TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

VALA JEYHANI
A WIRELESS DEVICE FOR AMBULATORY CARDIAC AND RESPIRATORY MONITORING - DESIGN CONSIDERATIONS AND ESSENTIAL PERFORMANCE

Master of Science thesis

Examiner: DSc. (tech) Antti Vehkaoja
Examiner and topic approved by the
Faculty Council of the Faculty of
Electrical Engineering
on 27th September 2017

ABSTRACT

VALA JEYHANI: A Wireless Device for Ambulatory Cardiac and Respiratory Monitoring - Design Considerations and Essential Performance
Tampere University of Technology
Master of Science thesis, 90 pages, 8 Appendix pages
September 20, 2017
Master's Degree Program in Electrical Engineering
Major: Medical Instrumentation
Examiner: DSc. (tech) Antti Vehkaoja
Keywords: wearable, ambulatory, ECG, electrocardiography, respiration, impedance pneumography, accelerometer, gyroscope, Bluetooth, BLE, measurement, monitoring, Android, medical, Holter

In recent years, utilization of mobile devices for tracking health parameters has increased. These devices are able to monitor different parameters such as heart rate, respiration patterns, amount of activity and energy expenditure. The devices specialized for medical applications provide more accurate measurements, assisting medical decision-makings and diagnosis procedures.

This thesis work presents the development of an ambulatory health monitoring system for measuring heart activity, respiration and movement. The developed system consists of a measurement unit, an Android application and a computer software. The measurement device, along with capturing the data from the required sensors, is also able to locally store and/or transmit the data wirelessly to a hand-held device. The designed Android application is responsible for receiving this data, reconstructing it and visualizing it in real-time. The computer software is developed to extract the locally stored information after the recordings.

The electronic design of the measurement unit is thoroughly described and the limitations are explored. Additionally, the structure of the implemented embedded software is illustrated and justified. Some brief overview of the structure of the Android application and the computer software is also provided.

The signal quality achieved by the system was evaluated and the power consumption was measured for different use cases. Our results showed that the developed system provides a competing signal quality comparing to devices in the market. Additionally, it has been shown that transmitting the data via Bluetooth Smart is less power hungry than storing it to a memory card. The report is finalized by mentioning the challenges faced during the development process.

PREFACE

The basis of this thesis work originally stemmed from the projects that I have been involved in at Tampere University of Technology. It was needed to design and develop a system for health monitoring, remotely measuring various parameters including cardiac activity, respiration and movement. This thesis work involves details about the system, an analysis of its performance and discussions about its bottlenecks and challenges.

The thesis, submitted for the degree of Master of Science, was sponsored by VitalSens project funded by the Finnish funding agency for innovation, Tekes and was done under supervision of DSc. Antti Vehkaoja in BioMediTech Institute and the Faculty of Biomedical Science and Engineering, Tampere University of Technology.

There were obstacles in the path of doing this thesis work and it took longer than I expected, but this design, its challenges and everything about it taught me a great deal of engineering and non-engineering lessons. And I am still as excited as the beginning of this work, but now more because of watching the system being used in various applications.

To my colleagues in VitalSens project and the department especially Jarmo Verho, Matti Mäntysalo and Tiina Vuorinen, I would like you thank for your help and great ideas. Special thanks go to my supervisor for giving me this chance to develop my skill in this work, his wonderful help, support and trust. Finally, I would like to thank my family especially my mother, who share credit on every goal I achieve and my girlfriend without whom it would not be possible to complete the work.

Tampere, 12.09.2017

Vala Jeyhani

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND SYMBOLS

AC	Alternating current
AD	Analog Devices
ADC	Analog-to-digital converter
AFE	Analog-front-end
API	Application program interface
AV	Atrioventricular
BGA	Ball grid array
BJT	Bipolar junction transistor
BLE	Bluetooth Low-Energy / Bluetooth Smart
CLK	Clock
CMRR	Common-mode rejection ratio
CPU	Central processing unit
DC	Direct current
DFU	Device Firmware Update
DLL	Dynamic-link library
DMP	Digital motion processor
DR	Data rate
CSV	Comma-separated values
ECG	Electrocardiography
EDF	European data format
EDR	ECG-derived respiration
EIP	Electrical impedance pneumography
EMI	Electromagnetic interference
ESD	Electrostatic discharge
GAP	Generic Access Profile
GATT	Generic Attribute Profile
GPIO	General purpose input/output
GSM	Global System for Mobile Communications
HCI	Host Controller Interface
HFXO	64 MHz crystal oscillator in nRF52832
HRV	Heart rate variability
I^2C	Inter-Integrated Circuit (identical to TWI)
IC	Integrated circuit

ID	Identity document
IIR	Infinite impulse response
INA	Instrumentation amplifier
iOS	Operating system from Apple
I/O	Input/output
LDO	Low-dropout
LED	Light-emitting diode
LOS	Line of sight
LP	Low power (related to ADS1296R)
MCM	Multi-chip module
MCU	Micro-controller unit
MEMS	Microelectromechanical systems
MISO	Master-in slave-out
MITM	Man-in-the-middle
MOSI	Master-out slave-in
NFC	Near-field communication
OOB	Out of band
OS	Operating system
OTA	Over-the-air
PC	Personal computer
PCB	Printed circuit board
PGA	Programmable gain amplifier
PWM	Pulse width modulation
RAM	Random-access memory
RC	Resistor-capacitor
RF	Radio frequency
RFI	Radio frequency interference
RGY	Red-green-yellow
RLD	Right-leg drive
RR	R-peak to R-peak
RTC	Real-time clock
SA	Sinoatrial
SD	Secure Digital
SINC	Sine cardinal
SPDT	Single-pole double-throw
SPI	Serial Peripheral Interface
SPS	Sample per second

STK	Short term key
SUT	System under test
SoC	System on chip
TWI	Two Wire Interface (identical to I^2C)
UART	Universal asynchronous receiver-transmitter
USB	Universal Serial Bus

C	Celsius
f	Frequency
I	Current
Q	Charge
R	Resistance
T	Temperature
t	Time
V	Voltage

1. INTRODUCTION

1.1 The Era of Mobile Devices

It is probably hard to imagine now that human being once dreamed about talking on phone, wirelessly while walking in a park. In 1906, Lewis Christopher Edward Baumer an English caricaturist published a cartoon in Punch magazine named "forecasts for 1907"[1], which was one of the first public appearances of the idea. “Development of wireless telephony”, the caption in the drawing explains. Although hand-held radio transceivers have been available since 1940, until 1973 the mobile telephony systems could only be found in vehicles. The world’s first mobile phone call was made about 40 years ago on April 3, 1973 by Martin Cooper informing a rival telecommunications company that he was speaking via a mobile phone[2]. This *mobile* phone that Cooper used to make the call weighted about 1 kg and used a brick-like battery. The battery could provide a call of 30 minutes and needed 10 hours to charge[3]. “something that would represent an individual so you could assign a number not to a place, not to a desk, not to a home but to a person” Cooper, renowned as the father of the mobile phone, said in his interview with BBC news on April 2013. This is a usual and normal element of our lives today. The phones that we carry nowadays have become smaller, and lighter with tens of additional features comparing to their ancestors.

There was a time that the digital LED watches needed pressing a button to display the time for a few seconds[4] to prevent the fast discharging of the battery. Now the a smart watch by Nokia, Steel HR, can work up to 25 days measuring heart rate, classifying our activities, analyze our sleep cycles and of course show the time. These would have not been possible without significant advancements in low-power

[1] The art can be found on punch.photoshelter.com/image/I00006GHuH4c0Ojo

[2] BBC New: www.bbc.com/news/technology-22013228

[3] Daily Express: www.express.co.uk/life-style/science-technology/388974/40-years-of-the-mobile-phone-Top-20-facts

[4] Wikipedia: Low-power electronics

electronics.

Nowadays several different devices have been added to the collection of the devices that we carry every day. Health trackers perform complicated computations, monitor our vital signs and still can work up to several days. These devices have now enabled us to know how many steps we take or how long we walk every day, how much calorie we burn, how fast our heart beats and how fast we breathe.

Health monitoring devices have been around for about a century. Willem Einthoven finished several prototypes of his string galvanometer for monitoring of heart activity. The original machine weighted about 270 kg and needed 5 people to operate it[5]. The ambulatory electrocardiography devices however began to appear commercially in 1960s. These devices, referred to as *Holter Monitors* named after Normal J. Holter, were designed to meet the need of capturing transient cardiac arrhythmia, which were not possible in a short measurement of heart activity. The first devices weighed 38 kg and was strapped on like a backpack. These devices were able to record 10 hours of measurement on a magnetic tape[6,7].

Today, the Phillips' DigiTrak XT Holter monitor only weighs 62 g, is able to record the measurements up to 7 days and is accompanied by a powerful software for automatic assessment of the recorded data. Other medical devices also have been moving to ambulatory applications. Beside electrocardiography, monitoring of blood pressure and blood oxygen saturation level are two examples[8].

1.2 The border for Ambulatory Medical Devices

The electronic shops today are full of health trackers that assist the customers in having a more thorough understanding of the daily activity. These devices can measure a variety of parameters including heart rate, distance and number of steps and are able to provide informative results. Depending on the location of these devices on the body, their methods of measurements and utilized algorithms, the accuracy of the information varies.

Another side of this rise is in the area of devices designed especially for medical

[5] Wikipedia: Willem Einthoven

[6] BCMJ: Ambulatory electrocardiography: The contribution of Norman Jefferis Holter

[7] ecgholtermonitor.com: Holter Monitor - The Brief History

[8] http://www.medicalexpo.com/prod/somnomedics/product-70128-520742.html

purposes. In these applications, the designers need to make compromises in their products to achieve the desired clinical validity of the measurements, in contrast to consumer health trackers. The location of the device, size, weight, number of connected wires are some changes that are noticeable when migrating from consumer products to medical ambulatory device.

When talking about an *ambulatory* medical device, it is important to define the goals and therefore the requirements and the compromises that need to be made. In medical devices, the quality of measurements, method of monitoring and accuracy of results are some parameters that take precedence over user's convenience, when it comes to making compromises.

In both of these categories, the devices make measurements, prepare the result (in whatever part of the whole system) and either store or transmit (or both) them to a peer device. It is worth mentioning that the transmission (which is sometimes a vital part of the system due to the limited resources in these low power devices) can cause problems with different levels of importance. In the next section, a more thorough discussion about the reason of the inclusion of hand-held devices in these systems is given.

1.3 The Need for Data Transmission

The ambulatory monitoring devices, beside a small area of applications used only for notifying the subject or others, are mostly intended to store the measured information to be reviewed and analyzed afterwards. Storing the data can be done in any node of the data transmission chain from the measurement device to the physicist/subject computer. Since there are some, if not all, physiological indicators that are usually required to be monitored and illustrated to the user in real-time, the first data node in the transmission chain can usually be the subject's hand-held device or the measurement unit itself if capable.

By utilizing the smart phones (or any other similar device) as a node in the system, not only the fundamental characteristics of the wearable devices including small size, affordable cost, weight, user comfort etc. are nicely maintained, but also a powerful source of computation, memory and wireless communication protocols are added to the measurement device. Display is a great advantage that cannot be provided in the measurement device itself. Another example is when the data is required to be uploaded to a cloud server. It can be done using the provided software layers in the smart phones, its WiFi or cellular modem and good energy resources.

As all the engineering stories, there are drawbacks in this scenario too. Data loss, limited distance between the phone and the measurement device and security are some examples. Sometimes additional hardware blocks are added to the measurement devices to bypass the need of the hand-held devices nearby them. Including GSM modems for sending the information directly to a cloud is one example.

Choosing the right architecture between all the possibilities is a significantly important factor of the design process and has large impact on the final performance of the system. The requirements of the application and the goals of the design impose the criteria for selecting the right system structure. According to the selected architecture, different wireless technologies are employed for either presenting the results to the user or storing the data. Classic Bluetooth, WiFi, GSM are such tools that can tackle specific requirements of the design.

Bluetooth Low Energy (BLE), which is also marketed as Bluetooth Smart (used interchangeably in this document), has drawn a lot of attention in ambulatory applications. BLE appeared first in the Bluetooth Core Specification 4.0 as an extension to its older brother, although with an entirely different lineage and design goals. Originally designed by Nokia Corporation as Wibree and then adopted by Bluetooth Special Interest Group (SIG), BLE was developed for applications with a tight energy and silicon budget. These design goals brought interesting features to BLE, differentiating it from the older wireless technologies and making it suitable solutions for applications such as wearable devices.

The more growing adoption of this technology in different devices (importantly smart phones and tablets), small chip area requirements, low cost, no intrusive licensing costs, availability and low power consumption are the features that make BLE a wise choice as the communication protocol between the measurement device and the next node of the system. “It’s the right technology, with the right compromises, at the right time”, mentioned Kevin Townsend et al. in their book about BLE[1].

BLE also has limitations. The throughput is the first limitation of BLE. Despite the upper limit of data transmission rate in the BLE specifications, this value is reduced in real-world practices by several factors, including but not restricted to bidirectional traffic, protocol overhead, CPU and radio limitations, and the specifications of the protocol stack implemented by different semiconductor companies. As an example, for nRF52832[9], with S130 *SoftDevice* [10] version 2.0 (compliant with Bluetooth 4.1) the theoretical data throughput is about 15 kByte/s. Although, the practical data rate is about 5 to 10 to kByte/s [1].

Operating range is another matter that should be taken into consideration. The transmit power is often configurable and usually between -30 and 0dBm. Although it might be possible to extend the range of BLE communication up to 30 meters *line-of-sight* (LOS), the practical range is smaller [1]. Incidentally, the transmit power in the new versions of the ICs (nRF52840 for example) coupled with Bluetooth version 5.0 release recently (about December 2016) has boosted the data rate and transmit range. In this project, Bluetooth 4.2 has been employed.

Based on the aforementioned short explanation of BLE, it is clear that what advantages it can bring to a system. We used BLE as the communication protocol between the measurement unit and the hand-held device. The data, along with settings, status flags and some other parameters are transferred between these two nodes. An extra powerful feature is Device Firmware Update (DFU) Over-The-Air (OTA) that provides a convenient way for upgrading the units. A more detailed technical overview (although still brief) is given in 2.4. For more information, refer to the Bluetooth Core Specifications.

1.4 Objectives

This thesis work focuses on the development of a remote monitoring system for measurement of cardiac activity, respiration and movement. This work was a funded by the Finnish Funding Agency for Innovation (Tekes) and several Finnish companies as a part of VitalSens project (funding decision number: 40103/14). Therefore the requirements of the system are defined based on the goals in that project. Additional features are also considered in the system to qualify it for other applications.

The system includes a miniaturized low-power measurement device that is able to measure the three aforementioned parameters. The measurement device is capable of communicating with an Android device for real-time monitoring of these parameters

[9] The nRF52832 is a *system on chip* (SoC) IC from Nordic Semiconductor that runs all the required layers of BLE stack and is armed with a powerful ARM processor and several peripherals.

[10] SofteDevice is the Bluetooth Smart protocol stack implemented by Nordic Semiconductor for concurrent multi-link roles.

and setting the measurement configurations. Additionally the device features a local flash memory for storing large amount of the measured data, which can be extracted by a computer software. The Android application and the computer software are also developed in this work[11].

The objectives in this work can be categorized in two main parts. The first part has been to develop the system with a smooth and accurate functionality, qualified for different applications. The other goal to achieve has been to study the challenges and limitation of this system. The discussions on these matters are mostly mentioned in chapter 5 and include our experience of developing this system. Additionally, the quantitative evaluation of the measured signal quality and power consumption of the system is included in chapter 4.

After a short background about ECG and respiration measurements, motion data acquisition and BLE in chapter 2, we go through the design details of the electronics and short descriptions of the computer and Android apps in chapter 3. Finally, the results and discussions are provided in chapters 4 and 5.

[11] A video demonstrating the system can be found here something

2. THEORETICAL BACKGROUND

In this chapter, a background about electrocardiography and its measurement techniques, measurement of respiration and BLE is presented. For the rest of the report, it is assumed that the reader has some background in electronics since its introduction is completely omitted in this report.

2.1 Electrocardiography

Electrocardiography (ECG) refers to measurement and graphical representation of the bio-potential signals generated by cardiac activity. Heart follows a repeating pattern initiated at the sinoatrial (SA) node located at the upper part of the right atrium and finalized with ventricular repolarization. Each part of the cardiac activity creates specific electrical patterns that are sensed with ECG measurement systems.

An ECG signal consists of several waves, each corresponding to a specific event in the cardiac cycle (see Figure 2.1). The SA node produces electrical impulses that travel through the atria and lead to a contraction in the two atrial chambers. This creates the P wave in the ECG signal. Then the electrical impulses reach at the atrioventricular (AV) node located at the lower end of the right atrium. After being delayed for a short time at the AV node, the electrical impulses enter the bundle of His, which is bifurcated to left and right bundle branches. The entrance of the impulses to the bundle branches cause contraction (depolarization) in the ventricles. The result is a combination of two fast upward and downward peaks in the electrocardiogram, named QRS complex. At the same time as the QRS complex, the atria also relax but the electrical footprint of it is very infirm and therefore, the QRS complex can be assumed to be mainly related to ventricular contraction. The T wave represents the relaxation (repolarization) of the ventricles. Figure 2.2 summarizes the heart cycle process [2].

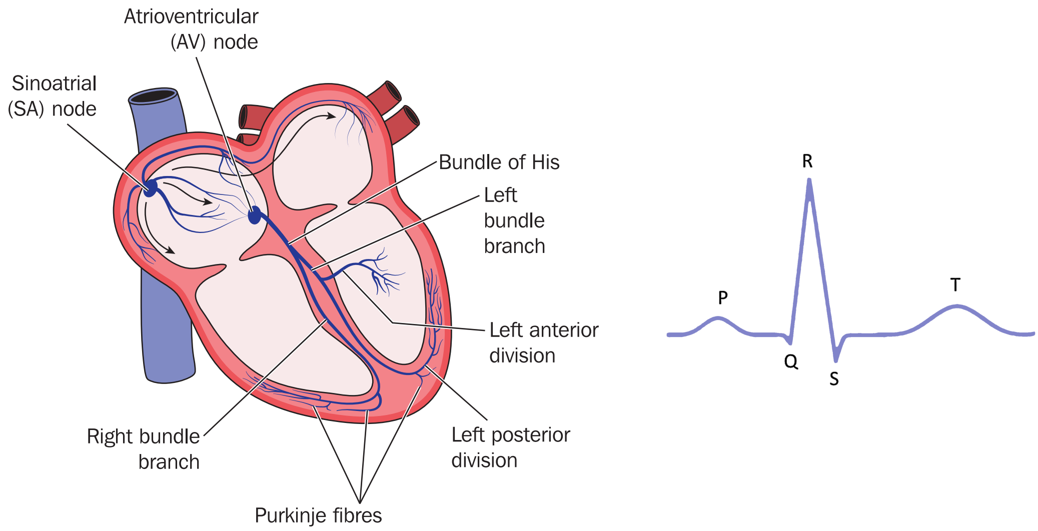


Figure 2.1 *Anatomy of the heart (left) and ECG signal waves (right). Copyright (for the left image): hfsimaging / 123RF Stock Photo*

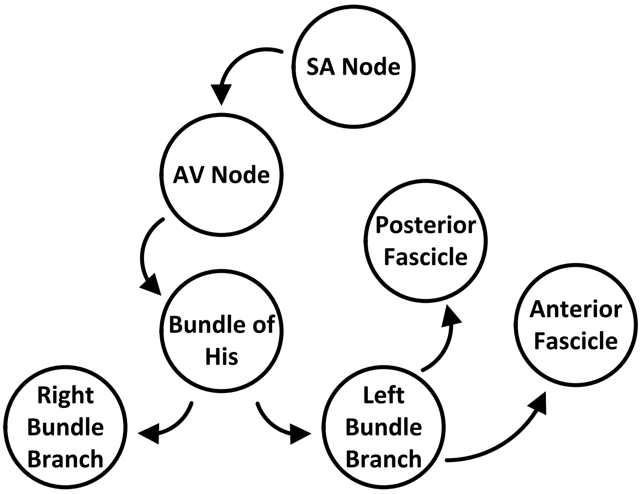


Figure 2.2 *Conduction pathway sequence of the heart*

At an instant of time, the heart activity is usually modeled as an electric dipole moment vector, resulted from the summation of several electrical activities by individual groups of cells, each having a specific direction and magnitude [3]. The projection of this vector on the ECG leads is what actually generates the aforementioned waves. A brief overview of ECG leads is given in the next section, 2.1.1.

A complete cycle of the cardiac activity is called a heartbeat. A series of heartbeats reveals the timing characteristics of the heart and is used for calculation of heart rate and analysis of heart rate variability (HRV). HRV is a popular marker of the operation of nervous system.

The ECG signal discloses many factors of heart activity. The morphology, duration of the waves (and intervals) and the amplitude are some elements that are commonly considered for extracting the desired information. For instance, in 1985, Pan and Tompkins utilized the slope, amplitude and width of the QRS complexes for detection of R peaks [4].

2.1.1 ECG Leads

The heart activity can be characterized by measurements from either the cellular level or body surface. The ECG signal looks morphologically different depending on the location of the electrodes. More specifically, the electrode locations form the leads that capture the projections of the heart activity.

A lead is referred to as the difference in voltage between a pair of electrodes [3]. The direction of a lead vector (the lead itself is considered as vector) is defined by the position and orientation of the heart and the electrodes forming that lead. Thus, the electrical activity generated by the heart may deflect the ECG signal depending on the lead vector and its own direction and size. When a depolarization propagates towards a lead it produces a positive deflection if it is towards the positive side of the lead. The opposite is true for repolarizations [5]. Additionally, a wave may not be captured by a specific lead if its propagation is perpendicular with respect to that lead. The depolarization waves are generally steeper with larger amplitudes comparing to repolarization waves [3]. The modulus of the vector projected into the lead decreases as the electrodes retreat from the heart due to the rise in impedance and therefore, the captured ECG signal has a smaller amplitude [3].

The ECG is typically measured with several leads including *unipolar* and *bipolar* leads. A bipolar lead is a differential measurement between two electrodes. On the other hand, a unipolar lead is a measurement of a single electrode with respect to a potential, which is usually constant during the whole cardiac cycle [3]. Not all the leads are actually *measured* (using an acquisition hardware). Some of leads, which are required due to their additional clinical value, are computed based on other leads. The three augmented leads in the standard 12-lead system are examples of

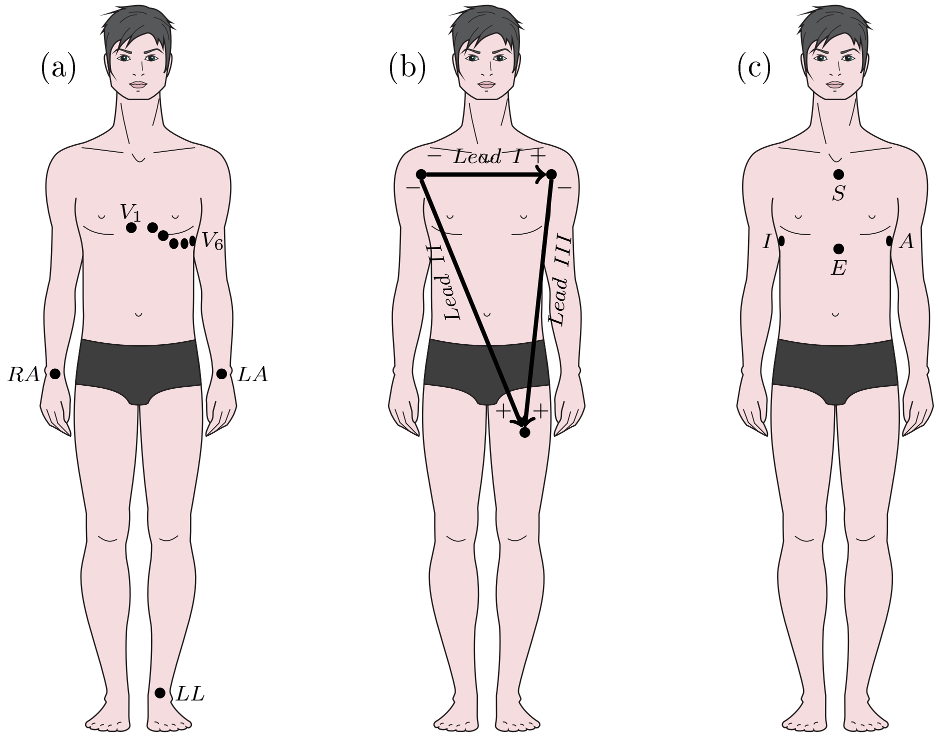


Figure 2.3 *The electrode locations in the ECG standard 12-lead system (a), the three limb leads in the standard 12-lead system (b) and the electrode locations for the EASI lead system (c). In (a), the V_1 to V_6 electrodes show the chest electrodes and the other three are the limb electrodes: right arm (RA), left arm (LA) and left leg (LL). Copyright: pushinka / 123RF Stock Photo*

this type of leads.

There are several standardized lead systems that are utilized not only on the basis of maximized information content, but also depending on their clinical value and practical considerations. The standard 12-lead system is probably the most frequently used system all around the world. This system is a combination of three bipolar limb leads, six precordial leads and three augmented limb leads. The history of limb leads goes back to the Einthoven string galvanometer[1]. The three electrodes are placed at right arm, left arm and left leg, constructing an equiangular triangle known as Einthoven's triangle. The six chest leads are measured between each electrode and usually Wilson Central Terminal that is calculated based on the voltage of the right and left arms and the left leg. The augmented leads, which can also be considered as unipolar leads [3], are computed from the limb leads. Figure 2.3

[1] The description and pictures of this device can be found on website of university of Toronto at utsic.escalator.utoronto.ca/home/blog/instrument/einthoven-string-galvanometer/

shows the electrode locations in the standard 12-lead system as well as the leads constructed by the three limb electrodes.

EASI lead system is another system that has interesting characteristics for ambulatory measurement devices. In 1988, Dower et al. synthesized 12-lead ECG based on only four electrodes: three from the Frank vectorcardiography [6] lead configuration and one additional electrode positioned over the upper end of the sternum [7]. Since then, several researches have focused on characterizing the accuracy of the synthesis and different solutions to diminish the difference. About ten years later, in 1997, Drew et al. ran a test on detecting the presence of ischemia on 151 patients. Their findings showed that EASI (and its derived standard ECG lead) resulted in 150 of the cases correctly classified with no false positive (only one false negative) [8]. Accuracy of QT interval measurement [9], PR and QRS intervals [10] and ST segment depression during exercise monitoring [11] is also reported to be high when the EASI system is compared to the standard 12-lead system. Several approaches from linear regression to artificial intelligence and support vector machine have been evaluated for deriving the standard lead system from EASI set. Figure 2.3 shows the electrode locations in the EASI lead system. The electrodes E, S, A and I are positioned at lower sternum, manubrium, standard 12-lead V5 location (left axilla) and right axilla, respectively. Recently, the interests are to minimize the electrode area even further [12].

The EASI lead set has two important advantages in ambulatory measurements: it employs only four electrodes and is much more convenient for the subject to wear it for long term measurements; and the amount of resulted information is much lower comparing to the standard 12-lead system, which makes it easier for low-power systems to handle, process, store or transmit this data.

2.1.2 ECG Acquisition

The ECG has an amplitude of a few millivolts, and may be buried in a much larger noise depending on the measurement environment. Therefore, the measurement system must be able to reject this noise in a reasonable amount and extract the ECG signal (high signal-to-noise ratio). The voltage is measured between two electrodes (or one electrode and one reference potential) as a differential measurement. Therefore, in many applications, instrumentation amplifiers (INA) are the best choice.

The instrumentation amplifiers, beside from amplification, have several interesting characteristics that make them suitable for biopotential measurements. High common-mode rejection, low input bias current, low noise, and low offset voltage are some of the most important parameters that must be considered when selecting them for this application.

The amplification gain strongly depends on required resolution and the analog-to-digital converter's (ADC) dynamic range. It is important to remember that although the ECG itself is not more than a few millivolts, the variations in the DC of the signal (mostly due to changes in half-cell potential of the electrodes) can take the signal out of range and result in amplifier saturation.

Before the amplifier, there must be a subject-protection circuit that also acts as passive filters that are needed for meeting the requirements of the analog to digital conversion, namely anti-aliasing filters. Usually in battery-operated devices, this protection circuit is implemented as a barrier to the passage of current from the device to the subject. The values are commonly decided based on the maximum supply voltage in the system. Along with the passive filters there are also *circuit* protection elements that limit the coming voltage from outside world to the sensitive components of the system.

In digital systems, the output of the amplifier stages is fed to an ADC to convert them to digital samples, which are then read by the CPU of the system. Aliasing, Nyquist theory and digitization are important issues to take into consideration about this stage of the circuit. A description about different parts of a general ECG measurement device is given in Webster's medical instrumentation book, chapter 6 [13]. A more practical guide in using in-amps is published by Analog Devices, which can be found online[2][14].

Some parts of this circuit may be replaced by a single analog-front-end (AFE) IC. AFEs implement most of the required parts of the circuit including amplifiers, ADCs, and even more advanced elements such as right-leg drive, Wilson central terminal and lead-off detection. Choosing between the two options is based on the acceptable compromises. In short, using AFEs greatly miniaturizes the system while it might impose less control over some of the circuit elements and increase the power consumption of the system.

[2] Refer to this link http://www.analog.com/media/en/training-seminars/design-handbooks/designers-guide-instrument-amps-complete.pdf

2.2 Respiration Measurement

Pulmonary evaluation is an important measure to assess the subject's condition. Various parameters such as respiration rate and lung volume can be extracted from raw ventilation data, although their accuracy is affected by the method of the measurement. There are different solutions currently used for measuring the respiration that are associated with several trade-offs between their usability, accessibility, and finally the accuracy of the measurement.

Spirometry, providing the most accurate measurement, is the gold standard method that directly measures the gas flow of breathing. Gauging the affected parameters of breathing air flow introduces alternative techniques for accessing this information. Temperature, humidity and CO_2 are examples of these parameters. The temperature can be measured by a mask worn in facial area [15] or infrared thermography [16], providing the possibility of a contactless monitoring solution. Acquiring the variations of pressure [17] is another approach that has been validated at least for adults [18]. Capnography [19], electrical impedance pneumography (EIP) [20] and inductance pneumography [21] are other techniques used for pulmonary monitoring. Additionally, other physiological signals such as electrocardiogram [22] and photoplethysmogram [23, 24] have been used for extracting respiration information. An extensive study on various techniques of respiration monitoring has been published by AL-Khalidi et al. [25].

As previously highlighted, different techniques have their own advantages and disadvantages. Among all other, EIP has interesting advantages that make it a very suitable choice for ambulatory applications. EIP, by excluding the facial mask from the measurement setup, provides a convenient monitoring for the subject. Another advantage of EIP is that the acquired data is straightly a respiration waveform, although it does not measure the breathing flow directly. This is while some of the other methods like ECG-derived respiration estimation (EDR) require some additional and sometimes computationally expensive processing algorithms to extract the respiration parameters. An important benefit of using EIP for respiration monitoring appears in the systems that also monitor the ECG. In such systems, the EIP can be sensed with the same electrodes that are used for ECG measurement. This excellence provides the user with the pulmonary information with no necessity of additional electrodes or other sensors. However, inclusion of a dedicated circuitry in the hardware for this measurement is inevitable.

As many other methodologies, the EIP technique has some drawbacks too. Since this measurement is straightly dependent to impedance changes, it is prone to movement artifacts. For instance, an arm abduction results in a significant change in the impedance waveform, obviously depending on the location of electrodes. This artifact can be suppressed by improving the electrode location, area and arrangements (bipolar, guarded bipolar, tetrapolar and guarded tetrapolar) [26, 27]. Another challenge is the effect of electrode locations. Clearly, the amplitude of the respiration-related changes in thoracic impedance are different across the chest. Lahtinen et al. reported that the best signal quality is achieved by placing the electrodes on the left and right flanks [28]. While this is the best measurement condition when there is no electrode area limitation, in small-area electrode systems those electrode locations might not be accessible. A study has shown that the S-A electrode pair provides the most accurate respiration rate estimation when using EAS (a subset of EASI) electrode set[29].

The respiration-related changes in the electrical impedance over the thoracic area are due to two main factors: the variation of the amount of the gas in the lungs in inhale and exhale phases and the variation in the length of the signal path in result of expansion and contraction in the chest cavity. In EIP measurement, these changes are sensed by feeding a carrier signal to the body, measuring the voltage across the measurement electrodes and finally extracting the respiration signal. This powerful mechanism of measurement, which is widely used in instrumentation to detect the small AC signals with a low signal-to-noise ratio, is called *synchronous detection*.

Systems featuring synchronous detection make a use of an auxiliary signal, named *carrier*. The parameters of this carrier signal, usually the amplitude, is affected by the signal of interest (respiration here). More specifically, the carrier is modulated, usually amplitude modulated, in the system under test (SUT). The desired signal is then extracted by demodulating the measured signal.

Modulation technique introduces several advantages over measuring the DC (or low frequency signals). It allows measuring very weak signals, buried in the noise floor. Moving the signals of interest away from low frequency noise increases the signal-to-noise ratio, allowing small signals to be measured with higher precision. Offset and drift errors in amplifiers are such noise sources that significantly affect on the measurement of low frequency small signals. Additionally, according to medical standards, a more powerful current can be applied to the body when using high frequencies, resulting in an increase in measurement precision [30].

When modulating a signal, all of its bandwidth is moved around the carrier frequency. In this phase, the low frequency noise and the interference are irrelevant to the signal of interest and therefore can be safely filtered out. Then, the measured signal is demodulated and a copy of the main signal is created at the origin of the spectrum. This signal is then low-pass filtered to extract the signal of interest. Figure 2.4 shows all the four steps including the spectrum of the each signal for an example case. In this case, the carrier signal is a sinusoidal signal with the frequency of 10 kHz. The signal of interest has a frequency of 3 Hz. As shown in this figure, in both modulation and demodulation phases, a copy of the input signal bandwidth is copied to $(\omega_c + \omega_i)$ and $(\omega_c - \omega_i)$, where ω_c and ω_i are the frequencies of the carrier and the main signals, respectively.

In simple hardware configurations, a square wave is preferred to a sinusoidal wave as the modulation signal due its simplicity. The demodulation signal has the same waveform and frequency as the modulation signal. A very important and sometimes challenging issue is that the phase delay in the signal path introduces a phase mismatch between the modulation and demodulation, causing problems in extracting the signal of interest. This problem is handled by some hardware techniques such as *demodulation with blocking*. However, some manual adjustments depending on the final system might be needed.

2.3 Posture Estimation and Motion Analysis

Human body makes a broad range of movements with different properties that may assist the medical diagnoses, event detection and patient behavior analysis. Nowadays, there are various approaches to get information about patient movements that are chosen depending on the application, resources and the type of required information.

One of the tools among several is using MEMS (Microelectromechanical systems) sensors such as accelerometers and gyroscopes for tracking the subject's activity. A variety of information can be derived from these two measurements such as step counting (the device is also known as pedometer) and gate analysis[31, 32, 33], gesture recognition [34, 35, 36] and sleep analysis [37]. Accompanied by other measurements (or even singly), MEMS sensors are also used for estimating level of activity

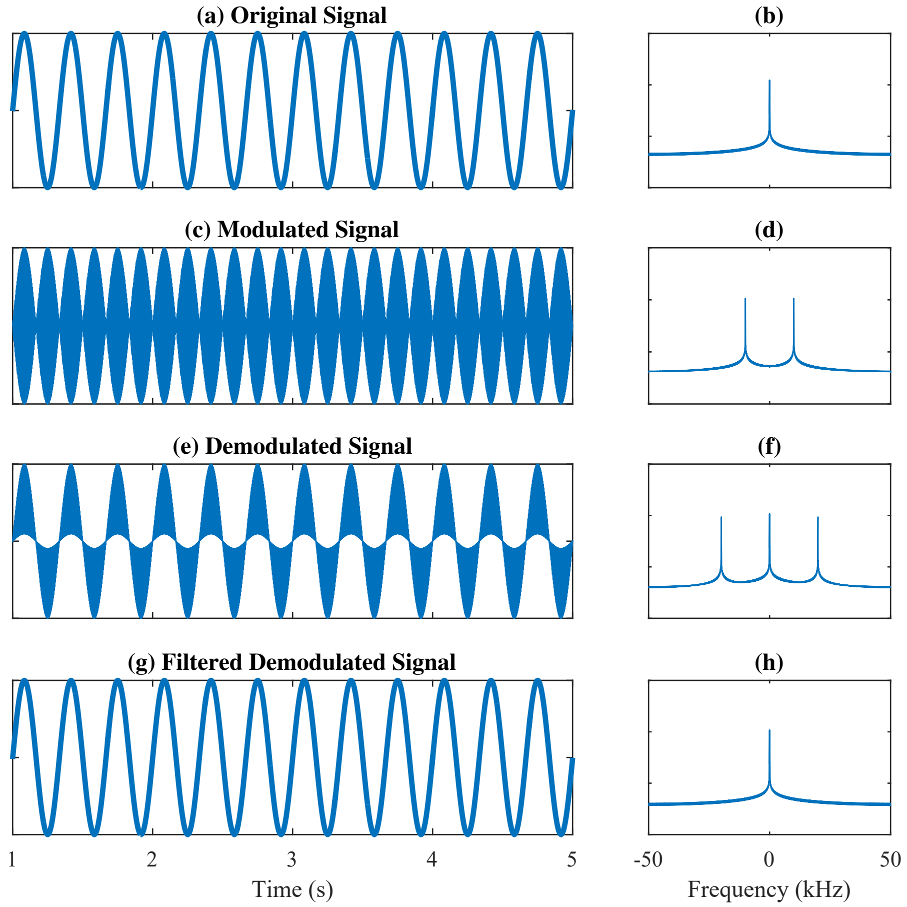


Figure 2.4 *Amplitude modulation: (a) The signal of interest in SUT; (b) the spectrum of the signal; (c) carrier signal with the low frequency signal in (a); (d) spectrum of the modulated signal; (e) demodulated signal; (f) spectrum of demodulated signal; (g) reconstructed signal; (h) the spectrum of the reconstructed signal*

and energy expenditure [38, 39, 40]. Additionally, MEMS sensors are employed in algorithms for boosting the signal quality of other measurements. For instance, accelerometer can be employed to remove the movement artifacts from the ECG signals adaptively [41].

In the light of these use cases, it was clear what benefits the MEMS sensor can bring to our device and therefore we selected the accelerometer and gyroscope as one of the measurement sources of the device. Accelerometers are able to measure static and dynamic acceleration that can directly measure subject's static orientation and dynamic movements, respectively. Gyroscope on the other hand, provides a zero output when having no movement but equips the device with measuring of angular velocity. There are situations in which gyroscope assists the accelerometer for a better understanding of the movement.

The accelerometer and gyroscope sensors have emerged in different packages with different features, meeting various requirements. Some are simple accelerometers providing an analog voltage that require other components such as filters, and an ADC in their signal path to be able to measure them digitally. Others, referred to as analog-front-ends (AFE), provide all the needed elements in one package to some extent, perform the desired actions and send out the information as digital data packets. In these cases, the main controller of the device needs to first configure the sensor with its desired parameters and just collect the data. One of the advantages with this type of approach is the miniaturization, which is a critical requirement in ambulatory applications (like AFEs for ECG measurement). From the negatives, one can mention the extra features provided by these sensors that might be unnecessary in a project and therefore an elevation in power dissipation of the sensor. This solution might increase the complexity of the development process and sometimes can limit the designer's control over some of the features.

We chose MPU9250 from TDK[3]. A powerful motion tracking multi-chip module (MCM) featuring a complete analog-front-end for measuring accelerometer, gyroscope and magnetometer. The IC collects all the requested information and put them on its output port. Furthermore, this chip is equipped with a digital motion processor (DMP) that offloads the processing from the host processor[4].

2.4 Bluetooth Low Energy

In this work, Bluetooth Low Energy (BLE) is used as the bridge from the measurement unit to a hand-held device. A brief description of this protocol is given in the rest of this section.

A system running a BLE protocol stack consists of three main building blocks: application, host and controller. Application is the part of the device and interacts with the upper layers of the BLE protocol stack. Host and controller include the upper and lower layers of the protocol stack, respectively. The communication

[3] Previously InvenSens that were recently acquired by TDK.

[4] The magnetometer and DMP were not utilized in this work.

between the host and the controller is done through Host Controller Interface (HCI) that is thoroughly defined by Bluetooth standards. This standardization enable different hosts and controllers to talk to each other, regardless of the manufacturers.

BLE is offered in different configurations; SoC, Dual IC over HCI and Dual IC with connectivity device [1]. SoC is a one-package solution that runs all the required layers, including the application part. The SoCs offered by Nordic Semiconductor and Texas Instruments are examples if this form. This configuration is usually preferred in small sensor nodes of the system in which heavy miniaturization is required. Dual IC over HCI provides the possibility to run the application and host layers of protocol stack in the processor and implement the controller as a separate unite. This configuration is usually used in smart phones and tablets that already include powerful CPU to easily run the protocol stack.

The third possibility is the case in which one IC runs the application and communicates with another IC running host and controller layers of the protocol stack. This configuration is similar to the traditional modular solutions for Bluetooth, GSM and WiFi wireless communications. Note that the communication between the two parts in this configuration is not under specifications of SIG and takes place according to the specific protocol chosen by the manufacturer such as Universal asynchronous receiver-transmitter (UART), I^2C and SPI.

The data transmission in a BLE communication can be done using two different ways: *connection* or *broadcasting*. Broadcasting is the connectionless way of sending data out, in a one-way direction, to any device that is listening. Two different roles are defined in broadcasting: *broadcaster* and *observer*. The broadcaster sends non-connectable advertising packets and observer repeatedly scans for these packets. There can be several observers in the network and every device interested in those packets can pick them up. The drawbacks of broadcasting include limited payload length, lack of security or privacy provisions and the restricted one-way path of data transmission. Connections on the other hand, establish a secure duplex tunnel of data transmission between the devices in the network. There are two roles defined in connections: *central* and *peripheral*. The central continuously scans for connectable advertising packets. It then picks the advertising packet coming from the appropriate peripheral (the one that it is supposed to be connected to) and initiates the connection. The peripheral sends the connectable advertising packets at specific radio channels in defined time intervals. Peripheral finally accepts the connection request from the appropriate central and the connection parameters are negotiated between them.

A connection consists of individual connection events, happening at an agreed connection interval. In each connection event, both of the peers send packets to each other, regardless of having any data to send. Therefore, to avoid overuse of the bandwidth and the power resources, the connection interval should be chosen according to the needs of data transmission. If no valid packet is received by a peer after a certain amount of time (connection supervision timeout), the connection is considered lost.

The parameters of the connection are controlled by the central, although the peripheral can request for updating the connection parameters. Central can then either accept or reject that request. In general, connections consume less power comparing to broadcasting and are the only way to negotiate parameters of data transmission [1].

The peripheral can tell the central that it has more data pending to be sent and if possible, the central will establish another connection event in the same connection interval[5]. The master[6] may however refuse to establish more connections in a connection interval regardless of the peripheral request. It is important to note that the number of connections per connection interval is not equal between different devices and different Bluetooth core specifications.

There are two important profiles namely Generic Access Profile (GAP) and Generic Attribute Profile (GATT). GAP defines the way that the two peers talk to each other. It specifies how the devices control the connection, discovery, security etc. GATT on the other hand, deals with data structure and adds a data abstraction to its lower layer (Attribute Protocol). The GATT data objects (known as GATT-based profiles), which can be employed by different applications, implement a structure for the user data along with the required information. Services are built according to the needs in different applications[7] and provide a standardized communication protocol between different devices. Conceptually a GATT service can be thought as a class in any modern object-oriented programming language [1].

[5] This is done by setting the *MD* field in the PDU (protocol data unit) header

[6] In a BLE connection, the devices are either master or slave. A master initiates a connection and controls it later.

[7] A list of generalized GATT services can be found in: https://www.bluetooth.com/specifications/gatt/services

From the version 4.1 of the specification, the BLE network can consist of different centrals and peripherals. In other words, each node in the network can be either a central, peripheral or both and each central can be connected to multiple peripherals and vice-versa.

Readers interested in more details about BLE may refer to the book *Getting started with Bluetooth Low Energy* [1] or the Bluetooth core specifications provided online.

3. SYSTEM DESCRIPTION AND DESIGN CONSIDERATIONS

The architecture of the developed system is illustrated in Figure 3.1. The device collects the measurements and along with storing them, it may transmit them to an Android device. The Android device reconstructs and monitors the received information in real-time, giving the user a fast representation of the data being recorded by the measurement unit. After the end of the measurement session, the device can be connected to a computer to retrieve its recorded signals. This chapter provides an overview on the design process of the system, with a focus on its electronics. A video demonstrating the system functionality may be found here: www.spiritcor9d.xyz.

3.1 Hardware

The hardware of the device includes four main blocks to integrate the functionality needed in this work. Figure 3.2 shows the block diagram of the hardware. The microcontroller (MCU) is the core of controlling and processing. It communicates with the AFEs to acquire the digitized ECG, impedance pneumography and motion signals. The ECG AFE measures three channels of ECG and one channel of impedance pneumography, which pass through analog switches and signal conditioning analog passive circuits. The signals are then sent via serial peripheral interface (SPI) protocol to the MCU. Another SPI channel is reserved a the Secure Digital (SD) memory card.

A USB connector is shared for both ECG recording and data transmission to computer. When in the USB mode, the switches connect the USB jack to the USB interface IC that is connected to the UART of the MCU. The presence of USB host is automatically detected and the device switches to USB mode. When unplugged, the device goes to its default mode i.e. the ECG measurement mode.

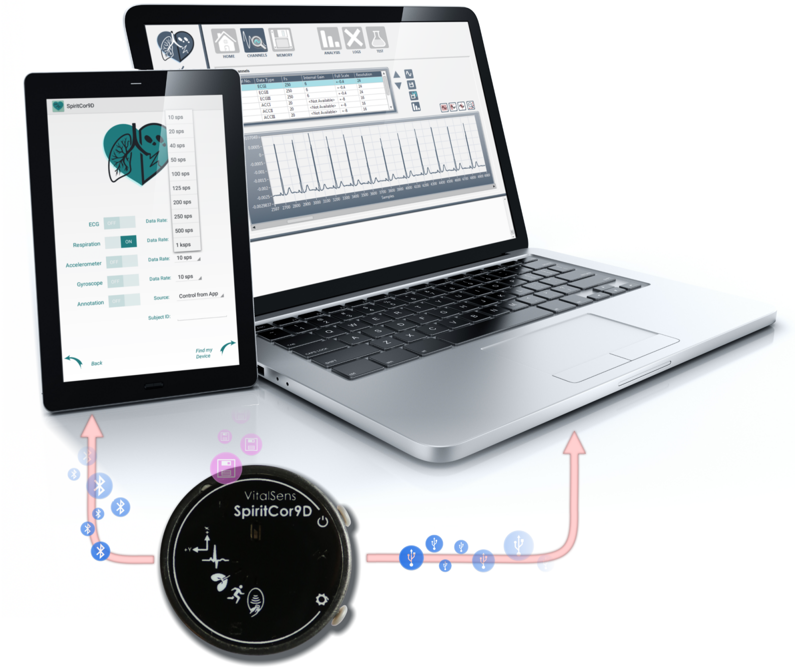


Figure 3.1 *Architecture of the designed system. The communication between the measurement unit and the Android device is done using BLE. Universal Serial Bus (USB) is the protocol between the measurement unit and the computer. The enclosure is designed by a third party company.*

The motion sensor is a 9-axis motion tracking device, measuring accelerometer, gyroscope and magnetometer. The sampling rate, bandwidth and the state of each sensor (enabled or disabled) can be configured. The digital samples are sent to MCU via Inter-Integrated Circuit (I^{2}C[1]) protocol. Due to the limited digital communication channels of the MCU[2], this I^{2}C channel is also shared with power management circuitry and I/O expander IC. The power management circuit is able to monitor the level of battery, recharge the battery and control the output voltage level.

Bluetooth low energy (BLE) is utilized as the communication bridge between the device and an Android hand-held device. BLE is used for not only sending the measured signals but also configurations, measurement settings, current time and date and other user- and system-controlled parameters. The hardware is also armed

[1] Also referred to as TWI. Both I^{2}C and TWI are used interchangeably in this document.

[2] The nRF52382 has several peripherals such as three SPI interfaces and Two Wire Interface (TWI) interfaces, but they share the same ID and may not be used simultaneously. Therefore, only three of the interfaces can be used at the same time. In datasheet of nRF52832, chapter 15.2, table 16 shows the details related to this matter.

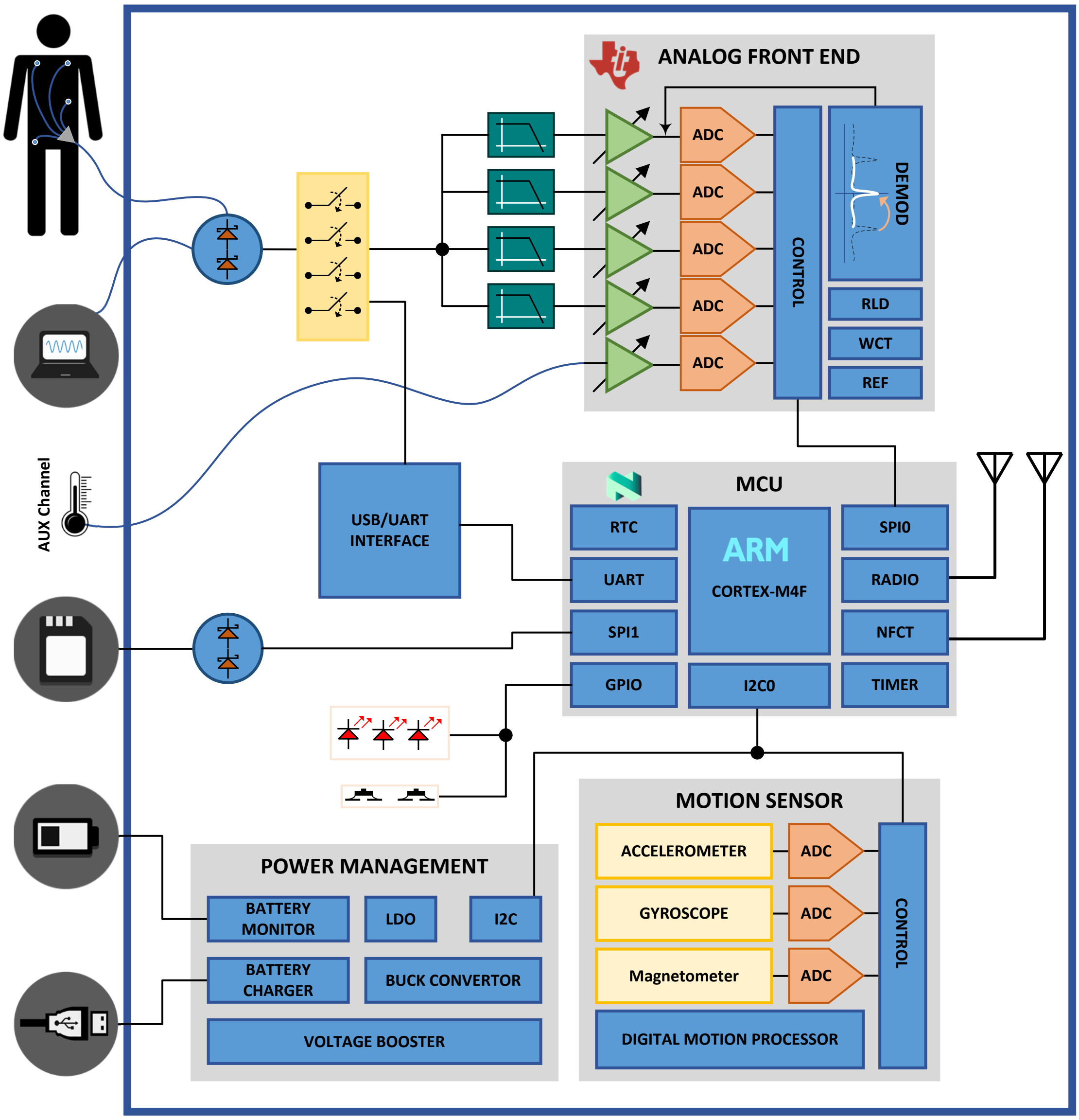


Figure 3.2 Block diagram of the hardware

with near field communication for future developments that can be useful for instance in tackling the security issues in BLE.

The rest of this chapter describes the important parts of the circuit. For a detailed description of the components, refer to their datasheets.

3.1.1 Controlling Unit

An nRF52832 from Nordic Semiconductor acts as the central controlling and processing unit of the device. This chip is chosen due to its very low power consumption, its well-designed BLE stack and the good support and documentation that the company provides. The MCU's main tasks are:

- power mode selection and battery charging controlling
- selecting the USB connector as either the electrode connection or the USB communication
- collecting the ECG, EIP, accelerometer and gyroscope data
- saving (into the SD card) or transmitting (via BLE) the collected data
- performing essential signal processing algorithms
- sending the stored data to a computer

There is one micro USB connector in the device that is used for data transmission to and from a computer or electrode connection for ECG and EIP measurements. A transistor-based circuit is designed to detect the presence of a connection to a computer. Therefore, the device is able to turn itself on and switch to USB mode automatically when it is connected to a USB host device.

The MCU collects the digitized samples from the analog-front-end (AFE) ICs and performs the required tasks on them. The communication protocol for receiving the ECG and EIP signals is SPI. Collecting movement samples is performed through I^2C.

Storing the data is done by communicating to a μSD card. The MCU buffers the measured signals and writes them to the memory card through SPI using a custom defined data structure. This structure is able to preserve the whole data (with no data loss) and keep all the information of the measurements, subjects and the device itself.

The rest of this section gives a more detailed insight about some of the blocks/-components used in the device including the ECG and EIP measurements (3.1.2), movement measurements (3.1.3), power circuit (3.1.4), analog switches (3.1.5), μSD card (3.1.6), and the printed circuit board (3.1.7).

3.1.2 ECG and EIP Measurement

Two physiological signals are measured by the device: electrocardiography (ECG) and electrical impedance pneumography (EIP). Four electrodes (excluding the right-leg drive electrode, which is not mandatory) can be connected to the device. This configuration allows acquisition of three channels of signal, all measured with respect to the fourth electrode. This number of channels produces up to 6 different leads ($\binom{n}{2} = \frac{n(n-1)}{2}$, where n is the number of electrodes) which can be computed by the MCU or as an offline or online post processing.

The main block of the measurement circuit is an ADS1296R analog-front-end (AFE) from Texas Instruments, which performs the signal amplification, filtering and digitization. This component consists of six analog input channels from which one of them is configured for EIP measurement, three for ECG measurement and one as auxiliary[3] channel. The last two channels are disabled. The converted data along with some additional information (lead-off and GPIO status) is then sent to the controller unit in a stream of bits . Table 3.1 shows some of most important characteristics of this component.

Table 3.1 *ADS1294R specifications. The values are typical unless otherwise noted.*

Parameter	**Value**
Input capacitance	$20pF$
Input bias current ($T_A = 0°C$ to $70°C$)	$\pm 1nA$
Bandwidth	$32 - 237kHz$ (depends on gain)
Resolution	$19 - 24Bits$ (depends on data rate)
Data rate	32000 SPS (max.)
CMRR ($f_CM = 50Hz$)	$-115dB$
Internal reference voltage	$2.4V$
Internal reference voltage accuracy	0.2%
Internal clock frequency	$2.048MHz$
Internal clock frequency accuracy ($T_A = 25°C$)	0.5% (max.)

One important factor is the uncertainty of the internal clock frequency, which is as high as 0.5 % in the room temperature of 25 °C. This is the best case since the

[3] This channel, although provided in the printed circuit board (PCB), is not available to the user and is utilized for development purposes.

value is larger in other temperature values. This uncertainty has a significant impact on the measurement since it directly affects on the accuracy of the sampling rate, which is vital in analog-to-digital conversion and post processing of the measured signals. The AFE contains a delta-sigma ADC that samples the signal at 512 kHz (assuming the component is set at the high-resolution mode) and then the discrete samples are decimated to the required data rate. Assuming that 500 SPS is the desired sample rate for the system, the decimation is done by a factor of 1024. This implies ±2.5 samples error per second (one sample error each 0.4 seconds) and thus, ±3 seconds (1500 samples) uncertainty in a 10-minute measurement. To address this problem, an external oscillator is provided as the alternative clock source. With this clock source, there would be 15 samples error in a 10-minute measurement. It is worth noting that the inclusion of the external clock source compromises the power consumption of the system. Therefore, there is a trade-off between the clock accuracy and the lifetime of the battery during the measurement. Taking this into account, the selection of the clock source is performed by the MCU according to the preferences of the user and the defined mode of measurement.

Although the AFE performs most of the actions needed for the conversion, a passive circuit is needed before the AFE. This circuit provides anti-aliasing filtering, subject protection and input ground return. A related point to consider is that the performance of the AFE (such as its CMRR) is degraded by this unavoidable circuit.

Figure 3.3 shows the circuit used before the AFE's inputs. An important issue that must be taken into consideration is the limiting current resistors. In Figure 3.3, the R1+R2 and R3+R4 resistors act as the safety resistors. The maximum voltage in the device is 4.5 V (fully charged battery). Therefore, in the worst case a current of 4.5 V / (39.2 kΩ × 4) = 28 µA passes through the body. For analysis and choosing the values, the anti-aliasing requirements can be first considered, which involves a discussion about the frequency response of the AFE internal circuit too.

In the AFE, the analog signal is first amplified by a programmable gain amplifier (PGA) and then digitized by a Delta-Sigma ADC at data rate of $f_{MOD} = f_{CLK}/4$ for high-resolution (HR) mode and $f_{MOD} = f_{CLK}/8$ (LP) for low-power mode, where f_{MOD} and f_{CLK} denote the ADC's modulator input sampling frequency and the frequency of the signal on CLK pin of the AFE, respectively. Assuming $f_{CLK} =$ 2.048 MHz, the resulting data rate would be 512 kHz and 256 kHz for HR and LP modes, respectively. The output of the ADC is then resampled by a digital dec-

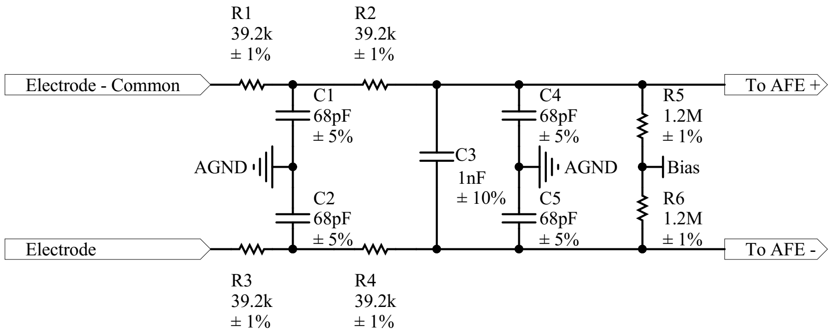


Figure 3.3 *The analog passive circuit used for ECG signal conditioning. Similar circuits can be found in Analog Device's guide to INA [14].*

imation filter to lower the rate and improve the resolution. The decimation filter also provides anti-aliasing filtering, which reduces the complexity of the analog anti-aliasing filters that are needed before the AFE.

The decimation filter is a third-order low-pass sine cardinal (sinc) filter with a variable decimation rate. The decimation rate can be configured by the registers in the AFE and its value affects the overall output data rate (f_{DR}) and the noise of the signal. More specifically, there is a tradeoff between the noise level and data rate: the higher the data rate the less is the noise-free resolution. The digital filter has a $-3\,\mathrm{dB}$ bandwidth of $0.262 \times f_{DR}$ but the passband repeats itself at multiples of f_{MOD}. Figure 3.4 shows the frequency response of the decimation filter for two data rates: 32 kSPS (DR[2:0] = 000) and 500 SPS (DR[2:0] = 110) assuming that the chip is set to perform in HR mode. DR is a field in the CONFIG1 register of the AFE and is used to choose between different values of data rates.

The repetition of the passband of the decimation filter frequency response implies the need of an appropriate passive anti-aliasing resistor-capacitor (RC) filter before the signal undergoes the digitizing process. According to Figure 3.4, the RC filter must have enough attenuation at f_{MOD} to achieve the desired performance. Therefore, the cut-off frequency of the RC filter depends on the clock frequency of the AFE (f_{CLK}) and the mode that the AFE operates in (HR or LP). Assuming that $f_{CLK} = 2.048\,\mathrm{MHz}$ and the operation mode is HR, for having -40dB attenuation at 512 kHz the cut-off frequency of a single stage RC filter should be located at 5.12 kHz. With a 68 pF capacitor and neglecting the loading effect of the two stages, the cut-off

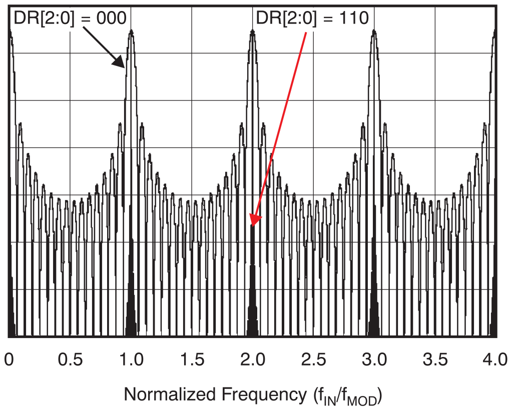


Figure 3.4 *Transfer Function of the On-Chip Decimation Filters of the ADS1298R extended to* $4 \times f_{MOD}$. *Copyright: from datasheet of ADS1296R*

frequency of the filter would be 60 kHz with an attenuation of −40 dB.

The mismatch between the components in the two differential lines create differences between the filters and therefore converts the common-mode components to differential. To diminish this effect, an extra capacitor (C3) is used, which creates a low-pass filter with a lower cut-off frequency comparing to the aforementioned filter. More specifically, the circuit in Figure 3.3 has two cut-off frequencies: a common-mode cut-off frequency (60 kHz), which is seen between each line and the ground of the circuit, and the differential cut-off frequency, which is seen between the two lines. The differential cut-off frequency is calculated according to equation 3.1. As a result, the differential cut-off frequency of the filter is located at 2 kHz.

$$f_{DIFF} = \frac{1}{2\pi R_2(2C_3 + C1)} \tag{3.1}$$

The common-mode filter is an anti-aliasing filter that also acts as a radio frequency interference (RFI) filter. The resistors in this filter are also needed for limiting the current flowing to the body, especially in case of a failure of an electronic component. Additionally, the differential filter attenuates the frequencies higher than 2 kHz that improves the adverse effect of component mismatching on the CMRR of the whole analog differential circuit.

It is worth mentioning that the minimum size of the resistors in this filter (e.g. R2) must be decided according to the medical standards. This is while in other applications of instrumentation amplifiers a much smaller resistor is recommended for RFI filtering circuit to lower the Johnson noise.[4]

Two large resistors R5 and R6 provide the ground return for the circuit that allow a discharge path for input bias current. Both inputs are referenced to the *Bias* potential, which is approximately half of the reference voltage (to bias the inputs to the middle of full-scale) used by the internal ADC of the AFE and is provided by an external IC. Due to the mismatch between the resistors (especially that the resistors are large), the input bias currents are not identical and there is a small offset voltage error. This error can be reduced by putting a resistor (about one-tenth of R1) between the input lines [14].

The maximum gain of the PGAs is 12 and the internal reference voltage is 2.4 V. Thus, assuming that the maximum amplitude of the measured signal (ECG + artifacts) is 50 mV, the output of the PGA (input of the modulator) is 0.6 V. This ensures that the dynamic input range of the ADC is enough even without a high-pass filter. Avoiding the high-pass filter allows keeping all the low frequency components of the ECG signal, which are useful in some analysis such as ST segment elevation.

The respiration is measured via modulating the impedance changes of the body caused by the respiration. The modulation signal is fed to the body and the measured signals are then demodulated to extract the respiration component. Figure 3.5 depicts the passive circuit used for respiration. Again, R2 and R3 provide the path for input ground return. The resistors R1 and R4 limit the current fed to the body. A high-pass filter is constructed by the C2, C3 and R2 with a cut-off frequency of 61.6 Hz. This filter isolates the modulated respiration signal from cardiac activity and is low enough to keep the modulated signal intact and therefore guarantees a reasonable demodulator gain[5].

3.1.3 Motion Sensor

The sensor used for motion tracking is MPU9250 from TDK. This IC measures nine axes of movement including three axes of accelerometer, three axes of gyroscope and three axes of magnetometer. The MPU9250 is a multi-chip module (MCM)

[4] Johnson noise (Johnson-Nyquist noist) is the electronic noise that is related to the temperature of the component. For instance a first order low-pass RC filter in each line with 2 kΩ resistors will add a Johnson noise of 8 nV/$\sqrt{\text{HZ}}$; this will be 18 nV/$\sqrt{\text{HZ}}$ with 10 kΩ resistors [14].

[5] Similar circuits can be found in the ADS1296R datasheet and TI's SBAA181 document.

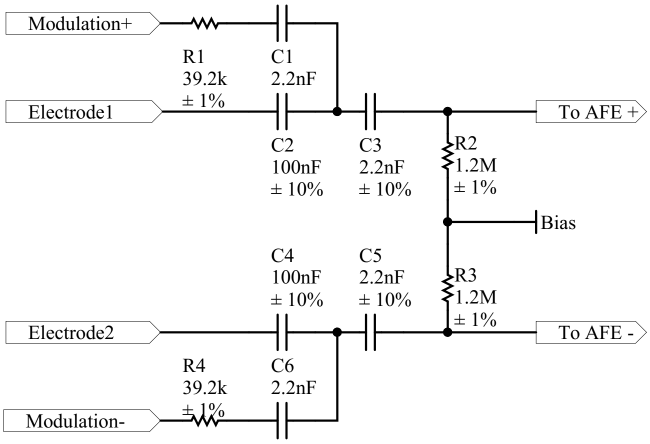


Figure 3.5 *The analog passive circuit used for respiration signal conditioning*

that integrates two dies. One die houses the accelerometer and the gyroscope and the other houses the AK8963 3-Axis magnetometer from Asahi Kasei Microdevices Corporation. This device also includes a digital motion processor (DMP) unit, which supports advanced motion processing tasks such as gesture recognition and pedometer. The low power DMP can offload some processing load from the host CPU and is useful in applications where the time or battery resources of the host CPU is tight. The DMP and magnetometer were not employed in this work.

The sensor provides two alternative communication protocols, SPI and I^{2}C. Due to limitations of the peripheral resources of the MCU, the I^{2}C is chosen which is shared between other I^{2}C clients. However, this restricts the data transmission speed to 400 kHz while it could be 1 MHz (2.5 times faster) with SPI.

3.1.4 Power

Power management in the wearable devices is a challenge since there are different and sometimes contrasting requirements. The battery lifetime is one of the most important factors. The capacity of a battery is limited because of its size limitation. Therefore, optimizing the power consumption of the system plays the main role in addressing this issue. Accurate monitoring of the battery, sufficiently good charging process, optimized and well managed power modes of the system and providing low power consuming and low noise voltage regulators are some of the other challenges in designing the power supply circuit for wearable devices.

The heart of the power management circuit in the designed device is BQ25120 from Texas Instruments, which handles charging and monitoring the battery, power supervisory and regulating two controllable voltage outputs. BQ25120 is able to communicate with the MCU of the system via I^{2}C protocol. By this communication, different features of the IC including the level of the output voltages and different thresholds can be set. It is also able to notify the processor about different events such as battery levels or input voltages. Power supervisory here means the ability to shut down the whole system if the input voltage is less than a particular value to avoid unexpected behavior from the circuit.

One important factor to have a highly optimized power management procedure is to have a complete control over different parts of the circuit. For instance if the user is not interested in the motion signals, all the parts of the circuit relating to the motion signal acquisition can be turned off (or put in standby mode). Some of the ICs used in the device provide an efficient power down mode, which can be used for this purpose. There are some other blocks such as SD card that do not include a (direct) access to put them to power down mode (although there might be a software solution to *decrease* the current consumption).

To address this problem, several load switches (different from the analog switches mentioned in section 3.1.5) are used for different blocks of the circuit. These switches are controlled by the MCU and disconnect the power from individual parts of the circuit if needed. The typical quiescent current of these switches is 100 µA.

A 5 V operating voltage is needed for analog switches since they carry USB signals when in USB mode. This voltage is provided by a voltage booster, which can be also put to power down mode and therefore, shut down the analog switches. Although the analog switches are always needed when the device is on, this solution helps minimizing the power consumption when device is off and therefore prolonging shelf life of the device.

The charging is automatically enabled when the presence of a host USB device is confirmed, which is done via sensing the VBUS pin of the USB connector by BQ25120. When BQ25120 detects a voltage higher than the set threshold for charging, it notifies the MCU and the required operations are performed by the MCU. Detection of the voltage at the VBUS pin of the USB connector is not so straightforward though. This complication is created due to the employment of the USB connector for two different tasks in the device i.e. ECG/impedance measurement and USB communication. See section 3.1.5 for a detailed discussion about this problem.

3.1.5 Dual Purpose USB Interface

One of key features of the designed device is its capability of being connected to electrode wires and the USB port through one micro USB connector. For this purpose, two analog switches were used to be able to switch between analog ECG signals (when the device is in measurement mode) and the digital data signals (when the device is in USB mode). The switches must have special characteristics to be able to preserve signal integrity for both modes. This section describes the specifications of the analog switches that are eligible for this task and briefly explains the chosen components.

Analog switches are offered with different number of channels having various switch types such as single-pole single-throw (SPST), single-pole dual-throw (SPDT) etc. For USB application, where only the two data pins are shared with other devices a two-channel SPDT analog switch is sufficient. However, in this work, there are four electrode wires connected through USB connector (excluding RLD), which implies the need of the more number of channels or multiple two-channel components. Most of the USB-compatible switches from Analog Devices consist of two channels. There exist a small number of quad switches such as ADG774 that unfortunately are not offered in small enough packages. Therefore, two ADG787 devices are employed in this work.

One important factor for an analog switch in USB applications is its On Resistance (R_{ON}), which explains the resistance of the channels when the switch is closed. The USB communication is a $45\,\Omega$ system, which indicates a source impedance of $45\,\Omega$ for the driver and a termination resistance of $45\,\Omega$ to ground in the receiver. Figure 3.6 shows the USB $45\,\Omega$ connection with the switch in the data lines. The source transmits $6\,\text{V}$ signals and the receiver should receive signals with an amplitude of $3\,\text{V}$. The $45\,\Omega$ system does the voltage division. However, when adding the switches, R_{ON} of the switch decreases the voltage received by the receiver. As an example, if the switch has a $5\,\Omega$ On Resistance the received voltage by the receiver will be $2.84\,\text{V}$.

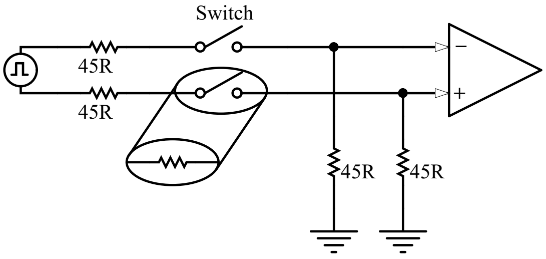


Figure 3.6 USB 45Ω *system with resistance model of switch*

Another important factor is the On Resistance flatness. The changes of R_{ON} due to either bias voltage, temperature, or supply sources cause difference in the amplitude seen by the receiver. The significant variations of R_{ON} with bias, causes difference in the rise- and fall-times of the signals. This increases the amount of jitter in the signal.

The next important factor is matching of On Resistance between channels. Since the USB signals are differential, any difference in the capacitance and resistance between two lines of data can cause serious problems. This is also important from the ECG measurement point of view. Any mismatch in a pair of electrodes decreases the common-mode rejection ratio of the circuit. It is worth mentioning that although multiple switches (here there are two switches) might imply some impedance difference between channels, this difference is smaller than the wire impedance or the tolerance of the input resistors of ECG conditioning circuit. Therefore, having multiple switches does not affect the ECG quality measurement. Both of the USB data lines are connected to one of the switches.

The bandwidth of the switch should be also taken into consideration. There are three speed categories for USB: Low-, Full- and Hi- Speed, which correspond to 1.5, 12 and 480 Mbps, respectively. Different USB versions support several of bit rates. USB1.1 supports Low- and Full-Speed and USB 2.0 supports all three speed categories mentioned. Therefore, the bandwidth requirements depend on the category of the speed considered in the system. The UART to USB converter IC (FT230XQ) supports up to full-speed communication that is also supported by the switches. The bandwidth of the switches are larger than the desired bandwidth of the ECG signal.

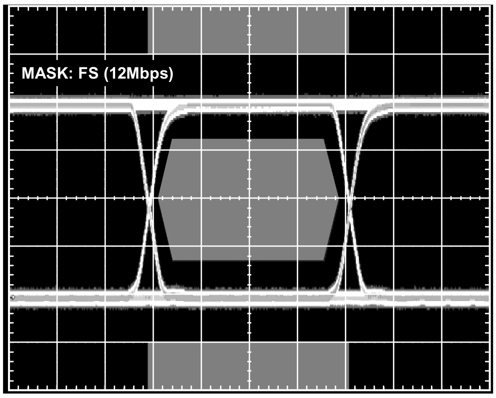


Figure 3.7 *Eye pattern of ADG787, 12 Mbps,* $V_{DD} = 4.2V$*. Copyright: from datasheet of ADG787*

Many of these specifications are evaluated using an *eye* diagram[6]. For instance, the jitter and mismatching of the differential lines of the signals appear as the thickness and the vertical symmetry of the eye diagram, respectively. This diagram can be compared to USB specifications to see if the system (including cables, connectors, PCB tracks, and switches) satisfies the requirements. Figure 3.7 shows the eye diagram of ADG787.

ADG787 is a single supply dual analog switch from Analog Devices (AD). Table 3.2 shows the specification of this IC. It is worth mentioning that in this family of AD's switches, when there is no VDD present, the state of the switch cannot be guaranteed, unlike conventional mechanical switches such as relays.

Table 3.2 *ADG787 specifications*

Parameter	**Typ.**	**max**
On Resistance R_{ON} (Ω) ($V_{DD} = 2.7V$)	4	5.75
On Resistance Match Between Channels ΔR_{ON} (Ω) (V_S = 1.5 V)	0.07	0.3
On Resistance Flatness (Ω) ($V_{DD} = 2.7V$)	1.6	2.3
-3dB Bandwidth (MHz)	145	-
Source Off Leakage Current (pA) ($V_{DD} = 3.6V$)	±10	-
Channel On Leakage Current (pA) ($V_{DD} = 3.6V$)	±10	-
Power Current (μA) ($V_{DD} = 3.6V$)	0.005	1

Figure 3.8 shows the schematic of the input section that includes switches, USB connector and ECG electrodes. It can be seen that the four lines of the ECG

[6] Refer to the document from ON Semiconductor

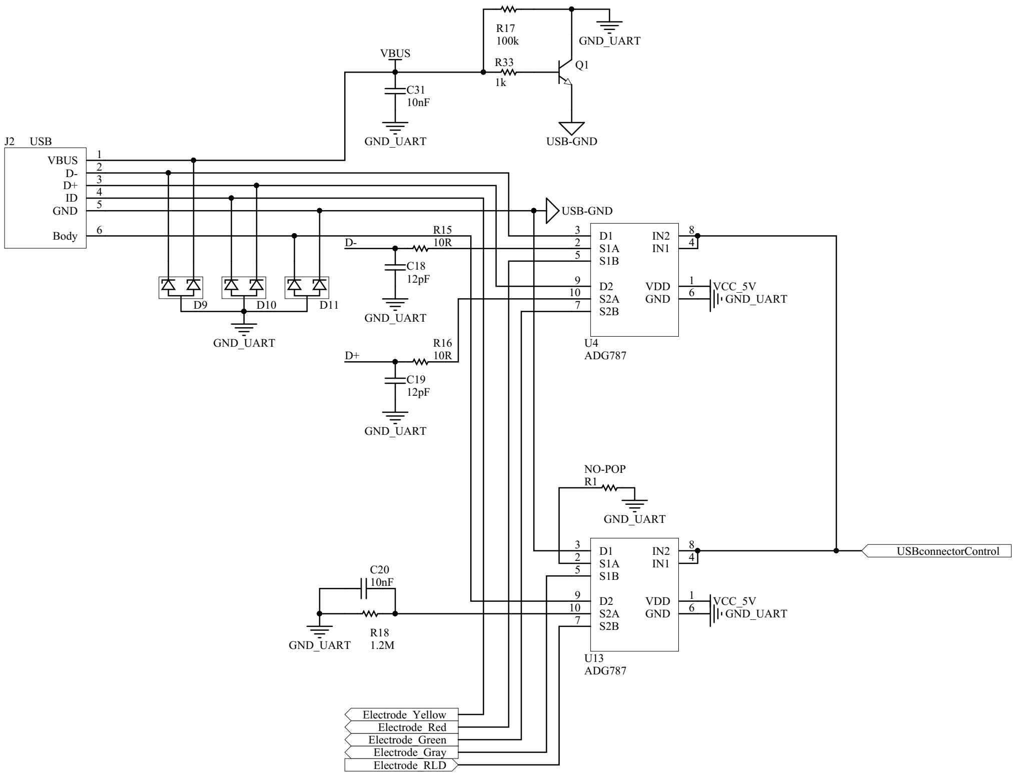


***Figure 3.8** Circuit of analog switches and USB insertion detection*

electrodes pass through two ADG787 switches. Note that the V_{CC} pin of the USB connector does not go through any of the switches. The other three pins needed for USB communication are connected to switches. There is another pin in the USB connector that is free and is directly connected to one of the ECG measurement inputs.

The shield of the USB connection is used for right-leg drive (RLD). This pin is connected to the rest of the system through one of the switches. This is done to overcome possible problem of voltage conflicting when the device is connected to an USB host device, in which the shield is most often connected to the GND and should not be driven by another source.

One of the challenges in this configuration is the detection of a USB host device connection. As can be seen in Figure 3.8, when the device is in ECG mode, the switch number 1 of U13 is connected to the Gray electrode. In this case, if a USB host device, which sources the V_{CC} pin by a 5 V voltage, is connected to the system, its presence cannot be sensed since the ground of the USB connection is not connected to the ground of the system. In other words, when the device is in ECG mode, the USB V_{CC} pin is floating. To tackle this issue, the ground pin of the USB is shorted to the ground of the circuit through a BJT transistor (Q1) that is enabled when the 5 V is present at the VBUS pin. When there is a voltage source present at VBUS pin of the USB connector, the transistor is in saturated mode and connects the digital ground of the system to the GND pin of the USB connector. Thus, the power management IC can evoke an event in the microcontroller. When in off mode, the transistor ideally disconnects its emitter, which is connected to gray electrode, from its base and collector. The transistor used for this purpose is MMBTH10L from ON Semiconductor, which has a low Base-Emitter and Base-Collector capacitance (about 0.7 pF).

The diodes after the USB connector are for protecting the inputs of the analog switches and the other components in the signal chain from the input overvoltage e.g. ESD events. The diodes are ultra-low capacitance unidirectional ESD protection diodes with leakage current of 10 nA at reverse voltage of 5 V from NXP Semiconductors. The diodes have a reverse working voltage/ reverse standoff voltage (V_{RWM}) of 5 V.

Since the operating voltage of the AFE is much less than 5 V (about 3 V), the internal protection diodes of the AFE could be in conducting state while the external ones are in off mode. This situation, in which the internal diodes or even the component can be damaged, is averted by limiting the current via the resistors in series (2 × 39.2 kΩ) to a level well below the accepted value (10 mA). On the other hand, for the impedance measurement path those resistors cannot be tailored since it directly affects the impedance measurement or even saturates the measurement circuit. This problem is addressed by providing additional 3.3 V protection diodes for impedance measurement path.

3.1.6 MicroSD card

Secure digital (SD) card is a type of removable flash memory for storing data, which is used in variety of portable devices such as cameras, mobile phones, media players and medical wearable devices. SD cards are available in different capacities and packages. Micro SD card is suitable for small devices. The benefits of SD cards include the following items.

- SD cards have non-volatile memory, which can keep the data while there is no power source present in the circuit. This also denotes the possibility of keeping the data with no power consumption.
- SD cards are available in lightweight and small size packages (microSD).
- They are available in high storage capacity, currently up to 2TB.
- SD cards allow fast access that is the key feature in high-speed applications such as high-resolution video frames with high frame rate.

There are two alternatives for communicating with a microSD card; SD mode and SPI mode. In this work, SPI mode is used and all the descriptions of SD card communication is based on this protocol in this report. Table 3.3 lists all the pins of a μSD card and their role in SPI protocol.

***Table* 3.3** *MicroSD card pinout description*

Pin	SD Mode	Description	SPI Mode	Description
1	DAT2	Data Line [Bit 2]	RSV	Not Used
2	CD/DAT3	Card Detect/Data Line [Bit 3]	CS	Chip Select
3	CMD	Command/Response	DI	Data In (MOSI)
4	VDD	Supply Voltage	VDD	Supply Voltage
5	CLK	Clock	SCLK	Clock
6	VSS2	Supply Voltage Ground	VSS2	Supply Voltage
7	DAT0	Data Line [Bit 0]	DO	Data Out (MISO)
8	DAT1	Data Line [Bit 1]	RSV	Not Used

From Table 3.3, it can be seen that there can be up to four data lines in SD mode, which allows the communication to be four times faster. For SPI mode though, four lines in total are needed, MOSI (Master Out Slave In), MISO (Master In Slave Out), SCLK (clock, which is driven by the controller) and CS (active low Chip Select). From operating voltage point of view, the compatibility of the logic levels can be guaranteed since both MCU and the SD card are powered with the same voltage source. It is worth noting that most of the SD cards allow operating voltage in the range of 2.7-3.6V. Although this should be confirmed in the initialization process by reading the OCR (operation conditions register of the SD card) register.

An interface IC, CM1624, connects the μSD card to the MCU. This chip provides

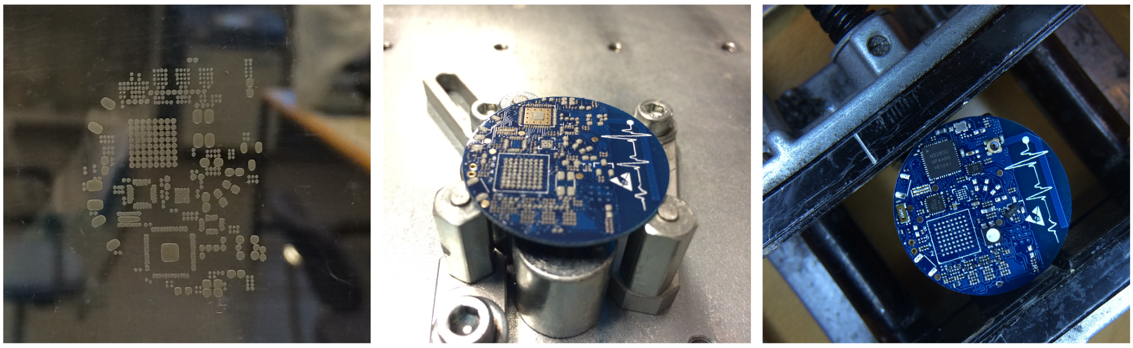


Figure 3.9 *Some stages of soldering the components for the prototype PCB*

not only protection diodes[7] for each line, but also an EMI filter and the required passive components such as pull-up resistors.

3.1.7 Printed Circuit Board

The first prototype batch of the printed circuit boards (PCB) where manufactured by a PCB company and the components were assembled manually using the university's facilities. For putting the solder paste on the PCB, a stencil printer from Stencils Unlimited was used. The components (about 200 in total) were placed on the board by hand. The board was reflowed in an oven twice, once for the components on the top layer and once for the components on the bottom layer. The BGAs were finally soldered using a hot air gun. Figure 3.9 shows some stages of the soldering process.

The final version of the device was ordered as assembled by the manufacturer after a couple of corrections and improvements. The PCB had four layers and a radius of 1.65 cm. The PCBs were panelized in a 28 cm×15.2 cm board including 3×5 devices. Figure 3.10 shows an illustration of the panel and an enlarged individual board. Each individual board had some supplementary components around it (for debugging purposes) and was separated by routes and breaking holes from the actual device. All the schematics and the PCBs were designed by Altium Designer DXP version 15.1.13.

[7] The diodes prevent high transient voltages resulted for example from hard plugin from reaching the MCU.

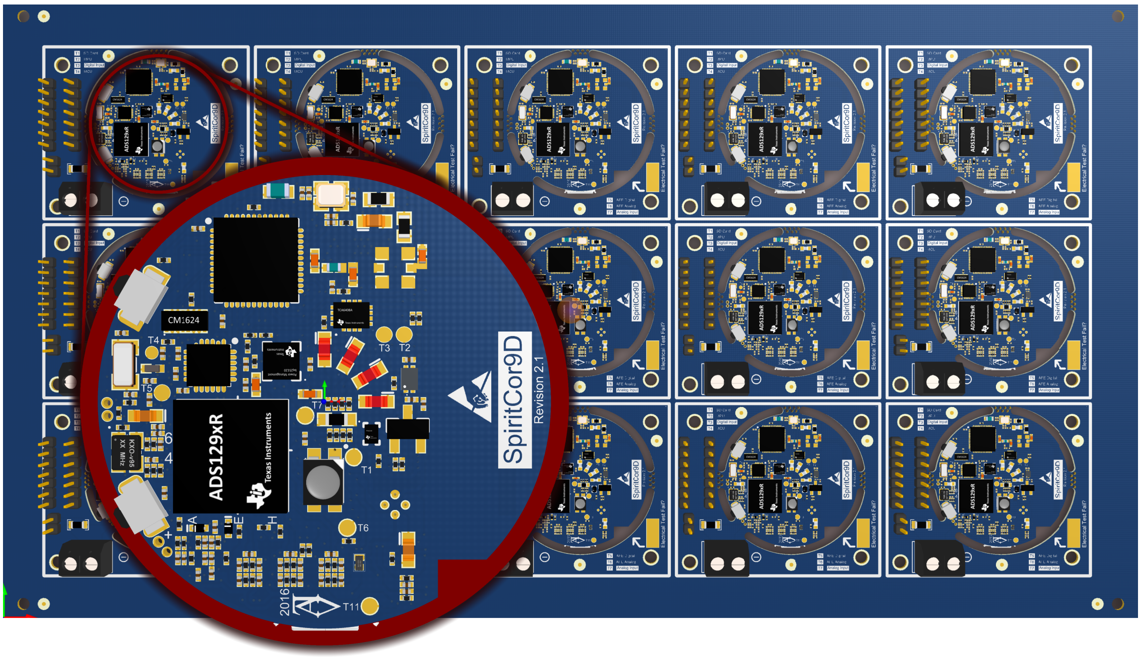


Figure 3.10 *A 3D view from Altium Designer of the Final panelized PCB*

3.2 Embedded Software

The device has a microcontroller that requires a firmware to handle all the required tasks. The microcontroller has a 32-bit ARM Cortex-M4F processor, 512kB flash memory, 64kB random-access memory (RAM) and additional peripherals. The programming has been done using GNU ARM toolchains and Eclipse. Flashing the MCU was performed through SEGGER J-Link debug probes using Nordic Semiconductor's tools such as nrfjprog and nRFgo Studio. The flash memory and RAM of the SoC are shared between the SoftDevice[8] and the application. Therefore, the application should be aware of memory resources occupied by the SoftDevice and leave them undisturbed for correct operation.

Figure 3.11 shows the structure of the embedded code from a high level of abstraction. The device is initially in power-down state. It jumps to the active state by either pressing the power button on the device or connecting it to the computer. The detection of both of these events are done using the sense capabilities of the

[8] SoftDevice is a precompiled and linked binary software that implements the required BLE (or other wireless protocols) stack. SoftDevice, which is developed by Nordic Semiconductor, is accompanied with required application program interface (API) that can be included in applications as high-level programming language interface.

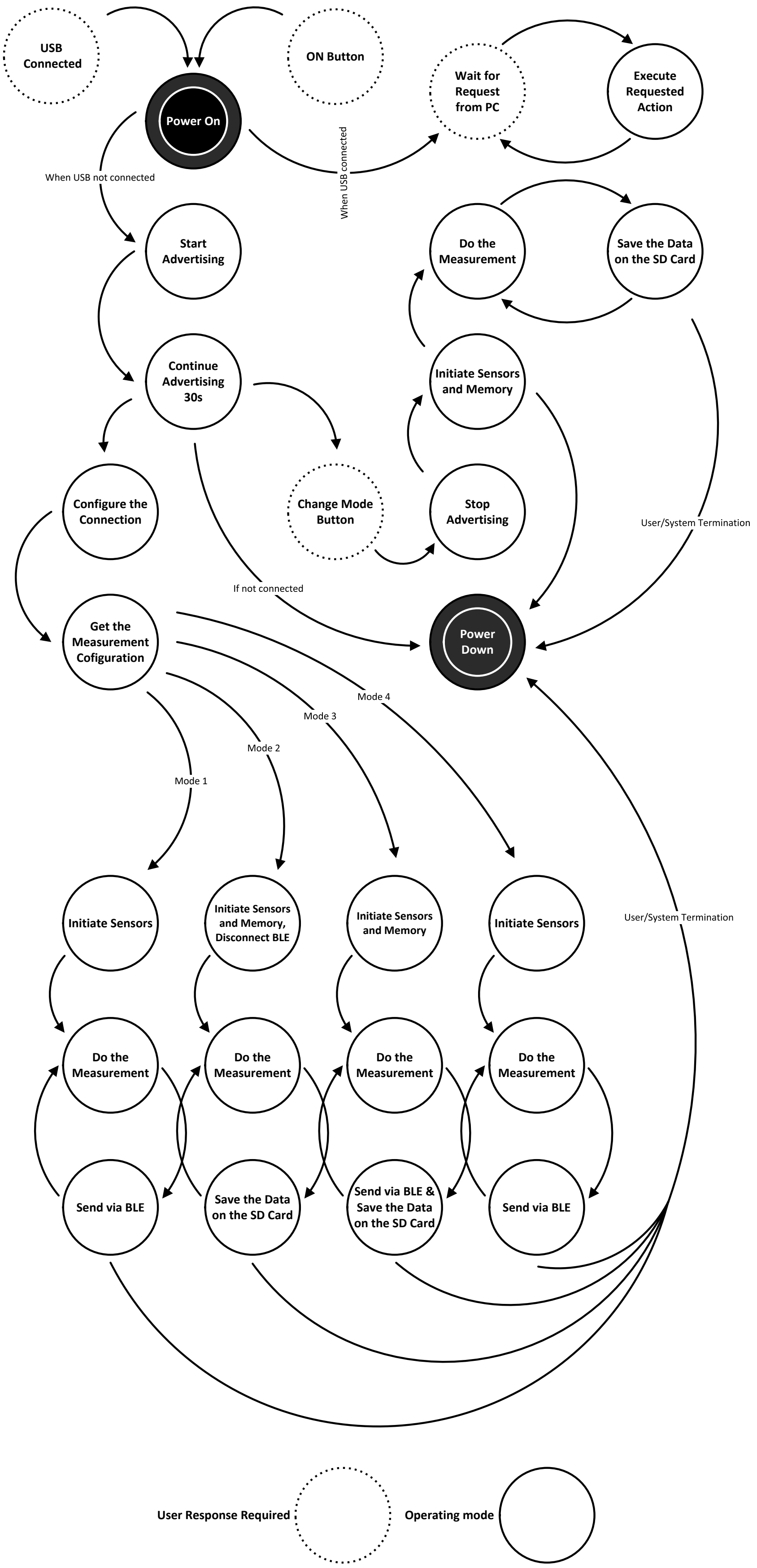


Figure 3.11 *Overall overview of the embedded software structure*

MCU's general-purpose input/outputs (GPIOs). When the device is connected to a personal computer (PC), it detects the presence of the PC, turns itself on and goes directly to *USB Mode*. After initializing this mode, the computer may send various requests, depending on the user's actions in the software and device responds to these requests. The response is either an announcement of its state or a report about the data stored in the memory. The initialization for the USB mode consists of managing the voltage of the different parts of the circuit, enabling the charger, connecting the USB switches and activating the SD card.

When the device is activated via the power button, it initializes the BLE and the device starts advertising. This state continues for 30 seconds and the device will go to power-down if no peer device sends a connection request to it. The user may terminate the advertisement and configure the device for *Mode 5* (see Table 3.4). The settings for mode 5 can be set using an Android device whenever it is available. These settings are then saved on the device so it can access them without the presence of the Android device.

Table 3.4 *Modes of the device. The USB mode (when connected to computer) is not considered in this list*

Mode	**Local Storage**	**Monitoring**	**Data Rate**
Mode 1	×	✓	Fixed
Mode 2	✓	×	Variable
Mode 3	✓	✓	Fixed
Mode 4	×	✓	Fixed
Mode 5	✓	×	Variable

The device can function in different modes, each with different pros and cons, satisfying various requirement in different applications. There are five modes of measurement (USB Mode is not considered as a mode). For enabling the first four modes, the user needs an Android device to start the measurement. The fifth mode can be started all from the device itself with no need of an Android device. Mode 1 is used only to observe the signals on an Android device. This mode is useful for checking the electrode-skin contacts as a verification step in the early stages of measurements. Second mode has flexibility in setting the measurement parameters. In this mode, the device disconnects itself from the Android device and continues with only storing the data on its local memory. This data can be then retrieved using the computer software. Since the Android resources on receiving and plotting the signals are limited, the sampling frequencies are not changeable when using modes

1, 3 and 4.

Mode 3 provides a safe mode of measurement. In this mode, the user can both see the signals on the Android device and store them locally (on the device memory). This mode is useful for the applications in which the measurements are short and therefore, the battery life is not critical. It also has the benefit of saving the data safely, since any problem in BLE connection would not affect the storing procedures. Mode 4 is implemented so that the Android device carries the load of storing the information. The device performs the same tasks as mode 1 and the Android device instead of only monitoring the signals, it also stores them. In this mode, the user might experience difficulties with not enough powerful Android devices. In both modes 3 and 4 the user should be in the range of BLE communication. However in the third mode as mentioned earlier, the device is able to store the data even when a disconnection happens. In this state, the device starts advertising and tries to reconnect to the previous central device. It is worth mentioning that although the data rate cannot be increased in some of the modes, the channels can be activated and deactivated by the user from the Android device. This gives the user an additional opportunity to decrease the power consumption and data size by deselecting the unwanted channels.

Depending on the selected mode, the initialization process of the measurement can be different. In short, the MCU turns on the required blocks of the circuit, sets the required voltage values of the different power source's outputs, turns on and initializes the SD card, sets the data rates of each measurement channel, sets the bandwidths (applicable to motion data), starts the real-time clock (to keep track of the measurement duration) and finally enables the required interrupts.

Terminating the measurement, depending on the mode of the operation, may have different reasons. In the first, third and fourth modes, the measurement can be ended from the Android device. In all the modes, the user can end the measurement by holding the power button for 3 seconds. There might be system-related terminations too. For instance, when the device detects a low battery level, it automatically turns itself off. The time and reason of the termination is stored in the local memory and is read by the computer software along with the stored data.

The process of reading the data from the data converters, sending or/and storing it are the three dependent tasks that are implemented asynchronously. Reading the battery level, managing different errors and responding to user inputs e.g. pressing buttons are other events that the MCU must handle. A more detailed insight about the procedures is given in the next subchapters.

3.2.1 Sending the Data via BLE

For sending the data from the device (peripheral in the BLE communication) to an Android device (central), UART service is utilized due to the match of its design with this application. BLE UART is a proprietary BLE service that sets up an RX and a TX, with *write* and *notify* properties, respectively. The notify property enables a fast data transmission from a peripheral to the central since the packets are not acknowledged by the application layer in the Central. However, this feature comes at the price of higher probability of losing the data on air. Although, this complication is addressed well in defining the applications of the system and the modes that it employs for different needs. For more information about BLE protocol stack see [1].

The implementation of UART service from Nordic Semiconductor lets 20 octets in each packet[9]. Every packet sent from the peripheral has a packet header and a packet counter, leaving even smaller space for the data. This allows the central to simply recognize each packet and take the required action. With this implementation, the data from different sources e.g. ECG and accelerometer is packed individually, simplifying the sending signals with different sampling rates. A packet of a specific data type is completed when enough number of samples from that signal is gathered. The packet may also be a command or a request from/to the central.

The most important characteristic that should be taken into consideration about sending continuous and real-time data via BLE is that the data traffic might depend on various parameters and thus is not constant and predictable (similar to other protocols of data transmission, even via cable). Changing parameters may be the amount the interfering radio frequency (RF) signals or noise, the distance of the two devices and the available processing power of the central device. The important conclusion is that sending the data must be implement as a completely separate procedure than the procedure of reading the data from the sensors. The MCU should handle the transmission when it can, the receiving device is ready and the connection is able to accept and transmit the data.

[9] The limitation comes from the Bluetooth core specification. Refer to vol 3, part F, section 3.4.7.1 in Bluetooth core specification version 4.2

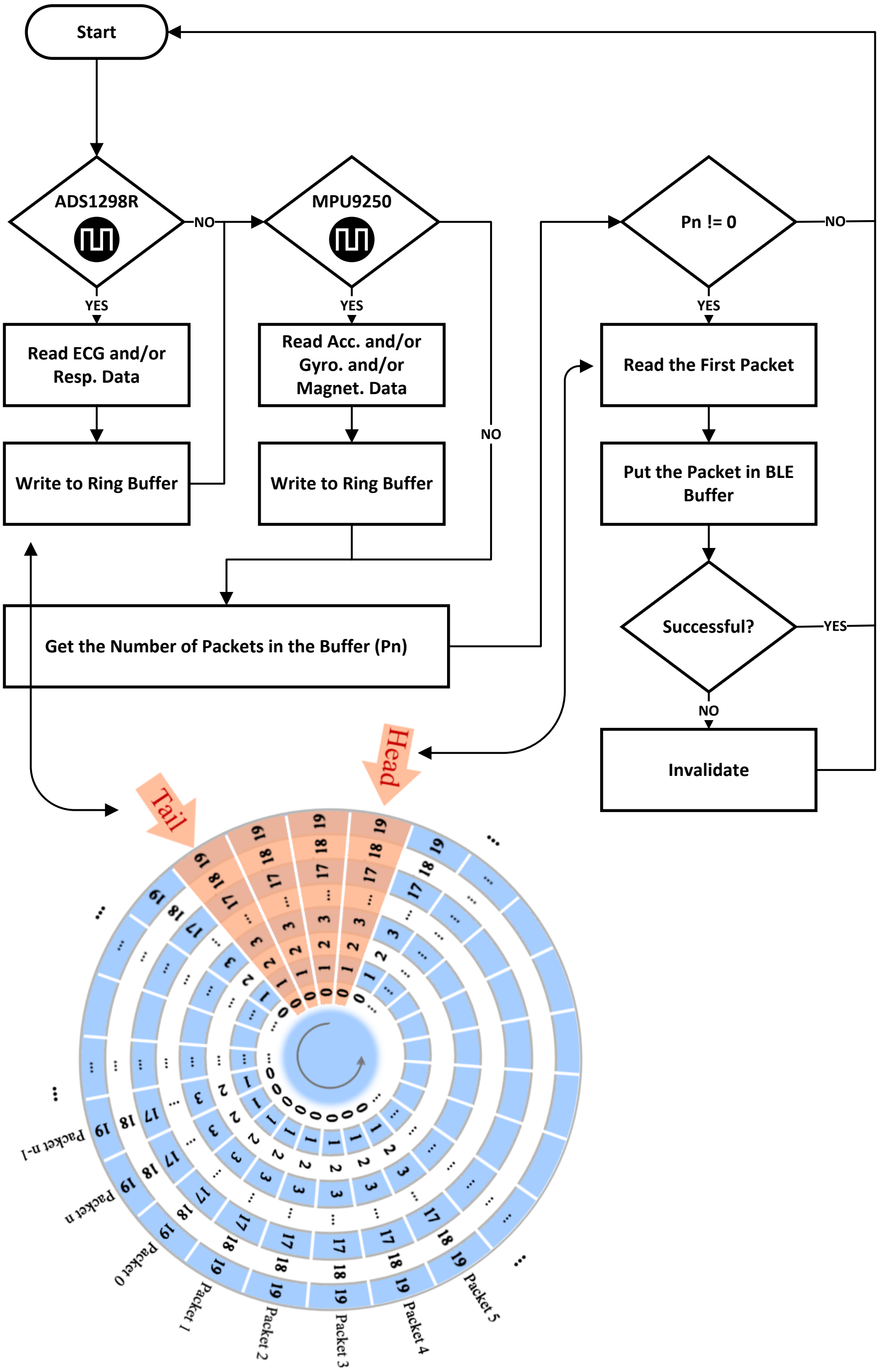


Figure 3.12 Overall overview of BLE data transmission procedure. When an interrupt comes from the AFEs, the MCU starts reading the required amount of bytes and then it writes it to the ring buffer. If there are packets left in the ring buffer, the MCU reads them one by one and puts them in the BLE buffer. If the BLE buffer does not accept them, the MCU makes sure to take back that packet to the ring buffer (invalidate them). Note that the ring buffer shown here is only for one data type e.g. ECG channel 1. There are multiple instances of this buffer for different channels.

A good solution to this condition is using a ring buffer. Figure 3.12 shows a simplified illustration of the procedure of putting BLE packets into the BLE buffer (different than the ring buffer itself) and the structure of the ring buffer. The ring buffer consists of n sections, each including 20 bytes. In other words, this buffer holds words of twenty bytes, not a single byte. The length of 20 bytes in this buffer is due to the previously mentioned data length in the communication. Therefore, each BLE packet, which is 20-byte long, is put into this buffer whenever it is ready to be transmitted.

Every write procedure increments the *tail* and the *head* is increased by one when a read takes place. The difference between the locations of these two pointers indicates the number of packets left in the buffer waiting to be transmitted. When the buffer is close to getting full, the process sets a flag indicating an error in the communication. Besides the importance of the reason of this error, it is very crucial that the CPU handles it in a fluent and proper way.

Writing to the buffer takes place when enough number of samples are gathered from the interface ICs. For instance, the output of ADS1296R includes 18 octets of data; six measurement channels each having a resolution of 24 bits. Only four channels are used in the device and the other two channels are always in power-down. Only 16 bits of each sample is transmitted via BLE (to save some resources in this low power device), thus, each packet can include three samples of data from all the measurement channels, consuming 18 bytes of the packet. Note that every BLE packet needs some space for additional information such as packet counter and packet header.

Reading from the buffer occurs when there is some data left in it and when the CPU has time for it. When the data is read from the ring buffer it is put into the BLE buffer. If this procedure fails, meaning that the BLE buffer has no space left for this packet, the packet should be returned to the ring buffer.

The significant benefit of using a ring buffer is its usage of pointers. Contrarily, a simple buffer must shift all the data fields when a single part of it is read or written. Ring buffer avoids this expensive memory relocation by tracking the pointers (head and tail) and simply overwriting the already-read sections. This memory relocation is more compute-intensive especially when having a word instead of a byte for each section of the buffer.

A very important factor in the implementation of this scenario, which is not pointed out in Figure 3.12, is that most of the parts of the code are designed to be executed in the *non-blocking* mode. For instance reading data from ADS1296R requires a transmission of 21 bytes on the SPI bus. This is done by starting the SPI bus for reading the required amount of bytes. After this point the CPU is free to execute the rest of the code e.g. start another transmission on another interface. In this way, the CPU avoids any blocking and makes sure that it can handle all the coming interrupts at the right time. The SPI starts reading the data and notifies the CPU when it has all the required data in its buffer and ready to be picked up by the CPU. In the worst scenario that the transmissions from all the sensors must happen at the same, two SPI buses are reading and writing data from/to AFE and to SD card, I^2C protocol is reading from MPU9250, and the MCU can rest in the power down mode.

3.2.2 Storing the Data

Saving the data to the local SD card requires gathering 512 byte of data, due to the page size of the memory card. Each page of data has a header that indicates the type of the data, number of bytes of data in that page and a page counter. Each page represents only one data type and is written to the SD card when a 512-byte long array in RAM is filled with data. At this time, the writing to SD card starts and the new coming samples are stored in the second 512-byte long array. Repeatedly, when the second array is filled with data, it is transmitted to the memory and the new samples are now written to the first array. Therefore, two 512-byte long arrays are needed for each data type.

This implementation makes sure that writing to the memory and reading the new samples can happen at the same time and individually for each signal that is being measured. Although, it might be a RAM hungry implementation for MCUs with limited resources. Assuming that there are seven signal types, 4 kB of RAM is needed.

Figure 3.13 shows a simplified version of the saving implementation. If an interrupt from ADS1296R is received, the MCU starts reading the data. After the reading is completed, the data is put into the free array of the buffer. If any of the sections (arrays) of the buffer (for any of the data types) is full, the CPU iterates in the implemented state machine for saving that array. This process starts after labeling

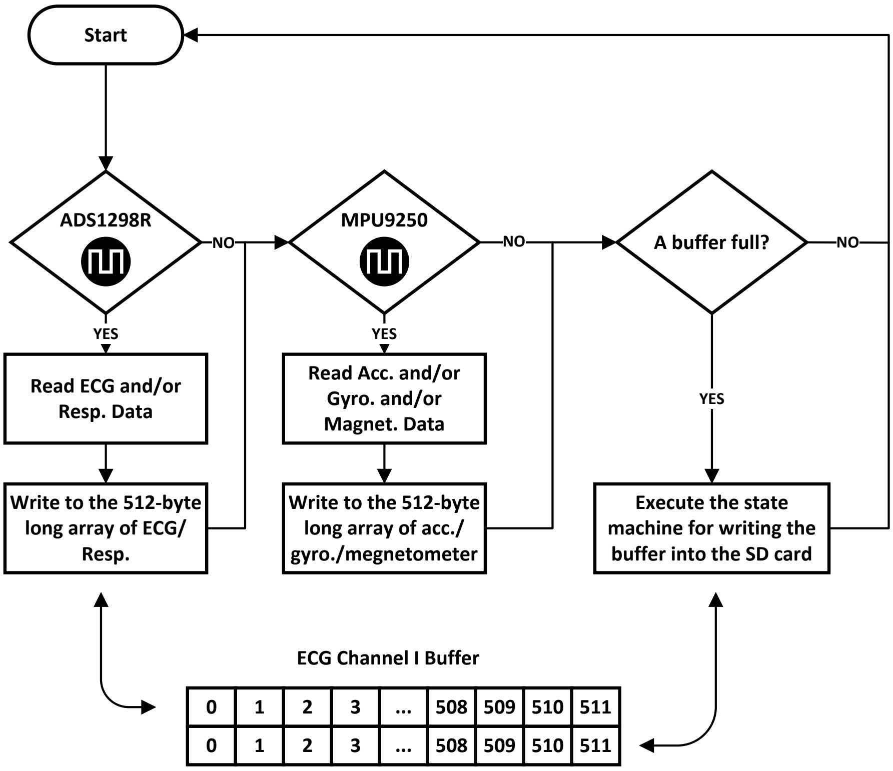

Start
ADS1298R
NO
YES
Read ECG and/or Resp. Data
Write to the 512-byte long array of ECG/ Resp.
MPU9250
NO
YES
Read Acc. and/or Gyro. and/or Magnet. Data
Write to the 512-byte long array of acc./ gyro./megnetometer
A buffer full?
NO
YES
Execute the state machine for writing the buffer into the SD card
ECG Channel I Buffer
0
1
2
3
...
508
509
510
511
0
1
2
3
...
508
509
510
511

Figure 3.13 *Overall overview of the locally storing procedure of the measured data. When the data from any of the AFEs comes, the MCU reads it and puts it in the buffer. If the buffer is full, the MCU starts saving that 512byte-long of the buffer to the SD card.*

that array with an appropriate header.

There is one page of information of the current measurement saved at the beginning of each measurement. This info page stores the information about the enabled channels in the measurement: data rate, resolution, internal gain (if available for the enabled channels) and full-scale. Some information related to the measurement sessions are also stored in this page: start and end time of the current measurement, and battery levels at the start and end of the measurement.

When the data is to be sent to a computer, the user needs to know what measurements are made with the device, the time and date of the measurements, subject IDs and possibly other information, before starting extracting a measurement. To avoid going through all the memory sectors to find the starting point of the measurements, a specific page is considered to keep track of all measurements. This page that is located in sector 2 of the memory stores information about the start and stop time of the measurement, subject ID, start address of the measurement data, mode of the measurement and the reason of termination. Reading this single page and transferring it to the computer reveals all the information of all the measurements. It is worth mentioning that although this brings a very suitable way of having an abstract about all the measurements, it brings a limitation for the number of possible measurements, depending on the amount of information saved for each measurement and number of pages considered for this information.

There are other blocks of information in particular sections of the SD card that the MCU uses for other matters. For instance, it includes some hard-coded values about the device's name, serial number, software and hardware versions and other information that may be helpful in identifying the devices. Another example is the page used for storing the errors that the device faces during the measurements. These error logs can be extracted, analyzed and if desired sent to developers via the computer software.

3.2.3 Optimizing the Power Consumption

Reducing the power consumption increases the battery lifetime and therefore the user can perform longer measurements with the device. To have an efficient battery usage, not only the circuit needs a careful design, but also the CPU's code requires various considerations.

As noted in chapter 3.1.4, there are several load switches placed in the hardware to be able to turn off the unused parts of the circuit. The MCU controls these through an I/O expander chip (since there are limited I/Os provided by nRF52832). Consequently, it also needs to make sure about a proper power-up of these parts of the circuit.

With the aid of the power management IC, BQ25120, the voltage output of regulators can be controlled through I^{2}C communication. Lowering the operating voltage might decrease the current consumption of the system. Although, this is limited by the component with the highest minimum operating voltage, which is the SD card. Its minimum supported operating voltage is 2.7V, when the storing mode is in use.

All the unused sections of the circuit should be put in the low power mode. The unused channels of ADS1296R, modulation and demodulation blocks when the respiration is not needed, the unnecessary sensors in MPU9250 e.g. magnetometer are some examples.

Using the internal clock of the ADS1296R decreases the power consumption of the system too. The external crystal provides a more accurate clock for the system and therefore a more precise sampling rate, which can prevent problem in the post processing of the signals e.g. when synchronizing the data with another device. Therefore, a compromise needs to be made between having a longer measurement or a more accurate one.

In the modes that the BLE transmission is enabled, lowering the amount of sent data leads to a longer connection interval and therefore a longer time that the BLE module can stay in power down mode. Note that for achieving a larger connection interval, the peripheral device needs to request it from the central device. The BLE module needs to wake up to respond to the central device at the time of the connections, even if there is no data to be sent.

Finally, putting the MCU into the standby state has a significant effect on the battery lifetime. The processor must first make sure that all the procedures e.g. writing to the SD card is completed and then execute the WFE[10] ARM instruction.

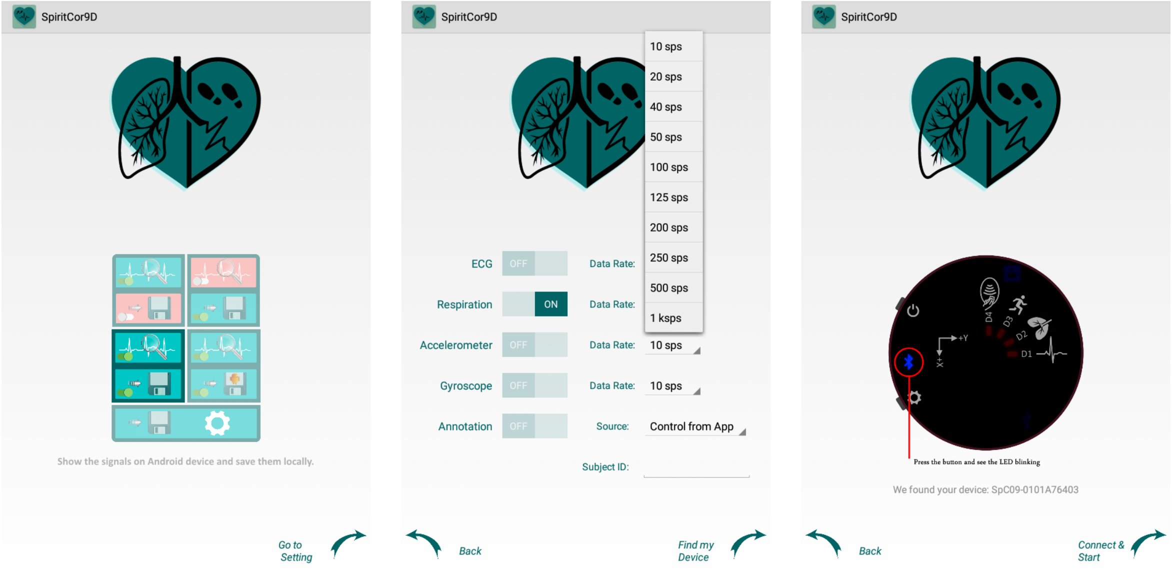
When the device is turned off by the user, all the blocks of the circuit should be turned off by the MCU by either disabling the load switches or activating the power-down pins of the ICs. All the blocks of the MCU, such as UART, SPI, I^{2}C and the BLE module, should be turned off. Then the CPU itself is put to its deepest power saving state[11], after enabling the sense mechanism for the GPIOs that are connected to the power button and USB detection circuit. These signals can wake up the CPU. Note that after waking up from the system off mode, the CPU resets and the required initialization phases should be handled again.

3.3 Android Application

An Android application is designed for this system to be able to start and control a measurement session, and monitor the measured signals. This application is designed using Android Studio and with API 18 (Android 4.3). The BLE UART service implementation of the UART application provided by Nordic Semiconductor is employed.

[10] WFE is an ARM instruction that suspends the execution in the CPU until an interrupt is triggered or an imprecise data abort happens (unless masked by the CPSR A-bit). For more information see ARM's documentation.

[11] This mode is called system off mode, which is enabled using the *SYSTEMOFF* register.



ECG OFF Data Rate:

Respiration ON Data Rate:

Accelerometer OFF Data Rate: 10 sps

Gyroscope OFF Data Rate: 10 sps

Annotation OFF Source: Control from App

Subject ID:

We found your device: SpC09-0101A76403

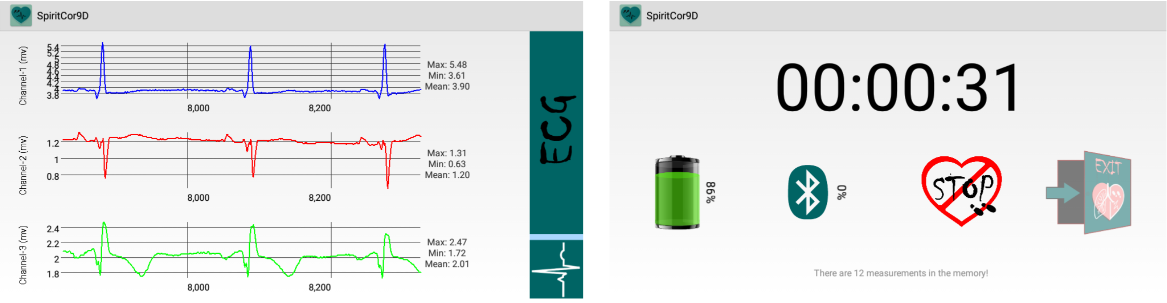
Figure 3.14 *Some selected pages of the Android application. The user starts by choosing the mode, required channels and their settings and subject ID. The application then starts scanning for appropriate devices and shows the serial number of the found device to the user. The user can connect to that peripheral device and start the measurement.*

The application consists of two main parts that are implemented as different *activities*[12]: measurement configuration and monitoring. In the measurement configuration section, the user selects the desired mode and several measurement settings such as the required channels and their sampling rates. The monitoring section, which is launched only if one of the modes 1, 3 or 4 is chosen, is designed to monitor the signals, show some statistics and control the measurement session e.g. stop the measurement. The *GraphView* package is used for plotting the signals and *ViewFlipper* configures a smooth jump between several pages in each part of the application.

Figures 3.14 and 3.15 shows some selected pages of the application. Figure 3.14 shows the configuration section of the application, in which the user selects between the modes and then the data rates. Figure 3.15 shows the measured ECG signals and the page in which some information from the measurement and the peripheral device is shown. In this page, the user can terminate the measurement and kill the application.

Figure 3.16 shows the steps that the application travels through according to the user settings. The user can change the data rates only if the second mode is chosen (or when configuring the fifth mode). In the case of modes 1 and 3, the sample rates

[12] For more information on activities in Android, refer to Android references.



There are 12 measurements in the memory!

Figure 3.15 *An example of the ECG monitoring in the Android application.*

are chosen to default values by the Android application to make sure that it works smoothly in receiving and updating the plots. This limitation comes from the limited bandwidth of the BLE and the expensive computation that the Android device performs in updating the plots[13]. More information about these limitations is given in chapter 5.2. The subject ID is not needed in the first mode since this mode does not store the data anywhere. After setting these variables, the application sends all the configurations along with the current time to the peripheral device and waits for an update about the status of the peripheral device in the beginning of the session. This information includes the current battery level and the number of measurements already stored in the local memory of the peripheral device. Then if the second mode was chosen, the peripheral device disconnects from the tablet/phone and continues doing the measurements and storing them. Otherwise, it jumps to monitoring pages.

When switched to monitoring activity, the data is continuously sent to the Android device. The application gets the packets, separates them according to their headers, reconstructs the values (since the values are in 2's complement format) and plots them graphically. If some packet is dropped on air, the two devices simply skip this error (If it is very important issue for the user, there are modes in which this problem never affects the stored data). Although, the application is able to recognize this missed packet based on the packet counter.

The start time of the session is always sent from the Android device. After that, the peripheral device keeps track of the time using its real-time clock (RTC) block. This block is disabled at the end of the measurement session for the sake of the power consumption.

[13] A workaround to this problem is to decimate the signals to a fixed sampling rate (independent of the chosen data rate). This feature is considered in the list of future updates of the system.

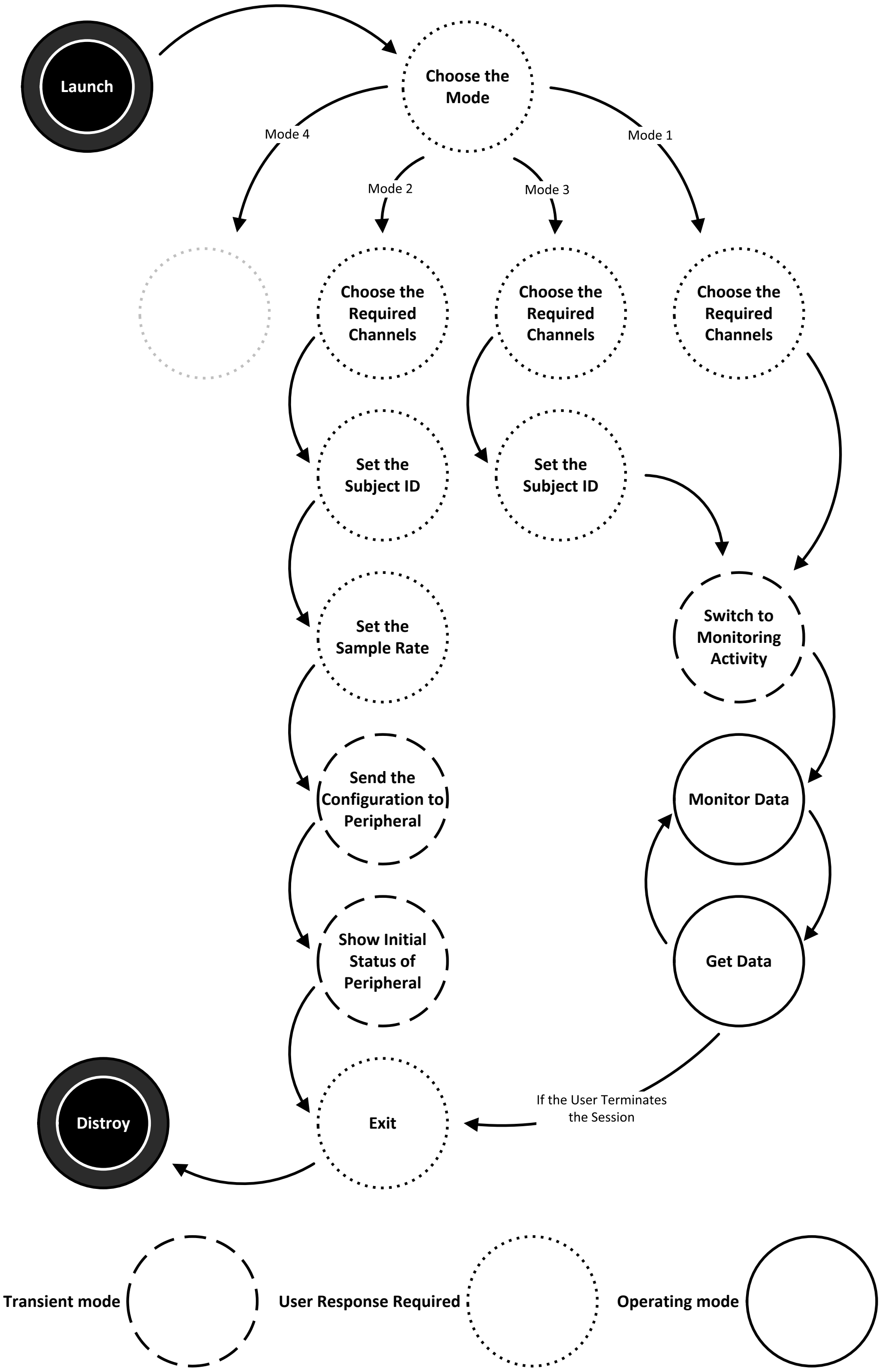

Launch
Choose the Mode
Mode 4
Mode 2
Mode 3
Mode 1
Choose the Required Channels
Choose the Required Channels
Choose the Required Channels
Set the Subject ID
Set the Subject ID
Set the Sample Rate
Switch to Monitoring Activity
Send the Configuration to Peripheral
Monitor Data
Show Initial Status of Peripheral
Get Data
Distroy
Exit
If the User Terminates the Session
Transient mode
User Response Required
Operating mode

Figure 3.16 *Overview of the Android application structure*

When the user starts the measurement from the Android application, it starts scanning for the devices with specific fields in their advertisement packets. Therefore, the application is able to recognize its matching devices. The information used from the advertisement packet for this procedure includes the name of the device, manufacture identifier and data. The data field holds the serial number of the device that can contain an encrypted code for a more secure finding procedure of matching devices. This serial number is shown to the user so that he/she can compare it with the serial number printed on the enclosure of the peripheral devices. In development phase, we experienced that sometimes the Bluetooth module is not able to find any advertising device. To solve the problem, the application always restarts the Bluetooth module before it starts scanning.

3.4 Computer Software

The device stores the measured data in its local memory. Therefore, a tool is needed to retrieve this data and store it in whatever format that the user needs e.g. EDF[14] [42], dat[15], CSV[16] etc[17]. The software can also provide additional features such as detailed monitoring of the battery level (instead of the three-level LED on the device), memory management, signal processing and analysis (not implemented at the time writing this report), hardware tests and reading error logs.

The software is designed using LabVIEW from National Instruments. The reason for adopting LabVIEW for this purpose was its high capabilities in graphical designs and its ability to employ other programming languages. Most of the processing parts are implemented in pure C that are built as DLLs, so that the LabVIEW software has easy access to them. The DLLs are implemented using Dev-C++. Some other parts are scripted using MATLAB that are called using MATLAB Nodes in LabVIEW[18].

[14] European data format

[15] MIT format. Refer to Physionet data collection for more information on this data format.

[16] Comma-separated values

[17] The choice between the data formats is vital in preserving the full resolution and all the information provided by the device. For instance, EDF format only accepts 16-bit resolution and therefore, 8 bits of resolution is lost in ECG and respiration signals. Although, it is able to store all the measurement information (e.g. subject id) in the same file as the data. On the other hand, MIT data format needs to create another file for measurement information (.hea), but it can achieve storing 24-bit data. The CSV format needs to save the values in a reconstructed way and therefore the number of decimal digits might limit the resolution. When CSV is selected, the software needs to create a separate info file that includes the measurement information.

[18] These parts will be transferred to C in future since MATLAB Nodes need MATLAB compiler to run which is problematic for the end users.

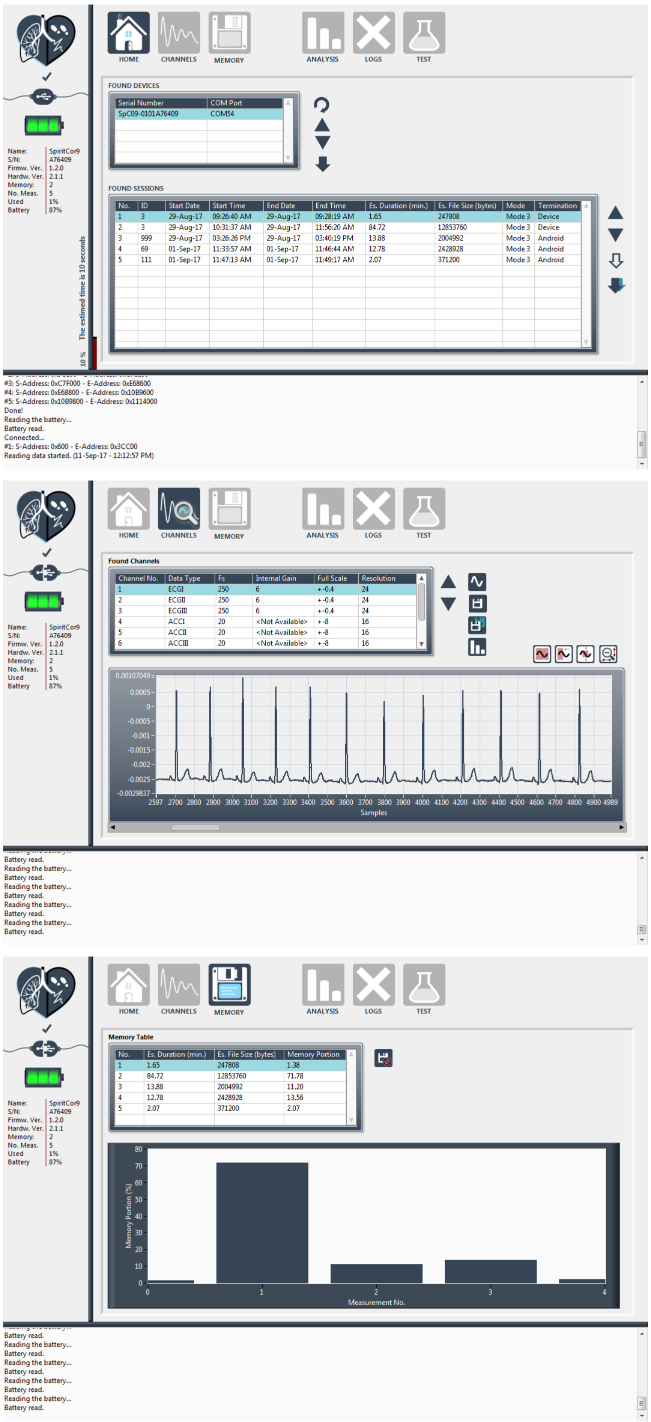
The software has three main parts: session information, channel information and memory management. The devices are automatically detected by the software by reading some registry keys and finding the name of the connection device. Therefore, the user does not need to find the COM port number of the device manually.

The protocol of communication between the computer (software) and the device is UART that has a request-response format. The computer always initiates a conversation and the device answers with the appropriate answer. For instance, the computer asks for the battery level and the device answers with the appropriate data.

The first picture in Figure 3.17 shows the first page of the software. As can be seen, all the information of the measurement sessions are shown in this page: subject ID, start time and date, end time and date, size and duration estimation, used modes and the reason of termination. It is worth mentioning that extracting this information does not need going through all the memory sectors. For more information about the location of this information in the local memory, see section 3.2.2.

The second picture depicts the page that the user is taken to after retrieval of a session. In this page, the details of the selected session e.g. sampling frequency and resolution are shown. The user can plot and export the channels. The export can be done for each channel individually or for all together (the number of created files depend on the chosen format). The sessions can also be exported all together. Additionally, some tools for an easy inspection of plotted signals are provided. After saving the desired sessions/channels, the user can empty the memory of the device (the bottom picture).

The software is implemented so that the RAM usage is very small. This is achieved by writing temporary files in the windows temporary directory. After receiving the data from the device and completion of the writing the temporary files, the software starts reconstructing them, frame by frame, and creating new temporary files. The software keeps track of all the created temporary files, their names and contents and delete them after closing the software. Therefore, the size of the available RAM on the computer is not a bottleneck for the proper functioning of the software. However, plotting the signals might require some large amount of RAM, depending on the size of the signal being plotted.



Name:	SpiritCor9
S/N:	A76409
Firmw. Ver.	1.2.0
Hardw. Ver.	2.1.1
Memory:	2
No. Meas.	5
Used	1%
Battery	87%

FOUND DEVICES

Serial Number	COM Port
SpC09-0101A76409	COM54

FOUND SESSIONS

No.	ID	Start Date	Start Time	End Date	End Time	Es. Duration (min.)	Es. File Size (bytes)	Mode	Termination
1	3	29-Aug-17	09:26:40 AM	29-Aug-17	09:28:19 AM	1.65	247808	Mode 3	Device
2	3	29-Aug-17	10:31:37 AM	29-Aug-17	11:56:20 AM	84.72	12853760	Mode 3	Device
3	999	29-Aug-17	03:26:26 PM	29-Aug-17	03:40:19 PM	13.88	2004992	Mode 3	Android
4	69	01-Sep-17	11:33:57 AM	01-Sep-17	11:46:44 AM	12.78	2428928	Mode 3	Android
5	111	01-Sep-17	11:47:13 AM	01-Sep-17	11:49:17 AM	2.07	371200	Mode 3	Android

```
#3: S-Address: 0xC7F000 - E-Address: 0xE68600
#4: S-Address: 0xE68800 - E-Address: 0x10B9600
#5: S-Address: 0x10B9800 - E-Address: 0x1114000
Done!
Reading the battery...
Battery read.
Connected...
#1: S-Address: 0x600 - E-Address: 0x3CC00
Reading data started. (11-Sep-17 - 12:12:57 PM)
```

Name:	SpiritCor9
S/N:	A76409
Firmw. Ver.	1.2.0
Hardw. Ver.	2.1.1
Memory:	2
No. Meas.	5
Used	1%
Battery	87%

Found Channels

Channel No.	Data Type	Fs	Internal Gain	Full Scale	Resolution
1	ECGI	250	6	+-0.4	24
2	ECGII	250	6	+-0.4	24
3	ECGIII	250	6	+-0.4	24
4	ACCI	20	<Not Available>	+-8	16
5	ACCII	20	<Not Available>	+-8	16
6	ACCIII	20	<Not Available>	+-8	16

```
Battery read.
Reading the battery...
Battery read.
Reading the battery...
Battery read.
Reading the battery...
Battery read.
Reading the battery...
Battery read.
```

Name:	SpiritCor9
S/N:	A76409
Firmw. Ver.	1.2.0
Hardw. Ver.	2.1.1
Memory:	2
No. Meas.	5
Used	1%
Battery	87%

Memory Table

No.	Es. Duration (min.)	Es. File Size (bytes)	Memory Portion
1	1.65	247808	1.38
2	84.72	12853760	71.78
3	13.88	2004992	11.20
4	12.78	2428928	13.56
5	2.07	371200	2.07

```
Battery read.
Reading the battery...
Battery read.
Reading the battery...
Battery read.
Reading the battery...
Battery read.
Reading the battery...
Battery read.
```

Figure 3.17 *Computer software. From top to bottom: Home page, Channels page and Memory Management Page*

4. EVALUATION OF SYSTEM PERFORMANCE

4.1 ECG Signal Quality

There are several important features of a mobile physiological monitoring device that should be evaluated. In this section, the quality of the measured signals by the device and its power consumption profile is presented. The quality of the measured signals were tested with several methods. As the first step, the measured signals with four different data rates are depicted in Figure 4.1. The data rates are 250 sps, 500 sps, 1 ksps and 2 ksps for the four panels in the figure, respectively. The measurements for 250 sps and 500 sps are done using the low-power mode of ADS1296R and internal clock source and the other two use the high-resolution mode and the external clock source.

The signals in Figure 4.1 show good noise performance of the device in even high data rates such as 2 ksps. Note that the architecture of the internal ADC of the ECG AFE implies averaging over fewer samples when the data rate is higher, which increases the level of the noise. This can be seen in panel (d) of Figure 4.1 that includes small fluctuations. The number of the samples averaged in low-power mode is half comparing to the high-resolution mode.

The four signals in Figure 4.1 were measured consecutively from the same subject laying on bed using conventional Ag/AgCl electrodes from Ambu (Bluesensor R-00-S/25) and a custom electrode cable, which included the required USB connector.

As a continuation of evaluating the quality of the measured signals, we compared our device with a device that already exists in the market. For this purpose, we chose *Faros 360* from Bittium Technologies Ltd[1]. Faros 360 is able to measure ECG up to 1000 sps with a resolution of 24 bits. Although, since its output is stored in

[1] Previously Mega Electronics Ltd, which was acquired by Bittium Technologies Ltd, a subsidiary of Bittium Corporation on November 10, 2016.

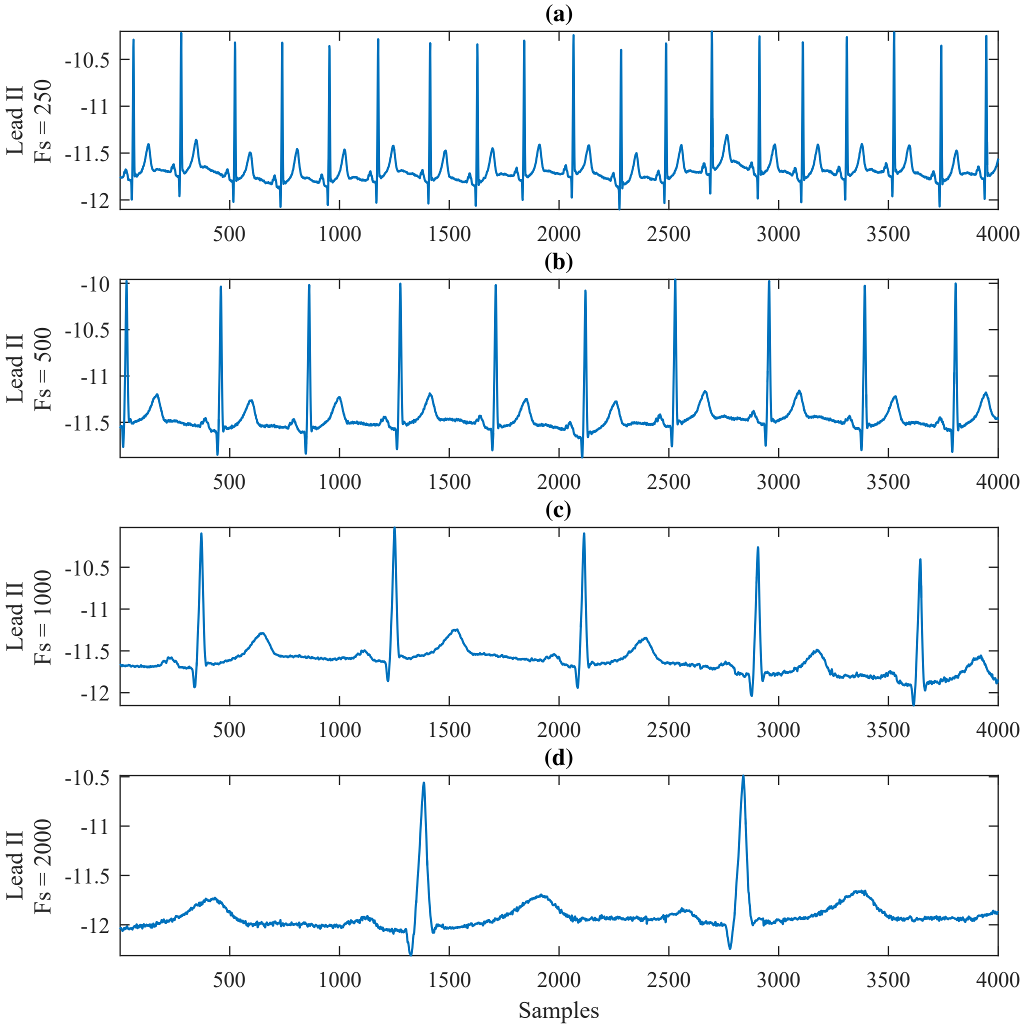


Figure 4.1 *Measured ECG signal by the device with data rate of (a)* 250 sps*, (b)* 500 sps*, (c)* 1 ksps *and (d)* 2 ksps*.*

the EDF format, only 16 bits of signals are preserved[2].

[2] The EDF format is capable of storing only 16-bit data. Although, this word length reduction during storing does not necessary mean that the resolution of the signal is lost. In many applications, this 8 bit of data (assuming a 24-bit ADC in the measurement chain) is dropped from the most significant part of the word. We think this is also the case with Faros 360 since the offset of the signal is perfectly zeroed. When removing the DC of the signal, it is usually fine (if the peak-to-peak value of the signal meets the criteria of the new dynamic range) to decrease the word length without losing resolution. It is worth mentioning that no information about this filtering was found in the header of the EDF file stored by Faros 360.

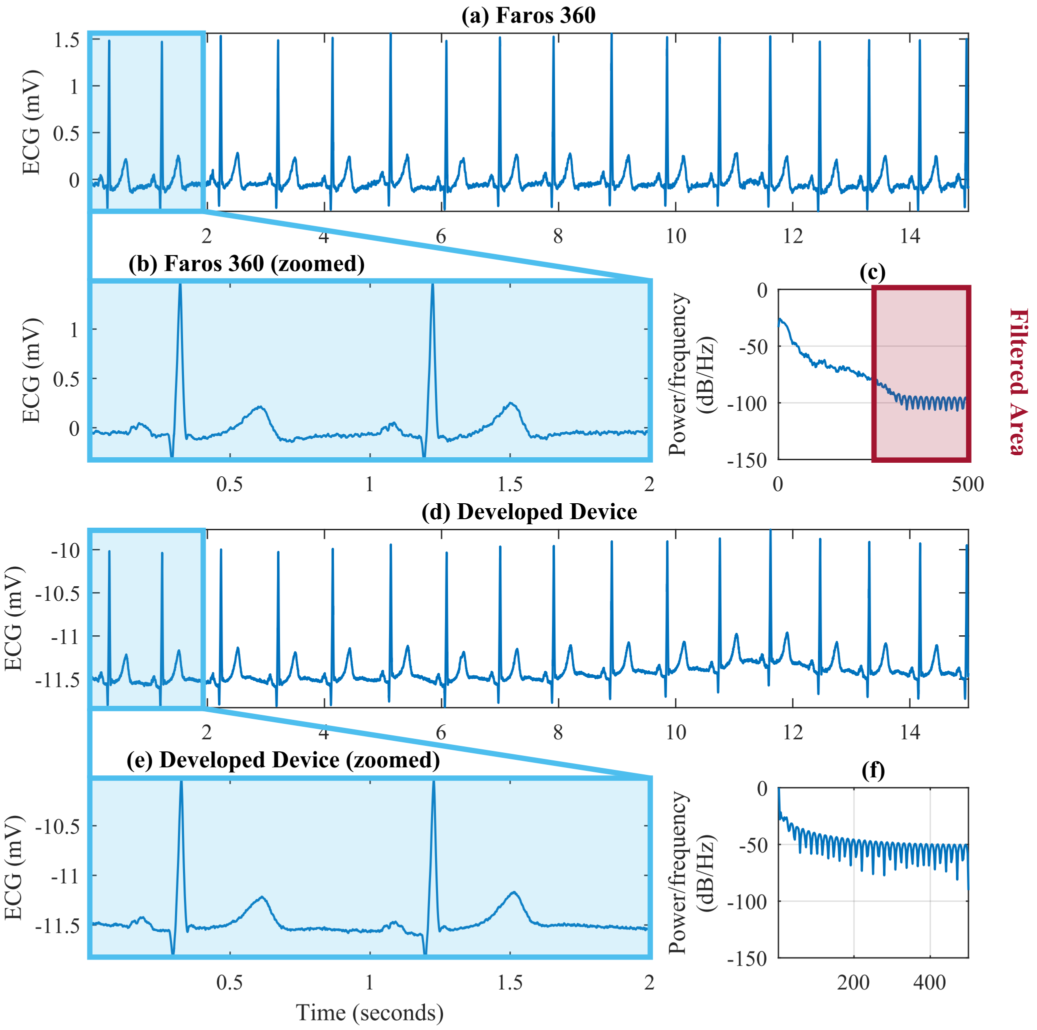


Figure 4.2 *Comparison of ECG signals measured by the developed device and Faros 360: (a) ECG measured by Faros 360 (15 seconds); (b) magnified view of the first two seconds of panel a; (c) power spectrum of the signal in panel b; (d) ECG measured by the developed device; (e) magnified view of the first two seconds of panel d; (f) power spectrum of the signal in panel e.*

The measurements were done simultaneously with the two devices. The electrodes used by the two devices were placed as close as possible to each other on the right arm and left leg to measure the lead II of the standard 12-lead ECG. The data rate was set to 1 ksps in both of the devices. Then, the two measured signals were synchronized using the inter-beat (RR) interval series of the heart. The R-peak detection was done using the Pan-Tompkin method [4].

It is important to remember that the bandwidth of the measured signal with the developed device is $0.262 f_{DR}$, where f_{DR} is the data rate. Therefore to have a fair comparison, we filtered the signal from Faros by a third order Butterworth low-pass filter with a cut-off frequency of 250 Hz. The filtering was done using the forward-backward technique to eliminate the undesired effects of the IIR filter. No other filters were applied on any of the signals.

Figure 4.2 shows the results. The first three panels (a, b and c) show the measurements from Faros 360 and the rest illustrate the ones gathered from the developed device. The signals in the a and b panels (and respectively the ones in b and e panels) are completely synchronized in time and one may compare them morphologically. Although, as explained earlier, the offsets are not equal since the Faros data is seemingly high-pass filtered. It can be seen that the quality of the measured signal by the developed device is no worse than the one from Faros, if not even better (there are actually less fluctuations in panel d). This illustration also confirms the process of handling the digital samples in the whole chain of the system, since the amplitude range of the two signals is the same.

In the third phase of the quality evaluation of the measured signals, we ran a test to rate specifically the noise of the device. In this test we bypassed the effect of the body, electrodes and electrode cables by simply short-circuiting the two inputs of a channel close to the USB connectors. For Faros 360, the two inputs were shorted using a resistance of 196 kΩ to simulate the electrode wire resistance in Faros[3]. The measurements were done in the same environment for both of the devices.

Figure 4.3 shows the result of the measurements. The developed device clearly imposes a better noise performance comparing to Faros 360 by about 18 µV lower peak-to-peak noise level. This evaluation may also be used to assess the number of noise-free bits of the measurement chain, which is largely affected by the quality of the circuit and PCB design. In the developed device and in a normal living environment (not a controlled room with a controlled level of noise[4]), at least 18

[3] In Faros 360, the patient safety resistors are placed in the electrode wires. The resistance between each snapper and its corresponding USB pin is 100 kΩ. Therefore, connecting two snappers together would build a 200 kΩ resistor (two 100 kΩ in series).

[4] Performing this measurement in a controlled environment would reveal the level of noise generated only by the device itself bypassing the variable effect of the environment noise.

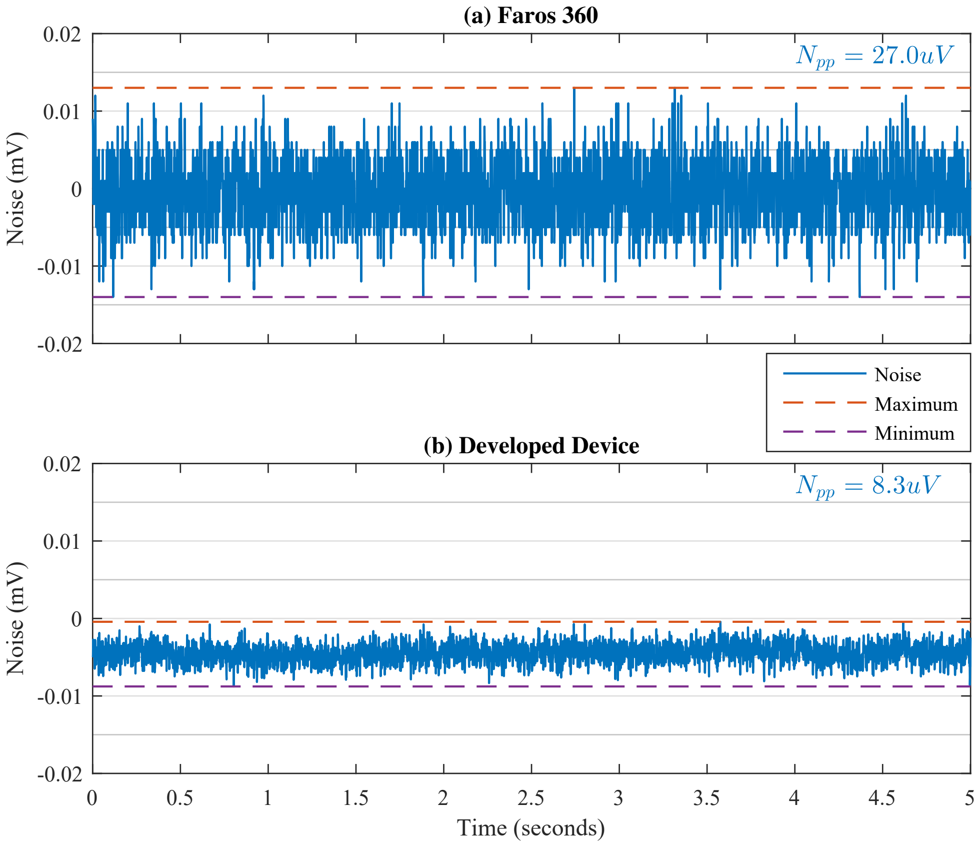


Figure 4.3 *Comparison of the level of noise in the signals measured by the developed device and Faros 360: (a) noise in the measurement by Faros 360; (b) noise in the measurement by developed device. The maximum and minimum of the signals are marked by two dashed lines. The peak-to-peak values for the measured noise in Faros 360 and the developed device are* 27.0 µV *and* 8.3 µV*, respectively.*

bits[5] of the 24 bits of the ADC are invulnerable against noise[6].

4.2 Respiration Signal Quality

As mentioned in section 2.2, one of the important parameters in measuring electrical impedance pneumography is the phase delay between the modulation and demod-

[5] The effective number of bits (sometimes referred to as flicker-free resolution) can be calculated according to the full-scale range of the ADC or simply by using the formula $\ln(\frac{fullScaleRange}{noise_{pp}})/\ln(2)$.

[6] According to the datasheet of ADS1296R, the input-referred noise is 5 µV peak-to-peak. The whole measurement chain in the developed device (excluding the electrodes and electrode cables) has noise level of 8.3 µV peak-to-peak.

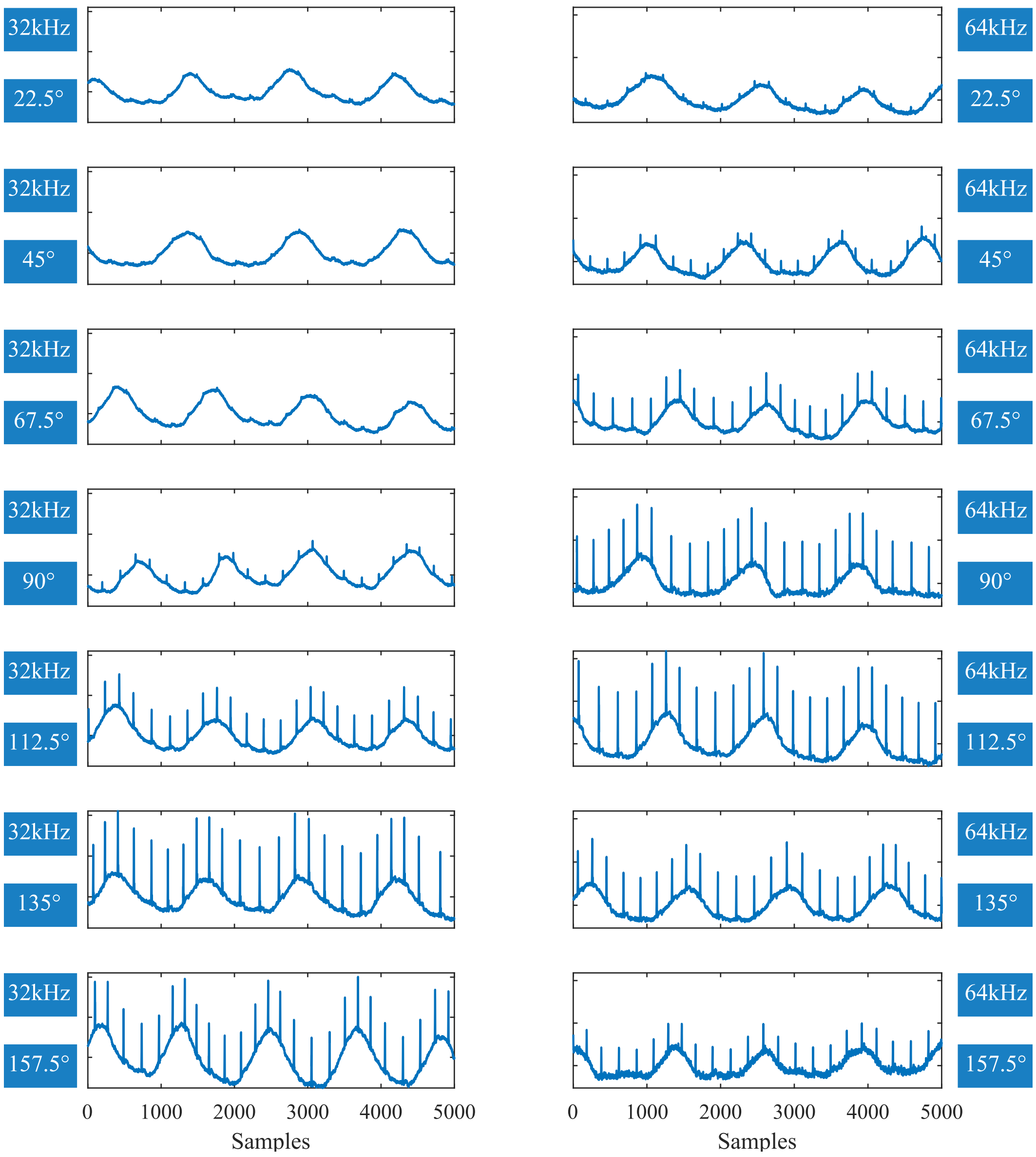


Figure 4.4 *Evaluation of the effect of frequency and phase of the modulation-demodulation circuit on EIP measurement. The modulation frequency is* 32 kHz *and* 64 kHz *for the left panels and right panels, respectively. Each row of the panels indicate one value of demodulation phase.*

ulation blocks of the measurement circuit. The whole signal path including the passive circuit, analog switches, cables and the body itself may change the phase difference. Therefore, it is necessary to test the device to understand which settings of the modulation demodulation pair provides the best signal quality.

Figure 4.4 illustrates the effect of changing the phase and frequency of the modulation-demodulation pair. Fourteen EIP measurements were performed, covering all the possible options provided by the AFE. The small boxes beside each panel shows the frequency of modulation (either 32 kHz or 64 kHz) and the phase values. The signals are measured with an internal pre-amplifier gain of 6, data rate of 250 Hz and are depicted with no filtering applied on them. However, the mean of the signals were removed for the ease of illustration. The y-axes have arbitrary values, but show the exact same range in all the panels. The measurements were done using bipolar electrode configuration and the electrodes were placed on the left and right midaxillary lines, lower than underarms.

In several of the cases, there are undesirable sharp artifacts that are due to the heart activity. It is worth mentioning that the results differ depending on the electrode location and the electrode configuration. Even though it would be possible to remove these artifacts by filtering later, a decision was made to choose such settings that minimize the coupling of cardiac activity to the signal. Therefore, frequency and phase were set to 32 kHz and 67.5°, respectively. A related point to consider is that the datasheet of ADS1298R states that a phase of 112.5° yields the best performance of the impedance measurement from the noise point of view.

For a deeper evaluation of what is being measured by the circuit, the EIP was compared to the result of a flow thermography device. The setup for this measurement included an NTC thermistor placed inside a breathing mask that was put in front of the mouth and nostrils. The thermistor was connected to a resistor in series and powered by an additional battery (providing a separate and floating source of voltage) to convert the resistance to voltage. The output voltage (with its floating ground) was fed to one of the ECG channels of the device and was measured along with the EIP signal. The internal gain of that specific ECG channel was set to 2 to prevent saturation in the measurement path.

Depicted by Figure 4.5, the result shows a significant correlation between the two measurements, as expected. The first panel shows the original EIP signal. The EIP signal filtered by a band-pass IIR filter with the cut-off frequencies of 0.06 and 1 Hz[7] is shown the middle panel. The filtering was performed using forward-backward technique to avoid any distortion caused by the non-linear phase of the IIR filter. The third panel shows the result of the flow thermography, which was filtered with

[7] The range of respiration rate is considered between 4 bpm and 60 bpm.

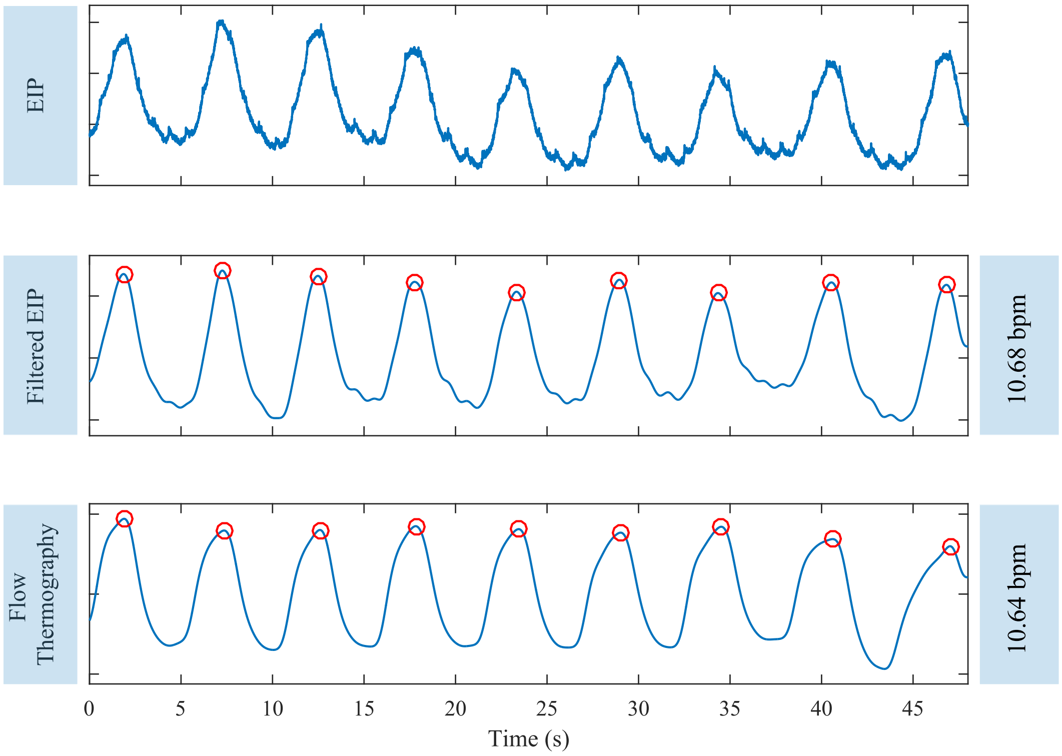


Figure 4.5 *A comparison of EIP and flow thermography. The first and second panels show the raw filtered versions of EIP, respectively. The panel on the bottom depicts the result of flow thermography. The respiration rate estimated from the two panels is shown on the right hand side of them.*

same approach as the EIP.

It is noticeable that the morphology of the signals are different, but needless to say, they provide the same rate that is the primary measure in assessment of the subject's health status especially in the case of hospital patients. To indicate this fact quantitatively, we ran a respiration rate estimation algorithm on both of the signals. The technique is named *Advanced counting* method, which was originally introduced by Schafer et al [43]. The performance of the algorithm has been evaluated and compared with its peers [44]. The results of the respiration cycle length estimation were 10.68 bpm (breath per minute) and 10.64 bpm for EIP and thermography, respectively[8].

[8] Although, the respiration rate is usually estimated in a shorter frame to be able to assume that the signal is stationary.

4.3 Power Consumption

The battery lifetime is an important parameter for the user and may explicitly define for what applications the device is suitable. There are two factors affecting on the lifetime of the device: the battery capacity and the power consumption of the device. The capacity of the battery is limited by its physical size and therefore cannot be chosen freely for wearable devices, which require a miniaturized design. The power consumption of the device, although restricted by the components chosen by the designer according the required functionality of the device, is affected by design decisions made during the development and must be controlled and tested carefully. For more information, refer to sections 3.1.4 for some hardware considerations and 3.2.3 to see how the firmware is tuned to reduce the power consumption.

The current consumption of the device was measured using a $10\,\Omega$ resistor in series with the battery and an oscilloscope measuring the voltage across the resistor in particular frames. The frame length was chosen according to duration of the longest event in the device to obtain the average current consumption of a full operation cycle. For example if the writing to the local memory is the least frequent event performed by the device and is executed every 1 s, then the frame length of the measurement was chosen to 1.2 s to be able to capture the whole event. However, for faster events more number of frames are considered in the calculations. It is worth noting that for slow events (such as sending accelerometer data with data rate of 20 Hz), a longer frame of the oscilloscope was required that limits its sampling rate and therefore increases the probability of missing fast occurrences. For each part of the code, which has a recursive nature[9], 10 measurements were performed and their mean and standard deviation were calculated. All the frames of the measurement were stored to a USB flash memory and transferred to a computer to be processed by MATLAB. In the following, the calculation procedure of power consumption is explained. First, the charge has been computed using

$$Q(C) = \delta t \sum_{k=0}^{n} i, \tag{4.1}$$

which integrates the measured current signal (i with n points) over time. δt shows the time steps and equals to $\frac{1}{f_s}$. Charge is in coulomb (C) or equally in *times* ×

[9] The transient parts of the code were not tested for power consumption.

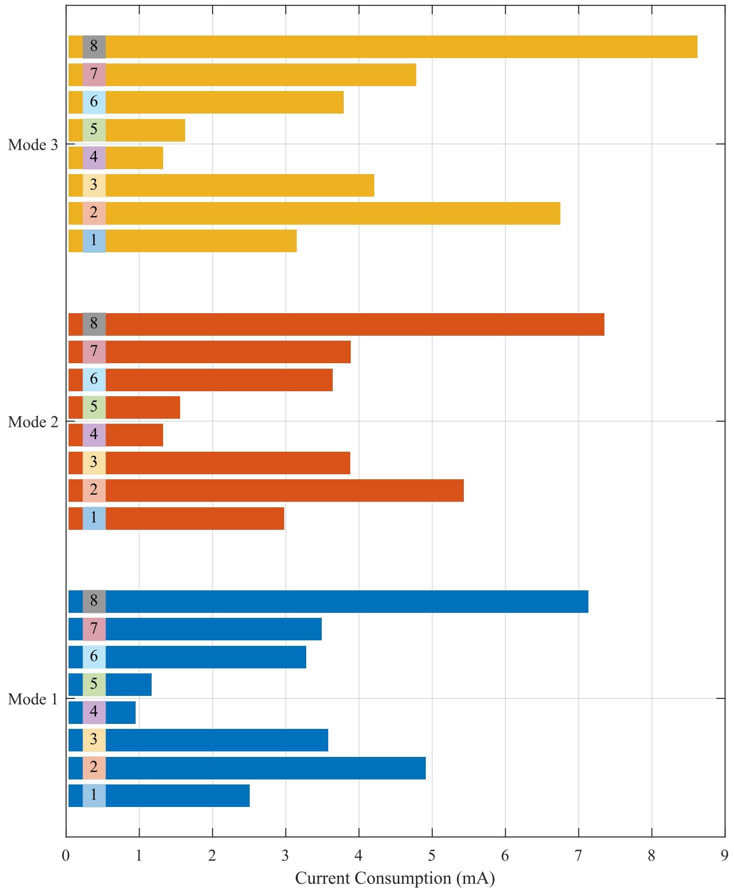


Figure 4.6 *The current consumption in all the operation modes and cases. The values for each mode are grouped together. The numbers on each bin shows the case number.*

ampere × *second*. Then the current consumption is calculated by

$$I(A) = Q/t_{frame}, \tag{4.2}$$

where t_{frame} is the time of the considered frame which has a natural number of the longest event in the recorded measurement.

The current consumption has been measured for all the operation modes 1, 2, and 3. Modes 1 and 4 are the same from current consumption point of view so mode 4 has been skipped. In each mode, various measurement channels with different data rates can be activated, creating a large set of possibilities from which only a selected set is evaluated and reported in this document. As a result, there are three modes, each including 8 cases. Each case represents a real use case that might be of user's

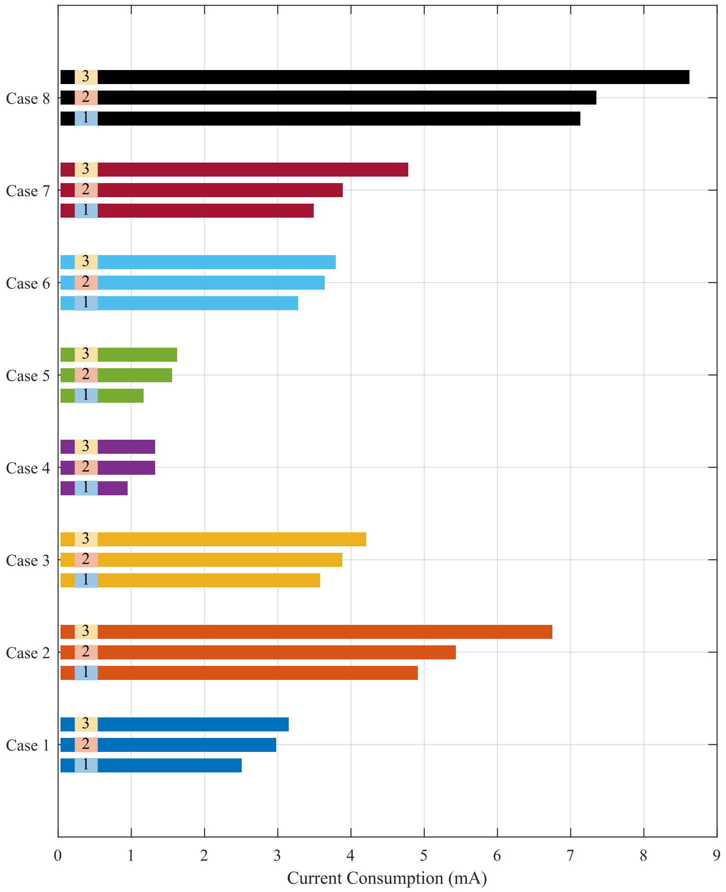


Figure 4.7 *The current consumption in all the operation modes and test cases. The values for each case are grouped together. The numbers on each bin shows the mode number.*

interest. Sections 4.3.1, 4.3.2, 4.3.3 and 4.3.4 represent detailed results of current consumption in each mode. Refer to Tables 4.2, 4.3 and 4.4 for more information on the 8 scenario cases.

Before getting into the details of the measurement, see Figures 4.6 and 4.7 for the final results. Figure 4.6 shows the current consumption for all the eight cases, grouped for each mode. It can be seen that in all the modes, the device consumes more when more channels are enabled and higher data rates are used. Figure 4.7 shows the same result but has grouped the values according to the cases. In all the cases, transmitting the data via BLE (mode 1 and 4) produces the smallest current consumption comparing to other modes. And obviously, both storing and transmitting (mode 3) is the most power hungry mode. Note that in all the modes and cases, the resolution of the data sent to the Android device is limited to 2 octets while the SD card writes 1.5 times larger amount of data when ECG and respiration are being recorded (3 octets).

Case 4 is a special case that due to the low data and therefore long frame of measurement, suffers from low sampling frequency of oscilloscope. In this case, high frequency events might have been missed by the measurement setup and therefore, the current consumption measured for both modes 2 and 3 are equal. Notice that according to the Nordic Semiconductor's online power Profiler[10], the difference between these two measurements is less than 80 µA.

4.3.1 Initiating a BLE Connection

The device wakes up in advertisement mode in which, it starts waiting for a central device to ask for a connection. The advertisement payload is 19 B long[11], the TX power is set to 0 dBm and the operating voltage is 2.8 V. The advertisement interval is chosen to be 40 ms to make the connection establishment quick. It is true that increasing the advertisement interval will diminish the power consumption in each cycle of advertisement (since the BLE block sleeps for a longer time), but on the other hand this might increase the required time of advertisement due the smaller probability of overlapping of advertisement and scanning. Table 4.1 shows the result of the measurements.

Table 4.1 *Current consumption of BLE advertisement with* 40 ms *interval.*

Phase of functioning	Current Consumption
Advertisement	742±14 µA

Figure 4.8 illustrates the power consumed by the device in an advertisement event. The 40 ms interval between these advertisements is excluded from the image. The light blue signal (the one that is noisy) is the main measurement that is then smoothed by a 20th other median filter (the oscilloscope acquires 500 ksps). The darker blue signal shows the result of the filtering. Different events of advertisement are marked. For their description, refer to the figure caption.

[10] https://devzone.nordicsemi.com/power/

[11] There are three data structures: flags, device name and TX power. The data length of device name is 11 bytes. For more information refer to GAP definitions in the BLE book by Townsend *et al.* [1]

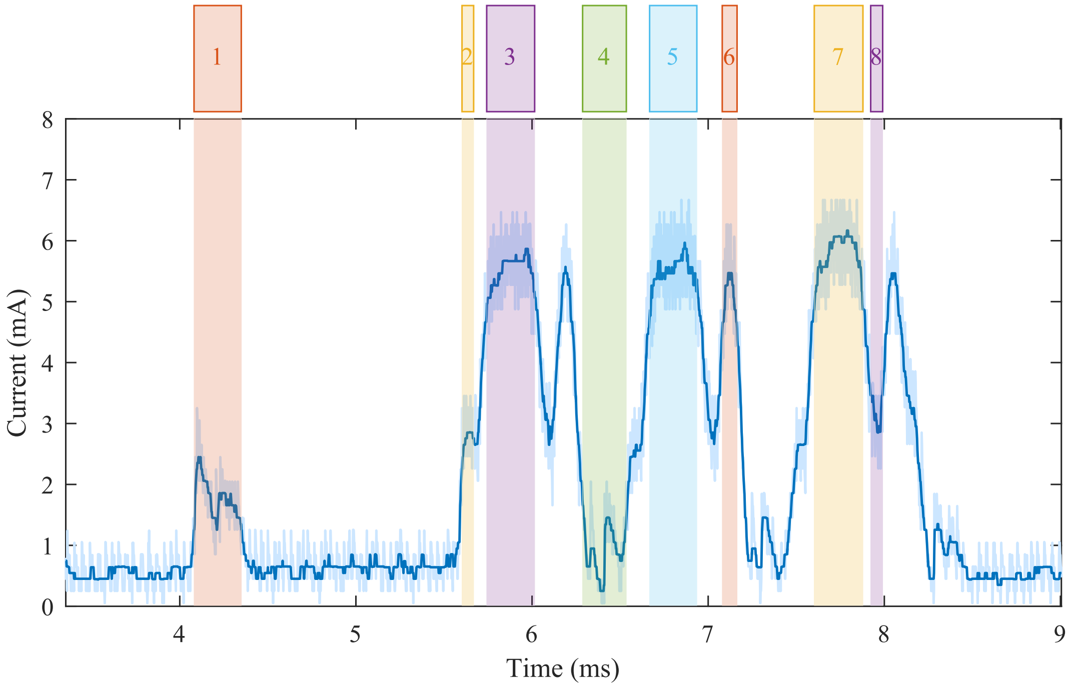


Figure 4.8 *Power consumption measurement for BLE advertisement. The long time* (40 ms) *between the advertisements is not included. The dark blue line shows the smoothed signal. Different sections are labeled with numbers: (1) Pre-processing and HFXO ramp, (2) radio starting and CPU involvement, (3) transmission in channel 37, (4) standby and post processing, (5) transmission in channel 38, (6) receiving, (7) transmission in channel 39 and (8) switching the radio.*

4.3.2 Mode 1: Transmitting the Data

In mode 1, after activating the required sensors and channels, the device only sends the captured information to the central device via BLE. The data is not stored locally and therefore the SD card is completely off. The BLE connection intervals are chosen depending on the data rate to optimize the performance and save the bandwidth if not needed. Table 4.2 shows the results.

Consider case number 1 (2$^{\text{nd}}$ row of the table). In this case, the motion sensor is in sleep mode and only three channels of ADS1296R are active (for measuring ECG). The BLE connection interval is set to 12.5 ms and the transient events are skipped. At lower levels, SPI, I^{2}C, timers (excluding RTC[12]) and interrupts in the MCU are active. The data rate of the AFE is set to 250 Hz. Transient events include measuring the battery and transmitting an *information packet* to the central device,

[12] Real time clock. This block is used to keep track of the duration of the measurement.

Table 4.2 *Average current consumption of the device while acquiring different sensor's outputs with several data rates and connection intervals in mode 1. All the measurements except respiration include three channels. The last row shows the case when ECG, respiration, accelerometer and gyroscope are enable. The resolution of all the channels is 16 bits.*

Ind	**Active Measure.**	**Clock Source**	**Data Rate (sps)**	**Connection Interval (ms)**	**Current Consumption (mA)**
1	ECG[1]	AFE Int. Clk	250	12.5[4]	2.55±0.006
1*	ECG[1]	AFE Int. Clk	250	37.5[5]	2.51±0.015
2	ECG[2]	Ext. Osc.	1000	8.75	4.91±0.011
3	ECG & Resp.[3]	AFE Int. Clk	250	37.5	3.58±0.013
4	Acc.	Auto-selection[6]	20	75[7]	0.95±0.003
5	Acc.	Auto-selection[6]	100	60	1.17±0.005
6	Acc. & Gyro.	Auto-selection[8]	Acc.:50 Gyro.:50	60	3.28±0.006
7	ECG & Acc.	Acc.: Auto[6] ECG: Int. Clk	Acc.:100 ECG:500	17.5	3.49±0.057
8	All	Acc.: Auto[8] Gyro.: Auto[8] ECG: Int Clk. Resp.: Int. Clk.	Acc.:100 Gyro.:100 ECG:500 Resp.:500	8.75	7.13±0.036

[1] The low-power mode of ADS1296R is used.

[2] The high-resolution mode of ADS1296R is used.

[3] The modulation and demodulation block used modulation frequency of 32 kHz with a phase delay of 67.5°.

[4] Considering a powerful signal strength and well-synchronized clocks, one connection is established in each connection interval.

[5] Considering a powerful signal strength and well-synchronized clocks, three connections are established in each connection interval.

[6] The best clock source is automatically chosen by the IC based on availability of gyroscope. In this case, since gyroscope is disabled the internal 20 MHz clock is selected, which has a smaller power consumption but a larger uncertainty.

[7] Note that every second packet has an empty payload.

[8] Since the gyroscope is enabled, it is used as the clock source.

* This case is an additional measurement for this mode and is not included the measurements for the other modes and therefore in Figures 4.6 and 4.7.

which happen every 2 minutes.

Let us have a closer look to the timing of the system in this case. The AFE triggers the interrupt of the MCU every 4 ms and a BLE packet is ready every 12 ms[13].

[13] An ECG packet has 20 bytes including three samples from each channel, each 16 bits (two octets), packet counter and packet header. The real resolution of the AFE's ADC is 24 bits and the least significant octet is simply dropped. High-pass filtering the signal and dropping the most

Therefore, most of the packets are sent right after they are ready. However, in real-world practices there are some errors that bring uncertainty to this claim. For instance, if the noise level in the measurement environment is high (or if the distance between the two peer devices is long), the number of unsuccessful transmissions increase and hence the BLE communication might trigger a second connection in one connection interval. The inaccuracy of the clocks is another example of this error. In the performed tests, the rate of occurrence of this issue was not high and hence did not have a large impact on the power consumption. We ensured that the device functions in a stable condition and verified that in the measured frames.

According to the last explanation, one can conclude that the power consumption of the device is not static and does not depend only on its design, but also the environment noise level and the distance between the device increase the power consumption since the peripheral needs to establish more connections or request for a smaller connection interval to be able to transmit its accumulated data buffer.

An example frame from the power consumption is illustrated in Figure 4.9. Similar to the previous section, the smoothed signal is shown with dark blue color. The events 2 and 3 are the moments that the AFE notifies the MCU and the MCU reads the ready data. Event 5 shows a connection (For more details visit the Nordic Semiconductor's online power Profiler). Event 1 is a spike generated by the LDO or DC/DC blocks in the MCU that are functioning in the refreshing mode[14].

Note that the connection interval value is chosen based on the estimated packet rate. A smaller connection interval would result in a waste of resources (bandwidth and power resources) since the CPU must be awake for a longer time on average to handle the additional connections in which no data is sent. Now what will happen if the connection interval is larger? The next measurement of power consumption with only ECG enabled, with a data rate of 250 Hz and a connection interval of 36 ms shows the result of this scenario.

With a smaller connection interval comparing to the rate of coming packets, the BLE stack would establish more than one connection per connection interval. In this case, there are more packets sent while the stack (along with the needed clock source and

significant part (as large as possible) would be the optimal practice but is skipped here, since other modes of the device can be used for a high-resolution measurement.

[14] The power management part of the MCU switches between these two whenever one of them is more optimized from the power consumption point of view.

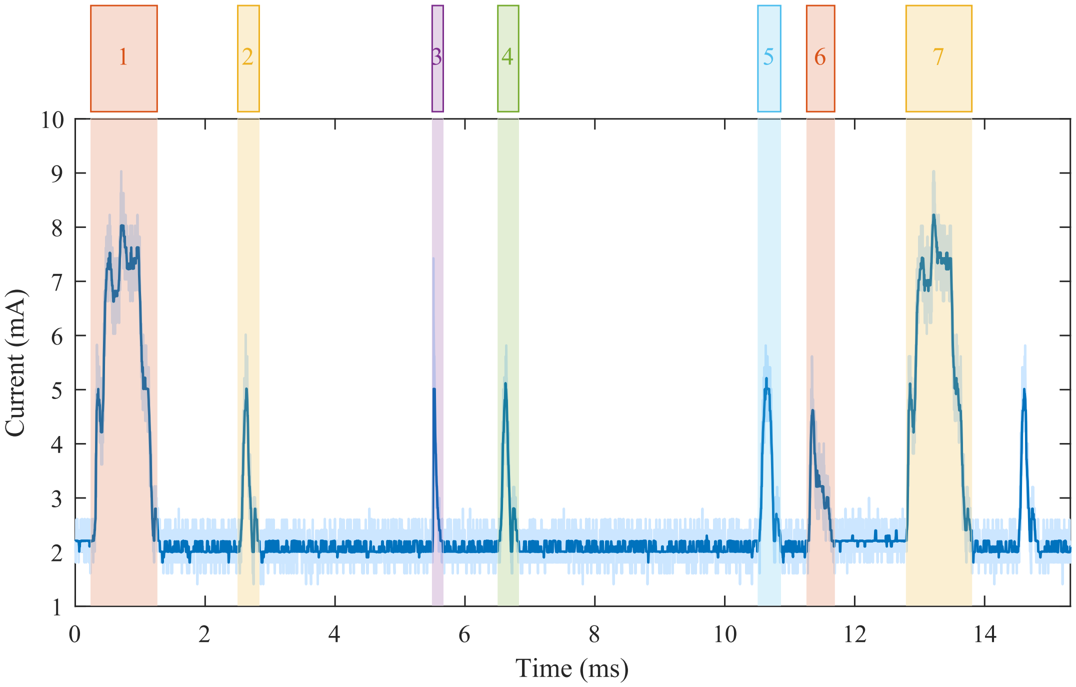


Figure 4.9 *Power consumption measurement for mode 1 with only three channels of ECG active. Different sections are labeled with numbers: (1) connection, (2) data from AFE is ready and read by MCU, (3) result of LDO or DC/DC blocks of the MCU functioning the refreshing mode, (4) same as 2, (5) same as 2, (6) pre-processing and HFXO ramp (getting ready for connection) and (7) connection.*

CPU) needs to power up and get started only once. This would decrease the power consumption. Note that the number of connections per connection interval is limited and varied between different hand held devices with different operating system (OS) version. For instance, some of the devices do not allow more than three connections per connection interval, while this number goes up to six in some other devices. A very important note to remember when designing these limits is that the bandwidth should have some room left for emergency cases so that devices are able to establish more connections if needed to compensate for the accumulated packets in the buffer due to a poor connection.

The matters mentioned above clearly show that the connection parameters not only need to be chosen based on the data rate gathered from the sensors, but also strongly require to be dynamically negotiated with the central device depending on the needs of the peripheral device.

With increasing the data rate of ECG to 1 kHz, the current consumption of the device expands to 4.6 mA. It should be noted that this elevation is not originated from the AFE but is due to use of the external oscillator and the larger number of established connections.

The measurements 5, 6 and 7 show some cases of using only activity measurement i.e. accelerometer and gyroscope. It can be seen that using only accelerometer, especially with low data rate, needs less than 1 mA. This can be further decreased using the low-power mode of accelerometer[15]. However, increasing the data rate increases the current consumption only by 200 µA. On the other hand, inclusion of gyroscope is costly but it elevates the clock source accuracy of the sensor.

4.3.3 Mode 2: Locally Storing the Data

In mode 2, the measured data is stored locally on the device's memory. Therefore, the BLE connection is terminated after the measurement has been started by the user and the required information packet[16] is received by the peripheral (the developed device). The device continues capturing the chosen signals until a fault happens (such as running out of battery) or the user turns off the device.

Every channel, along with the required header, is buffered in a 512-byte array of RAM (504 bytes are dedicated to storing data) and the CPU starts writing the buffers to the local memory when they are ready. Writing to the memory happens as fast as the card responds to its received commands. One SPI bus is dedicated to SD card[17] and the transmission from RAM to SPI's buffer is offloaded from the CPU by employing the EasyDMA[18].

In this mode, the local memory plays an important role in power consumption. Writing to memory cells produces large peaks in the current drawn from the battery. Therefore, the less frequent the writing to the memory is, the smaller is the power consumption. This is of course with the assumption that the card wastes only a small amount of power when no transaction is in progress.

[15] Refer to the datasheet of the IC

[16] The information packet that is sent from the central to the peripheral at the start of the measurement includes the activated channels, their data rates, subject id, time and date and the mode chosen by the user. This packet is the same for all the modes 1 to 4

[17] If the SPI bus is shared with other slaves in the hardware, the clocks received by the SD card prevents it from going to sleep mode.

[18] EasyDMA is a module implemented by some of the peripherals in the MCU. This module allows the peripherals to have direct access to RAM.

Table 4.3 *Average current consumption of the device while acquiring outputs of different sensors with several data rates and writing them on the local memory in mode 2. The current consumption for both the* 1 GB *ATP card (column A) and* 2 GB *Transcend card (column B) are shown. All the measurements except respiration include three channels. The last row shows the case when ECG, respiration, accelerometer and gyroscope are enable. The resolution of ECG and respiration is 24 bits and the resolution of accelerometer and gyroscope is 16 bits.*

Ind	**Active Measure.**	**Clock Source**	**Data Rate (sps)**	**A: Current Consumption (mA)**	**B: Current Consumption (mA)**
1	ECG	AFE Int. Clk	250	2.98±0.015	4.57±0.15
2	ECG	Ext. Osc.	1000	5.43±0.07	8.86±0.91
3	ECG & Resp.	AFE Int. Clk	250	3.88±0.025	6.56±0.107
4	Acc.	Auto-selection	20	1.33±0.015	1.43±0.019
5	Acc.	Auto-selection	100	1.56±0.019	2.18±0.13
6	Acc. & Gyro.	Auto-selection	Acc.:50 Gyro.:50	3.64±0.015	3.98±0.035
7	ECG & Acc.	Acc.: Auto ECG: Int. Clk	Acc.:100 ECG:500	3.89±0.032	6.74±0.13
8	All	Acc.: Auto Gyro.: Auto ECG: Int. Clk Resp.: Int. Clk	Acc.:100 Gyro.:100 ECG:500 Resp.:500	7.35±0.085	10.29±0.081

The SD cards depending on the manufacturer and their capacity impose various levels of power consumption. The page write peak current, automatic standby (if provided at all) current, the time in which the card enters the sleep mode if no activity on the clock pin, are some examples that complicates the process of choosing the right SD card. The larger obstacle is the lack of this category of information in their datasheets.

Since this work is not oriented around comparing the SD cards, only two types of cards were tested to show how seriously this difference of cards can affect on the overall lifetime of the device. A 1 GB card from ATP and a 2 GB card from Transcend were used. These tests were done with the exact same device, firmware and measurement procedure. Table 4.3 shows the results and Figure 4.10 visualizes them.

According to Figure 4.10 there is a significant difference between the two SD cards, especially when the rate of writing to them is high. Consider case number 7. Using

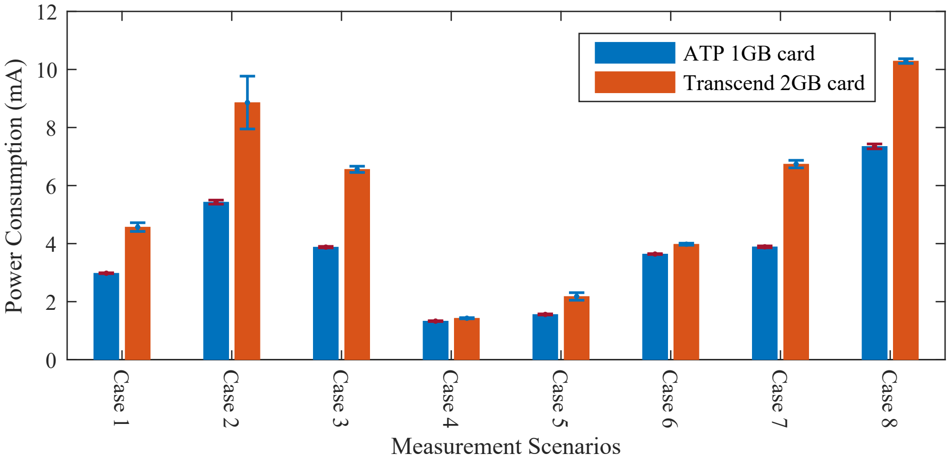


Figure 4.10 *A comparison of the power consumption of two different SD cards. The blue and red bars correspond to the* 1 GB *card from ATP and the* 2 GB *card from Transcend, respectively. For details about measurement cases, refer to Table 4.3. The small lines at the top of each bar indicate the standard deviation of each measurement.*

the card A, the device consumes 3.9 mA while with the card B this value increases up to 6.7 mA. By using a 300 mA h battery, the device can stay on for about 77 and 45 hours with employing the cards A and B, respectively. In other words, employing a wisely chosen SD card increases the battery lifetime for 32 hours.

Figure 4.11 shows a detailed demonstration of the reason that the card from ATP has a lower power consumption. These two measurements are both done for case 1 in Table 4.3. The current consumption of the device using the ATP card is depicted in the bottom panel. Both cards require almost the same amount current peak while writing a page. However, the ATP card goes to sleep mode much faster, causing the average current consumption to be lower.

Figure 4.11 also demonstrates the timing needed in measuring three channels of ECG with the data rate of 250 sample/s. The samples from the AFE are captured by the MCU every 4 ms and are loaded to the buffer. The buffer can contain up to 504 bytes of data. Therefore, it takes 672 ms to fill the buffer with 168 samples of 3 octets. The parts labeled by † show when the writing to memory pages takes place. There are three peaks since there are three channels and therefore three buffers are filled up at the same time.

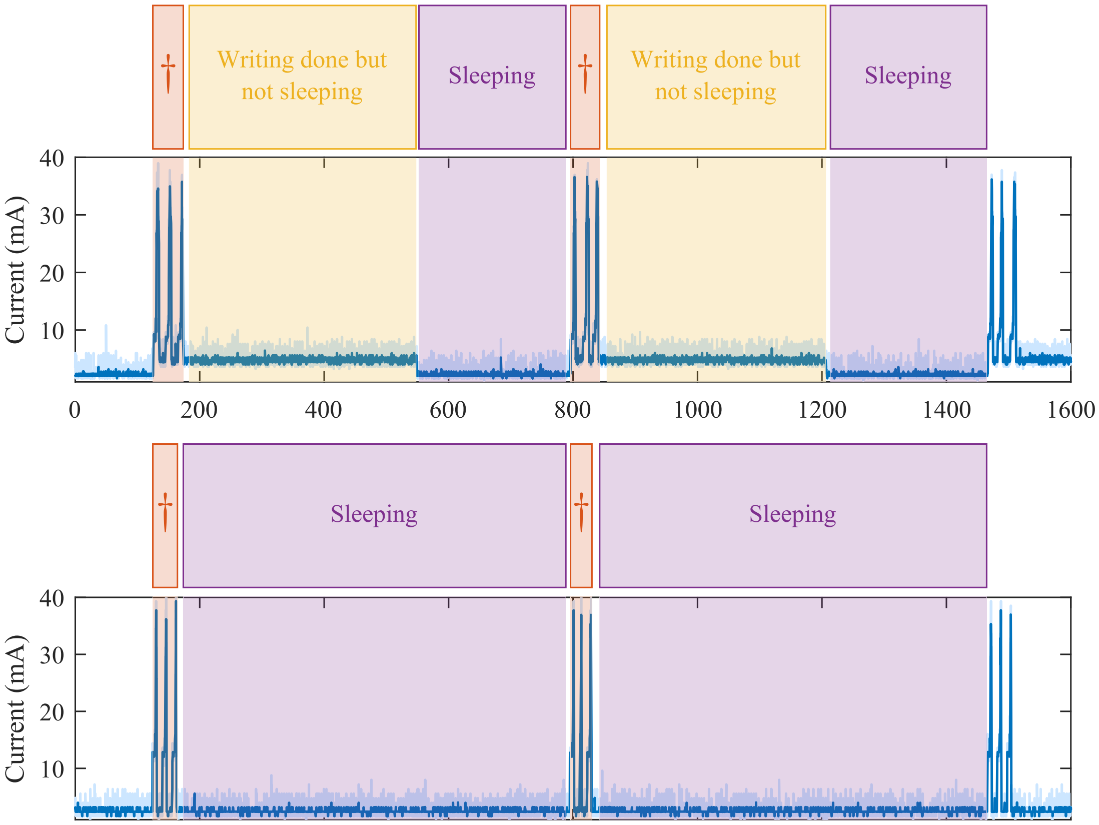


Figure 4.11 *An example of the power consumed by two SD cards: a 2GB from Transcend (top panel) and a 1GB from ATP (bottom panel). The parts labeled with a light violet color show the time that the SD cards are in sleep. The regions labeled by † indicate the moments that the cards are writing three pages of memory.*

4.3.4 Mode 3: Both Sending and Storing the Data

The device provides another mode to both record the signals and monitor them in an Android device. As expected, this mode consumes a larger amount of current comparing to modes 1 and 2 and is preferred only when short measurements are of interest. In these scenarios, while the user is continuously able to observe the measurement signals, the data is also stored locally to avoid loss of the recorded data due to complications in the wireless communication.

Once more similar measurements as the two previous sections were made. This time only one SD card that had the better power consumption characteristics was used. The acquired samples were stored with their full precision, however they are truncated for sending via the BLE, similarly for the previous measurements in modes

Table 4.4 *Average current consumption of the device while acquiring different sensor's outputs with several data rates and writing them on the local memory and transmitting them via BLE (mode 3). All the measurements except respiration include three channels. The last row shows the case when the ECG, respiration, accelerometer and gyroscope are enable. The resolution of the ECG and respiration is 24 bits for storing and 16 bits for transmission. The resolution of accelerometer and gyroscope is 16 bits for both writing to the memory and sending to the Android device.*

Ind	**Active Measure.**	**Clock Source**	**Data Rate (sps)**	**Connection Interval (ms)**	**Current Consumption (mA)**
1	ECG	AFE Int. Clk	250	37.5	3.15±0.017
2	ECG	Ext. Osc.	1000	8.75	6.75±0.080
3	ECG & Resp.	AFE Int. Clk	250	37.5	4.21±0.070
4	Acc.	Auto-selection	20	75	1.33±0.010
5	Acc.	Auto-selection	100	60	1.63±0.009
6	Acc. & Gyro.	Auto-selection	Acc.:50 Gyro.:50	60	3.79±0.011
7	ECG & Acc.	Acc.: Auto ECG: Int.	Acc.:100 ECG:500	17.5	4.78±0.043
8	All	Acc.: Auto Gyro.: Auto ECG: Int. Clk Resp.: Int. Clk	Acc.:100 Gyro.:100 ECG:500 Resp.:500	8.75	8.62±0.029

1 and 2. Table 4.4 contains the gathered results.

5. DISCUSSION AND CONCLUSIONS

5.1 Summary

In this work, we designed a system for ambulatory physiological measurements providing ECG, respiration, accelerometer and gyroscope measurements. The device has the capability of both storing the measured data and transmitting it to a peer device via BLE. Besides the development of the system, this work has assessed the power profile of the device, explored different use case scenarios and studied the quality of the signals measured with the device.

In this project, we also designed an Android application and a computer software package. The Android application is used to configure the required settings of the measurement and the computer software is needed to read and manage the locally stored measurement data. The system has been extensively tested also in real use and it has shown a reliable performance.

The system is currently being used in two different research project: VitalSens[1], which is the main sponsor for this thesis work and Turre2[2]. The system has been also used in Oulu hospital for long-term monitoring of cardiac patients, including 12 subjects each monitored for approximately 18 hours.

The developed device is already outperforming some of the existing devices in market in signal quality and the provided features. The signal quality of ECG has been compared to Faros 360 (from Bittium Technologies Ltd) and the results are competing. The quality of respiration measurement was evaluated by comparing it to results from a thermistor placed into a breathing mask.

The current consumption of the device was also evaluated for different modes and

[1] More information can be found on http://www.tut.fi/en/research/research-fields/biomedical-sciences-and-engineering/research-projects/vitalsens/index.htm

[2] A project for improving the monitoring of dog well-being and enhance the human-dog relationship. More information can be found on https://turre.fi/?lang=en

use cases. The results reveal that one of the biggest challenges is saving massive amount of data locally. The BLE communication appears as a less power-hungry solution comparing to saving the data on an SD card. Our results also verified the direct relationship between data rate and power consumption.

The next couple of sections discuss the problems faced during the development of this system that are considered as the current bottlenecks and targets of future optimization of the device. In section 5.3, we discuss what other features the system can benefit from and what is still vital to be added to it if it is intended to be further developed for wider use.

5.2 Challenges with Wearable Devices

As we walked this road, there were difficulties, technology limitations and most importantly engineering trade-offs that formed the specification of the system. The next four sections speak of the challenges related to usability, battery lifetime, wireless transmission bandwidth and transferring the data to the computer.

5.2.1 Usability

Although nowadays we are seeing the fashion factors impacting the design outcome of the wearable devices, we can probably still stick to the fact that the user's comfort is the one that rules the design criteria in all the stages from choosing the components to designing the enclosure of the device. The size and the weight of the device, its connection to the electrodes, its location on the body, the complication in getting the device running and the probability of user errors in doing the measurements (especially for clinical self-monitoring purposes) are some of the factors that heavily affect on the number of devices sold in the market.

The device designed in this work is a circular unit with a radius of 16.5 mm. The tallest components on the top layer and button layer are the red-green-yellow (RGY) LED with a height of 1.83 mm and the USB connector with a height of 2.51 mm, respectively. The PCB itself with all the components assembled on it (except the battery and including an SD card) weighs 3.7 g.

The new technologies in electrode set can bring a huge amount of simplicity to the system. For example, the electrode reported by Vuorinen et al. [12] is a beneficial solution for measuring EAS lead set. This electrode set can be used with the designed device with an optimized position of the respiration electrodes within the EAS leads [29].

5.2.2 Battery Lifetime

As the results showed, one of the biggest challenges is the memory for storing the measurements. SD cards impose complications on estimating the power consumption and vary a lot depending on their version and manufacturer. This problem could be diminished by using a flash memory IC such as MT29F4G01ADAGDSF from Micron, which proposes a through description of its detailed performance.

The ECG analog-front-end itself drew a large proportion of the power. One alternative to using an AFE for measuring ECG is using discrete components i.e. separate amplifiers, ADCs etc. Although this approach might consume a smaller amount of current, it could need a large area of PCB, especially if all the features of ADS1298R (e.g. impedance pneumography measurement) were to be implemented. The AFE used in this project includes several additional features that were not used. These features are obviously implemented using some hardware blocks that increase the power dissipation even if they are not enabled. As a result, other AFEs with features closer to the needs of the project could improve the power consumption.

Another serious power consumer part that was also an absolute need was the external clock source for the AFE. The limited range of the allowed clock frequency by ADS1298R was the reason that we could not provide this clock from the MCU using a pulse width modulation (PWM) block[3], which could have diminish the power consumption dramatically. Thus, AFEs with more accurate internal clock or the possibility of accepting a clock signal from the MCU could improve the power consumption. In this project, although ADS1296R satisfies the needs (three channels of ECG and one channel of respiration), using ADS1294R would have lowered the power consumption. In application in which only one channel of ECG is needed (such as fitness trackers that mostly analyze only the heart rate), using ADS1292(R) would be the most suitable option since it features the lowest power dissipation among the ADS129x family.

The motion sensor used in this project, although having a nice and straightfor-

[3] The clock source of the MCU could not be changed due to the requirements of the BLE stack

ward functionality, suffers from the lack of complete documentation and the developer community support from the manufacturer. The sleep power consumption of MPU9250 is one example of the problems that we faced, which never reached to what is claimed in the datasheet.

Our results show that storing the data locally is more costly from power consumption point of view. Therefore, one can conclude that storing the data on a hand-held device is more optimized than storing it locally. However, it is important to remember the complications in this scenario. Data loss is higher when having a wireless tunnel before storing the data. The hand-held device must always be kept close to the measurement unit. Otherwise, the measurement should be terminated and the measurement unit must wait for its peer to return to the range and perform the data storing. And finally, the battery lifetime of the hand-held device is what is usually not considered. If the hand-held device is continuously receiving high rate data, probably monitoring it (and therefore, updating the screen with a high rate) and doing some calculations, it might not be able to accompany the measurement for a long time.

5.2.3 Wireless Data Transmission Bandwidth

The maximum data rate transmitted by the Bluetooth low energy is limited by different factors such as minimum connection interval, maximum permitted number of connections per connection interval and the structure of the service in use. Beside the restrictions applied by the designers (SIG group), different devices, depending on their hardware and software implementations, can limit these boundaries even further. Therefore, it is important to remember that the amount of data to be transmitted is limited and might not be suitable for all the medical requirements. For instance, considering a connection interval of 7.5 ms and 6 connections per connection interval, the UART service can send up to 800 packet/s, assuming that the protocol is in its best shape and never needs a re-transmission. Sending three channels of ECG with data rate of 3000 sample/s, would require 1000 packet/s, with the possibility of fitting three samples of each channel in one packet. However the above notion is true only in theory and in practice the limits are much lower due to complications in the communication. In addition, the designer would prefer to leave some room in the bandwidth e.g. for packet re-transmission.

BLE is designed to provide significantly lower power consumption comparing to its older brother classic Bluetooth (also known as Bluetooth BR/EDR), especially in small data rate applications. Nevertheless, transmission of large amount of data will dramatically increase the power consumption since the hardware must wake up more frequently from sleep to handle the connection. Therefore, it is important to pay attention that the power consumption gets large even if using BLE.

Another important matter to remember when using hand-held devices as the receiving and monitoring unit in the system is that these devices are also limited in battery lifetime, especially when asking them to do heavy computations such as updating the screen with fast rates. It is worth mentioning that requesting heavy computations like asking an Android device to plot six channels of data with a data rate of 2 ksample/s increases the probability of the application being crashed. Thus, it is important to note that the receiving point is not a powerful computer with many resources. However, in *semi-stationary* applications in which the user can stay in a closed environment, the central device can alternatively be a computer assisted by a BLE dongle.

5.2.4 Transferring Data to the Computer

Storing the data in the local memory of the device provides the most reliable way of recording the measurements. However, transferring the large amount of data from the local memory to a computer can take longer than the user would desire. This process requires reading the memory pages and sending the data to a computer either by a USB block implemented in the MCU or through a UART/USB converter, if not mentioning other events like the extra conversations between the computer and the device implemented for hand shaking purposes.

Consider the following scenario. It is desired to do a long-term measurement with a clinically acceptable measuring configuration. For instance, the subject enables measuring three channels of ECG and respiration with a data rate of 500 sps and accelerometer with a data rate of 50 sps. Let us assume that the device is able to do this measurement for 20 hours. This implies 453.2 MB considering that the ECG and respiration samples are 3 octets and accelerometer samples are 2 octets. Now if the UART block is sending with the baud rate of 1 Mbit s^{-1}, the transmission takes about 63 minutes. Note that we have not considered many other items here such as the additional bits in the UART communication, the time between each transmitted byte[4], the hand-shaking communications between the two peer, the additional information associated with the data, which is stored as header in each page of the memory [5] and the dependency between the two states: reading from memory and sending to the computer. As a result, the time needed to transfer the data to the computer can take much longer than the reported number (even up to 4 hours[6]).

There are different workarounds for this issue. Using an MCU with integrated USB transceiver is an example. Data compression can also benefit the whole system including faster data transmission to computer. To read about other advantages of using an optimized method of compression refer to section 5.3.

5.3 Future Work

As shown in this work, the biggest concern to optimize the battery lifetime of the device is the amount of information that it either stores or sends to a peer device. In both scenarios, compressing the signals by loss-less compression methods can bring a huge benefit to the system and still preserve the integrity of the data. By this tool, not only the frequency of writing to the SD card and BLE connection would be declined, but also the size of the stored data is reduced, which results in less memory occupation and faster transmission of the measurements to the computer.

The selection of the right compression method is based on several properties of them. Their effect on power consumption, associated computational complexity with them and their compression ratio are some important aspects that should be explored.

Inclusion of real-time analysis algorithms is another item that adds an extended number of features to the device and qualifies it for some specific use cases. The heart rate and heart rate variability (HRV) is one example. HRV is simple in calculation and yet a powerful tool in assessing one's health condition. It can be executed with a very low power consumption since the required sampling frequency of ECG

[4] This can be modified in the driver of the USB, but cannot be smaller than a certain value to prevent the corruption of the data

[5] This is a very vital part of data and cannot be neglected. The data in each page of the memory is labeled with some certain values that are needed by the computer to be able to reconstruct the data.

[6] With optimization in the driver and full concurrency in reading from the memory and transmitting it to the computer, so that the UART block always has a ready page to be transmitted to the computer.

measurement for HRV analysis can be as low as 50 Hz [45]. Respiration cycle length, step counting and gesture estimation are some other helpful attributes that can be simply generated by the CPU. As a use case example, calculating the heart rate locally in the device and sending it via the dedicated heart rate BLE service[7] compatibles the device with hundreds of Android and iOS phone applications.

Other internal real-time analysis methods would extend the possible usage scenarios of the device even further. As an example, connecting an ST-elevation detection algorithm could turn the device to a medical event recorder. Apart from these examples, a ton of software features and algorithms can be added to a wearable device. However, it is always a matter of resources and design goals.

The computer software in this system can experience a series of updates too. The analysis part of the software is still blank but would be capable of bringing a world of statistical reports, analysis and descriptive diagram to the user. The software can also be equipped with more complicated features such as recognizing the recorded ECG leads [46].

The android application is still missing the ability of storing the data. Beside from storing the data on the phone or tablet's memory, the application may be furnished with some cloud data storage possibilities, to not only prevent data loss in the hand held device, but also to enable the memory-limited devices accompany the measurement unit without difficulties. Extending the application to other operating systems especially iOS is vital from the user side.

There are also some feature in the hardware that are not still employed. The possibilities provided by the AFE in the ECG measurement such as generating the Wilson Central Terminal for the measurement of the chest leads of 12-lead ECG is one example. NFC connection is another item that is vital in data security matters. There are three methods in a BLE connection that the two peers can negotiate the Short Term Key (STK), out of which only *Out Of Band* (OOB) approach is possible in this device that provides protection against *man-in-the-middle* (MITM) attacks[8] [1]. OOB requires some means other than the BLE radio to transfer the encryption key. The NFC in this device is provided exactly for this purpose.

[7] See the Bluetooth full list of adopted GATT profile and service specifications on https://www.bluetooth.com/specifications/gatt

[8] *Passkey Display* is another method that is safe against MITM but needs a display on both of the devices.

The antenna of the device should be more evaluated and tuned for the especial use case of the device, which is when the device is attached to the human body. "The radiation pattern always includes a deep null because the radiation cannot penetrate the body", wrote Tiiti Kellomäki in her doctoral thesis [47]. It is obvious that the body de-tunes the antenna, but it should be in a controlled way and its limitations and affects on this device should be examined.

BIBLIOGRAPHY

[1] K. Townsend, C. Cufí, Akiba, and R. Davidson. *Getting started with Bluetooth low energy: tools and techniques for low-power networking.* O'Reilly Media, Inc., 2014.

[2] R. Gupta, M. Mitra, and J. Bera. *ECG acquisition and automated remote processing.* Springer India, 2014, "20".

[3] L. Sörnmo and P. Laguna. *Bioelectrical signal processing in cardiac and neurological applications.* Academic Press, 2005, "416".

[4] J. Pan and W. J. Tompkins. "A real-time QRS detection algorithm". In: *IEEE transactions on biomedical engineering* 3 (1985), pp. 230–236.

[5] J. Malmivuo and R. Plonsey. *Bioelectromagnetism: principles and applications of bioelectric and biomagnetic fields.* Oxford University Press, 1995. Chap. 15, "416".

[6] E. Frank. "An accurate, clinically practical system for spatial vectorcardiography". In: *circulation* 13.5 (1956), pp. 737–749.

[7] G. E. Dower, A. Yakush, S. B. Nazzal, R. V. Jutzy, and C. E. Ruiz. "Deriving the 12-lead electrocardiogram from four (EASI) electrodes". In: *Journal of electrocardiology* 21 (1988), S182–S187.

[8] B. J. Drew, M. G. Adams, M. M. Pelter, S. F. Wung, and M. A. Caldwell. "Comparison of standard and derived 12-lead electrocardiograms for diagnosis of coronary angioplasty-induced myocardial ischemia". In: *The American journal of cardiology* 79.5 (1997), pp. 639–644.

[9] J. P. Martínez, P. Laguna, S. Olmos, O. Pahlm, J. Petersson, and L. Sörnmo. "Accuracy of QT Measurement in the EASI-derived 12-lead ECG". In: *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE.* 2006, pp. 3986–3989.

[10] M. D. Klein, I. Key-Brothers, and C. L. Feldman. "Can the vectorcardiographically derived EASI ECG be a suitable surrogate for the standard ECG in selected circumstances". In: *Computers in Cardiology 1997.* 1997, pp. 721–724.

[11] C. L. Feldman, G. MacCallum, and L. H. Hartley. “Comparison of the standard ECG with the EASIcardiogram for ischemia detection during exercise monitoring”. In: *Computers in Cardiology 1997*. 1997, pp. 343–345.

[12] T. Vuorinen, A. Vehkaoja, V. Jeyhani, K. Noponen, A. Onubeze, T. Kankkunen, A. K. Puuronen, S. Nurmentaus, S. P. Preejith, J. Joseph, T. Seppänen, M. Sivaprakasam, and M. Mäntysalo. “Printed, skin-mounted hybrid system for ECG measurements”. In: *Proc. 6th Electronic System-Integration Technology Conference (ESTC)*. IEEE, 2016, pp. 1–6.

[13] J. Webster. *Medical instrumentation: application and design*. John Wiley & Sons, 2009. Chap. 6.

[14] C. Kitchin and L. Counts. *A designer’s guide to instrumentation amplifiers*. Analog Devices, 2004. Chap. 6.

[15] N. Verginis, M. J. Verginis, and R. S. C. Horne. “Scoring respiratory events in paediatric patients: Evaluation of nasal pressure and thermistor recordings separately and in combination.” In: *Sleep Med.* 11.4 (2010), pp. 400–405.

[16] A. K. Abbas, K. Heimann, K. Jergus, T. Orlikowsky, and S. Leonhardt. “Neonatal non-contact respiratory monitoring based on real-time infrared thermography”. In: *Biomed Eng Online* 10.1 (2011), p. 93.

[17] J. M. Shneerson. *Sleep medicine: a guide to sleep and its disorders*. John Wiley & Sons, 2009.

[18] R. Thurnheer, X. Xie, and K. E. Bloch. “Accuracy of nasal cannula pressure recordings for assessment of ventilation during sleep”. In: *Am J Resp Crit Care* 164.10 (2001), pp. 1914–1919.

[19] B. H.C. V. Schéeleand and I. A.M. V. Schéeleand. “The measurement of respiratory and metabolic parameters of patients and controls before and after incremental exercise on bicycle: supporting the effort syndrome hypothesis?” In: *Appl Psychophys Biof* 24.3 (1999), pp. 167–177.

[20] J. J. Freundlich and J. C. Erickson. “Electrical impedance pneumography for simple nonrestrictive continuous monitoring of respiratory rate, rhythm and tidal volume for surgical patients”. In: *Chest Journal* 65.2 (1974), pp. 181–184.

[21] S. L. Hill, J. P. Blackburn, and T. R. Williams. “Measurement of respiratory flow by inductance pneumography”. In: *Med Biol Eng Comput* 20.4 (1982), pp. 517–518.

[22] L. Mirmohamadsadeghi and J. M. Vesin. “Respiratory rate estimation from the ECG using an instantaneous frequency tracking algorithm”. In: *Biomed Signal Proces* 14 (2014), pp. 66–72.

[23] P. A. Leonard, J. G. Douglas, N. R. Grubb, D. Clifton, P. A. Addison, and J. N. Watson. “A fully automated algorithm for the determination of respiratory rate from the photoplethysmogram”. In: *J Clin Monitor Comp* 20.1 (2006), pp. 33–36.

[24] K. V. Madhav, M. Raghuram, E. H. Krishna, N. R. Komalla, and K. A. Reddy. “Extraction of respiratory activity from ECG and PPG signals using vector autoregressive model”. In: *IEEE International Symposium on Medical Measurements and Applications Proceedings*. IEEE, 2012, pp. 1–4.

[25] F. Q. AL-Khalidi, R. Saatchi, D. Burke, H. Elphick, and S. Tan. “Respiration rate monitoring methods: a review”. In: *Pediat Adol* 46.6 (2011), pp. 523–529.

[26] A. V. Sahakian, W. J. Tompkins, and J. G. Webster. “Electrode Motion Artifacts in Electrical Impedance Pneumography”. In: *IEEE Transactions on Biomedical Engineering* BME-32.6 (1985), pp. 448–451.

[27] R. S. Khandpur. *Handbook of biomedical instrumentation*. Tata McGraw-Hill Education, 1992.

[28] O. Lahtinen, V. P. Seppä, J. Väisänen, and J. Hyttinen. “Optimal electrode configurations for impedance pneumography during sports activities”. In: *4th European Conference of the International Federation for Medical and Biological Engineering*. Springer, 2009, pp. 1750–1753.

[29] V. Jeyhani, T. Vuorinen, K. Noponen, M. Mäntysalo, and A. Vehkaoja. “Optimal short distance electrode locations for impedance pneumography measurement from the frontal thoracic area”. In: *Proc. XIV Mediterranean Conference on Medical and Biological Engineering and Computing*. Springer International Publishing, 2016, pp. 1138–1143.

[30] P. Regtien, F. V. D. Heijden, M. J. Korsten, and W. Otthius. *Measurement science for engineers*. Elsevier, 2004.

[31] K. Tran, T. Le, and T. Dinh. “A high-accuracy step counting algorithm for iPhones using accelerometer”. In: *Signal Processing and Information Technology (ISSPIT), 2012 IEEE International Symposium on*. 2012, pp. 000213–000217.

[32] V. Genovese, A. Mannini, and A. M. Sabatini. “A Smartwatch Step Counter for Slow and Intermittent Ambulation”. In: *IEEE Access* (2017).

[33] C. Soaz and K. Diepold. “Step detection and parameterization for gait assessment using a single waist-worn accelerometer”. In: *IEEE Transactions on Biomedical Engineering* 63.5 (2016), pp. 933–942.

[34] S. Chernbumroong, A. S. Atkins, and H. Yu. “Activity classification using a single wrist-worn accelerometer”. In: *Software, Knowledge Information, Industrial Management and Applications (SKIMA), 2011 5th International Conference on.* 2011, pp. 1–6.

[35] W. Song, C. Ade, R. Broxterman, T. Barstow, T. Nelson, and S. Warren. “Activity recognition in planetary navigation field tests using classification algorithms applied to accelerometer data”. In: *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE.* 2012, pp. 1586–1589.

[36] A. Dutta, O. Ma, M. P. Buman, and D. W. Bliss. “Learning approach for classification of GENEActiv accelerometer data for unique activity identification”. In: *Wearable and Implantable Body Sensor Networks (BSN), 2016 IEEE 13th International Conference on.* 2016, pp. 359–364.

[37] R. J. Cole, D. F. Kripke, W. Gruen, D. J. Mullaney, and J. C. Gillin. “Automatic sleep/wake identification from wrist activity”. In: *Sleep* 15.5 (1992), pp. 461–469.

[38] M. Altini, J. Penders, R. Vullers, and O. Amft. “Estimating energy expenditure using body-worn accelerometers: a comparison of methods, sensors number and positioning”. In: *IEEE journal of biomedical and health informatics* 19.1 (2015), pp. 219–226.

[39] J. H. Choi, J. Lee, H. T. Hwang, J. P. Kim, J. C. Park, and K. Shin. “Estimation of activity energy expenditure: accelerometer approach”. In: *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the.* 2006, pp. 3830–3833.

[40] T. Matsumura, V. T. Chemmalil, M. L. Gray, J. E. Keating, R. L. Kieselbach, S. B. Latta, N. Occhialini, E. Kinnal, S. O’Toole, and R. A. Peura. “Device for measuring real-time energy expenditure by heart rate and acceleration for diabetic patients”. In: *Bioengineering Conference, 2009 IEEE 35th Annual Northeast.* 2009, pp. 1–2.

[41] W. Dargie. “A stochastic fusion technique for removing motion artefacts from the measurements of a wireless ECG”. In: *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2016 IEEE International Conference on.* 2016, pp. 449–454.

[42] B. Kemp, A. Värri, A. C. Rosa, K. D. Nielsen, and J. Gade. “A simple format for exchange of digitized polygraphic recordings”. In: *Electroencephalography and clinical neurophysiology* 82 (1992), pp. 391–393.

[43] A. Schäfer and K. W. Kratky. “Estimation of breathing rate from respiratory sinus arrhythmia: comparison of various methods”. In: *Annals of Biomedical Engineering* 36.3 (2008), pp. 476–485.

[44] V. Jeyhani, T. Vuorinen, M. Mäntysalo, and A. Vehkaoja. “Comparison of simple algorithms for estimating respiration rate from electrical impedance pneumography signals in wearable devices”. In: *Health and Technology* (2016), pp. 1–11.

[45] S. Mahdiani, V. Jeyhani, M. Peltokangas, and A. Vehkaoja. “Is 50 Hz high enough ECG sampling frequency for accurate HRV analysis?” In: *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE.* 2015, pp. 5948–5951.

[46] S. Mahdiani, J. Vanhala, and J. Viik. “A novel generic algorithm for robust physiological signal classification”. In: *XIV Mediterranean Conference on Medical and Biological Engineering and Computing 2016.* 2016, pp. 1044–1049.

[47] T. K. a ki. “Effects of the human body on single-layer wearable antennas”. In: *Tampere University of Technology. Publication-Tampere University of Technology. Publications; 1025* (2012).

APPENDIX A. CALCULATION OF THE VALUE OF RESISTORS FOR I^{2}C PROTOCOL

The pull-up resistors for I^{2}C protocol play an important role in devising a compromise acceptable by both the power consumption and the speed characteristics of the protocol/system. The higher value for resistors impose a smaller power consumption (when the lines are driven low by either the master or the slaves) but builds up a large time constant in conjunction with the capacitance of the bus and therefore may cause the bus not to reach its expected level before the next coming edge. On the other hand, a strong pull-up might prevent the devices from being able to drive low the bus and increases the power consumption. In this case, due to the small time constant of the RC model (R is the pull-up resistor and C is the capacitance of the bus) the bus is suitable for faster data/clock transmission.

A range for the pull-up resistors can be calculated as following. According to the datasheets, the accepted voltage levels by the master and the three slave devices are mentioned in Table 1.

Table 1 *High and Low voltage thresholds and line capacitance for the devices in I^{2}C network*

Device	**High Level (min.)**	**Low Level (max.)**	**C$_{\text{i - SCL}}$ (max.)**	**C$_{\text{io - SDA}}$ (max.)**
nRF52832	0.7 × VDD	0.3 × VDD	3 pF	3 pF
TCA6408A	0.7 × VDD	0.3 × VDD	7 pF	8 pF
MPU9250	0.7 × VDD	0.3 × VDD	10 pF	10 pF
BQ25120	0.275 V †	2.475 V †	10 pF	10 pF

† Assumed that the pull-up voltage is 3.3V

Assuming that the threshold for high and low levels for all the devices are $0.7VDD$ and $0.3VDD$, respectively, the following can be calculated:

$$V_{t_1} = 0.3 \times V_{DD} = V_{DD}(1 - e^{\frac{t_1}{RC}}), \tag{1}$$

$$V_{t_2} = 0.7 \times V_{DD} = V_{DD}(1 - e^{\frac{t_2}{RC}}), \tag{2}$$

Then t_1 and t_2 can be calculated:

$$t_1 = 0.36RC, t_2 = 1.20RC \tag{3}$$

The maximum rise time for I^{2}C can be expressed as

$$t_r = t_1 - t_2 = 0.8473RC \tag{4}$$

Therefore, the maximum pull-up resistor can be calculated by

$$R_{(max)} = \frac{t_r}{0.8473 \times C}, \tag{5}$$

where t_r is 300 ns for fast mode of I^{2}C (400 Hz).

The minimum value for the pull-up resistor is a function of V_{DD}, minimum sink current and the maximum low output and can be calculated using equation 6.

$$R_{(min)} = \frac{V_{DD} - V_{OL}(max)}{I_{OL}}. \tag{6}$$

The minim sink current (I_{OL}) for the fast mode is 3 mA.

The bus capacitance in Equation 8 is the total capacitance of I^{2}C devices in the bus and the capacitance built in the PCB traces and vias.

The capacitance of microstrips, striplines and vias can be calculated using the MATLAB scripts included at the end of this appendix. Table 2 shows the calculated values using these scripts via the length of the traces reported by Altium software for each bus (SDA and SCL), separately.

Assuming 8 pF for both lines of TCA6408A and considering the maximum PCB capacitance, the total capacitance for both SDA and SCL lines can be estimated by

$$C_{I^2CBUS} = 3 + 8 + 10 + 10 + 1.5 \tag{7}$$

With an additional capacitance headroom of 10pF, the maximum pull-up resistor is

$$R_{(max)} = \frac{300ns}{0.8473 \times 42pF} = 8.4k\Omega, \tag{8}$$

Table 2 *The capacitance estimated from the PCB elements*

		SCL	SDA
Trace Length (in.)	Top Layer	0.62	0.51
	Mid. Layer	0	0.12
No. Via		0	2
Trace Height above Ground (in.)	Top Layer	0.02	0.02
	Mid. Layer	-	0.01
Trace Width (in.)		0.006	0.006
Trace Thickness (in.)		0.0014	0.0014
Permittivity of Material		4.5	4.5
Capacitance (pF)	Top Layer	0.89	0.72
	Mid. Layer	0	0.22
	Via	0	0.50
	Total	0.89	1.44

In this project, the values closer to R_{max} are of interest since they impose a smaller power consumption that also preserves the signal integrity for the required speed. The closest standard resistor for the maximum pull-up resistor is $8.2k\Omega$.

```
function [ZMSTRIP, CMSTRIP, LMSTRIP, T, TperInch, Ttolerance]=...
    microStripParameters(h, w, t, er, x, tolerance)
% microStripParameters-Calculates characteristics of a microstrip
% The function implements the formulas from the following book:
% "High-Speed Digital Design, A HandBook of Black Magic"
% Howard W. Johnson & Martin Graham
% www.sigcon.com
% There are two types of these calculations in the book: 4
% "accurate" which are mentioned in Appendix C and "estimated"
% which are explenationed in Chapter 4.5
%
% Syntax: Z = microStripParameters(h,w,t,er,x)
%         [Z, C] = microStripParameters(h,w,t,er,x)
%         [Z, C, L] = microStripParameters(h,w,t,er,x)
%         [~, ~, ~, T] = microStripParameters(h,w,t,er,x)
%         [~, ~, ~, T1, T2] = microStripParameters(h,w,t,er,x)
%         [~,~,~,T1,T2,T3] = microStripParameters(h,w,t,er,x,tol)
%
% Inputs:
%   h           Trace height above ground (in.)
%   w           Trace width (in.)
%   t           Trace thickness (in.)
%   er          Relative permitivity of material between trace
%               and ground plane (dimensionless)
%   x             Trace length (in.)
%   tolerance     Includes three parameters showing the tolerance
%                 for h, w and er, respectively. If this input is
%                 included the Ttolerance output is available.
%
% Outputs:
%   ZMSTRIP       Characteristic Impedance (OHM)
%   CMSTRIP       Capacitance (pF)
%   LMSTRIP       Inductance (nH)
%   T             A table listing all detailed specifications
%   TperInch      A table listing important parameters per inch
%   Ttolerance    A table showing the changes with specified
%                 tolerance
%
% Example:        [Z, C, L, T1, T2, T3] = microStripParameters...
%                            (h, w, t, er, x, [0.002, 0.002, 0.1])
%
% Other m-files required: none
% Subfunctions: none
% MAT-files required: Only basic mathematical functions
%
% Author: Vala Jeyhani,
% Tampere University of Technology, Dept. of Automation Science
%   and Engineering
% email address: vala.jeyhani@yahoo.com
% December 2016; Last revision: 12-Dec-2016

  value(1,1) = eeff(h, w, t, er);
  value(2,1) = we(h, w, t);
  value(3,1) = zmstrip(h, w, t, er, 1);
  value(4,1) = lmstrip(h, w, t, x) * 1e9;
  value(5,1) = cmstrip(h, w, t, er, x) * 1e12;

  ZMSTRIP = value(3,1);
  CMSTRIP = value(5,1);
  LMSTRIP = value(4,1);

  description = {'Effective Relative Permittivity'; ...
    'Effective Width (in.)'; 'Impedance (Ohm)'; ...
    'Inductance (nH) (total)'; 'Capacitance (pF) (total)'};
  parameters = {'EEFF'; 'WE'; 'ZMSTRIP'; 'LMSTRIP'; 'CMSTRIP'};
  parameters = categorical(parameters);
  T = table(parameters, value, description);

  clear value
  value(1,1) = pmstrip(h, w, t, er) * 1e9;
  value(2,1) = lmstrip(h, w, t, 1) * 1e9;
  value(3,1) = cmstrip(h, w, t, er, 1) * 1e12;
  description = {'Propagation Delay (ns/in.)';...
    'Inductance (nH/in.) (per in.)';...
    'Capacitance (pF/in.) (per in.)'};
  parameters = {'PMSTRIP'; 'LMSTRIP'; 'CMSTRIP'};
  parameters = categorical(parameters);
  TperInch = table(parameters, value, description);

  if nargin > 5
    clear value
    if length(tolerance) < 3
      error('The tolerance must have 3 elements');
    end
    dh = tolerance(1);
    dw = tolerance(2);
    der = tolerance(3);

    value(1,1) = zmstrip(h + dh, w - dw, t, er - der);
    value(2,1) = zmstrip(h - dh, w + dw, t, er + der);
    value(3,1) = cmstrip(h + dh, w - dw, t, er - der, x) * 1e12;
    value(4,1) = cmstrip(h - dh, w + dw, t, er + der, x) * 1e12;
    value(5,1) = lmstrip(h + dh, w - dw, t, er - der) * 1e9;
    value(6,1) = lmstrip(h - dh, w + dw, t, er + der) * 1e9;
    description = {'Impedance Upper Range (Ohm)';...
      'Impedance Lower Range (Ohm)';...
      'Capacitance Upper Range (pF)';...
      'Capacitance Lower Range (pF)';...
      'Inductance Upper Range (nH)';...
      'Inductance Lower Range (nH)'};
    parameters ={'ZMSTRIP-UP'; 'ZMSTRIP-LOW';'CMSTRIP-UP';...
      'CMSTRIP-LOW'; 'LMSTRIP-UP'; 'LMSTRIP-LOW'};
    parameters = categorical(parameters);
    Ttolerance = table(parameters, value, description);
  else
    Ttolerance = NaN;
  end
end

%% Effective relative permittiviy
function EEFF = eeff(h, w, t, er)
  % For skinny traces
  if w < h
    temp = ((1 + (12 * h/w))^(-0.5))+(0.04 * ((1 - (w/h)) ^ 2));
    E = ((er + 1) / 2) + (((er - 1) / 2) * temp);
  else
    % For wide traces
    temp = ((1 + (12 * h / w)) ^ (-0.5));
    E = ((er + 1) / 2) + (((er - 1) / 2) * temp);
  end
  EEFF = E - (((er - 1) * t / h) / (4.6 * (sqrt(w / h))));

  % When w/h is skinny, you get the average of the PCB
  % permittivity, er and the permittivity of air. When w/h is
  % wide (the trace is very close to ground plane) you get er.
end

%% Effective Trace Width (in.)
function WE = we(h, w, t)
  % For skinny traces
  if (2 * pi * w) < h
    WE = w + ((1.25 * t) / (pi) * (1 + log(4 * pi * w / t)));
  else
    % For wide traces
    WE = w + ((1.25 * t) / (pi) * (1 + log(2 * h / t)));
  end
end

%% Characteristic Impedance (OHM)
function ZMSTRIP = zmstrip(h, w, t, er, checkAccuracy)
  if nargin > 4
    if checkAccuracy == 1
      if (((t / h) > 0) & ((t / h) < 0.2)) == 1
        if (((w / h) > 0.1) & ((w / h) < 20)) == 1
          if ((er > 0) & (er < 16)) == 1
            disp(['Impedance: Accuracy of better',...
              'than 2 percent is guaranteed']);
          else
            warning(['Impedance: Accuracy of better than 2',...
              'percent is NOT guaranteed']);
          end
        else
          warning(['Impedance: Accuracy of better than 2',...
            'percent is NOT guaranteed']);
        end
      else
```

```
        warning(['Impedance: Accuracy of better than 2',...
          'percent is NOT guaranteed']);
      end
    end
  end

  WE = we(h, w, t);
  EEFF = eeff(h, w, t, er);
  % For skinny traces
  if w < h
    ZMS = 60 * log((8 * h / WE) + (WE / 4 / h));
    % For wide traces
  else
    temp = (WE / h) + 1.393 + (0.667 * log((WE / h) + (1.444)));
    ZMS = 120 * pi / temp;
  end
  ZMSTRIP = ZMS / sqrt(EEFF);
end

%% Propagtion Delay (s/in.)
function PMSTRIP = pmstrip(h, w, t, er)
  EEFF = eeff(h, w, t, er);
  PMSTRIP = 84.72 * sqrt(EEFF) * 1e-12;
end

%% Inductance (H)
% A dummy er value. Doesn't matter for inductance calculations
function LMSTRIP = lmstrip(h, w, t, x)
  ZMSTRIP = zmstrip(h, w, t, 1);
  PMSTRIP = pmstrip(h, w, t, 1);
  LMSTRIP = PMSTRIP * ZMSTRIP * x;
end

%% Capacitance (F)
function CMSTRIP = cmstrip(h, w, t, er, x)
  ZMSTRIP = zmstrip(h, w, t, er);
  PMSTRIP = pmstrip(h, w, t, er);
  CMSTRIP = PMSTRIP / ZMSTRIP * x;
end
```

Program 1 *Code for calculating the characteristics of the PCB micro strips*

```
function [C, L, T] = viaParameters(d, d1, d2, h, t, er)
% viaParameters-Calculates the capacitance and inductance of via
% The function implements the formulas from the following book:
```

```
% "High-Speed Digital Design, A HandBook of Black Magic"
% Howard W. Johnson & Martin Graham
% www.sigcon.com
% See Chapters 7.2 and 7.3
%
% Syntax:    [C, L, T] = viaParameters(d, d1, d2, h, t, er);
%
% Inputs:
%   d             Diameter of via
%   d1            Diameter of pad surrounding via (in.)
%   d2            Diameter of clearance hole in ground plane (in.)
%   h             Length of via (in.)
%   t             Thickness of printed circuit board (in.)
%   er            Relative electric permeability of circuit board
%   material

% Outputs:
%   C             Capacitance (pF)
%   L             Inductance (nH)
%   T             A table listing all detailed specifications
%
% Example:        [Z, C, L, T1, T2, T3] = microStripParameters...
%                     (h, w, t, er, x, [0.002, 0.002, 0.1])
%
% Other m-files required: none
% Subfunctions: none
% MAT-files required: Only basic mathematical functions
%
% Author: Vala Jeyhani,
% Tampere University of Technology, Dept. of Automation Science
%     and Engineering
% email address: vala.jeyhani@yahoo.com
% December 2016; Last revision: 12-Dec-2016

C = 1.41 * er * t * d1 / (d2 - d1);
L = 5.08 * h * (log(4 * h / d) + 1);

description = {'Capacitance (pF)'; 'Inductance (nH)'};
parameters = {'C'; 'L'};
parameters = categorical(parameters);
value(1,1) = C;
value(2,1) = L;
T = table(parameters, value, description);
```

Program 2 *Code for calculating the characteristics of the vias in PCB*