual PPG signal looks like) then, theoretically, the time between one systolic PPG peak to the next (and likewise for any other important component of a PPG signal) would be approximately equal to the time between consecutive R ECG peaks.

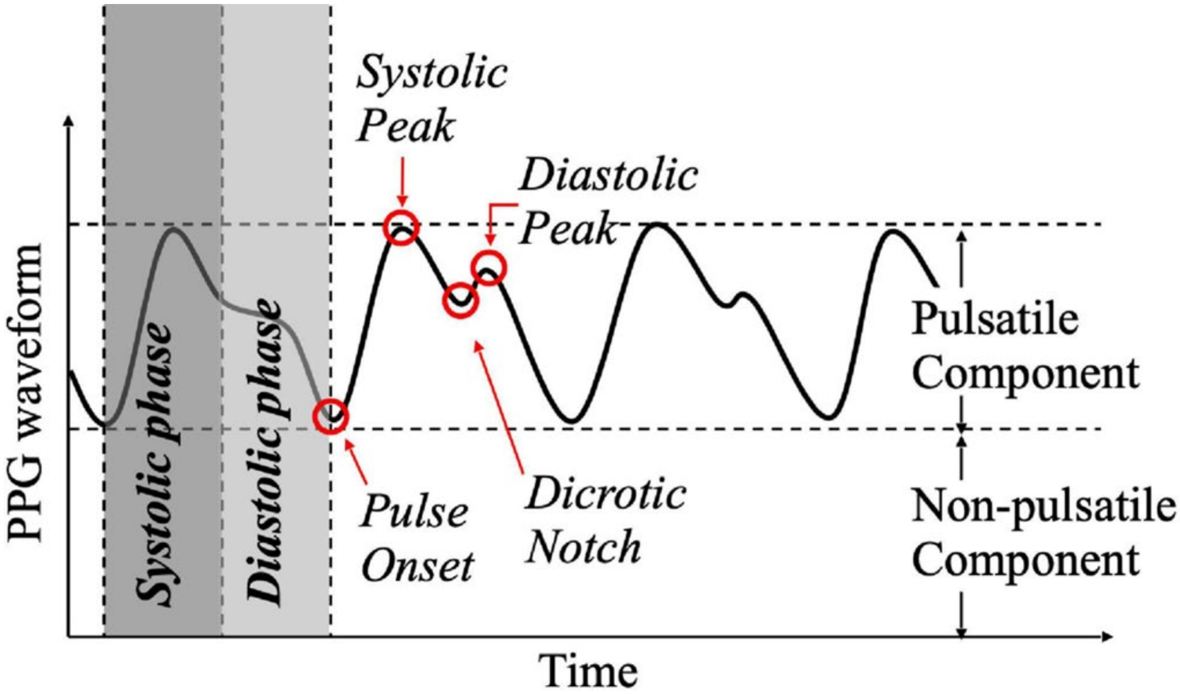


Figure 2 – Major Components of a Period from Typical PPG Signal

However, no matter what other useful features get implemented to help validate the mQRS that the algorithm supposedly identifies, it is vital to have the flexibility to explore such options without too many constraints in the system. Because of this, the proposed system in this work would like to retain a semblance of an embedded system, but include more memory and higher core clock speeds among other features. It will be closely based on the monitoring system, specifically its patch, that utilized Extended Kalman Filtering as its primary algorithm [2].

In summary, the goal of this thesis is threefold. First, an embedded system with added resources is created and configured to facilitate the data collection from multiple sensors, Lullaby algorithm execution on the local microcontroller, and transmission of results wirelessly to a mobile device. Second, the Lullaby algorithm's general performance on this more powerful system (specifically on the microcontroller) is examined. Third, the system's data collection scheme is leveraged to test the feasibility of using PPG (maternal) heart rate to help identify any missing ECG (maternal) peaks that would aid in the Lullaby algorithm's identification of fetal QRS complexes (fQRs's).

CHAPTER 2 – HARDWARE CHOICES

2.1. Overview

Sensors capture data non-invasively from signals that the mother and fetus naturally produce. However, the data coming in is a combined signal, consisting of the maternal signal and fetal signal. To extract the fetal signal, a microcontroller runs an algorithm in charge of filtering and processing the data. Once extracted, the data is forwarded via a wired connection to a Bluetooth Low Energy (BLE) module, which then sends the data wirelessly to the subject's mobile device for examination. **Figure 3** shows the system in terms of its main components and a summary of the interactions between them.

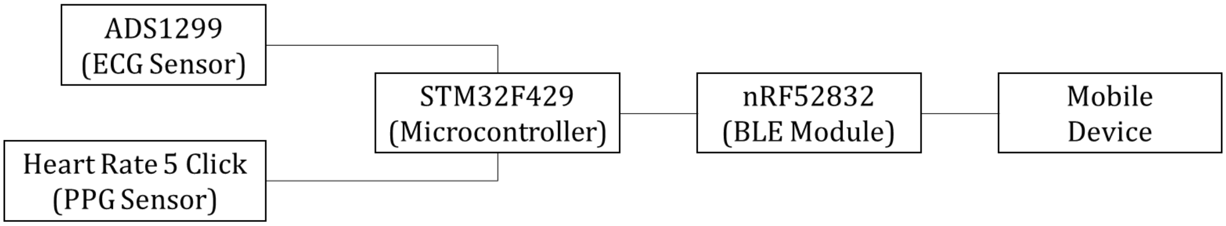


Figure 3 – System's Hardware and Connections

2.2. ADS1299

For this project, the ADS1299 was chosen to serve as the ECG sensor in the system. Up to 8 channels of data can be captured in the sensor, with each channel given a dedicated programmable gain amplifier (PGA) and analog-to-digital converter (ADC) [11]. The amplifier's amount of gain can be programmable and set up to 24. The converter has 24-bit accuracy and is simultaneous-sampling. The data can be outputted at several possible rates, ranging from 250 to 16,000 samples per second (SPS). Besides accepting ECG signals as the input signal, the sensor also helps capture data for EEG and even provides test signals internally generated to validate correct setup and check for any noise. It uses Serial Peripheral Interface (SPI) to send its collected and processed data to a microcontroller or other compatible device.

One competitor for an ECG sensor is the AD8233 from Analog Devices [12]. The product focuses on acquiring the signal from its inputs, passing it through the instrumentation amplifier, and then processing the data via high-pass filtering (and optional low-pass filtering for further noise reduction). However, this product does not contain an ADC; rather, its output is an analog ECG signal that will be passed as input to some converter. For this device, a 12-bit ADC is commonly utilized, just like the one available in this system's microcontroller. However, the selected chip for the fetal system already utilizes a 24-bit ADC, which can provide increased accuracy in the data compared to the Analog Devices product. Because the electrocardiogram signals will be recorded from the abdomen, where the signal is relatively weak compared to the chest and heart area, the resolution of the converter would preferably be as high as possible.

2.3. Heart Rate 5 Click

The PPG sensor installed in this system is mikroBUS's Heart Rate 5 Click (HR5 Click) [13]. The light-emitting diodes (LEDs) and photodiodes are located next to each other on the same surface, making this device a reflection PPG sensor [14]. It illuminates the blood under subjects' skin using up to 3 different LEDs: green, red, and infrared (IR); one of 2 different photodiodes are activated to read the light data for either heart rate monitoring (HRM) or peripheral capillary oxygen saturation monitoring (SpO2). Once the data is collected, it gets passed to the heavily utilized AFE4404 IC made by Texas Instruments. It can amplify the input current using the built-in trans-impedance amplifier (TIA), convert the sensor data to digital representation with its ADC, and perform ambient subtraction to clean up and finalize the PPG signal on at least the data from the green LED [15]. Its primary clock can be selected to either use the 4MHz internal oscillator or an external clock running between 4 and 60MHz. This will then run other secondary clocks, used for the ADC and timing engine (TE). Ultimately it can output PPG data at highly programmable sampling rates, between 10 Hz and 1 kHz, that can be read from its registers via an Inter-Integrated Circuit (I2C) interface.

One strong competitor is the MAX30101 from Analog Devices and Maxim Integrated. Instead of the HR5 Click, which composed of several components to perform the sensor's operations, the MAX30101 provides an entire system for capturing, processing, and even transmitting the data all in one package [16]. Like the AFE4404, several of its properties are all programmable; for example, the ADC's resolution is flexible between 15 and 18 bits, and its sampling rate can set to be one of 8 different values between 50 and 3,200 SPS. While it does take 5V to operate the LEDs, the rest of the system (ambient light cancellation, ADCs, digital filters, I2C communications, etc.) can operate off of 1.8V. This is one of the reasons why this product makes for an excellent solution should ultra-low power be a concern for the system. Yet it also makes adaptability to standard devices slightly more complicated, as most microcontrollers operate on 3.3V or 5V. This usually includes their GPIO pins and communication interface pins, which will not accept the 1.8V used by the MAX30101 to output its I2C traffic and trigger its interrupts. So, it would be necessary to add components like voltage-level translators or bus transceivers to adjust for the higher voltage levels that the microcontrollers are expecting. While this has been done specifically with the MAX30101 in products such as the Heart Rate 4 Click and Hexiwear, the topic of signal accuracy still remains an issue. Even in the Hexiwear application, the device could only offer rough estimates for heart rate despite the PPG sensing being performed at ideal locations such as the fingertip and wrist [17].

2.4. STM32F429

For this project, the STM32F429 from STMicroelectronics has been selected to perform the microcontroller's tasks of data collection and primary computation. It offers a large amount of memory, with 2MB of flash and 256KB of SRAM, and an external memory controller should even more memory be required [18]. Its core is an ARM 32-bit Cortex-M4 capable of clock frequencies up to 180MHz. Other relevant features include more than 160 I/O ports all capable of handling interrupts, a 16-stream DMA controller to offload computation from the main processor, 3 I2C interfaces, and 6 SPI interfaces with communication speeds as high as 45Mbits/s.

One strong competitor in the market is Silicon Labs' EFM32 Giant Gecko Series 1. It similarly has more than 100 I/O pins, several options to utilize for the DMA controller, and a handful of various communication interfaces that are available for use [19]. First to appeal to this project's needs was its low power usage. A comparison of the maximum current consumption between the STM and EFM was done, examining both chips at their highest power mode, with all peripherals disabled, and at an operating temperature of 25°C. Both provide differing options for clock frequencies, so only the range of frequencies between 16 and 72 MHz were used for calculations. In addition, the STM board's datasheet only provided max current in units mA, so their values were converted to μA and then divided by their corresponding frequencies. In this range, the Silicon Labs device used an average of 125±4 uA/MHz[1] while the ST Electronics device used an average of 407±121 uA/MHz; this makes the EFM approximately 3 times as power conserving as the STM. More details for this can be found in **Figure 4** and **Table 1**. It also excels in terms of memory capacity, providing as much flash as the ST board and double the RAM in certain devices. While it does also support a connection to an external memory device, the extra memory is limited for data used by a thin film transistor (TFT) controller that operates with RGB interfaces and displays. For this fetal system which relies on a general and large memory space to run its algorithm and displays any data to a mobile app, the ST board with its more general-use external memory capability is preferred. However, its core brings the most concern; while it is an ARM 32-bit Cortex-M4 processor just like the STM32F429, its maximum clock speed is only 72MHz, which is less than half of its competition. In order to prepare the system for its real-time application, a higher clock speed device would be preferred instead.

[1] Calculations for the mean/average are formatted in this document as *<average>*±*<1-standard-deviation>*.

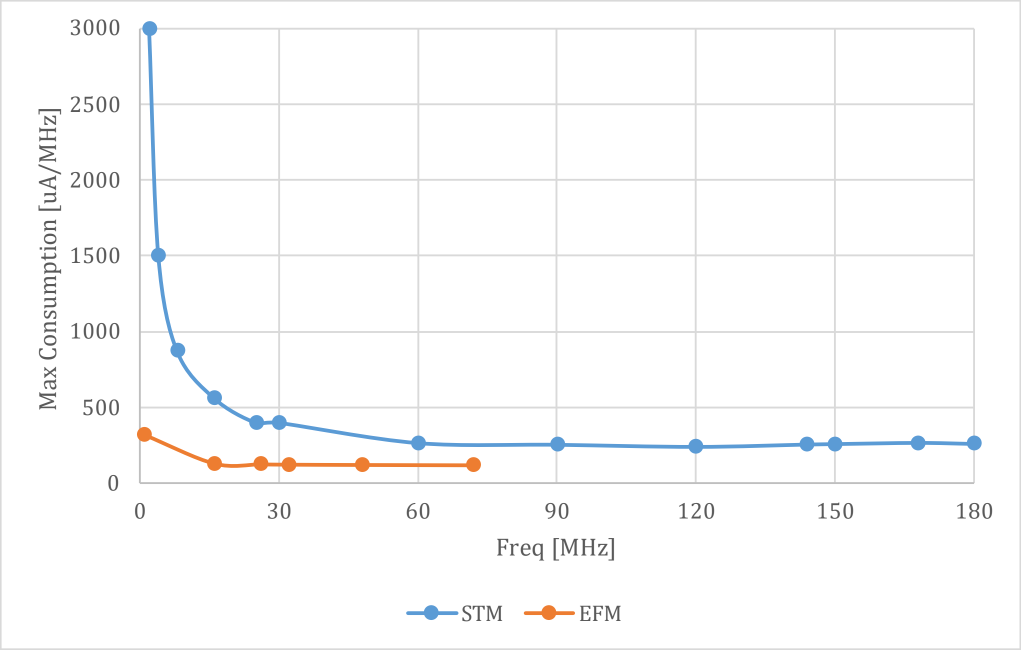


Figure 4 – Graphed Current Consumption, STM32F429 vs. EFM32GG11

Table 1 – Tabulated Current Consumption, STM32F429 vs. EFM32GG11

Device	Frequency [MHz]	Max Current Consumption [uA/MHz]
STM32F429	2	3000
STM32F429	4	1500
STM32F429	8	875
STM32F429	16	563
STM32F429	25	400
STM32F429	30	400
STM32F429	60	267
STM32F429	90	256
STM32F429	120	242
STM32F429	144	257
STM32F429	150	260
STM32F429	168	268
STM32F429	180	261
EFM32GG11	1	319
EFM32GG11	16	131
EFM32GG11	26	126
EFM32GG11	32	124
EFM32GG11	48	122
EFM32GG11	72	120

2.5. nRF52832

For the system's BLE module, the nRF52832 from Nordic Semiconductor was selected. It carries a 2.4 GHz transceiver for its Bluetooth communication, which has a -96 dBm sensitivity in low energy mode [20]. The maximum current is 5.4 mA during RX and 5.3 mA during 0 dBm TX; overall its TX/output power ranges between -20 dBm and 4 dBm, with 4 dBm steps. Several of each communication peripheral is installed, including most importantly SPI with which there are up to 3 of them.

The EFR32BG24 family of Bluetooth-capable SoC's competes well with the Nordic chip. It similarly carries a 2.4GHz radio and has up to 3 SPI peripherals, but has a better sensitivity of -105.7 dBm [21]. It can also use less current during its operations, being able to use at most 4.4mA during RX and 5.0mA during 0dBm TX. But its output power is not as flexible as the nRF's, providing only options for 0 or 10 (or 19.5dBm, depending on the specific device). With this range, it also leans towards the side of heavier power consumption than the Nordic chip which is not preferred for a power-saving system such as this.

CHAPTER 3 – IMPLEMENTATION

3.1. ECG Sensor

In terms of the ECG sensor, one crucial aspect about its hardware must be determined before programming. Inputs in the ADS1299 can be set up to be either differential, where measurements are done with respect to an electrode on the subject, or single-ended, where measurements are done with respect to a patient "signal reference and bias" (SRB) pin. The latter is there to help establish a reference input (positive or negative pin) that is particularly helpful when using multiple channels, so that all channels' data is with respect to that voltage. However, because this project currently uses only 1 channel from the sensor, it is decided to have differential inputs to reduce the complexity of setup. In addition, the type of power supply that the ADS can operate with can either be a unipolar 5 V one or a bipolar ±2.5 V one. Because the expected input signal is not just differential but also is centered around 0V (in the ideal situation), the ADS needs to be operated with the bipolar supply.

First, the ECG sensor is programmed to power up as generally dictated by its ADS1299's datasheet regarding the general flow for this procedure. Once power is provided to the sensor, the clock select (CLKSEL) pin must be set to 0 or 1 depending on whether an external clock will run the sensor or not, respectively. Regardless of the choice, the clock should run at 2.048 MHz. The system is configured to utilize its internal oscillator for the clock; as such, CLKSEL is directly tied to the sensor's main digital voltage pin (DVDD) so that it is always set as 1 and so that no firmware needs to be written for this portion of configuration. Afterwards, the other GPIO pins need to be initialized to a starting value of low or high. In particular, the power-down (PWDN) and reset (RESET) pins need to be toggled in order to trigger a power-on reset and give enough time for the internal oscillator to start up should it be used. A delay of 18 periods of the main clock is given afterwards for the ADS1299 to be configured. The device will then start up in read data continuous mode (RDATAC) but must be immediately stopped with a SPI command in order to write settings into the ADS's registers. To confirm that this mode has been actually stopped, the current system additionally has the microcontroller read the sensor’s ID register; this also serves as a way to confirm that the correct device is being programmed.

Now that registers can be written to and read from, all the relevant registers are edited to program the ADS1299 as intended. To begin, there are 3 registers in charge of general settings such as the output data rate, internal reference buffer state, and BIAS amplifier settings. For the project, the general settings were configured to: provide data at 250 SPS, enable the internal reference (which outputs a low noise 4.5 V signal), use an externally fed BIASREF signal, and disable any other BIAS related settings. Then, there are 8 registers (one for each channel) where each one controls the setup of its corresponding channel with settings like its on/off state, PGA gain, and expected input signal. Only Channel 1 was powered on for the fetal system, with a gain of 24 and expectancy of normal electrode input. The ADS1299 provides other registers related to BIAS measurements and lead off detection, but they will not be covered here due to their irrelevance to the fetal project.

Now that the ADS1299 has been configured, the START pin is set high so that the sensor can begin capturing and converting data. The device is also set back into read data continuous mode so that data is automatically placed in the output register each time a new sample has been collected. **Table 2** summarizes the main settings used to configure the ECG sensor.

Table 2 – ECG Sensor Configuration Settings

Parameter	**Value**
Input Type	Differential
Power Supply Type	Bipolar
Power Supply Voltage	±2.5 V
Clock Source	Internal Oscillator
Sampling Rate	250 SPS
Internal Reference Buffer State	Enabled
Num. Channels	1
PGA Gain	24
Channel Input	Normal Electrodes

3.2. PPG Sensor

Next, the PPG sensor is initialized. The microcontroller sets the sensor's reset pin low for 25-50 µs and then high again. Then its diagnosis register is first written to; optionally, this is where a reset can be performed again but with software rather than the reset pin hardware. To be sure that the reset was successful, the software reset is performed as well for the fetal system. This system also does a quick check with its I2C peripheral to make sure communications are ready between the two devices before proceeding with the remaining setup. Next is the settings register, where the clock source is selected as either the internal 4 MHz oscillator or external source and where components like the ADC and TIA can be either enabled normally or under dynamic power-down mode. The LED current range is also determined here to be between 0-50 mA or 0-100 mA. In this project, the sensor uses the internal clock, keeps the ADC in power-down mode for energy conservation, leaves the TIA running as normal, and chooses the 100mA range for its LEDs.

The sensor's timing control within its receiver is programmed next, which contains configurations spanning over 30 registers in charge of 3 primary elements: choosing whether to read data from the infrared LED, controlling the length of measurement pulses, and deciding the timing engine's clock speed. First to address is the IR LED, which can either be enabled and involved during measurements (3-LED mode) or disabled (2-LED mode). To provide context, 4 measurements are made per cycle: one per active LED to measure the light from said LED, and the remaining others to measure the ambient light. So should the IR LED be disabled, ambient subtraction can just be performed on the data from the red LED. But the system has chosen to utilize the IR LED under the possibility that the deeper penetration of infrared light past the skin may be beneficial when being applied at the abdominal region [22]. Second, the times of each measurement pulse involved in the timing control must be set. For each of the 4 measurements, there are at most four pulses that represent when to: turn on the corresponding LED (if applicable), gather the sample via the photodiode, signal the ADC to prepare for conversion, and let the ADC convert the collected data. One additional pulse occurs after the 4 measurements that signals the receiver to enter dynamic power-down mode; essentially, no measurements are taken for the duration of this pulse. Each time (start and end, with respect to a single cycle) for all these pulses is designated a register to store its value into. For the project, the times selected were chosen from the default settings that minimized active time and power consumption [15]. The very end of the entire cycle is dictated by what is denoted as the PRPCT register, which is one of the driving factors in deciding the sampling rate. It is used for a counter acting as the "sampling rate clock", behaving almost like a timer in that it: starts at 0, increments every period of the TE's clock, and resets to 0 after it reaches a "high value” (the PRPCT). The high value can be programmed via this register to any 16-bit value (up to 65,535 in decimal); while it could potentially achieve sampling rates as low as 4 Hz and as high as the TE’s clock, it is recommended to choose a valid PRPCT value such that its rate is within the recommended range. Third, the TE's clock speed is determined, which is the last timing control setting to program yet also another important factor in deciding the sampling rate. This is controlled by dividing the ADC clock by a power of 2 (as much as 16); as such, the programmer is given for all intents and purposes a 3-bit register to determine the division. A clock divider of 1 is selected to run the timing engine at 4 MHz for this project. When combined with a PRPCT value of 39,999, the sensor is programmed to output data at 100 SPS.

A few remaining settings need to be adjusted in order to finalize the PPG sensor's setup. For example, the NUMAV field controls how many times the same measurement's data is converted in the ADC; an average of all conversions is then stored in the output register. This can help reduce any noise that could be present (at the cost of a longer pulse during data conversion), but only in the converter. The system is configured to use an average from 4 attempts for each sample. Another important setting is regarding the TIA gain, which has several programmable resistance values to choose from between 10 kΩ and 2 MΩ. These values are always used when collecting data in the LED1 and AMB1 phases, while a secondary set of gain values can be used for the other phases optionally. For the fetal system, the primary gain is set to 50 kΩ and the secondary gain is used and set to 25 kΩ. At this stage, the amount of current going through all 3 LEDs is determined as well. Each LED's current can be configured in a 6-bit field, providing 64 available values equally separated between the minimum and maximum of the current range. For this system, the green LED receives 24 mA while the red and IR ones each receive 4.8 mA.

Table 3 summarizes the main settings used to configure the PPG sensor.

Table 3 – PPG Sensor Configuration Settings

Parameter	**Value**
ADC Conversions Per Sample	4
ADC Power Setting	Power-Saving
TIA Power Setting	Normal
Measurement Optimization	Min. Active Time
Clock Source	Internal Oscillator
Primary Clock's Speed	4 MHz
ADC Clock's Speed	4 MHz
TE Clock's Speed	4 MHz
PRPCT (Num. TE Periods)	39,999
Sampling Rate	100 SPS
IR LED Status	On
TIA Gain (Green LED, Green Ambient)	50 kΩ
TIA Gain (Red LED, Red Ambient or IR LED)	25 kΩ
LED Current (Green)	24 mA
LED Current (Red, IR)	4.8 mA

3.3. MCU

First to configure for is the ECG sensor. 2 GPIO pins are configured in standard, output mode for the PWDN and RESET pins needed to initially reset the ADS1299. To allow for subsequent communication in configuring the sensor and obtaining data from the sensor, a SPI peripheral is prepared in full-duplex master mode and SPI mode 1 (clock polarity low, clock phase high). The data is expected to have a size of 8 bits (with the MSB first) and a transmission rate of 3.125MBits/s baud rate which is divided by 2 from the 6.25MHz clock that the microcontroller’s APB2 (Advanced Peripheral Bus) uses. The SS pin is not configured via the STM's hardware NSS signal, but rather through a simple GPIO output pin. Afterwards, another 2 GPIO pins are configured, one for the sensor's START pin in normal output mode and one for the DRDY pin in "external interrupt" (EXTI) mode. Trigger detection for the EXTI pin occurs on the falling edge, and its corresponding NVIC interrupt is enabled. When the interrupt handler runs, the sensor's data is read over SPI in blocking mode and saved into a data array for later processing.

Next to set up for is the PPG sensor. To transmit the reset pulse that the Heart Rate 5 Click requires before setup, a GPIO pin is set up to a standard, output mode. For the primary communication with the PPG sensor, an I2C peripheral from the MCU is configured, running at a clock speed of 100kHz. When the sensor transmits its ADC ready interrupt signal, it is via a connection to a GPIO pin on the microcontroller set up in EXTI mode. This EXTI pin detects its trigger on the rising edge of the signal, as is expected from the sensor's AFE4404, and its corresponding NVIC interrupt is enabled. When the interrupt handler runs, the sensor's data is read over I2C in blocking mode and saved into a data array for later processing. While unnecessary due to the sensor's primary clock being its internal oscillator, a clock pin from the MCU is provided and connected to the PPG sensor. It is outputted at a frequency of 4MHz, which is divided by 4 from the STM's 16MHz high-speed internal RC oscillator.

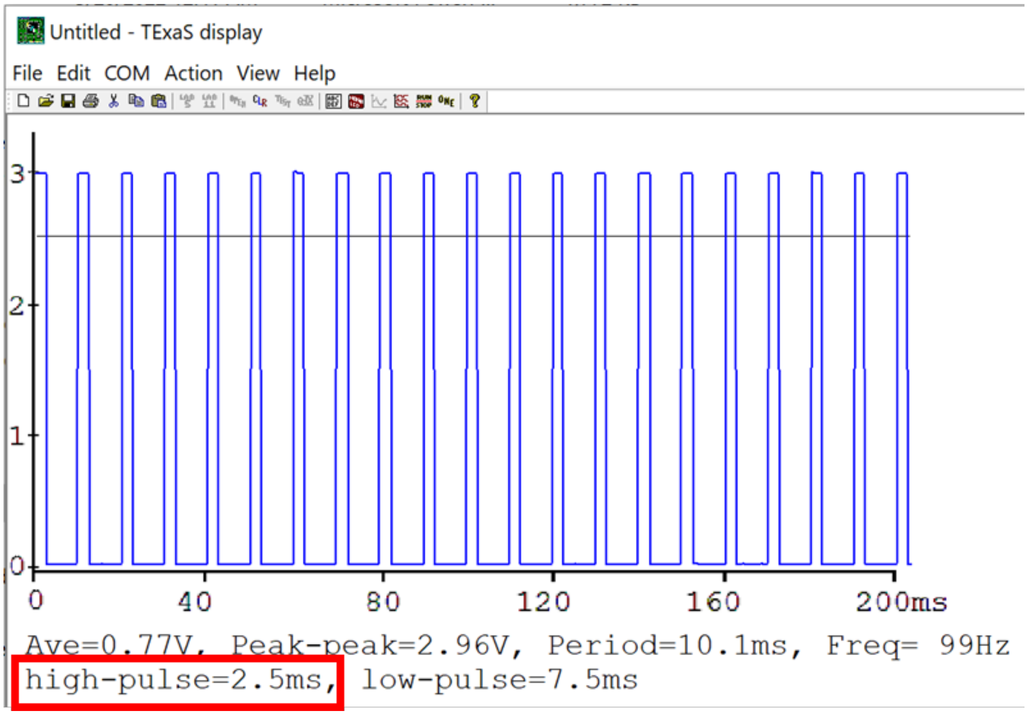
Finally, the connections to the BLE module are made; due to the simplicity of the system, only a SPI peripheral needs to be set up. It is prepared in full-duplex master mode and SPI mode 1 (clock polarity low, clock phase high). The data is expected to have a size of 8 bits (with the MSB first) and a transmission rate of 1.5625MBits/s baud rate which is divided by 2 from the 3.125MHz clock that APB1 uses. The SS pin is not configured via the STM's hardware NSS signal, but rather through a simple GPIO output pin. In addition, direct memory access (DMA) has been configured on this peripheral to more quickly transmit data to the BLE module.

All other peripherals and pins in the STM are disabled by default. **Table 4** provides a short summary of the pins used in the microcontroller.

Table 4 – MCU Configuration Settings

Parameter	**Value**
Num. GPIO Pins (Normal)	6
Num. GPIO Pins (EXTI)	2
Num. SPI Peripherals	2
Num. I2C Peripherals	1
Master Clock Output Speed	4 MHz
DMA-Enhanced Peripheral	SPI (BLE Module)

One aspect to bring attention to is the time it takes for the microcontroller to read data from its sensors. To test the timing of the interrupt handlers, a blank project in STM32CubeIDE (the integrated development environment accompanied by the ST board) was made with the sensor configuration library and simple code to: set a GPIO output high, call the I2C or SPI function for reading the data, set the GPIO output to low, and delay for several milliseconds. An oscilloscope is connected to the GPIO pin that can measure the time that its signal is high, which corresponds to the time needed for the function to execute. For this particular experiment, the onboard scope from TI's MSP-EXP432P401R was used; it has a sampling rate of 10kHz and 8 bits of precision. Texas Instruments also provides the TExaS Display program that can used directly with this scope and automatically provides average readings for frequency, peak-to-peak voltage, and most importantly the time for its high-pulse. See **Figure 5** for a snapshot of this program when measuring the PPG read time. This experiment showed that, per function call, it took 0.1ms and 2.5ms to read ECG data via SPI and PPG data via I2C, respectively.



Ave=0.77V, Peak-peak=2.96V, Period=10.1ms, Freq= 99Hz
high-pulse=2.5ms, low-pulse=7.5ms

Figure 5 – PPG Data Read Time Captured by MSP’s Scope and TExaS Display

3.4. Algorithm

The Lullaby algorithm was originally written with MATLAB. Since this algorithm needs to run on the microcontroller, MATLAB Coder was used to convert the code to its C equivalent. The main issue to overcome with this was to reduce the amount of variables declared in the program, as this was using up more RAM than the microcontroller would be able to handle. Because of this, the algorithm code was split up into 4 different functions before conversion. All functions and outputted C code were then stored in 2 different libraries: the "1 second Lullaby" library, which extracts the maternal and fetal signals from the combined, abdominal ECG (aECG) signal, and the "4 second Lullaby" library, which performs fQRS extraction.

Timing tests were conducted with the primary functions for each library; similar to the tests for reading sensor data, an oscilloscope was connected to a GPIO pin where the signal's high time is the function execution time. In each test case, the function ran 5 times and the average and standard deviation of the execution times were calculated. The experiment was done without any data collection to isolate the performance of just the function, and also with data collection to analyze its real-time capabilities during expected use. Several different types of data were also provided as input for the functions: a "blank signal" (where all values were 0), an ECG signal obtained from a male subject's wrists via electrodes (i.e., without a fetal signal), and a simulated maternal-and-fetal ECG (mfECG) signal. For example, when focusing on non-fetal signals with data collection present, the extraction of maternal and fetal signals took 97.8±2.7 ms and running Lullaby took about 2772.4±71.8 ms. **Table 5** shows a summary of the findings from this experiment.

Table 5 – Execution Times for Algorithm Functions

Algorithm Part	**Data Collection?**	**Input Data**	**Avg. Exec. Time [ms]**	
Separate Data Into Maternal/Fetal Signals	No	Blank	66.5	
		Non-Fetal	67.9	
		Simulated	72.3	
	Yes	Blank	91.9	± 0.0
		Non-Fetal	97.8	± 2.7
		Simulated	97.2	± 2.3
Template Subtraction & fQRS Extraction	No	Blank	28.9	
		Non-Fetal	1844.2	
		Simulated	2303.9	
	Yes	Blank	41.2	± 0.0
		Non-Fetal	2772.4	± 71.8
		Simulated	2779.8	± 86.1

Several important parameters regarding the algorithm were adjusted due to the current system configuration. First, the portion of the algorithm that will choose the best data between multiple ECG channels is removed, since only 1 channel from the ECG channel is used for data at this time. Also, the sampling rate for data being used in the algorithm has been adjusted from 1000 SPS to 250 SPS. If ECG data is read every 4ms, it would not conflict with the long 2.5ms needed to read PPG data. Recall that the data is read in blocking mode, which will pause other program execution until the data has been fully read. Should ECG data be read even as little as two times faster (every 2ms), then the microcontroller will be too preoccupied with reading PPG data to handle the ECG interrupt signal in time. While this would reduce the amount of data collected, 250 SPS is a standard rate when dealing with ECG signals that has been shown to still provide enough data points for time domain analysis [23]. Finally, at the end of running Lullaby, the amount of time that the 4-second window is shifted has been adjusted from 1 second to 4 seconds. From the timing tests done in **Table 5** with the Lullaby algorithm and simulated mfECG data, it takes about 2.8 seconds to finish processing each 4-second window. If left with a 1-second shift, the amount of data waiting to be processed will quickly accumulate and exceed the microcontroller's memory capacity. In order to prevent such an overflow and ensure a real time performance of the algorithm, a larger shift is given with ample time for the algorithm to run and for 1-second segments' maternal and fetal signals extracted.

CHAPTER 4 – RESULTS AND ANALYSIS

4.1. ECG Sensor Output with Simulated ECG

Two primary datasets were used as the simulated mfECG signal collected from the abdomen. The first comes from the Database for the Identification of Systems (DaISy). The dataset consists of 10 seconds of data collected at 250 SPS from 8 different channels, of which 5 of them were recorded from the subject's abdominal area [24]. Channel output column 1 was selected for use in this project. The second one is sourced from PhysioNet, under the Abdominal and Direct Fetal ECG Database [25]. The dataset from each subject consists of 5 minutes of data collected at 1000 SPS from 6 different channels, of which 4 of them were recorded from the subject's abdominal area and 1 from the fetal scalp (a typical location for obtaining accurate fetal ECG [26]). One of the abdominal ECG signals from subject 1 (see the "r01" files) was selected to be used in the simulator.

The simulator device is adapted from a previous project led by the thesis committee chair’s research group that had outputted zebrafish ECG signals. An Arduino Mega 2560 board is hardcoded with values for several periods of the signal, which are outputted repeatedly in a loop. Those values are then sent via SPI to the MCP4921 digital-to-analog converter (DAC), which can output voltages up to the 5V that it's provided; however, resistors are installed to adjust the voltages to the range expected from human ECG signals.

Basic processing is done offline to ensure the data is suitable for the simulator's code. First, a specific portion of the original signal is selected manually. Obtaining an integer number of periods of the signal is critical; this includes not just the maternal signal's period but also the fetal signal's. In addition, to prevent an abnormal transition between the end of a previous loop and beginning of a next loop, the start and end values of the portion should be approximately equal. If applicable, the signal is inverted to orient the ECG signal in the standard direction. Also if applicable, any outlier points are removed and replaced with the value from the previous data point. For example, the main outliers with the PhysioNet dataset were the ones at the 1 minute mark, 2 minute mark, etc.; this is most likely related to a timer from the system used when capturing this data that affected the ECG signal. Since the DAC will only accept non-negative values, the signal is first moved to the non-negative domain, by finding the minimum value from the data and shifting all data points by this minimum. The new minimum (being 0) and maximum are calculated again and used to find out how much to scale up/down the signal by in order to remap the signal to a range between 0 and 4095, the largest value that the 12-bit DAC can accept. Finally, these numbers are rounded to the nearest integer, which makes the data ready to be stored on the simulator. **Figure 6** shows a summary detailing this process, using a portion of the data from the DaISy set.

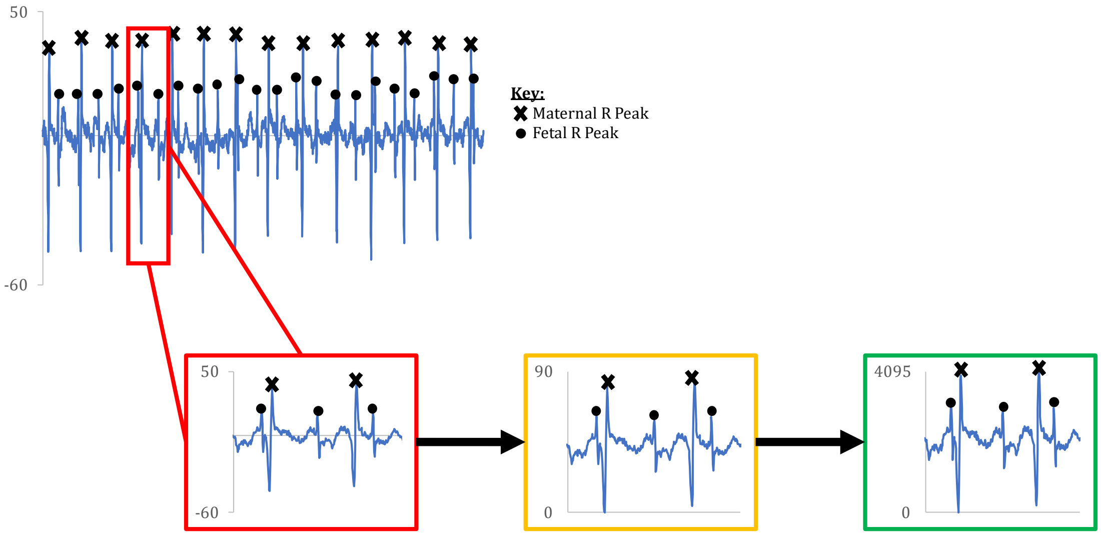


Figure 6 – Offline Processing of DaISy Abdominal ECG Data for Simulator Device

Since the expected signal's data is already known, a comparison can be done between the simulator's outputted signal (the one based off the original dataset, but remapped for use in the Arduino) and the signal captured by the ECG sensor (converted to voltages) to show how well the signal was retained. This can be done with a standard calculation such as cross-correlation [27], where the resulting coefficient can be between -1 (meaning the signals are opposite) and 1 (meaning the signals are identical); 0 represents no relationship between the 2 signals. As shown in Equation (1), one n-sized set of data (containing some arbitrary values x that average out to x') and another n-sized set of data (containing some arbitrary values y inside that average out to y') are needed as part of calculating the coefficient c. In preparation for this calculation, however, the signals do need further processing as well. If necessary, both signals need to be provided at the same sampling rate so that they both share data sizes of n. For example, the PhysioNet data needs to be downsampled from 1000 SPS to 250 SPS, which is simply done by keeping every 4th data point. Once sampling rates for both signals match up, the data just needs to be normalized or, in other words, remapped to a range from 0 to 1.

$$c = \frac{\sum_{i=1}^{n}(x - x')(y - y')}{\sqrt{\sum_{i=1}^{n}(x - x')^2 * \sum_{i=1}^{n}(y - y')^2}} \tag{1}$$

This process was applied with both the DaISy and PhysioNet datasets, where each dataset was tested 3 times non-consecutively; an average and standard deviation for the cross-correlation coefficient were calculated using these 3 attempts. For the DaISy set, sampled at a rate equal to the ECG sensor's, the coefficient was 0.987±0.009, which is close to 1 as expected. Now for the PhysioNet data, there was some concern that its higher sampling rate would cause a lower coefficient; if combined with a very elevated maternal or fetal heart rate, there would generally be less data points to represent peaks which is less chance for the sensor to sample enough of the peaks. But with this particular simulated signal displaying an average of 84 bpm for the mother (standard even during pregnancy [28]) and 127 bpm for the fetus (also within the standard range [29]), this should not be an issue. As such, the results presented a high coefficient of 0.985±0.002. These results help confirm that the signal is well retained from its source to the sensor; different criteria such as electrode motion artifacts [30] and skin-electrode contact impedance [31] would normally factor into a cross-correlation such as this, but are irrelevant for this particular experiment which does not directly involve human subjects at this time.

4.2. Algorithm Performance with Simulated ECG

Taking the next step through the system, the simulated data from the DaISy and PhysioNet datasets are passed as input for the Lullaby algorithm to identify fQRS indices. Each dataset was tested 3 times non-consecutively, each time with a single 4-second window of data. The performance will be determined naturally by whether the indices correspond to actual fQRS segments. In addition, because it is the main application of the algorithm, the fetal heart rate will be calculated using the indices and compared with the expected rate by checking the percent difference from the expected rate. To provide context, heart rates are calculated by finding the differences in time between each fQRS (via the data/graph), calculating the average difference, and deriving the heart rate in units beats per minute. The DaISy data shows 6 maternal peaks every 0.68 seconds and 9 fetal peaks every 0.45 seconds, corresponding to 88.132 bpm and 132.450 bpm respectively. The PhysioNet data also shows the same number of maternal and fetal peaks, but every

0.71 seconds and 0.47 seconds respectively; this is computed to an 84.746 bpm maternal heart rate and 126.850 bpm fetal heart rate. **Table 6** provides the preliminary data from these experiments.

Table 6 – Detected fQRS's and Calculated Fetal BPM from Lullaby

Dataset	Attempt	Num. fQRS	Fetal BPM
DaISy	Expected	9	132.45
DaISy	1	4	87.72
DaISy	2	1	N/A
DaISy	3	4	88.41
PhysioNet	Expected	9	126.85
PhysioNet	1	5	119.36
PhysioNet	2	4	88.41
PhysioNet	3	2	107.14

The DaISy data was tested first and, while it generally did not perform well, was processed by Lullaby in a surprising manner. Attempt 2 was only able to detect 1 fQRS in a stagnant portion of the signal i.e., not on any significant segment of the maternal or fetal ECG. The index does point to a local maximum that can only be seen in the combined signal, but not in either the maternal or fetal signal. One guess for such an abnormal fQRS was that the point was near the start or finish of the signal's looped portion, where the transition between loops could have affected the algorithm more than anticipated. However, the proposed fQRS index is closer to the middle of a loop so this guess is not likely. Attempts 1 and 3 were each able to detect 4 fQRS segments; while one or two of the proposed fQRS indices did mark a point close to a fetal peak, they altogether led to a calculated fetal heart rate of about 88.4 bpm (-33.7% difference from expected). However, upon closer inspection, it appeared that the detected fQRS's were consistently nearby mQRS's; each fQRS was about 150±14 ms before an mQRS; **Figure 7** shows this relationship with the results from Attempt 1. In addition, the calculated fetal heart rate from Attempt 1 and 3 only had -0.47% and +0.31% difference from the expected maternal heart rate, respectively. Despite the very close-to-ideal ECG signal and clear visual identifiers for fetal peaks that this particular dataset provided, the algorithm appeared to only detect the much more distinct maternal peaks during its computation. While impressive in its own right with calculating the maternal heart rate, it skews from the main goal of this project to analyze fetal activity.

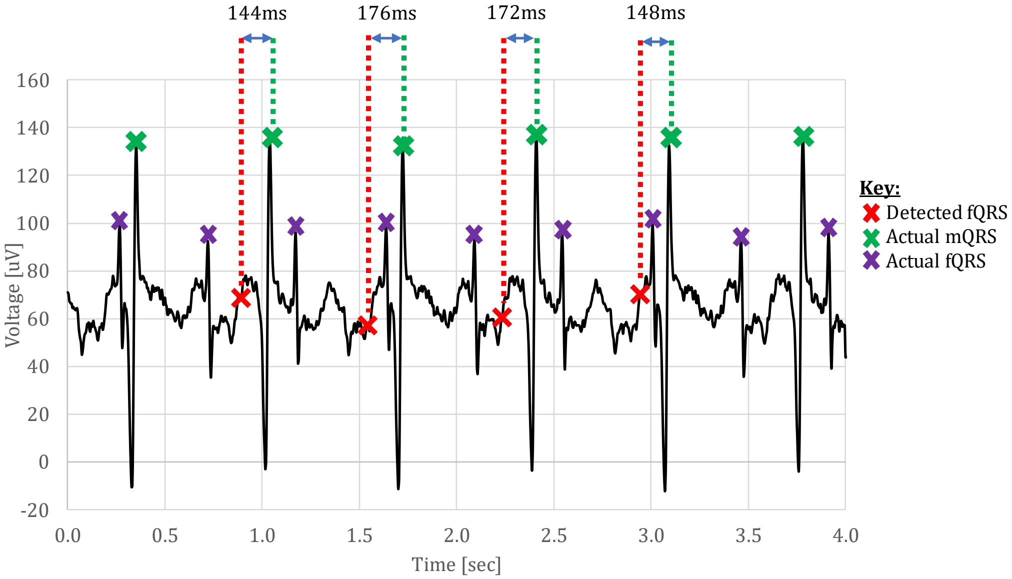


Figure 7 – Correlation Between Lullaby-Detected fQRS & Actual mQRS with DaISy Data

The PhysioNet dataset's performance with the Lullaby algorithm shows more promising results than the DaISy set, though misses enough fetal peaks to provide accurate results. Attempt 3 detects only 2 fQRS's, with only one of them being only 44ms away from the actual fQRS. Out of all the detected fQRS from using the DaISy data, which was an average of 140±38 ms away from the actual peak, this PhysioNet point has pinpointed the fetal peak the best thus far. Similar to the impressive (maternal) heart rate calculated from a relatively ideal signal, it does come as a surprise of better fetal peak detection despite more noise and instances of near-overlapping mQRS and fQRS. The other attempts confirm this as well, with Attempts 1 and 2 showing most of its fQRS's totaling an average of 53±17ms before the actual fQRS; **Figure 8** shows Attempt 1's results of this accurate pinpointing of fQRS's. However, the calculated heart rates from the 3 attempts vary highly, from Attempt 2's calculated 88.4 bpm (-30.3% difference from expected) to Attempt 1's calculated 119.4 bpm (-5.9% difference from expected). One possible explanation is that there are too little fQRS's detected by the algorithm. Especially with only a 4 second window to work with, the calculated heart rate can vary widely depending on how many data points there are to use in the calculation; in addition, the more consecutive fetal peaks that are extracted, the more accurate the average will be.

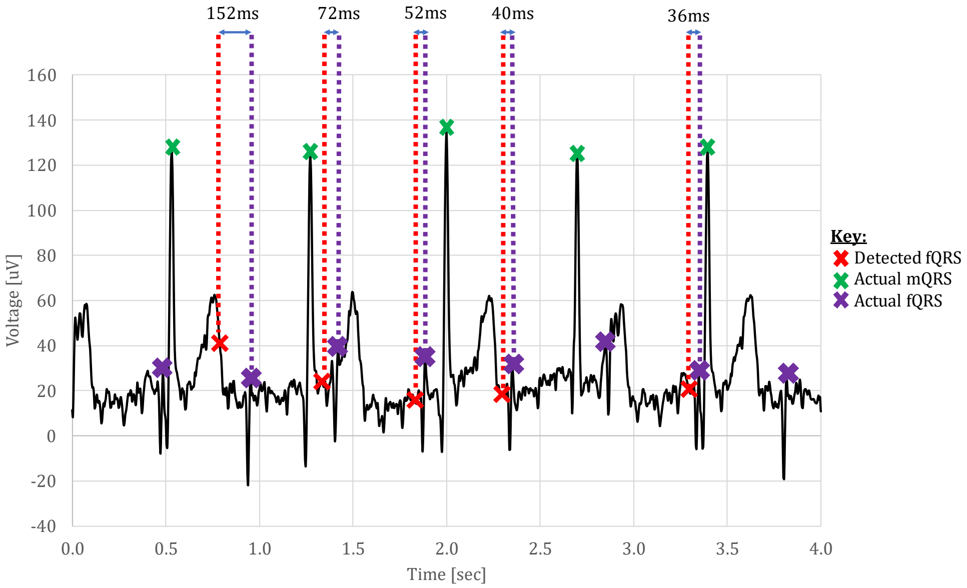


Figure 8 – fQRS Time Indices with PhysioNet Data, Lullaby vs. Actual

To further improve the performance of Lullaby, improving the computation time of the algorithm such that it can run with a 1-second shift as intended is vital. Recall that, due to the execution time of the algorithm in the system being close to 3 seconds, the need for a 4-second shift was necessary to avoid increasing lag in results. The same PhysioNet data from previous experiments were used to show the differences detecting fQRS's and calculating heart rate between a shift of 1 second and one of 4 seconds. First, 8 seconds (double the usual 4-second window) of the combined signal was collected via the current system. Then separating the maternal from the fetal signal and running the algorithm on windows t = 0-4, 1-5, 2-6, and 3-7 were done in separate projects that solely performed these actions in order to remove any potential issues with data collection, peripheral setup, etc. Any repeated indices from the 1-second shift version of the algorithm, and those as close as roughly 50ms away from already-discovered indices, were considered duplicate measurements and hence removed. The result is shown in **Figure 9**. With the 4-second shift, 4 fQRS's were detected, averaged about 527±51ms apart, and computed a fetal heart rate of 113.9 bpm (-10.6% difference from expected). With the 1-second shift, 3 more fQRS's were detected in the last second of the window and altogether averaged about 460±111ms apart; they computed the most accurate fetal heart rate of all other experiments at 130.4 bpm (+2.3% difference from expected). With the additional peaks that Lullaby is innately able to capture, heart rate calculations will naturally improve.

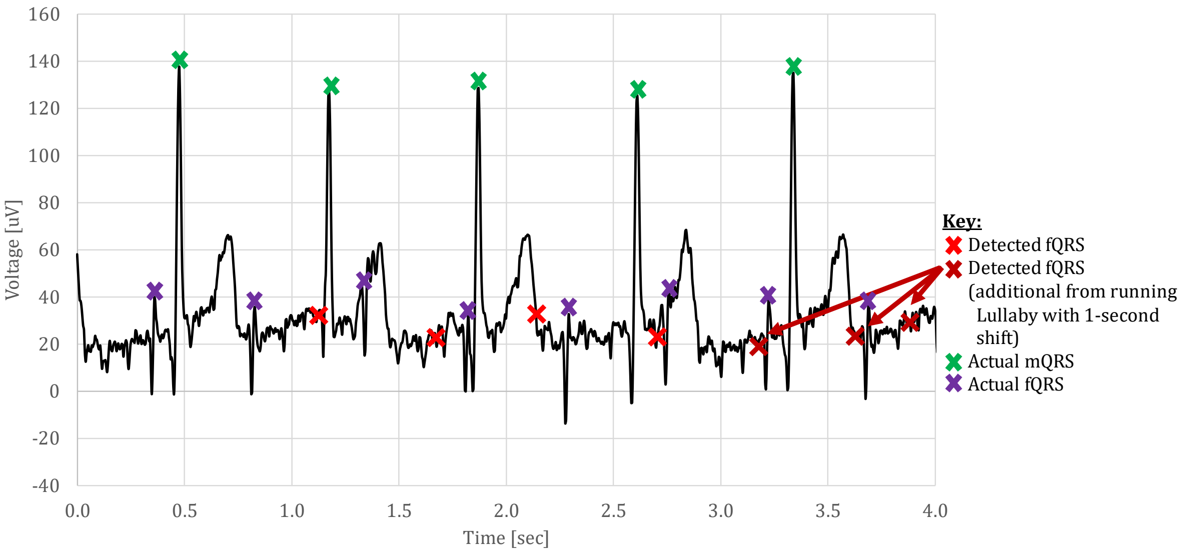
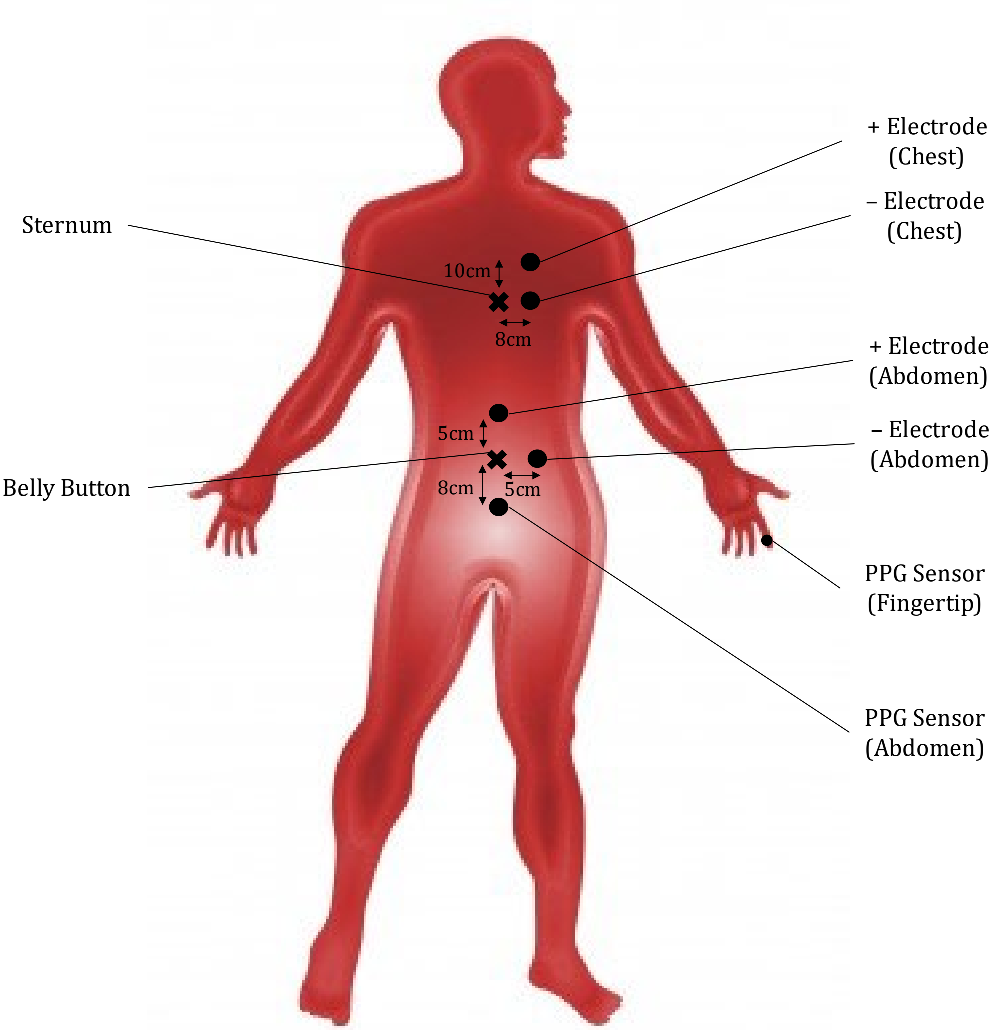


Figure 9 – Improved fQRS Detection with Lullaby using 1-Second Shift

4.3. PPG-Assisted ECG QRS Detection

Using the same fetal system, we specifically test its PPG sensor's performance when both PPG and ECG data were being collected simultaneously. However, the tests are for testing its feasibility specifically when data was collected from any human's abdominal area; at this time, obtaining mfECG data from pregnant subjects was not possible due to a need for official clinical trials first. 2 subjects volunteered to have their data collected, both male and in their early 20's, from 4 different sets of sensor and electrode locations, with 3 attempts per set. These sets are from testing all different possibilities of 2 electrode locations and 2 PPG sensor locations. Wet electrodes (3M Red Dot Electrodes 2560), most commonly used for measuring ECG and other human biopotentials [32][2], are placed either at the left pectoral muscle (for its proximity to the heart where the source of such signals arises from) or the abdominal area. The PPG sensor is placed either at the tip of an index finger or the lower abdominal area (due to the naturally thinner skin compared to other parts of the abdomen). When at the abdomen, the PPG sensor did require more pressure applied onto it by the subject in order to get at least some periodic signal capable of extracting heart rate from. Data is collected from areas such as the chest and fingertip because this is where signal strength is highest and most accurate for heart rate monitoring; it also sets a standard for adequate comparison when later examining abdominal data. **Figure 10** shows general locations for the electrodes and PPG sensor; note that the distances from the sternum and belly button are vertical lengths, not the length across the contours of the body. **Table 7** summarizes where data is being measured from in each set.

[2] Reproduced from book chapter, with permission from Springer Nature and cited here as [32]



"Human Anatomy" base image courtesy of vectorile at FreeDigitalPhotos.net

Figure 10 – Anatomical Locations for Placing Electrodes and PPG Sensor

Table 7 – Set Numbering for Sensor Locations during PPG & ECG Acquisition

Set Num.	**ECG Sensor (Electrodes)**	**PPG Sensor**
1	Chest	Fingertip
2	Chest	Abdomen
3	Abdomen	Fingertip
4	Abdomen	Abdomen

First, ECG data is compared between the heart and abdominal areas. The chest ECG signals also show QRS peaks being approximately 2mV tall. As expected, the aECG's amplitude is smaller, showing peaks around 500 μV tall (4 times smaller); see **Figure 11** for this comparison with the chest ECG. Despite this, the abdominal signal is still clear and its major segments are still discernable by the human eye. Recall that the simulator's signal showed ECG peaks with 150 μV amplitudes that could be detected from previous experiments. Thus, the system would most likely have no issues with signal strength when used for its application. The readings from the heart also show inverted T waves that the same subject's abdominal readings do not show. Heart readings such as this have been known to point towards signs of cardiomyopathy [33], but considering no official diagnosis of this condition in any of the subjects and the unorthodox placements of the electrodes, this inversion is disregarded for the remainder of this discussion.

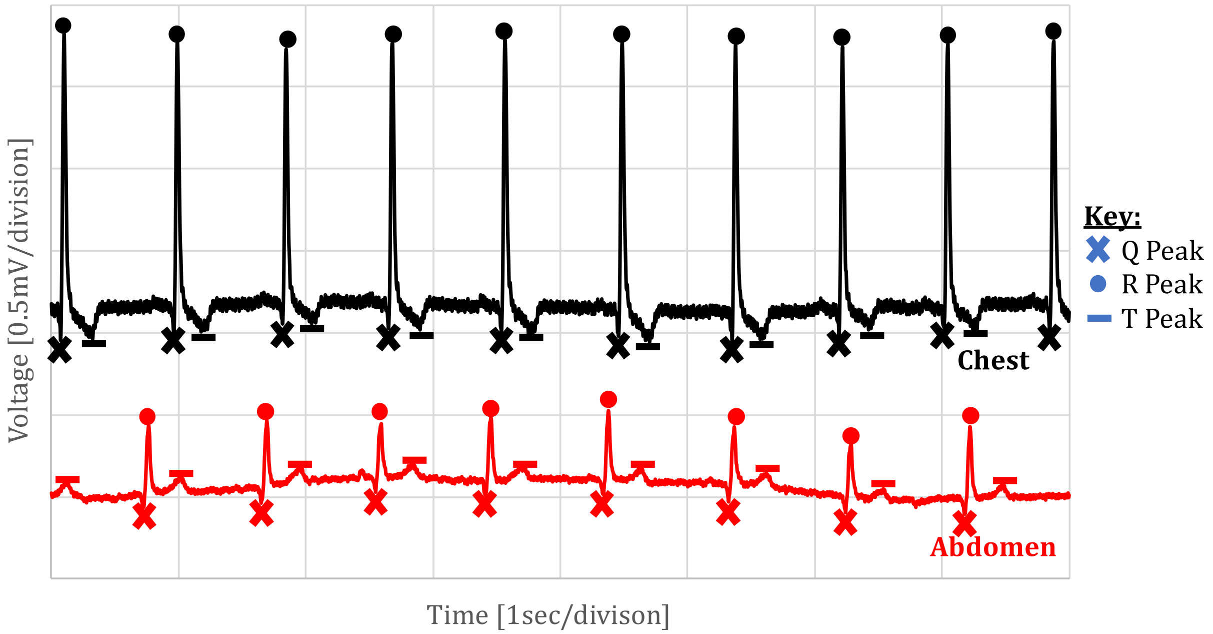


Figure 11 – Non-Pregnant Subjects' ECG Signals from System, Chest vs. Abdomen

Next, PPG data is compared between what was gathered at the fingertip and at the abdomen. When using the most common values from all fingertip measurements' attempts, it appears that the typical amplitudes for the signal retrieved are 2.5 mV or 5 mV as shown in **Figure 12**. However, the signal strength is not quite consistent; for example, the IR LED's data has recorded voltages as low as 1 mV and as high as 20 mV. Data regarding this inconsistency has been tabularized in **Table 8** and graphed in **Figure 13**. Fingertip measurements were able to show the major points of a typical PPG signal very clearly, especially from readings captured during the red and IR LEDs' activation. However, only the green LED's data showed the usual shape; the other data showed what appears to be an inverted signal. This is most likely due to the PPG sensor's configuration in 3-LED mode where only the green LED's data underwent ambient subtraction. But for most of the discussion, the graphs for the red and IR light will have their inversions undone to orient them correctly; only the shape and scale of the PPG signal (not the exact voltage values) will be relevant. Now when examining the abdominal PPG, the voltage readings from response to green, red, and IR LEDs have amplitudes of 1 mV, 0.2 mV, and 0.5 mV respectively; see **Figure 14** for one example of these readings. While these do not fluctuate greatly across different attempts, there is roughly a 5x reduction in the green LED signal and 10x reduction in the red and IR LED signals compared to the fingertip signals (see **Figure 15**). In addition, signal quality degrades severely at the abdomen (especially with the green LED one), where identifying the dicrotic notches and diastolic peaks is most likely impossible. Several attempts also show a highly fluctuating mean, which is most likely caused by motion artifacts and exacerbated by the small amplitudes of the PPG. As can be seen in **Figure 14**, the green LED's signal is especially degraded in the first 4 seconds; the times for pulse onsets and systolic peaks during this time frame had to be roughly estimated. But there is a consistent ability for the IR LED's light to return signals that can very easily target the pulse onsets and systolic peaks, which would be plenty to help monitor heart rate in this system.

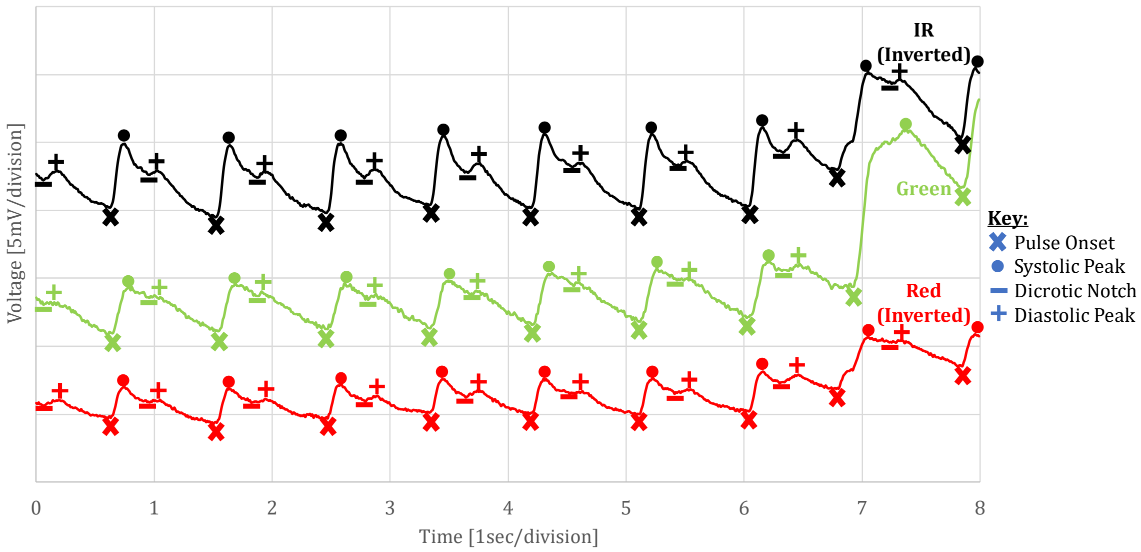


Figure 12 – Non-Pregnant Subjects' PPG Signals from System, Fingertip Only

Table 8 – Count for Various Fingertip PPG Amplitudes during PPG & ECG Collection

LED	Signal Amplitude [mV]	Num. Attempts
Green (w/ Ambient Subtraction)	**2.5**	**4**
	5.0	**4**
	7.5	2
	10.0	1
	12.5	1
Red	1.0	2
	2.5	**5**
	5.0	1
	10.0	3
	15.0	1
IR	1.0	2
	5.0	**5**
	10.0	4
	20.0	1

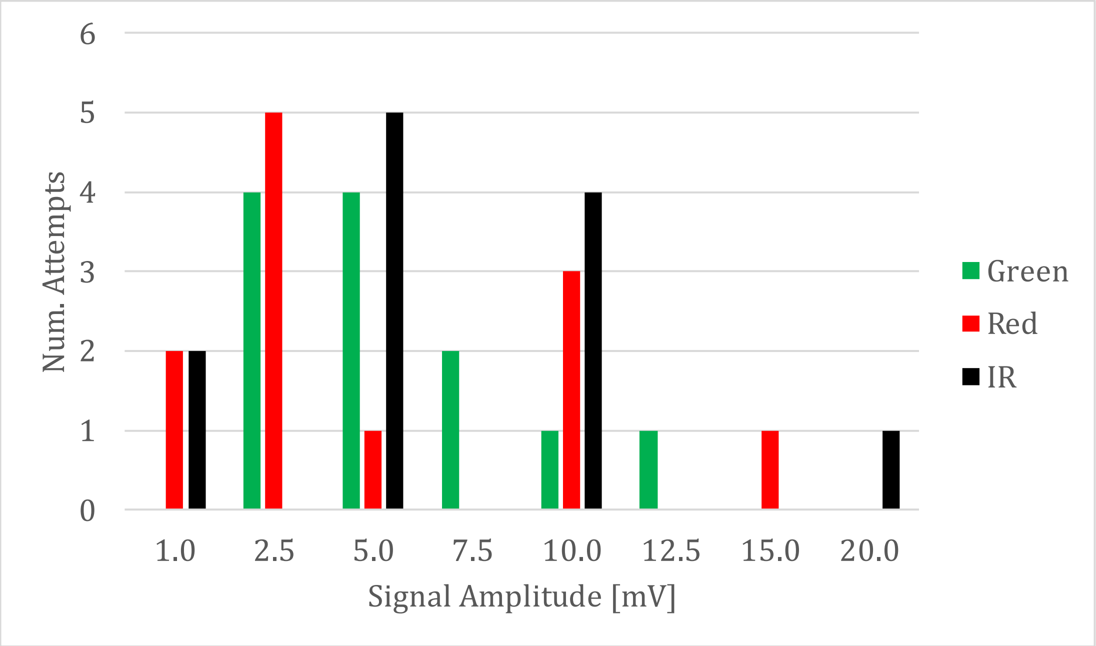


Figure 13 – Graph for Various Fingertip PPG Amplitudes during PPG & ECG Collection

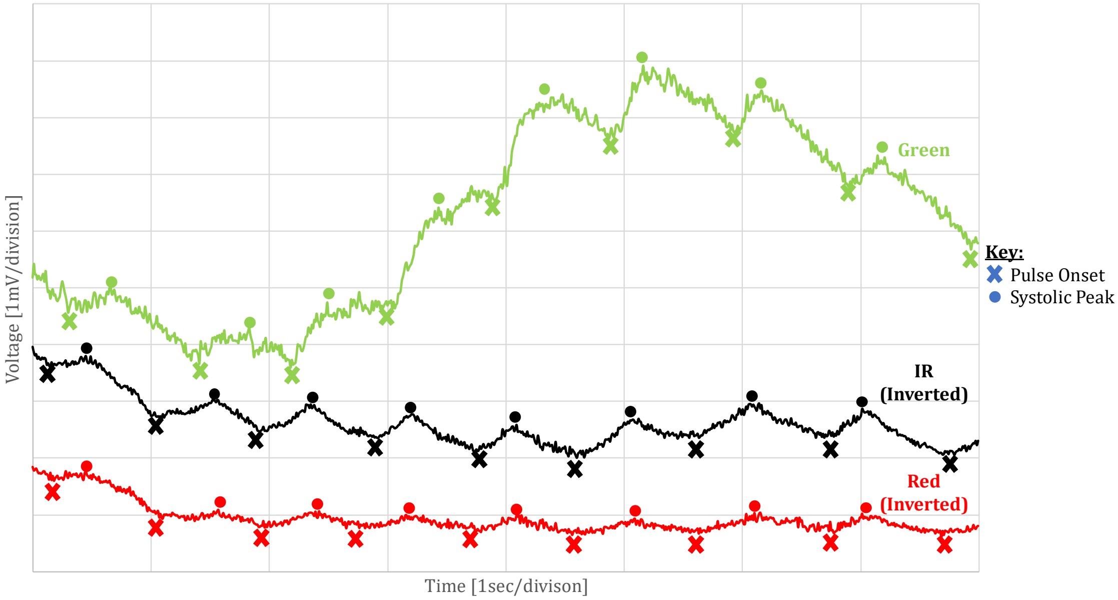


Figure 14 – Non-Pregnant Subjects' PPG Signals from System, Abdomen Only

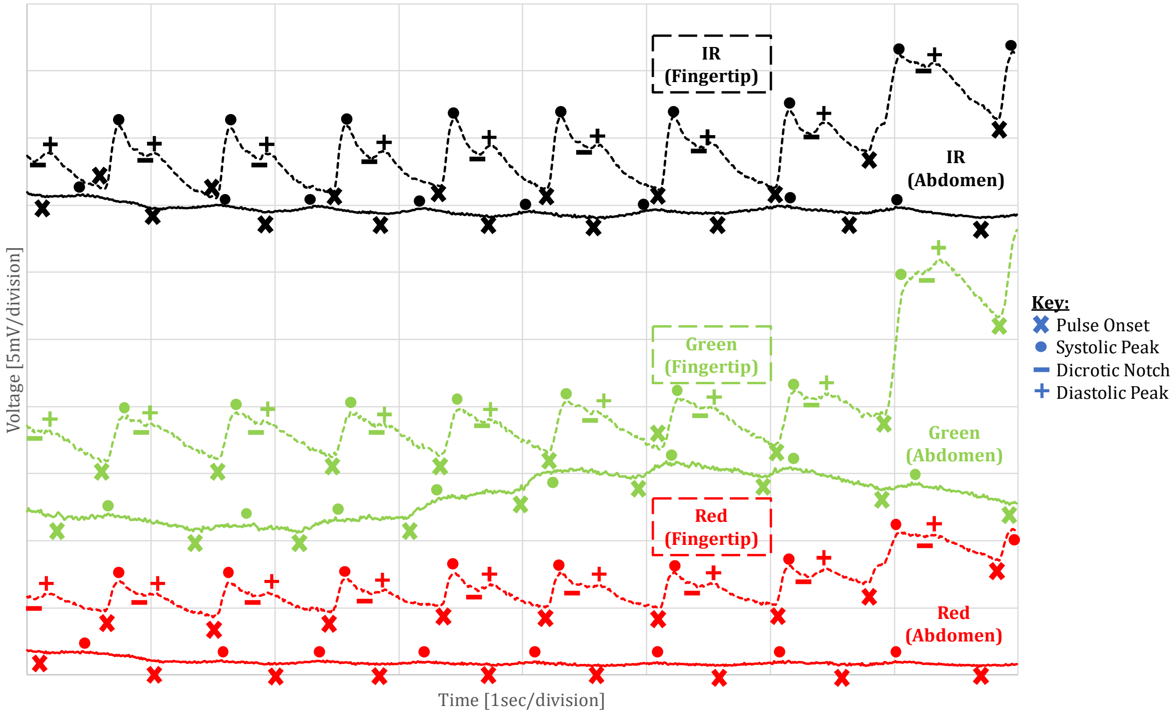


Figure 15 – Non-Pregnant Subjects' PPG Signals from System, Fingertip vs. Abdomen

Preliminary analysis is done to see how well the calculated heart rates from the abdominal region compare between ECG and PPG. The IR signal had the clearest indicators for peaks that could be used for finding heart rate, so only that was compared with the ECG signal. Looking at an 8-second window of data, and using the time between pulse onsets and time between systolic peaks, heart rates of 63.41 bpm and 63.93 bpm respectively were generated, both with less than 1% absolute percent difference from the ECG's calculated 63.63 bpm. Such values are excellent indicators for the potential ability of PPG to greatly assist in this particular system.

Some more analysis brings about specific relationships between ECG and PPG at the abdomen that may be beneficial. Given one specific attempt with a subject, the Q, R, and T waves' peaks from the ECG signal along with the pulse onsets and systolic peaks from the

PPG signal have their time indices marked. For each of the ECG peaks marked, the list of PPG points are compared to see what the average times between an ECG peak and PPG point are. As is shown from **Table 9** (highlighted cells indicate the smallest value in the column for the set of rows regarding data for one particular ECG peak), the ECG Q-peak lines up best with the PPG's pulse onset, as does the ECG T-peak with the PPG's systolic peak. Using the pulse onset to coordinate time indices for R-peaks (and possibly fQRS's) may certainly be possible as well. In addition to having the clearest PPG signal out of all other LEDs' signals, the IR data is generally closer in time to significant ECG segments and peaks; see **Figure 16** for a graphical representation of this. An initial hypothesis was that this was related to the timing control for the PPG sensor's configuration, where the time to sample from each LED was determined. However, the current configuration separates each sampling phase by roughly 400 periods of the 4 MHz timing engine, which is 100 μs; this differs greatly from the abdominally captured data which shows differences in the order of tens of milliseconds. Instead, this feature could perhaps be attributed to the wavelength of color for the LED; for most of the PPG points shown in **Table 9**, there is an indirect correlation showing smaller time difference associated with larger wavelengths.

Table 9 – Tabulated Time Difference Between ECG and PPG Points w/ Abdominal Data

ECG Peak	**PPG Point/LED**	**Time Difference [ms]**	
		Avg.	Std. Dev.
Q	Pulse-Onset/Green	165.56	64.31
Q	Pulse-Onset/Red	93.56	52.62
Q	Pulse-Onset/IR	66.22	43.52
Q	Systolic/Green	396.50	113.10
Q	Systolic/Red	344.25	66.85
Q	Systolic/IR	330.57	40.76
R	Pulse-Onset/Green	121.56	64.36
R	Pulse-Onset/Red	124.22	45.71
R	Pulse-Onset/IR	101.78	53.38
R	Systolic/Green	396.50	113.10
R	Systolic/Red	300.25	67.41
R	Systolic/IR	286.57	41.34
T	Pulse-Onset/Green	118.25	56.50
T	Pulse-Onset/Red	331.25	104.77
T	Pulse-Onset/IR	347.50	62.59
T	Systolic/Green	150.00	113.76
T	Systolic/Red	79.25	29.78
T	Systolic/IR	47.00	35.86

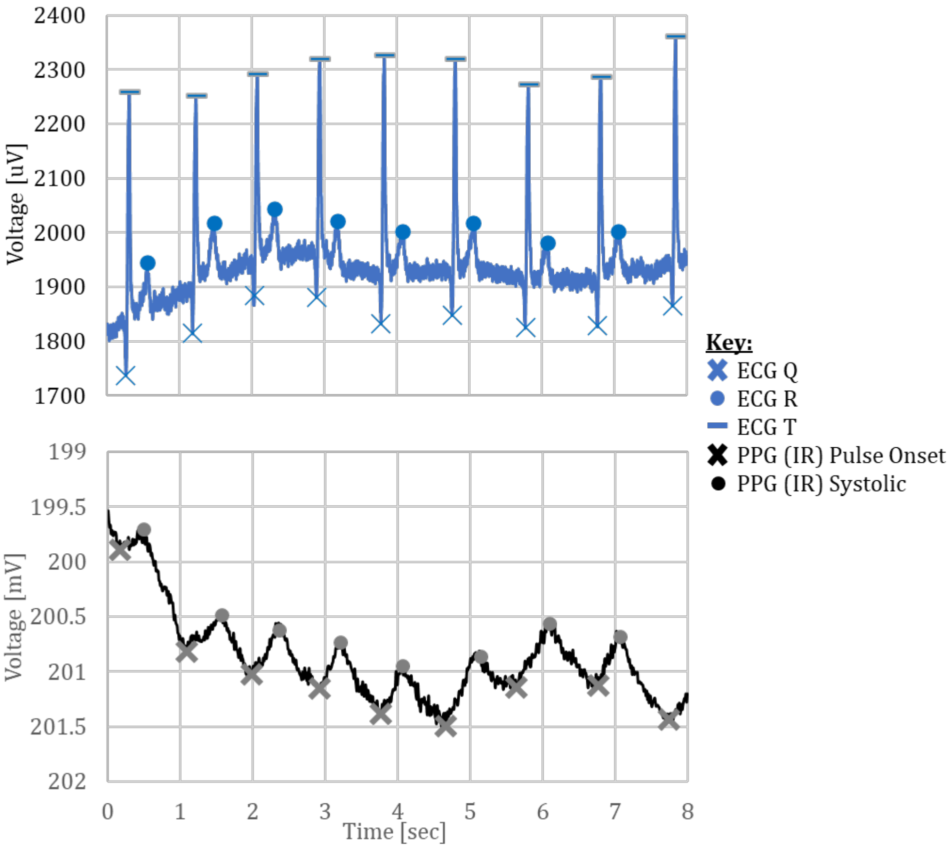


Figure 16 – Graphed Time Difference Between ECG and PPG Points w/ Abdominal Data

CHAPTER 5 – CONCLUSION

5.1. Summary

The hardware selection for this system lines up closely with the one for the adapted project's system, but there are a few key changes that were made to suit this project's specific needs. Most notably, there was the inclusion of the Heart Rate 5 Click to capture PPG data while being configured to utilize the IR LED, operate with minimum active time, and output data at 100 SPS. Then the tasks of the BLE module were split between simple transmission of data (still done at the module) and the remaining data collection and processing (now done at the more powerful STM controller). With the specific ST board selected, the 180 MHz max clock speed and 2 MB flash memory attempt to provide the best computation speed for Lullaby to run. At the same time, blocking functions for its communication with the sensors run well when capturing ECG data, but slow enough with PPG data to currently keep the ECG sensor at 250 SPS. It has been proven as well to be practical enough for signal capture, showing a 0.98 out of 1 cross-correlation coefficient when using simulated signals.

Implementing Lullaby with this system has produced unexpected results across all experiments. When testing the timing of the fQRS extraction functions, Lullaby needed about 2.8 seconds to finish processing a 4-second window; this has not only required a 4-second shift with what window of data is processed, but also resulted in fewer fQRS's being detected. The DaISy dataset seemed to work opposite what Lullaby intended and quite accurately calculated maternal, not fetal, heart rate. It's from the PhysioNet dataset where the issues of the algorithm performance start to arise; several fetal peaks can be detected with acceptable accuracy but not enough to find the fetal heart rate with as much confidence.

The benefits of including the PPG sensor in the system certainly revealed themselves, even when collecting data from non-pregnant subjects as a preliminary step. While signal strength and quality certainly degrade compared to when the sensors and electrodes were at more ideal locations, PPG (especially data from the IR LED) and ECG retain enough of their major components that heart rate analysis can still be possible. In addition, the heart rate relationship between PPG and ECG was confirmed strong with this particular experiment as well. Should the PPG data be used to eventually help with identifying mQRS's in the fetal application, there is promise that it would help a significant amount.

5.2. Future Work

The most important improvement that can be done to this system is to implement Lullaby with the 1-second shift as it was meant to originally run. Due to the operations performed in a 3D array as part of the algorithm, vectorizing the code is one way to help parallelize the code enough that could vastly improve performance. Efforts can also be made to confirm that the microcontroller is configured to run at the advertised highest core clock speed. This may also carry over to other components or peripherals in the microcontroller that are limiting the performance, such as the communication functions with sensors that are currently blocking.

Knowing the helpfulness of PPG in a fetal monitoring system such as this, there can generally be more research and experimentation done with the sensor's settings. With only several subjects who were involved in the PPG experiments, there can certainly be more and with various criteria like skin tone [34] and sensor/electrode locations. In addition, to bring the device closer to a one-size-fits-all product, it may be best if the sensor's settings were adjusted dynamically. Even Texas Instruments, the developers of the PPG sensor's main IC, recommends calibrations and adjustments on important variables labelled "AFE signal chain settings" such as TIA gain and LED current [15].

REFERENCES

[1] F. T. Ellington, B. Demirel, D. Jilani, M. A. Al Faruque and H. Cao, "Edge-based Real-time Fetal Electrocardiography Monitoring in the Home Setting," in *Computing in Cardiology*, Tampere, 2022.

[2] S. Sarafan, T. Le, F. Ellington, Z. Zhang, M. P. Lau, T. Ghirmai, A. Hameed and H. Cao, "Development of a Home-based Fetal Electrocardiogram (ECG) Monitoring System," in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society*, 2021.

[3] D. Jilani, T. Le, T. Etchells, M. P. Lau and H. Cao, "Lullaby - A Novel Algorithm to Extract Fetal QRS in Real Time Using Periodic Trend Feature," *IEEE Sensors,* vol. 6, no. 9, pp. 1-4, 2022.

[4] E. A. Ashley and J. Niebauer, "Conquering the ECG," in *Cardiology Explained*, London, Remedica, 2004.

[5] C. Lempersz, J. O. van Laar, S. B. Clur, K. M. Verdurmen, G. J. Warmerdam, J. van der Post, N. A. Blom, T. Delhaas, S. G. Oei and R. Vullings, "The standardized 12-lead fetal electrocardiogram of the healthy fetus in mid-pregnancy: A cross-sectional study," *PLoS One,* vol. 15, no. 4, 2020.

[6] A. Matonia, J. Jezewski, T. Kupka, M. Jezewski, K. Horoba, J. Wrobel, R. Czabanski and R. Kahankowa, "Fetal electrocardiograms, direct and abdominal with reference heartbeat annotations," *Scientific Data,* vol. 7, 2020.

[7] R. Kleiman, B. Darpo, R. Brown, T. Rudo, S. Chamoun, D. E. Albert, J. M. Bos and M. J. Ackerman, "Comparison of electrocardiograms (ECG) waveforms and centralized ECG measurements between a simple 6-lead mobile ECG device and a standard 12-lead ECG," *Ann Noninvasive Electrocardiol,* vol. 26, no. 6, 2021.

[8] N. Selvaraj, A. K. Jaryal, J. Santhosh and K. K. Deepak, "Assessment of heart rate variability derived from finger-tip photoplethysmography as compared to electrocardiography," *Journal of Medical Engineering & Technology,* vol. 32, no. 6, pp. 479-484, 2008.

[9] K. B. Gan, E. Zahedi and M. A. M. Ali, "Application of Adaptive Noise Cancellation in Transabdominal Fetal Heart Rate Detection Using Photoplethysmography," in *Adaptive Filtering Applications*, 2011, pp. 141-156.

[10] J. Park, H. S. Seok, S.-S. Kim and H. Shin, "Photoplethysmogram Analysis and

Applications: An Integrative Review," *Frontiers in Physiology,* vol. 12, 2022.

[11] Texas Instruments, "ADS1299-x Low-Noise, 4-, 6-, 8-Channel, 24-Bit, Analog-to-Digital Converter for EEG and Biopotential Measurements (Rev. C)," [Online]. Available: https://www.ti.com/lit/ds/symlink/ads1299.pdf. [Accessed 1 March 2023].

[12] Analog Devices, "AD8233 (Rev. D)," [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/ad8233.pdf. [Accessed 1 March 2023].

[13] MikroElektronika, "Heart Rate 5 click," [Online]. Available: https://www.mikroe.com/heart-rate-5-click. [Accessed 1 March 2023].

[14] M. A. Almarshad, M. S. Islam, S. Al-Ahmadi and A. S. BaHammam, "Diagnostic Features and Potential Applications of PPG Signal in Healthcare: A Systematic Review," *Healthcare (Basel),* vol. 10, no. 3, p. 547, 2022.

[15] Texas Instruments, "AFE4404 Ultra-Small, Integrated AFE for Wearable, Optical, Heart-Rate Monitoring and Bio-Sensing (Rev. D)," [Online]. Available: https://www.ti.com/product/AFE4404. [Accessed 1 March 2023].

[16] Analog Devices, "MAX30101," [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/MAX30101.pdf. [Accessed 1 March 2023].

[17] MikroElektronika, "Heart Rate 4 Click," MikroElektronika, [Online]. Available: https://www.mikroe.com/heart-rate-4-click. [Accessed 28 February 2023].

[18] ST Electronics, "STM32F429ZI - High-performance advanced line, Arm Cortex-M4 core with DSP and FPU, 2 Mbytes of Flash memory, 180 MHz CPU, ART Accelerator, Chrom-ARTAccelerator, FMC with SDRAM," ST Electronics, [Online]. Available: https://www.st.com/en/microcontrollers-microprocessors/stm32f429zi.html. [Accessed 1 March 2023].

[19] Silicon Labs, "EFM32 Giant Gecko Series 1 32-bit Microcontrollers (MCUs)," Silicon Labs, [Online]. Available: https://www.silabs.com/mcu/32-bit-microcontrollers/efm32-giant-gecko-gg11. [Accessed 1 March 2023].

[20] Nordic Semiconductor, "nRF52832 Product Specification v1.8," Nordic Semiconductor, [Online]. Available: https://infocenter.nordicsemi.com/pdf/nRF52832_PS_v1.8.pdf. [Accessed 1 March 2023].

[21] Silicon Labs, "EFR32BG24 Wireless SoC Family Data Sheet," [Online]. Available: https://www.silabs.com/documents/public/data-sheets/efr32bg24-datasheet.pdf.

[Accessed 1 March 2023].

[22] D. Castaneda, A. Esparza, M. Ghamari, C. Soltanpur and H. Nazeran, "A review on wearable photoplethysmography sensors and their potential future applications in health care," *International Journal of Biosensors & Bioelectronics,* vol. 4, no. 4, pp. 195-202, 2018.

[23] O. Kwon, J. Jeong, H. B. Kim, I. H. Kwon, S. Y. Park, J. E. Kim and Y. Choi, "Electrocardiogram Sampling Frequency Range Acceptable for Heart Rate Variability Analysis," *Electrocardiogram Sampling Frequency Range Acceptable for Heart Rate Variability Analysis,* vol. 24, no. 3, pp. 198-206, 2018.

[24] B. D. Moor, "[96-012] Cutaneous potential recordings of a pregnant woman," DaISy: Database for the Identification of Systems, 14 February 2006. [Online]. Available: https://homes.esat.kuleuven.be/~smc/daisy/daisydata.html. [Accessed 1 March 2023].

[25] A. L. Goldberger, L. A. Amaral, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals," *Circulation,* vol. 101, no. 23, pp. 215-220, 2000.

[26] J. Jezewski, A. Matonia, T. Kupka, D. Roj and R. Czabanski, "Determination of the fetal heart rate from abdominal signals: evaluation of beat-to-beat accuracy in relation to the direct fetal electrocardiogram," *Biomedical Engineering / Biomedizinische Technik,* vol. 57, no. 5, pp. 383-394, 2012.

[27] F. Rundo, S. Conoci, A. Ortis and S. Battiato, "An Advanced Bio-Inspired PhotoPlethysmoGraphy (PPG) and ECG Pattern Recognition System for Medical Assessment," *Sensors (Basel),* vol. 18, no. 2, 2018.

[28] L. Loerup, R. M. Pullon, J. Birks, S. Fleming, L. H. Mackillop, S. Gerry and P. J. Watkinson, "Trends of blood pressure and heart rate in normal pregnancies: a systematic review and meta-analysis," *BMC Medicine,* vol. 17, 2019.

[29] T. Ghi, E. Di Pasquo, A. Dall'Asta, A. Commare, E. Melandri, A. Casciaro, S. Fieni and T. Frusca, "Intrapartum fetal heart rate between 150 and 160 bpm at or after 40 weeks and labor outcome," *Acta Obstetricia et Gynecologica Scandinavica,* vol. 100, no. 3, pp. 548-554, 2021.

[30] M. A. Serhani, H. T. El Kassabi, H. Ismail and A. N. Navaz, "ECG Monitoring Systems: Review, Architecture, Processes, and Key Challenges," *Sensors (Basel),* vol. 20, no. 6, 2020.

[31] K. Goyal, D. A. Borkholder and S. W. Day, "Dependence of Skin-Electrode Contact Impedance on Material and Skin Hydration," *Sensors (Basel),* vol. 22, no. 21, 2022.

[32] T. Le, I. Clark, J. Fortunato, M. Sharma, X. Xu, T. K. Hsiai and H. Cao, "Electrocardiogram: Acquisition and Analysis for Biological Investigations and Health Monitoring," in *Interfacing Bioelectronics and Biomedical Sensing*, Springer, Cham, 2020, pp. 117-142.

[33] F. D'Ascenzi, F. Anselmi, P. E. Adami and A. Pelliccia, "Interpretation of T-wave inversion in physiological and pathological conditions: Current state and future perspectives," *Clin Cardiol,* vol. 43, no. 8, pp. 827-833, 2020.

[34] A. Puranen, "Effects of LED wavelength, intensity and skin tone on the performance of optical beat-to-beat heart rate monitoring," Tampere, 2021.