

SMART SHOE GAIT ANALYSIS AND DIAGNOSIS: DESIGNING AND
PROTOTYPING OF HARDWARE AND SOFTWARE

A Thesis

Submitted to the Faculty

of

Purdue University

by

Seshasai Vamsi Krishna Peddinti

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Electrical and Computer Engineering

May 2018

Purdue University

Indianapolis, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Maher Rizkalla, Co-Chair

Department of Electrical and Computer Engineering

Dr. Mangilal Agarwal, Co-Chair

Department of Mechanical Engineering

Dr. Mohamed El-Sharkawy

Department of Electrical and Computer Engineering

Approved by:

Dr. Brian King

Head of the Graduate Program

To my father who taught me that success can only be earned from hard work, my mother who taught me patience, my sister who taught me perseverance and to Madhumita for her unconditional love and support.

ACKNOWLEDGMENTS

This work would not have been possible without the support of my advisor, Dr. Mangilal Agarwal. I will forever be in debt to him for his patience, financial and emotional support, suggestions and encouragement to achieve my career goals. When everything I did went awry, I still remember the words you told me, "Never give up, work harder than before, don't let failures discourage you." Those words drove me to complete this thesis. I would like to thank my adviser Dr. Rizkalla, for being encouraging me from the first semester. As my teacher and mentor, he has taught me more than I could ever give him credit for here. The immense support you provided in writing the papers and thesis helped me to grow academically. Thank you for being my inspiration and role model. Also, I would like to thank Dr. Mohamed El-Sharkawy for all the suggestions during different stages of my thesis.

I am thankful for the ECE department for providing the financial support for four semesters. I would like to thank Sherrie Tucker on behalf of all the international students for her assistance before and after arriving to the University. I am grateful to all of those with whom I have had the pleasure to work during this and other related projects.

Nobody has been more important to me in the pursuit of this thesis than the members of my family. I would like to thank my parents, whose love and guidance are with me in whatever I pursue. They are the ultimate role models. Most importantly, I wish to thank Madhumita, for being with me throughout this journey.

TABLE OF CONTENTS

	Page
LIST OF TABLES	viii
LIST OF FIGURES	ix
ABSTRACT	xii
1 INTRODUCTION	1
1.1 Introduction	1
1.2 Biomechanics of lower body	2
1.3 Organization of thesis	8
2 CLASSIFICATION OF GAITS	10
2.1 Gait associated with Parkinson's disease	10
2.2 Gait associated with Hemiplegia	10
2.3 Gait associated with Diplegia	11
2.4 Gait associated with Ataxia	12
2.5 Neuropathic Gait	12
2.6 Myopathic Gait	13
2.7 Choreiform Gait	14
2.8 Gaits associated with Injury	14
3 HARDWARE SELECTION	16
3.1 Pressure sensors	16
3.2 Inertial motion sensor	18
3.3 Mode of Communication	19
3.4 Microcontroller	20
4 SOFTWARE IMPLEMENTATION	23
4.1 Introduction	23
4.2 Analog connections to motion sensor and FSRs	27

	Page
4.2.1 Data Collection	31
4.3 Working of SAADC/FSRs	34
4.3.1 Basic Setup	34
4.3.2 Data Collection	38
4.4 Working of MPU	39
4.4.1 Basic Setup	39
4.4.2 Interrupt Handling	41
4.4.3 Filtering Techniques	41
4.5 Working of BLE	44
4.5.1 Introduction to BLE in nRF52	44
4.5.2 Advertising	44
4.5.3 Basic Setup	46
4.5.4 Flowchart	49
4.6 Power Consumption	49
4.6.1 Power Consumption during Advertising	50
4.6.2 Power Consumption during Peripheral Connection	52
5 HARDWARE PROTOTYPING AND DESIGN	54
5.1 Introduction	54
5.2 Model 1	60
5.2.1 Schematic	60
5.2.2 PCB Layout	61
5.2.3 Drawbacks	62
5.3 Model 2	63
5.3.1 Schematic	63
5.3.2 PCB layout	64
5.3.3 Drawbacks	64
5.4 Model 3	65
5.4.1 Schematic	65

	Page
5.5 Effect of Relative Humidity and Temperature in Shoe Sole	66
6 PROTOTYPING OF THE CUSTOM PCB	68
6.1 Testing and Debugging	68
6.2 Verification of Working of the PCB	68
6.3 Initiation of the Board	70
7 RESULTS	73
7.1 Output from FSRs	73
7.1.1 Flexibility and Sensitivity	74
7.2 Output from the Motion Sensor	74
7.3 BLE Outputs	75
7.4 Improper Gait: Analysis and Diagnosis	75
7.5 Flowchart	77
7.6 PCB Layouts	77
8 DISCUSSION AND FUTURE WORK	114
8.1 Future Work	115
9 CONCLUSIONS	116
REFERENCES	117

LIST OF TABLES

Table	Page
2.1 Comparison between different Gaits	15
3.1 Comparison between modes of communication	19
3.2 Comparison between nRF51 and nRF52	21
3.3 Comparison between CC2640 and nRF52	22

LIST OF FIGURES

Figure	Page
1.1 Comparison of Gait terminology [12]	4
1.2 Zeno pathway- an example of pressure pathway	6
2.1 Person suffering from a Parkinson's Disease [32]	11
2.2 Person suffering from Ataxia [35]	12
2.3 Example of a Neuropathic Gait [37]	13
3.1 Placement of FSRs(Blue) and Motion sensor(Red) on the foot	17
4.1 nRF52 Development Kit	24
4.2 nRF 52 in JLINK mode	25
4.3 Loading nRF52 with SoftDevice via nrfjprog	25
4.4 Path at user defined variables	26
4.5 Keil working interface	28
4.6 nRF52832 pin assignments [67]	29
4.7 TWI Master Slave block diagram [67]	30
4.8 Comparison between three pin and two pin FSRs.	34
4.9 MPU9250-motion sensor used	40
4.10 Working of Kalman filtering technique	43
4.11 Communication via BLE in nRF52	45
4.12 nRF52 BLE advertising channels	46
4.13 Flowchart for BLE data processing	49
4.14 Smart shoe prototype design using nRF52 development kit as the base station	50
4.15 Power consumption during BLE advertising event	51
4.16 Power consumption during BLE peripheral connection event	52
5.1 Flowchart of PCB design flow	57

Figure	Page
5.2 Track Resistance calculator [83]	59
5.3 QFN48 internal LDO reference model [67]	61
5.4 Fabricated Custom PCB based on Model 1	62
5.5 Model 1 board parameters	63
5.6 Model 2 board parameters	64
5.7 Fabricated Custom PCB based on Model 3	65
5.8 Model 3 board parameters	67
6.1 Readings taken from oscilloscope	69
6.2 Bootloading with JLINK Debugger	71
6.3 Output received at JLINK Commander	71
6.4 Working set up of custom PCB with nRF52 Development board	72
7.1 Prototype of the smart shoe	78
7.2 Outputs obtained from FSRs with Gain=1/6.	79
7.3 Outputs obtained from FSRs with Gain=1/5.	80
7.4 Outputs obtained from FSRs with Gain=1/4.	81
7.5 Outputs obtained from FSRs with Gain=1/3.	82
7.6 Outputs obtained from FSRs with Gain=1/2.	83
7.7 Outputs obtained from FSRs with Gain=1.	84
7.8 Outputs obtained from FSRs with Gain=2.	85
7.9 Outputs obtained from FSRs with Gain=4.	86
7.10 Flexibility test of the FSRs.	87
7.11 COMSOL simulation of single FSR at 350KPa	88
7.12 COMSOL simulation of two FSRs at 350KPa	88
7.13 Roll and Pitch values of 2G accelerometer and 2000 DPS Gyroscope	89
7.14 Roll and Pitch values of 16G accelerometer and 2000 DPS Gyroscope	90
7.15 Output from nRF toolbox Application when the device is advertising.	91
7.16 Output from nRF toolbox Application-UART module when the device is connected.	92

Figure	Page
7.17 Output from nRF toolbox Application- sensor values received via BLE.	93
7.18 Output from nRF toolbox Application-UART module when the device is connected.	94
7.19 Output of a Normal Gait	95
7.20 Output of a Normal Gait in BLE	96
7.21 BLE results for Normal Gait Diagnosis	97
7.22 Output of a Parkinson's Gait	98
7.23 Output of a Parkinson's Gait in BLE	99
7.24 BLE results for Parkinson's Gait Diagnosis	100
7.25 Output of an Injured Gait	101
7.26 Output of an Injured Gait in BLE	102
7.27 BLE results for Parkinson's Gait Diagnosis	103
7.28 Output of a Hemiplegic Gait	103
7.29 Output of an Hemiplegic Gait in BLE	104
7.30 BLE results for Hemiplegic Gait Diagnosis	105
7.31 Flowchart of the code	106
7.32 Schematic of Board1	107
7.33 Layout of Board1	107
7.34 Schematic of Board2	108
7.35 Layout of Board2	109
7.36 Schematic of Board3 Part 1	110
7.37 Schematic of Board3 Part2	111
7.38 Schematic of Board3 Part3	112
7.39 Layout of Board3	112
7.40 Outputs from two FSRs and Motion sensor	113

ABSTRACT

Peddinti, Seshasai Vamsi Krishna. M.S.E.C.E., Purdue University, May 2018. Smart Shoe Gait Analysis and Diagnosis: Designing and Prototyping of Hardware and Software. Major Professors: Mangilal Agarwal and Maher Rizkalla.

Gait analysis plays a major role in treatment of osteoarthritis, knee or hip replacements, and musculoskeletal diseases. It is extensively used for injury rehabilitation and physical therapy for issues like Hemiplegia and Diplegia. It also provides us with the information to detect various improper gaits such as Parkinson's disease, Hemiplegic and diplegic gaits. Though there are many wearable and non-wearable methods to detect the improper gate performance, they are usually not user friendly and have restrictions. Most existing devices and systems can detect the gait but are very limited with regards of diagnosing them. The proposed method uses two A201 Force sensing resistors, accelerometer, and gyroscope to detect the gait and send diagnosed information of the possibility of the specified improper gaits via Bluetooth wireless communication system to the user's hand-held device or the desktop. The data received from the sensors was analyzed by the custom made micro-controller and is sent to the desktop or mobile device via Bluetooth module. The peak pressure values during a gait cycle were recorded and were used to indicate if the walk cycle of a person is normal or it has any abnormality.

Future work: A magnetometer can be added to get more accurate results. More improper gaits can be detected by using two PCBs, one under each foot. Data can be sent to cloud and saved for future comparisons.

1. INTRODUCTION

1.1 Introduction

Locomotion is a process by which a person moves from position to other. Humans are able to use only two limbs to facilitate them in walking. This is because of the evolution dating back to thousands of years. Locomotion associated with only two limbs is generally called as bipedal. It should be noted that the human infants crawl on all the four limbs. Slowly and gradually, they learn how to walk on two limbs. Since walking is learn by practice, it is usually visibly different when compared. However, the basic locomotion between all remains the same which will be discussed. Walking is a relative term, that can be done in many ways. The way in which the foot moves can be described as the gait or walk cycle. Walking is a type of locomotion where the remaining body of the individual is usually kept erect or straight and facilitates in forward or backward motion while maintaining the balance of an individual. Here, the balance is maintained by moving one leg forward while fixing the other leg to ground. Gait analysis has been the primary research area for many years. Analyzing the gait and using it to detect and diagnose issues, if any, helps the physicians in understanding the issue in the test subject with better accuracy. It has been used in the fields of sports [1], injury rehabilitation [2], detecting of various musculoskeletal diseases, and knee and hip replacements [3,4]. The applications can be further extended to the forensic departments [5] There is a bright future for usage of gait in biometrics [6]. Gait analysis can also be used for detection of issues such as Parkinson's disease [7], Hemiplegic gait [8] and Diplegic gait [9]. It can also be used for identity detection [10]. To better understand a human gait, the mechanics associated with the motion of the upper limb and hips are to be discussed.

1.2 Biomechanics of lower body

Femur is the longest bone in the human body. The joint that holds the top end of the Femur is called as the hip joint. This hip joint creates an angle which facilitates the motion of the body. The ideal angle made by the hip joint with respect to the Femur is about $120\text{-}135^\circ$ for adults. This joint helps in rotation, movement towards body mid-line and movement away from body mid-line. The bottom end of the Femur is connected to ankle joint. The angle formed between these two joints is about 7° with respect to Femur.

There are two types of forces acting at these joints; externally and internally generated. Externally generated forces are the forces acting based on gravity pull, while the internally generated forces are the forces acting based on muscular contraction. The bottom end of the Femur is connected to another joint called as knee joint, the largest joints in human body. This is very complicated joint that helps in giving a proper response based on the internally generated forces. This joint is very important in maintaining overall balance of our body.

Tibia and Fibula are two bones which connect knee joint to another joint, the ankle joint. This ankle joint has two other bones connected to its bottom, Talus and Calcaneus. Talus is the top part of the foot and calcaneus is the bone that supports the heel of the foot. The ankle joint facilitates dorsiflexion, which is the vertical upward movement of the foot and Plantar flexion, the vertical downward movement of the foot. Apart from this, it also helps in continuing the motions introduced by the other two joints. This joint plays a vital role in motion of the body.

Human Gait is accompanied by two phases: Stance Phase and Swing Phase. Stance phase is a cumulative action made by our foot while at least one part of it still being in contact with the floor. Swing phase is the duration during which the entire foot is not in contact with the floor anymore. A normal gait cycle consists of 60% Stance phase and about 40% swing phase at any given cycle. Gait or walk cycle can

be explained in many ways using different terms, depending upon the application. The basic gait cycle can be defined by one of the two ways [11]:

1. Using old terminology: The cycle starts when Heel strikes the floor, followed by foot flat condition, a condition when the toe and the heel are flat on the floor. The foot stays in this foot flat condition for a period of time. This is called as mid stance. This ends when the heel of the foot is in the air while the rest of the foot still in contact with the surface, called heel off. Now, the rest of the foot, including heel, but except toe are not in contact. This condition is called toe off. Toe off becomes mid swing when the complete foot is in air. This ends when the heel of the foot lands on the ground, thus marking the end of one cycle.
2. Using new terminology: Initial contact to the ground is made by the heel. Loading response is the duration during which the load slowly starts to shift from one leg to the other, as seen in Figure 1.1. The mid stance condition is similar to both cases. Then it enters terminal stance, a stage wherein the other foot completes its swing phase. The selected foot now enters the swing phase and gets ready to enter that stage. This condition is called as pre swing which is then accompanied by initial swing. Initial swing continues till the selected foot crosses the femur of the other foot. The selected foot then enters mid swing. The mid swing ends when the foot is about to strike its heel to the floor, signaling the end or termination of the swing, named aptly as terminal swing.

It can be seen that the old system provides terms closely associated with time whereas the new system uses terms that are closely associated with force. The selection of the terminology can be application specific as mentioned before. Though the wordings are different, both systems are standard in analyzing and detecting the gait. Apart from the discussed terms, there are still many other parameters that can be used to detect a gait, a few of those have been discussed below:

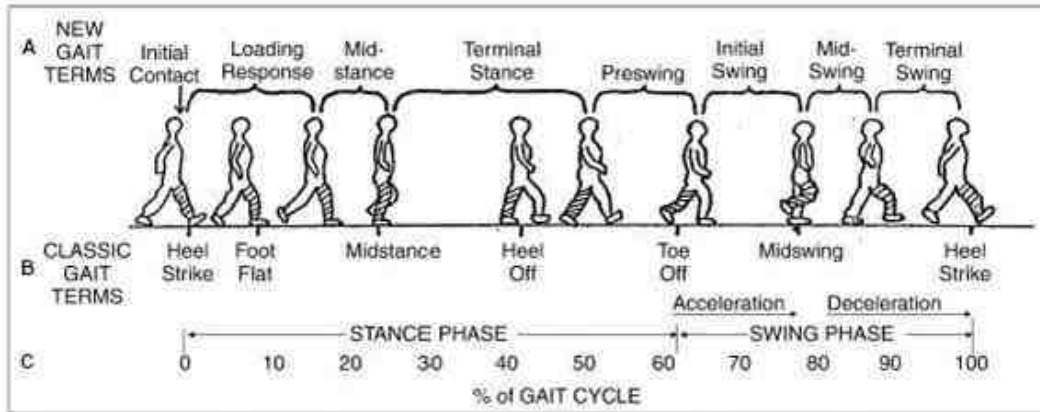


Fig. 1.1. Comparison of Gait terminology [12]

1. Velocity with which a person or a test subject walks plays a major role in the gait analysis. The velocity can help in differentiate and understand whether a person is running, walking or jumping.
2. Step length also plays a crucial role in analyzing gait better. It is the linear distance between the selected two linear points on the line of progression from the floor contact of opposite feet.
3. Stride length is similar to step length with the only difference being that, the stride length is related to a single foot whereas step length is related to opposite feet. Stride length is the linear distance between the selected two linear points on the line of progression from the floor contact of same foot.
4. Stride or step width is the distance between two lines that would cut the distance in two equal halves.
5. Cadence is the number of steps a person takes per unit time.
6. Stance and swing durations are the time duration when the foot is in contact and time duration when the foot is completely in the air, respectively.

7. Step time is the time taken from one foot to contact the floor to the other foot to be contacted to the floor. This is generally 0.8 seconds for a normal human gait.

Although obtaining these values would be helpful for diagnosis purposes, these were not directly considered to be the means of detecting the gait. The disturbance in gait may also be caused due to aging. A continuous wear and tear tend to cause gait issues. According to a research conducted by World Health Organization (WHO) given in WHO Global Report on Falls Prevention in Older Age report,” 28-35% of people who are 65 and above fall each year, and this increases to 32-42% for all the people above 70 years of age. Also, the number of times a person falls increases with age. Each year, 2.5 million older people are treated in emergency departments for fall injuries. Falls are the most common cause of traumatic brain injuries (TBI). Adjusted for inflation, the direct medical costs for fall injuries are \$34 billion annually.” [13] So, the possibility of method to rectify improper gait has been primary concern to many researchers. There has been many ways in which the gait can be detected by obtaining the information about one or more of the parameters discussed above. These methods are broadly distinguished as non-wearable and wearable analysis.

Before the widespread boom of the usage of electronic devices to detect gait, a physical therapist, unassisted by sensory data collection tools, used to analyze gait based on what he could see. These tests, still being used, provides a set of rules and parameters to be considered by the therapist. A few such tests are, a timed 25ft walk where gait observations are noted [14], a Tinetti Performance-Oriented Mobility Assessment where observations are collected as the subjects follow a sequence of actions, such as, walking on the predefined line on the floor, turning, walk back and sit on a chair [15], or using videotape to capture patients walking patterns and critique their gait parameters and facets [16]. While qualitative methods can detect abnormalities and are useful in rehabilitation and training, the role of qualitative visual observation and its accuracy is sometimes questionable.

There is a common misconception that gait abnormalities can be easily detected. More to the point, visual methods typically fail to provide accurate information regarding the location of the abnormality and the reasons underlying it. With the acceptance of using of electronic equipment into medical field, a large variety of devices have been designed to get the output of the gait.



Fig. 1.2. Zeno pathway- an example of pressure pathway

The traditional method for gait analysis is using multiple cameras to capture the motion of the test subject who walks on a platform which has an ability to measure Ground Reaction Force (GRF) [17], the amount of force exerted by the ground on the body that is in contact with it. Here, Patient walks on the platform and the cameras records his/her motion patterns and the sensors embedded on the platform helps us to get the data related to the force applied by the patient on to floor in contact. However, this technique is very costly and needs trained experts to read the data that is obtained from the platform. It cannot be used to measure more than a single stride, making it incompetent for multiple strides. This technique cannot be used outdoors as it requires a large area to implement it. This means that the patient must go to a professional motion laboratory to get the gait analysis to be done. This would be difficult for the patient and it takes a lot of time to get the data. Also, patients feel uncomfortable due the amount of equipment that must be worn to get the test done. There are many walk aiding robots, such as RT Walker [18], PAMM [19], etc. However, these tools are very complex to build and are expensive as well.

One might ask about the role of visual observation as there is a misconception that the gait abnormalities can be detected quite easily and since it is a main feature of a human body, it is easy to detect such variations in gait. However, it always fails to provide us with accurate information about where the abnormality occurred or the reason for it. Hence, visual observation is not an ideal tool to measure gaits. Some tools used to analyze gait often opt different types of image processing techniques. Laser range scanner is a type of image processing gait analysis technique which uses laser beams to analyze the gait of the test subject [20]. However, such a scanner is very costly to build and this requires many such scanners to obtain some accurate results. This furthermore increases the complexity and cost inefficiency for this technique. Infrared Thermography or infrared thermal imaging is another image processing technique which helps us to obtain gait information [21]. This uses infrared energy and the equipment used to generate it is also quite expensive. To obtain more accurate information, many such equipment need to be used as an array. Microsoft Kinect has also been used to recognize the gait [22]. The camera sensor used in the Microsoft Kinect has been used to detect the gait via Cellular Neural Network. A comparatively cost-efficient method, this has its own limitations. It cannot be used in real time settings as the device requires relatively dark images for better accuracy. Though the above mentioned techniques are very accurate, they are very costly, difficult to implement and most importantly, not portable. These can be categorized as non-wearable sensing equipment. To overcome the issues posed by these equipment, research on effective and portable methods to detect gait have been increased. Ultrasonic sensors are widely used to obtain the information about various parameters of gait [23]. Though these sensors provide accurate data, they are very expensive and heavy (about 80grams). The use of Electromyography for gait detection has been widely used [24]. This uses EMG signals from the limbs of a person and detects the gait based on those readings. However, this technique requires a trained expert to interpret the data.

The usage of wearable devices to analyze gait is primary concentration for research regarding this field. The reason to develop such devices is that these sensors occupy very less space and can be used at home. At the initial stages of development of such devices, the usage of universal serial bus (USB) cable was inconvenient for the users [25]. However, the inclusion of wireless transmission to these sensors helped in better freedom of movement to the individuals. These devices can usually send information to smartphone devices [26] or a personal computer via Bluetooth, wifi or zigbee. A number of such devices has been developed [27, 28]. However, it can only detect a gait but not diagnose it. Though some work has been done to diagnose the gait as well, a limited number of improper gaits could be detected by one device at a time [29]. Hence, there is a requirement of designing a wireless device that can detect more than one type of gait and provide a therapeutic feedback to diagnose the issue.

1.3 Organization of thesis

Chapter 2: Consists of the classification of gaits, identification of improper gaits based on sensors, and the improper gaits selected for the prototyping.

Chapter 3: Consists of comparison of different sensors which are preexisting, reason for selection of a sensor, comparison between various Micro controllers and the reason for selecting the one to be used in this thesis. It also discusses about the importance of BLE module.

Chapter 4: Discusses about the software implementation, Keil interface, analysis of different drivers used and working of the proposed code with nrf52 base station. It also discusses about the setup and working of sensors and BLE module.

Chapter 5: Discusses about the hardware prototyping and design. This chapter explains about the Custom Printed circuit boards designed and the criteria considered in finalizing the PCB. It also talks about the reason to select EAGLE as the primary tool to design the PCB. It talks about selection of certain types of integrated circuits and the reason to place it.

Chapter 6: Talks about the steps taken to make the custom PCB designed to work. The issues and the solution steps taken were discussed in it.

Chapter 7: Talks about results. It explains all the results obtained and how it is useful for this particular thesis. It also discusses about the overall impact the work done has for the field of Gait.

Chapter 8: Talks about discussions and future work. It further talks about the future implementation of the work done. It explains about the work that can be done on this project to make it more developed in the future.

Chapter 9: Talks about conclusions which briefs about the work done in this thesis.

2. CLASSIFICATION OF GAITS

There are a number of improper gaits caused due to multiple factors such as musculoskeletal or neurological issues. In this chapter, various improper gaits are discussed in detail and the concept used to detect and analyze the selected gaits has been described.

2.1 Gait associated with Parkinson's disease

Parkinson's disease is a type of neurological brain disorder which leads to slow and jerky motion of the body. According to a report by parkinsons.org, this disease is generally observed in people above the age of 60 [30]. It is also the 14th leading cause of death in United States. The jerkiness is caused due to the gradual loss of motor control. Gait associated with this disease is generally called as Parkinson's gait or parkinsonian gait. The Figure 2.1 shows the gait of a person suffering from Parkinson's disease.

A person suffering with this gait usually leans the upper body forward and drags the feet in swift motion. In normal gait, the heel contact occurs which is continued by toe off. However, in Parkinson's or Parkinsonian gait, the gait mostly consists of foot flat condition [31]. Analyzing the flat foot condition will help in detecting this abnormality of the gait.

2.2 Gait associated with Hemiplegia

Cerebral palsy is a brain disorder that occurs due to a damage in a part of the brain that controls muscles. Hemiplegia is a type of cerebral palsy that effects one vertical side of the body. This is not a progressive disease and hence, detection at

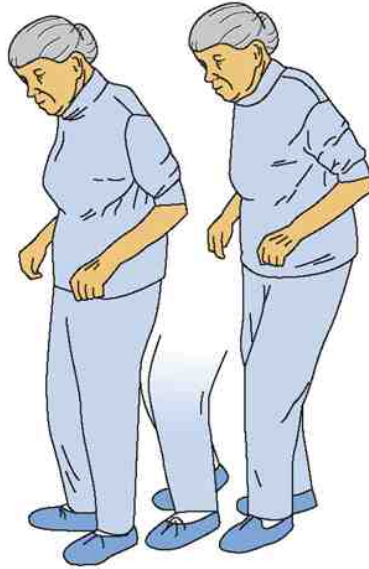


Fig. 2.1. Person suffering from a Parkinson's Disease [32]

an early age helps in providing a possible cure. A gait associated with this disease is called hemiplegic gait.

Person suffering from this gait generally tend to move the foot of the effected side away from the body in a circular motion. Using accelerometer and gyroscope to detect this deviation away from body helps in this gait diagnosis [8]. The following image explains the gait cycle of a Hemiplegic gait.

2.3 Gait associated with Diplegia

Diplegia is another type of cerebral palsy which is similar to Hemiplegia. The only difference between these two brain disorders is that, Hemiplegia effects only one vertical side of the body whereas Diplegia effects both the vertical sides of the body. Just like Hemiplegia, this disease can be cured with early detection and physical therapy. A gait associated with this gait is called as Diplegic gait.

Person suffering from this gait generally tend to move both feet away from the body in a circular motion. Just like Hemiplegic gait detection, an accelerometer and gyroscope on each foot can be used to analyze this gait [33].

2.4 Gait associated with Ataxia

According to National Ataxia Foundation, Ataxia is a regenerative neurological disease that can affect any age group [34]. This is a degenerative disease that restricts normal bodily movements. About 150,000 people are affected by it every year in United States. This type of disease cannot be cured with physical therapy and research for cure is still under progress. Gait associated with this disease is called Ataxic gait.



Fig. 2.2. Person suffering from Ataxia [35]

Person suffering from this gait generally tend to show wide range of gait abnormalities such as inability to balance, and difficulty in walking. This causes difficulty in detection of it via wearable devices as it needs a definitive boundary condition to analyze. The Figure 2.2 details about Ataxic gait.

2.5 Neuropathic Gait

In this gait, the person has difficulty in dorsiflexing the foot. This means that the vertical upward movement of the foot cannot be facilitated by the person. Here, the

foot is raised high before dropping it on the floor in foot flat condition. This is also called steppage or high stepping gait. This type of disease can be cured with regular physical therapy.

This gait can be detected by large pressure signals a heel and toes due to quick foot droppage.

2.6 Myopathic Gait

According to the National Institute of Neurological Disorders and Strokes, Myopathy is a neuromuscular disorder that leads to muscle cramps and stiffness [36]. Myopathy are usually inherited or acquired. Treat for myopathy is dependent on its type. Common treatments include surgery or physical therapy. This disease may be progressive or fatal.

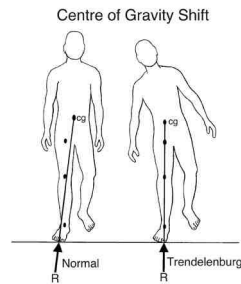


Fig. 2.3. Example of a Neuropathic Gait [37]

Gait associated with myopathy is called Myopathic gait. This occurs due to inability to stabilize the pelvis when the foot is lifted. Here, the body moves towards the right side when left foot is put forward to maintain stability. Similarly, the body moves towards the left side when the right leg is put forward to maintain the stability. This results in waddling motion.

This can be detected using motion sensors placed on the hips to detect the tilt in the body.

2.7 Choreiform Gait

Hyperkinesia is a term used to describe heavy and uncontrollable body movements associated with jerks and stiffness. This is one of the most common childhood psychiatric disorders [38]. It is also associated with Attention deficit hyperactivity disorder, also known as ADHD.

The gait associated with it is called as Hyperkinetic gait or Choreiform gait or Chorea. Due to its very irregular, jerky and involuntary movements of the upper and lower extremities, it is very difficult to detect this gait via wearable devices.

2.8 Gaits associated with Injury

Gait analysis plays a major role in injury rehabilitation, especially in the field of sports and athletics. Gaits related to different injuries show different results. Injury to knee or hips usually leads to very slow walking motion. Injury to foot leads to less air time for the affected foot. Usually, the injured person tends to avoid contacting the floor with the injured part of the foot.

Detection can be done based on the part of the foot, heel or toe, that is not in contact in the entire gait cycle. Depending on the application, sensors are selected based on the gaits that can be detected properly. Table 2.1 informs about the type of gait, case proposed to its possible detection and proposed sensors that can be used.

Chapter 3 will deal with the information about different hardware components used and the reason behind selection of each sensor was discussed in detail.

Table 2.1.
Comparison between different Gaits

Name of the abnormal gait	Case proposed for possible detection	Proposed Sensors used to analyze it
Parkinsons Gait	Continuous foot flat condition	Pressure sensors used to analyze it
Hemiplegic Gait	Circular motion when in air	Accelerometer and gyroscope on foot
Diplegic Gait	Circular motion when in air	Accelerometer and gyroscope on both feet
Ataxic Gait	Inability to balance, and difficulty in walking	Difficult to detect
Neuropathic Gait	High pressure readings during foot flat condition	Pressure sensors
Myopathic Gait	Waddling motion	Motion sensors at hips
Choreiform Gait	Jerky involuntary movements	Difficult to detect
Gaits due to Injury	Injured part of the foot not in contact	Pressure sensors

3. HARDWARE SELECTION

As discussed in the previous chapters, we require two types of sensors, namely, pressure sensor and inertial measurement unit sensor, i.e, accelerometer and gyroscope to get the readings from the gait of the test subject and analyze the output. A pressure sensor is required to get the readings regarding the pressure applied by foot which in turn helps in analyzing the human gait. There are a wide range of methods to detect the information obtained from sensors [39–41].

3.1 Pressure sensors

There are different types of pressure sensors, broadly divided into two types: the sensors which provide information based on the force applied, and the electronic sensors [42]. Electronic sensors generally use properties of the material that is based on thermal conductivity or ionization. These sensors are generally heavy when compared to sensors that work related to force applied. Also, these work for relatively lower pressures and are quite expensive, hence, force-based sensors are preferred. There are several such sensors available in the market. The difference in these sensors are based on the method opted by the sensor to collect the force related values. Though these have high precision, these sensors have restricted sensing capabilities and bulkier making it difficult to embed it on the shoe insole.

Force sensitive resistors are a type of sensors whose force can be recorded by varying its resistance. This can also be called as pressure sensitive resistors as, the data obtained from this sensor is taken from the its active sensing surface on which the force is applied. These slim and flexible sensors generally consist of four different layers, an electrically insulating plastic, an active area, plastic spacer and flexible substrate. These are not as accurate as other force-based sensors, but its slim size

and flexibility overcomes the drawback [43,44]. Hence, Force sensitive resistors, also called as FSRs, are used as the pressure sensors for the project. For this thesis, Flexiforce A201 sensors have been selected as the sensors to detect pressure [45]. It is a FSR that can detect up to 450N of force.

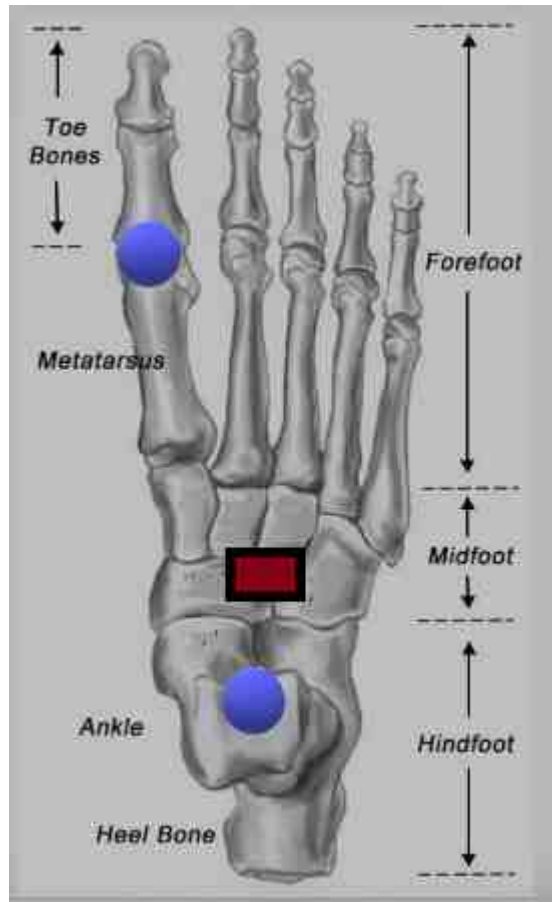


Fig. 3.1. Placement of FSRs(Blue) and Motion sensor(Red) on the foot

The placement of these sensors plays a major role in getting the desired output. The bottom part of the foot is divided into three different sections as shown in the figure, forefoot, midfoot and hindfoot. Forefoot consists of the toe bones and metatarsus, a group of bones between ankle and toes. Midfoot consists of bones supporting the arch on the foot. The hind foot consists of heel bone and direct contact with

ankle joint. Figure 3.1 shows two FSRs placed at 1st metatarsal region and at the heel bone along with the motion sensor placed closed to heel.

3.2 Inertial motion sensor

The angle at which the foot lands gives more information about gait and hands in the ability to analyze the data much better [46,47]. The data from accelerometer and gyroscope gives us roll and pitch. The data from magnetometer gives us values of yaw and provides us with accurate roll and pitch values. Hence, the usage of all three components is suggested. However, for the present application, roll and pitch values are enough to determine the required results.

The most commonly used and cost efficient IMUs currently available are IvenSense's 60x0 series [48]. This 60x0 series consists of two different types, namely, 6000 and 6050. These are also called motion processing units, in short, MPU. It consists of accelerometer and gyroscope. The major difference between these sensors is that MPU6000 works well for higher speed updates whereas MPU6050 is relatively slower [49]. There is another MPU in this series, MPU6500. This transmits data at very high frequency when compared with MPU6500, meaning that, its sensitivity is very high. It should be noted that all these MPU are of 6 degrees of freedom, 3 for accelerometer and 3 for gyroscope.

There is also MPU9x50 series, which has an added magnetometer in addition to accelerometer and gyroscope. MPU9150 has MPU 6050 with AK8975 compass [50]. This is a 9 degrees of freedom Motion processing unit, with 6 for MPU6050 and 3 for AK8975 compass [51]. MPU9250 is another MPU which uses MPU6050 with AK8963 [52,53]. This gives better resolution and better accuracy when compared with other MPUs. MPU9250 has been selected as the Motion sensor for this application. It can be placed anywhere on midfoot or hindfoot.

Using FSRs and Motion sensors, the human gait can be detected in various methods [54–57].

3.3 Mode of Communication

The data obtained from the above specified sensors needs to be scrutinized and processed. This processed data needs to be sent to the mobile device or desktop via BLE or Zigbee or Wifi. So, it is necessary to compare these three methods of communication and get the desired output. Table 3.1 compares the various popular modes data transmission [58, 59].

Table 3.1.
Comparison between modes of communication

	Bluetooth Low Energy	Near field communication(NFC)	ZigBee	Low Power WiFi
Frequency	About 2.5GHz	13.56MHz	About 8MHz or 2.5GHz	About 2.5GHz
Max potential data rate	1Mbps	425Kbps	250Kbps	54Mbps
Range	10m	10cm	>100m	30m
Complexity	Complex	Simple	Simple	Complex
Power profile	Days	Months to years	Months to years	Hours
Modulation	GFSK	ASK	BPSK and QPSK	64QAM

BLE or Bluetooth low energy works at a very high frequency compared to other methods. It also has a decent data rate but has lower range when compared with ZigBee or WiFi.

Also, the power profile is very low for BLE. NFC, as the name suggests works well only when it the device is within 10 cm. This is a simple system and lasts relatively very long but has lower data rate and limited range also poses a challenge.

ZigBee has a very flexible frequency range when compared to other means of communication. This lasts for a very long time, probably for years. This is very simple system with a range greater than 100 meters. However, it has the lowest data rate when compared to others at a mere 250Kbps.

Low power WiFi has the maximum data rate of all the four suggested methods. With a decent range of 30 meters, this complex system has its fare share of drawbacks as well. This has the highest power consumption when compared to others.

BLE is selected as the preferred means of communication. Though its range is not as good as ZigBee or WiFi, 10 meters will be sufficient for the project. Also, its power profile completely beats out WiFi. However, ZigBee lasts almost a year to the BLE that lasts for a few days. However, ZigBee has a very low data rate. With about four times better data rate than ZigBee, it stands out as the winner between these three. On the other hand, NFC has very good power profile, on par with ZigBee. Data rate is about 425Kbps, relatively bad but better when compared with ZigBee. However, the major drawback of this mode is its range. For our defined application, 10 cm cannot be an ideal range. Hence, BLE is our clear winner here.

3.4 Microcontroller

The mode of communication or data transfer is selected as BLE. Now, a microcontroller that has BLE embed in it is to be used. Nordic semiconductors and Texas instruments provide a wide range of such required microcontrollers. nRF series by Nordic semiconductors and CC series of Texas instruments are widely used as BLE microcontrollers for several applications. In Nordic semiconductors, nRF series is broadly distinguished as nRF51 and nRF52 series [60, 61]. The difference between the two systems was detailed in Table 3.2.

Table 3.2.
Comparison between nRF51 and nRF52

	nRF51	nRF52
CPU	ARM Cortex -M0	ARM Cortex-M4F
Flash memory	256KB	512KB
RAM	32KB	64KB
Clocks	Internal RC Oscillators	PLL
Serial interface	2x I2C master 2x SPI master/1x slave 1x UART CTS/ RTS No I2S No PDM 1x QDEC 3x (8/16/32) bits TIMER	2x I2C master/slave 3x SPI master/slave 1x UART CTS/ RTS 1x I2S 1x PDM 1x QDEC 5x 32 bits TIMER
Supply voltage	1.8-3.6V	1.8-3.6V
Gpio	31	32
Current	0.6uA when system off 2.6uA when system on	0.4 uA when system off 1.6uA when system on

With a powerful CPU of ARM Cortex M4F, better flash memory storage and RAM, nRF52 beats nRF51 in every field discussed above. Table 3.3 compares this with the most famous of Texas Instruments CC series, CC2640 [62].

With better core CPU, flash memory, RAM and lower power consumption, nRF52, is clearly better than CC2640. Though the sensitivity of CC2640 is overcoming its minor drawback, and therefore, the high sensitivity of nRF52 features the work.

nRF52832 is the microprocessor used within the the nRF52 micro-controller system. It has the same configuration as discussed throughout in addition to its ability

Table 3.3.
Comparison between CC2640 and nRF52

	CC2640	nRF52
Company	Texas Instruments	Nordic Semiconductors
CPU	ARM Cortex -M3	ARM Cortex-M4F
Flash memory	128KB	512KB
RAM	20KB	64KB
Current RX/TX	5.9mA/6.1mA	5mA/4.6mA
TX power	+5dBm	+4dBm
Receiver Sensitivity	-97dBm	-96dBm

to access near field communication along with BLE, and this makes it more attractive to use.

4. SOFTWARE IMPLEMENTATION

4.1 Introduction

In this chapter, the software implementation of the design will be discussed in detail. nRF52 has been selected as the MCU to get the desired output. nRF52 development kit is taken as the base station for this research work. This has both NFC and BLE.

The programming of nRF52 can be done on three major platforms, mbed os, Eclipse and Keil MicroVision. Though the code can be executed on mbed os, it is still in development for nRF52. Due to the extensive available library components that enable the user for proper design with flexibility, Keil μ vision it was selected as run time environment [63]. Before discussing the code, it is important to understand about a few concepts exclusive to nRF52. Most of the coding is done using C language.

According to Nordicsemis infocenter [64], A SoftDevice is a wireless protocol stack library for building System on Chip (SoC) solutions. These precompiled binary files work based on the wireless protocol specifications, so that what it needs is creating the application. This further helps in providing run-time memory protection, a way to provide access right to the system to modify or control the flow of it. This helps the run time environment, Keil, to use the memory allocated to the specific function without accessing the memory not allocated to it. SoftDevices also provides safety to the smallest programming instructions, called threads. In short, the SoftDevices act as the platform that helps the system to understand the usage of a program and its applications. The application of a SoftDevice is declared by the user via the header files. This declaration helps the computer to understand the specific application of nRF52 that is used in this study. Figure 4.1 shows the nRF52 development kit that was used.

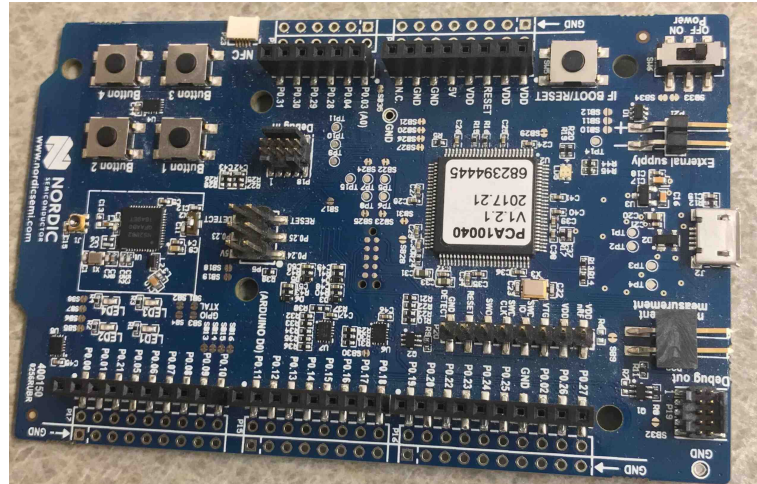


Fig. 4.1. nRF52 Development Kit

There are a wide range of such SoftDevices available in Nordisemi directories. nRF52 pca10040 development kit is being used as the base station. So, S132 SoftDevice, a BLE specific stack solution is used. This SoftDevice has a complete BLE5.0 compatible stack, Master Boot Record for over-the-air device firmware update, support for external power and low noise amplifiers as its key features. The use of Master boot record is that the SoftDevice, application and bootloaders can be updated separately with ease.

For this, a software development kit, also called as SDK, was downloaded onto the computer. nRF SDK v11.0.0 was used for this application. This SDK supports S132 v2.0.0 SoftDevice. To start the device, "J-Link OB-SAM3U128-V2-NordicSemi 170724.bin" is dragged and dropped after bootloading the device. The device is then shut down and turned on. Now, the following dialog box shown in Figure 4.2 pops up, indicating a successful bootloading. Now, a command line tools executable needs to be installed from nordicsemiconductors page. This command line tool is called as nrfjprog, a tool used to program via Command Prompt [65]. To verify if it is working properly, path at user defined variables needs to be set at as shown in the Figure 4.3 and Figure 4.4 as per [66].

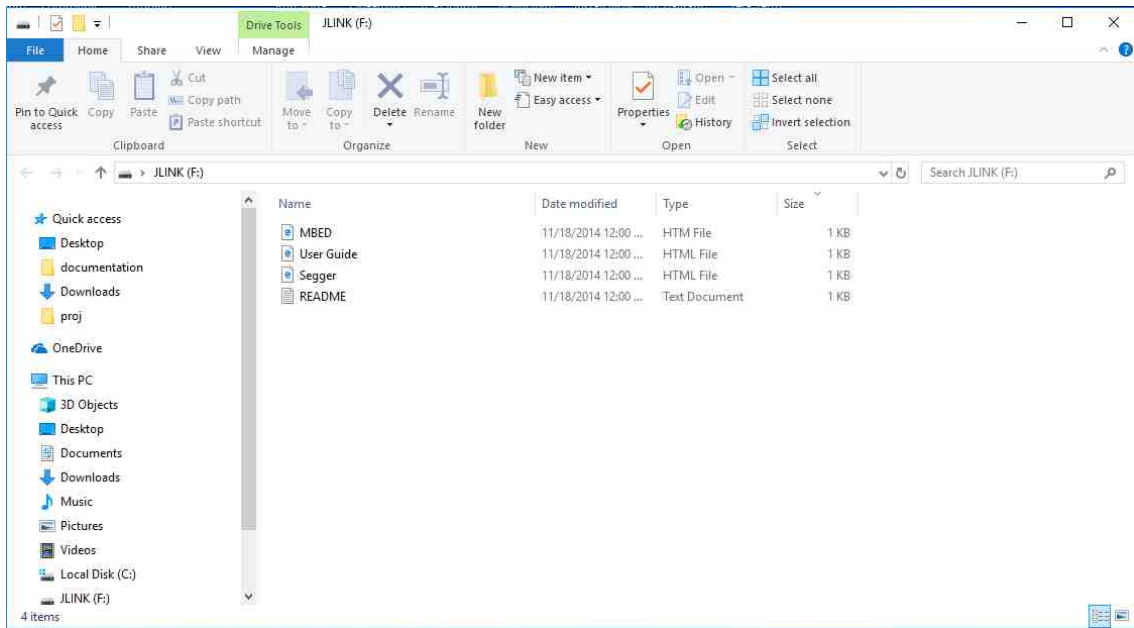


Fig. 4.2. nRF 52 in JLINK mode

```

Microsoft Windows [Version 10.0.16299.248]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Vamsi Krishna>nrfjprog --family nrf52 --eraseall
Erasing user available code and UICR flash areas.
Applying system reset.

C:\Users\Vamsi Krishna>nrfjprog --family nrf52 --program Documents\finalblework\components\softdevice\s132\hex\s132_nrf5
2_2.0.0_softdevice.hex
Parsing hex file.
Reading flash area to program to guarantee it is erased.
Checking that the area to write is not protected.
Programming device.

C:\Users\Vamsi Krishna>

```

Fig. 4.3. Loading nRF52 with SoftDevice via nrfjprog

It is located under environment variables, available under control Panel->System Properties->Advanced->environment variables. Using the two codes as specified in

Figure 4.3, the SoftDevice hex file is loaded into the development kit.

–family informs about the board family. This is defined by the user as nrf52.

–program command requires the location of the hex file on the system. This has to be defined correctly, else an error pops up.

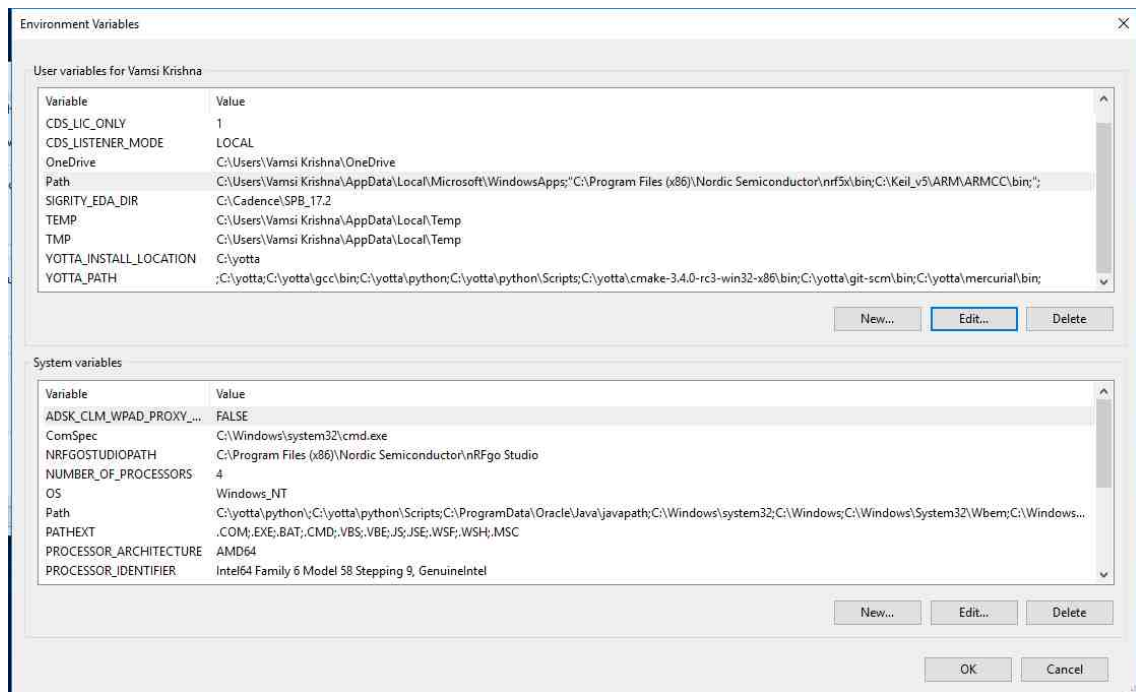


Fig. 4.4. Path at user defined variables

SDK already has all the applications related to the board used. When a keil project is opened for a specific application, A pop up asks for installation of all the different software required. This provides the Keil with all the required header files located at a single place. Figure 4.5 shows the working project when opened.

It has many folders in it. Importance of each such folder is discussed below.

1. Application: This folder has the main code in it. Any additional required codes that might be useful can be added here.
2. nRF_Drivers: This folder has the codes related to various drivers that would be declared in the project.

3. `nRF_Libraries`: This has all the common libraries that used to call up drivers.
4. `nRF_BLE`: This consists of the codes that assist in working for a specific application, like Bluetooth Low Energy.
5. `Board Support`: This has the codes that inform the system about pin allocation and declaration.
6. `nRF_SoftDevice`: This has the code that handles SoftDevice. This is a mandatory folder for any project.
7. `Seggerfiles`: This consists of SeggerRTT, Segger real time transfer. This can be used to check the output via Real Time Data Transfer.
8. `CMSIS`: Cortex Micro controller Software Interface Standard is a vendor independent hardware abstraction layer for ARM cortex processors. Declaring it helps the system to realize that the board has a cortex micro controller and needs to be communicated in a way it understands.
9. `Device`: This specifies about the type of the device used. For the selected board, it should be `nrf52`.

4.2 Analog connections to motion sensor and FSRs

nRF52832 has 8 analog pins as show in the Figure 4.6 [67]. For normal Micro-controllers, I2C communication protocol is most commonly used for data transfer between one or more masters and multiple slaves. It is used to communicate data between short distances within the same device. This requires only two signals to send or receive information. This is comparatively better than Serial Peripheral Interface, also known as SPI, which uses more number of pins leading to wastage of space and complex circuitry.

SPI and I2C exist in nRF52 board. Though SPI remains the same, I2C is generally called as TWI. TWI is the abbreviation for Two Wire Interface. This is to avoid the

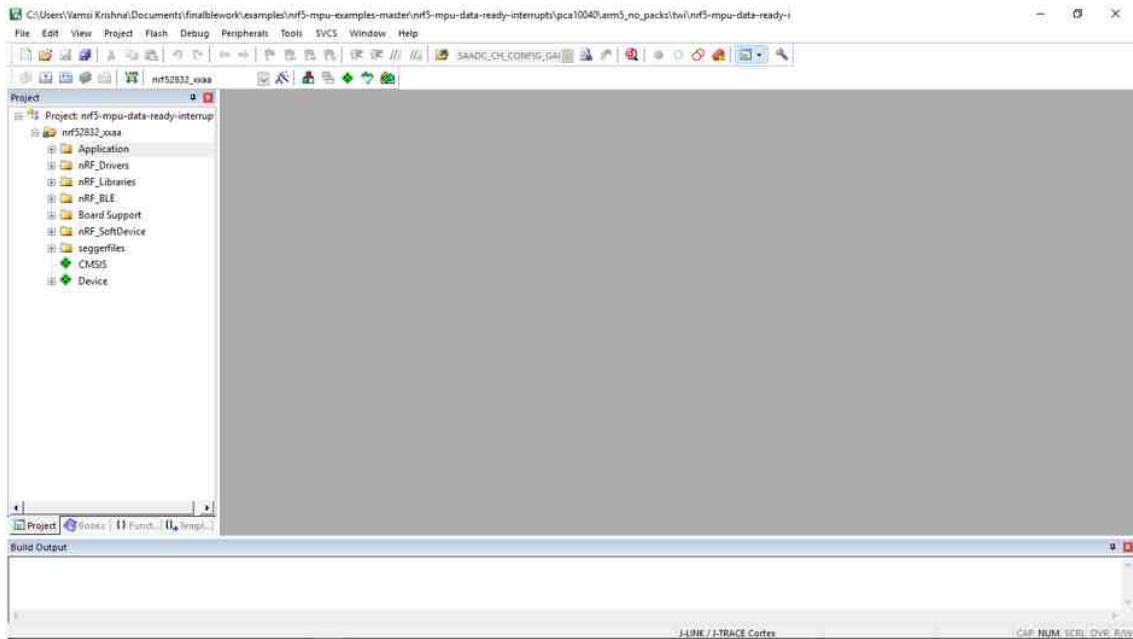


Fig. 4.5. Keil working interface

issues of copyright as the I2C is affiliated with NXP, Texas Instruments and Intel. TWI is defined for Atmel processors. The working is almost similar between these two protocols, with the basic function being the communication of data between master/masters and slaves. The basic working of a TWI setup is explained in figure 4.7.

For capturing the information of the motion sensor, MPU9250, TWI is used. As specified in the figure above, pins 25 and 26 of the micro controller, namely, SWDCLK and SWDIO are connected to SDA and SCL of the motion sensor. Also, to understand if there is a flow of the data from the motion sensor or not, INT pin from Motion sensor is connected to MCU. INT pin is the interrupt pin, that sends active high signal if the motion sensor is active. Else, it sends active low signal that the data being sent is not from the motion sensor.

In nRF52 development board, the SDA and SCL for the motion sensor is connected to P.04 and P.03. Interrupt is connected P.05. It should be noted that the maximum voltage of this sensor is about 3.3V. This is well within the supply voltage of nRF52

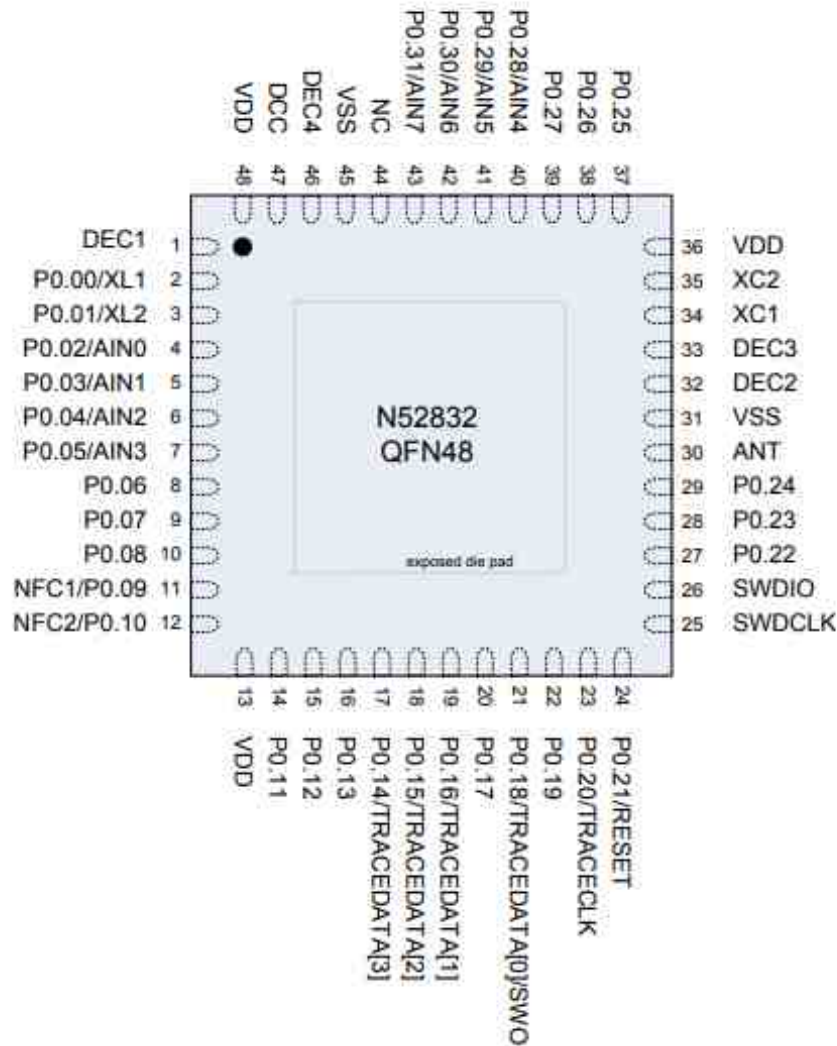


Fig. 4.6. nRF52832 pin assignments [67]

development kit. So, VDD of the MPU is connected to VDD of the development board. Also, MPUs ground is connected to the ground of the board. This sets up the required hardware connections for MPU.

As explained in the previous chapters, Flexiforces A201 has been selected as an FSR to be used. This is an analog sensor that gives force as an output by varying resistance. SAADC is the abbreviation for Successive Approximation Analog to Digital Converter [68]. The output from this analog sensor is sent to the SAADC and

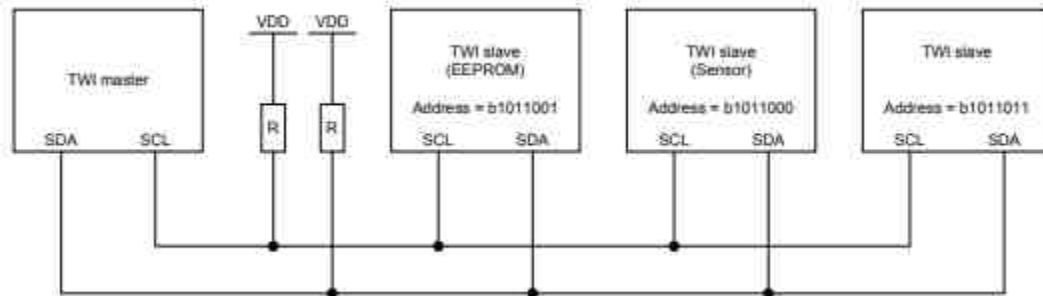


Fig. 4.7. TWI Master Slave block diagram [67]

the output is obtained. It is worth noting that the SAADC provides a wide range of options related to resolution, gain and oversampling conditions. Resolution can be 8-bit or 10-bit or 12 bit or 14 bit. Gain variation is an important key feature of SAADC. The gain can be varied as $1/6$, $1/5$, $1/4$, $1/3$, $1/2$, 1, 2 and 4. Oversampling is another feature provided by SAADC. The user can bypass this feature or use it from over 2x to over 256x. Usage of oversampling increases the overall noise performance.

The input is sent via analog channels, generally called as analog inputs. There are two types of modes in which information can be passed via these channels from the sensors:

1. Continuous mode: In this mode, the values are sampled individually, one after the other. Hence this is called as on-shot mode or continuous mode. The data received is easier to manipulate in this mode.
2. Scan mode: In this mode, multiple channels send the output at once. This is generally difficult when multiple analog channels are used. Due to the difficulty in controlling the data received and the necessity to use multiple channels, this method is not preferred.

Due to the ease in handling of the analog data, continuous mode is selected. Also, the analog inputs can be differentiated as eight single ended inputs or four differential inputs or combination of both. Single ended inputs are those whose voltage

is measured between VDD and a common ground. For example, if there are 20 single ended inputs, there will be 20 VDD or HIGHS and only one ground or LOW. Differential inputs are used when there is a requirement to compare voltage between HIGH and LOW. For an analog sensor used in this particular application, Single ended inputs would be sufficient to obtain the required accuracy. Also, Single ended inputs, also called as SE, are simple and easier to install.

Conversion of analog signals to digital signals follow:

$$RESULT = [V(P) - V(N)] * GAIN / REFERENCE * 2^{(RESOLUTION-m)} \quad (4.1)$$

Where,

V(P)= Voltage at input P

V(N)= Voltage at input N

Gain= Gain set for the analog channels

REFERENCE= Reference voltage

m=0 if CONFIG.MODE=SE

m=1 if CONFIG.MODE=Diff

For the discussed setup, as given in (4.1), V(P) is the analog output obtained from the sensor, V(N) is the ground and hence it is taken as Zero. GAIN is dependent on the user and REFERENCE is selected as 5V. Finally, m =0 as we have assumed that the readings are related to single ended mode. Since three of the seven Analog pins are taken by TWI, P.02 and P.31 are selected as the analog pins for the FSRs.

4.2.1 Data Collection

To get the data from the FSRs, SAADC needs to be informed with the working environment, Keil. There are a few key points that are necessary to achieve the working code. After downloading the SDK, activate the packet installer and select Nordic Semiconductor. This is also followed by activating the nRF52832_xxAA and downloading the required modules. This will also install the preexisting examples which includes SAADC, TWI and BLE template.

Uploading the SAADC example gives the basic head start towards the working of it. This has several predefined libraries. As explained before, the nRF_Libraries has the basic required libraries for any program to run under Keil interface.

Header files and c files are available in it. Header files give the declaration required whereas c files give the working function available for the user to call. App_error and app_fifo files help the working environment in handling the information based on the application. App_util_platform verifies the compatibility of the SoftDevice with the program. Nrf_assert deals with the ability to reset the device when the user requests it. Retarget helps in communicating or modifying the attributes of the board.

nRF_Drivers gives the access to Keil related to many drivers readily available in SDK. The drivers have to be defined in the nRF_Drivers folder before calling its function. All the drivers were based on [69]. Apart from the previously discussed drivers, the important drivers are:

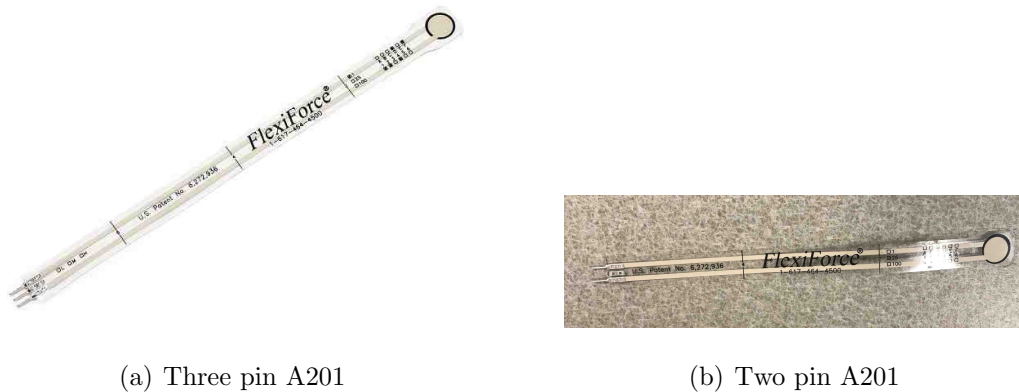
1. Nrf_drv_common: This driver has all the common definitions that are used in the other drivers. This plays a crucial role in providing a proper pathway to various drivers. This file is a mandatory one and contains information about interrupt request handler, SPI, TWI and SoftDevice verification.
2. Nrf_delay: This helps in providing a delay in data reception. This works well when there is a lot of data being sent in few seconds. Using functions such as nrf_delay_ms() will help in delaying the output received by a given amount of time in seconds.
3. PPI driver: Programmable Peripheral Interconnect, also called as PPI, helps in autonomous communication between peripherals via tasks and events. This is done while being completely independent of the CPU. It has a mechanism through which one task can be triggered due to an event in another peripheral which are not mutually exclusive. This synchronization is done using an internal clock which can be delayed till 52MHz. This driver plays a major role in sending and synchronizing information from FSRs.

4. **TIMER:** This can usually operate in two modes, as a timer or as a counter. Its base frequency is generally 16MHz. This driver is used to synchronize when PPI is called. It also can be used with any other driver to send information or data at a selected frequency. TIMER implements multiple capture/compare registers. This helps in allowing the user to time and control various events being called by various drivers. There are 5 TIMER components available, namely, TIMER0, TIMER1, TIMER2, TIMER3, and TIMER4. TIMER0-2 has 4 CC registers (CC [0..3]) and TIMER 3-4 has 6 CC registers[0.5]. For the selected application, TIMER0 has been selected to be used. There are two major tasks performed by these drivers:
 - (a) **Capture:** Capture task helps in saving the values in a counter at CC[n] register.
 - (b) **Compare:** This is generated whenever the counter is incremented and becomes equal to the value specified in one of the capture compare registers. Then, both the values are compared on the basis of bits.
5. **GPIOTE:** This is a General Purpose input output Tasks and Events that provides the access to the existing GPIO pins. We can set, clear or toggle these pins at rising edge or falling edge or any given change of input conditions.
6. **PSTORAGE:** Persistent data storage is a tool that is used for BLE. This helps in storing the exchanged information such as bond identification, GATT Server configuration, and/or GATT Server attribute handles.
7. **UART:** Universal Asynchronous receiver/transmitter, also called as UART. It plays a major role in transfer of data. There are four different UART pins which help in accessing the resources. RXD and CTS are the input UART pins whereas RTS and TXD are output UART pins. Transmission occurs when the TXD register is written with the information meant to be transmitted. When CTS is deactivated, transmission stops and resumes when it is activated.

4.3 Working of SAADC/FSRs

4.3.1 Basic Setup

FSR A201 has three pins to facilitate analog data reading. However, the middle pin does not have any functionality other than providing the balance when it is placed on a breadboard vertically. Hence, the middle part of it is removed using a wire cutter. The comparison can be seen in Figure 4.8.



(a) Three pin A201

(b) Two pin A201

Fig. 4.8. Comparison between three pin and two pin FSRs.

After setting up the project with drivers, libraries, board support, CMSIS and device folders, the project is ready with the basic setup. The basic SAADC code has a few preexisting functions that must be used to get the output. These include, a function to handle the UART events (`uart_events_handler`), a function to initialize UART (`uart_init`), and a function to handle the timer (`timer_handler`).

SAADC event sampling is initiated via the function, `saadc_sampling_event_init`. This function calls PPI in it and verifies it using app error function. Calling PPI helps in initializing different peripherals at the same time. This is done to accept the BLE module that we would be adding later. TIMER is another driver that is called for synchronization purposes. This is done to compare the results at 400ms. TIMER0 is allocated for SAADC. After this is done, PPI channel is then setup to inform that the timer compares event triggers sample task in SAADC. A final assignment and verifica-

tion of the PPI channels and TIMER has been done using app error. Further, SAADC channels are allocated to PPI channels via `saadc_sampling_event_enable` function.

`Saadc_init` is used for primary initialization of the SAADC channels. Two analog channels which are single ended are initialized for analog inputs, P.02 and P.31. These pins are declared as `NRF_SAADC_INPUT_AIN0` and `NRF_SAADC_INPUT_AIN7`. The following two lines help in declaring the above two pins:

```
nrf_saadc_channel_config_t channel_0_config =
NRF_DRV_SAADC_DEFAULT_CHANNEL_CONFIG_SE
(NRF_SAADC_INPUT_AIN0);
nrf_saadc_channel_config_t channel_1_config =
NRF_DRV_SAADC_DEFAULT_CHANNEL_CONFIG_SE
(NRF_SAADC_INPUT_AIN7);
```

After these declarations are done, `app_error` is used to verify proper initiations. Now, the calibration is done using the following commands,

```
NRF_SAADC->TASKS_CALIBRATEOFFSET = 1;
while (NRF_SAADC->EVENTS_CALIBRATEDONE == 0);
NRF_SAADC->EVENTS_CALIBRATEDONE = 0;
while (NRF_SAADC->STATUS ==
(SAADC_STATUS_STATUS_Busy <<SAADC_STATUS_STATUS_Pos));
```

The values obtained from the analog channels are sent into a buffer. A constant was added to the code in order to set the value of the buffer and enhance the system accuracy. `SAADC_SAMPLES_IN_BUFFER` is used to define the number of values that a buffer can store while processing data from analog input or channel. For two channels, this is selected as 2. The analog value from channel 0 goes to buffer pool [0] and the output from channel 1 goes to buffer pool [1]. The modification to the output received from analog channel is done in SAADC callback function. Here, first analog pin value is stored in buffer [0] and second analog value is stored in buffer [1]. `P_event` is used to note the event that has taken place. Using `p_event`, data from buffer is extracted individually and it is done as follows:

```
p_event->data.done.p_buffer[0]
p_event->data.done.p_buffer[1]
```

Sometimes, these analog channels produce outputs less than 0. So, in order to remove the possibility of error, values above or equal to 0 were only considered. Now, these analog values are to be converted to be converted into pressure readings using code available in [70]. Here, VCC/VDD is taken as 5V. This is because the voltage received from the USB to the board is about 5V. R_DIV is the resistor value used to provide linearity to the circuit. It is taken as 10 K Ω resistor.

There is another way of declaring analog channels. The alternative declaration can be done without using buffers. The analog values can be directly sent to UART. Using saadc values, declaration can be done as follows:

```
static nrf_saadc_value_t adc_value_0;
static nrf_saadc_value_t adc_value_1;
```

It can be said that `p_event->data.done.p_buffer[0]` is equal to `adc_value_0`. In other words, `adc_value_0` is the direct declaration of `p_event->data.done.p_buffer[0]`. The force is obtained from the sensors with follow the equation:

$$fsrV1 = p_event \rightarrow data.done.p_buffer[0] * VCC/1023.0 \quad (4.2)$$

where,

`fsrV1`= Voltage reading of FSR1

`p_event->data.done.p_buffer[0]`= Values from analog channel at pin P0.02

VCC= 5V

Here, the resolution is selected as 10 bits by default, and hence the value of 1023 which is 2^{10} .

Similarly,

$$fsrV2 = p_event \rightarrow data.done.p_buffer[1] * VCC/1023.0 \quad (4.3)$$

where,

`fsrV1`= Voltage reading of FSR1

p_event->data.done.p_buffer[0]= Values from analog channel at pin P0.31
VCC= 5V

To convert analog values to pressure readings, we go through the A201 FSRs linearity with respect to increase in force. This linearity can be differentiated as two parts based on the resistance acting. The following is the code that helps in accomplishing the required task:

```
fsrR = R_DIV * ((VCC / fsrV) - 1.0);
float fsrG = 1.0 / fsrR;
if (fsrR <= 600)
force = (fsrG- 0.00075) / 0.00000032639;
else
force = fsrG/ 0.000000642857;
```

where,

fsrR= Force resistance

fsrG= Force conductance

Force = force generated by the sensor with respect to acceleration = $1m/s^2$

We know that the Pressure is equal to force per unit area. Pressure is calculated from this force readings as follows:

$$Pressure = Force * 100 * 100/185. \quad (4.4)$$

This gives pressure in Pa. We need the pressure in KPa. So, we divide it with 1000 and multiply with $10m/s^2$ which the acceleration is due to gravity. This gives,

$$Pressure = Force * 10 * 10/185. \quad (4.5)$$

where,

area of the shoes sole is considered as $185cm^2$ for a shoe size of 7 US.

The output now obtained are pressure readings. These pressure values can be used to detect gait [71–73]. In the main function, saadc_sampling_event_init(), saadc_init() and saadc_sampling_event_enable() were called. SAADC can be accessed in two different ways. The only difference between these two ways is the sampling rate. While

direct declaration using `adc_value_0` gives synchronized data rate with respect to TWI drivers, declaration using buffers need external synchronization.

4.3.2 Data Collection

The output data is collected from UART via Tera Term and sent to MATLAB for data processing. Like discussed before, SAADC has two major properties, oversampling and gain. Varying the oversampling rate will take all the values in the buffer, readings from sensor1 and sensor2, and averages the cumulative readings. This will not serve the purpose as the readings of the sensors are to be taken separately. Hence, oversampling is not considered here. Gain is an important parameter used to vary the output based on our needs. Gain of an analog channel can be changed at `nrf_drv_saadc.h` header file. In the definition at "nrf_drv_saadc_default_channel_config SE" gain can be changed based on our requirement.

The smoothening technique uses the moving average filter to get a proper output. This is done via MATLAB to understand the peak possible values for a normal gait.

It is important to note down the criteria followed during the data collection from a test subject:

1. The data is taken from a shoe placed on a flat surface.
2. The test subject was asked to walk at a slow but comfortable pace.
3. The test subject was a 60 kg and has a shoe size of 7 US. This is done to get the minimum possible peak values such that it can be used for other shoe sizes as well.

To test the working of an FSR at pressure readings about 150KPa, a COMSOL output has been verified. The pressure readings obtained were used to provide the base cut off for the FSRs during analyzing of the gait.

4.4 Working of MPU

4.4.1 Basic Setup

To the above code, let us add the working code of an MPU as well. There are a few preexisting files available to be used as the basis for MPU functionality [74]. These files are used to make the MPU work with respect to TWI driver. There are a number of files that describe its registers, application and, working and declaration with respect to TWI driver. It has definitions for configuration of MPU (`mpu_config`), configuration of its interrupt pin (`mpu_int_config_pin`), enabling of interrupt, and initiation of MPU. Apart from the above functions, reading the accelerometer and gyroscope can be done via `app_mpu` code. These functions access the MPU's registers handling the accelerometer and gyroscope values. It also has magnetometer definitions but are not used in this application. `Nrf_drv_mpu_twi` code is used to define pins of MPU to nRF52 development kit.

Pins P0.03 and P0.04 are used as analog pins receiving SCL and SDA. Accelerometer and gyroscope can be selected from a wide range. The range can vary from 2G, 4G, 8G, and 16G. It is generally known fact that lower the value of g, higher the sensitivity. So, 2G and 4G are generally used to detect the very small variations. So, 16G is selected as the accelerometer range for this design. This is because the accelerometer need not be very sensitive for the application as it leads to complex readings at simple movements. Gyroscope can vary from 250DPS, 500DPS, 1000DPS and 2000DPS. 2000DPS is selected for this design. This is because the higher the DPS values, the less sensitive the device becomes with smaller variations.

Figure 4.9 shows the MPU9250-motion sensor used in the design. TWI driver is added to the existing code. Now in the main code, `mpu_setup` function is called. This function is used to initialize the MPU driver by verifying its initiation via app error. The configuration of the MPU is also done here. In the configuration table, several parameters can be changed depending upon users comfort. Accelerometer and gyroscopes values in x, y and z directions can be stated here with the users preferred

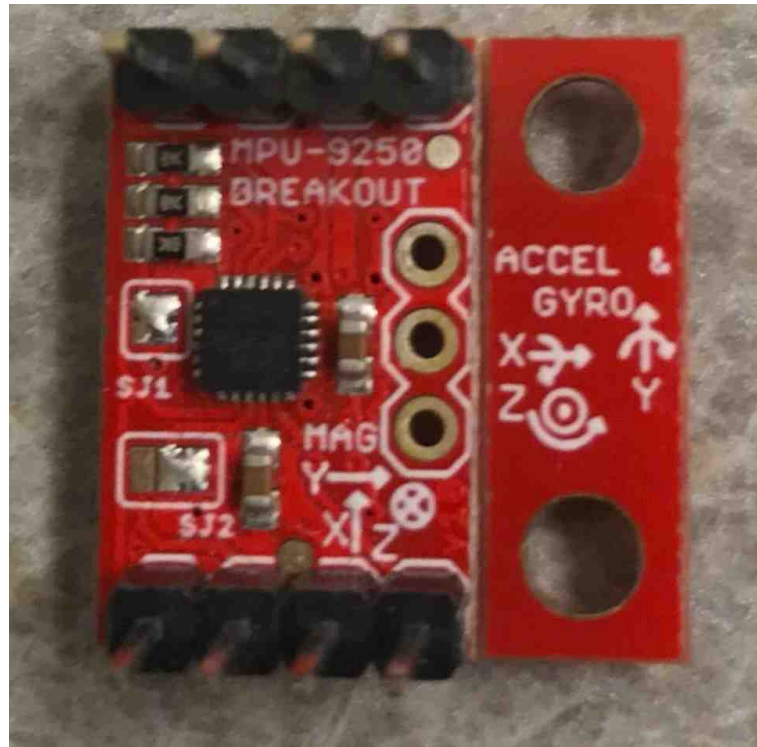


Fig. 4.9. MPU9250-motion sensor used

selection. Here, values 0 are taken for all the 6 degrees of freedom available to get accurate values. Sample rate can also be modified here using `p_mpu_config.smplrt_div` where `p_mpu_config` is the structure based on `mpu_config` and `smplrt_div` is the sample rate divider. Here, sample rate is calculated as:

$$\text{SampleRate} = \text{GyroscopeOutputRate} / (1 + \text{SMPLRT_DIV}) \quad (4.6)$$

where,

Sample rate is the number of samples sent per second.

Gyroscope Output Rate is the DPS selected for the application.

SMPLRT_DIV is the sample rate divider which can be modified by the user to get the desired sample rate.

4.4.2 Interrupt Handling

After the configuration of accelerometer and gyroscope is done, interrupt pin is configured and enabled. This is triggered whenever new information is sent from the MPU. This interrupt pin is set as true via interrupt pin handler function. The enabling of the interrupt pin is checked using GPIOTE function. GPIOTE function has to verify if this interrupt pin is high or not. However, there is a high possibility of other drivers to activate GPIOTE pins before the function triggers it. Multiple initiations causes the device to hardfault. In order to avoid such issues, the GPIOTE pins are initiated only when the `nrf_drv_gpiote_is_init() == false`. After verifying it using the GPIOTE pins, the GPIOTE driver sends information to the system saying that the interrupt pin is active or not. Finally, in the main code, `mpu_setup` is called.

4.4.3 Filtering Techniques

The direct usage of the values from the accelerometer and gyroscope poses errors. Accelerometer measures the readings obtained from various forces acting on it, thus leading to introduction of errors which may be due to even vibrations. This is because the readings obtained would be more than just the gravity vector. Usage of a low pass filter on the accelerometer reduces these errors to an extent. Gyroscope, on the other hand, drifts over time due to integration. This can be controlled by using High pass filters.

Complementary filter:

This simple filter works based on the equation 4.7:

$$y_k = \alpha * x_k + (1 - \alpha)y_{k-1} \quad (4.7)$$

where,

α is the filter gain which can be one of the three values:

When $\alpha=1$, then $y_k=x_k$ and raw data at 100% with no filtering is obtained

When $\alpha=0$, then $y_k=y_{k-1}$ and constant signal with 0% of the input data is obtained. With $\alpha=0.5$, then value of the output signal is half its previous value and half the new input signal value.

A complementary filter can manage both the high pass and low pass filter [75]. It should be noted that the accelerometer can show no drifting in long term and gyroscope can show no drifting in a short term. So, to make it work, a low pass filter is added to raw values of accelerometer while gyroscope raw data is integrated with respect to time. Low pass filter filters the high frequency data which is a direct result of vibrations in accelerometer. First order design was selected for this purpose. This is relatively a simple filtering technique. However, some drawbacks have been observed:

1. Only one measurement can be introduced to the filter at one time and output will be of the same nature as the input.
2. No information about accuracy can be obtained.
3. A lag can be observed in the filtered signal when compared with the input signal.

Kalman filter is a very complicated filter but is much more powerful when compared with respect complementary filter [76]. This filter needs a predefined setup model that directs the filter in its working. For any Kalman filter need prior knowledge of the design to be implemented. The first step is the prediction step where the future state of the system is predicted based on the current state. Simultaneously, an error correction model is introduced which informs the filter about the possible magnitude of the error. This propagation step makes sure that there is no lag due to the ability to predict future values.

This is followed by the update step which verifies with the error correction model to see if the obtained output from the previous step is correct or not. It is done when the information about the accuracy is provide to the entire filter. Figure 4.10 shows

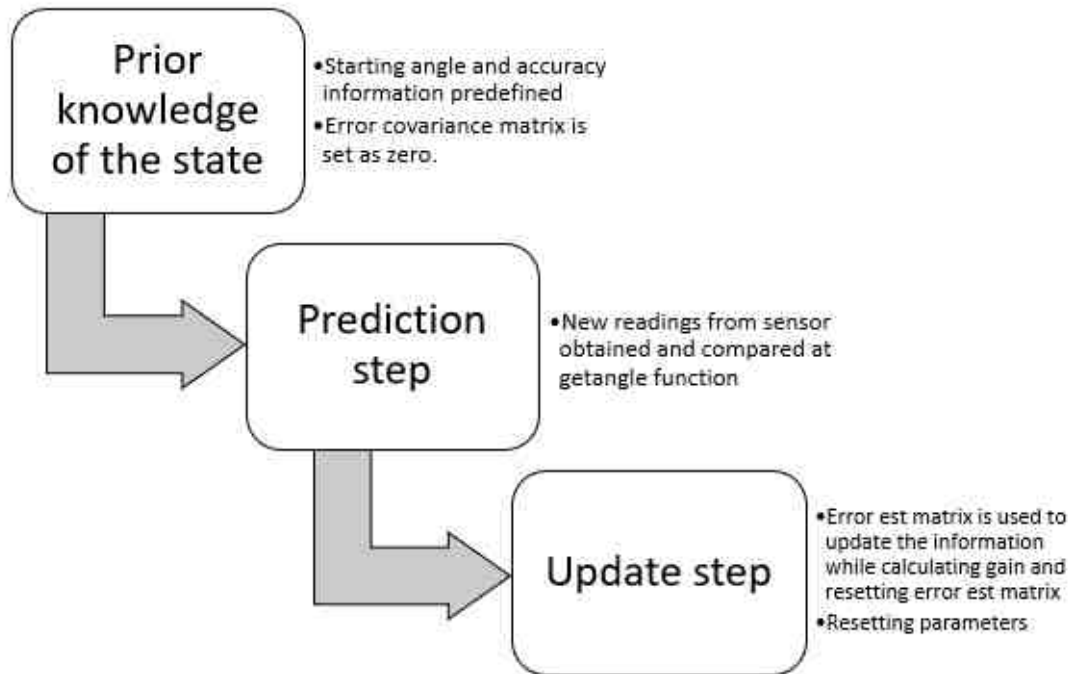


Fig. 4.10. Working of Kalman filtering technique

the Kalman filter model. This works like that of the previous filter but has many advantages:

1. Multiple measurements of different natures can be fed at any time.
2. Gain is chosen by the system itself.
3. Much smoother response with relatively no lag.

Both the filtering techniques were compared, and the output is available in the results section. The obtained output from these sensors are used as a cutoff to detect Hemiplegic gait, which can also be used for Diplegic Gait.

4.5 Working of BLE

4.5.1 Introduction to BLE in nRF52

This is a very low power protocol which can work up to one year on a coin cell battery when not advertising. The connection can also be maintained for a long time and its duration can be modified by the user. It consists of a very flexible topology which consists of several roles [77]:

1. **Broadcaster:** this is generally used as a transmitter only. It helps in broadcasting the information, hence the name broadcaster. The broadcaster sends a non-connectible data, meaning that broadcaster cannot be connected. The purpose of it is to tell the other devices of its presence.
2. **Observer:** This acts as a receiver only. The information is received by it are the advertising packets from broadcasters and peripherals.
3. **Peripheral:** This generally supports slave role. Peripheral generally advertises and informs the other devices of its presence.
4. **Central:** This generally supports the master role. This can only send the connection request for establishing a connection. Central and peripheral is generally called a master/slave pair where, central is the master while the peripheral is the slave. Central device can be connected to multiple peripherals. However, one device can function as peripheral and central as well. Figure 4.11 shows the communication via BLE in nRF52.

4.5.2 Advertising

Broadcasting of data is called as advertising. Advertising of BLE data is one way to inform devices about the presence of a BLE. Broadcaster or a peripheral advertises this information. The data sent via this is either connectible or non-connectible. A

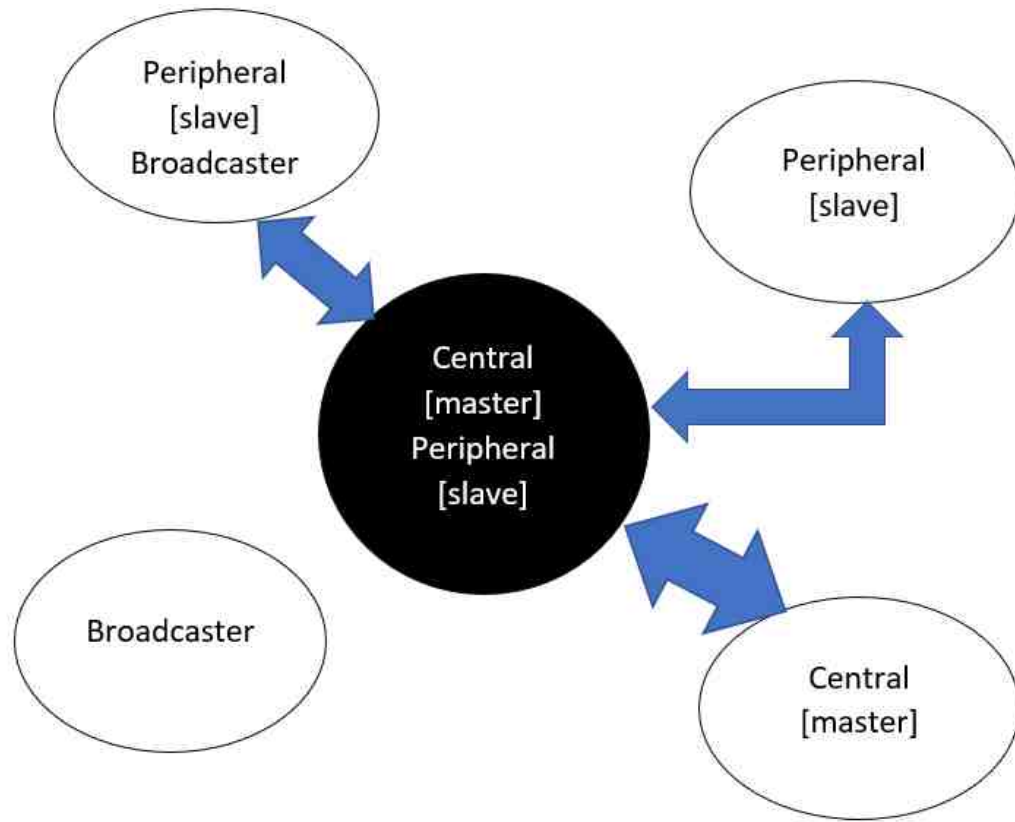


Fig. 4.11. Communication via BLE in nRF52

Scanner or a server/central picks up this signal. There are three advertising channels and it is transmitted to all these channels. Figure 4.12 shows the three advertising channels. It is spread out, and this is done to avoid interference.

When a peripheral advertises data, the scanner should be able to identify the device that is advertising. The addresses are of 48 bits and are of two types:

1. Random: This is a random number generated by the peripheral which follows a set of rules given by Bluetooth Special Interest Group (BT SIG). These addresses can be of two types, static and private. Static addresses typically never change until the change is requested. This change can happen only upon power cycling. A Private address can be resolvable or non-resolvable. If resolvable, the address can change for every given interval. This provides additional security during advertising.

- Public: This is an address which has the 48-bit universal LAN MAC addresses which can be obtained from IEEE. This will be a unique address that generally can identify the chip vendor.

Universal unique identifier, also called as UUID helps the scanner to detect the application that the device is being used for. The following figure shows the connection requests and connection establishment mechanisms.

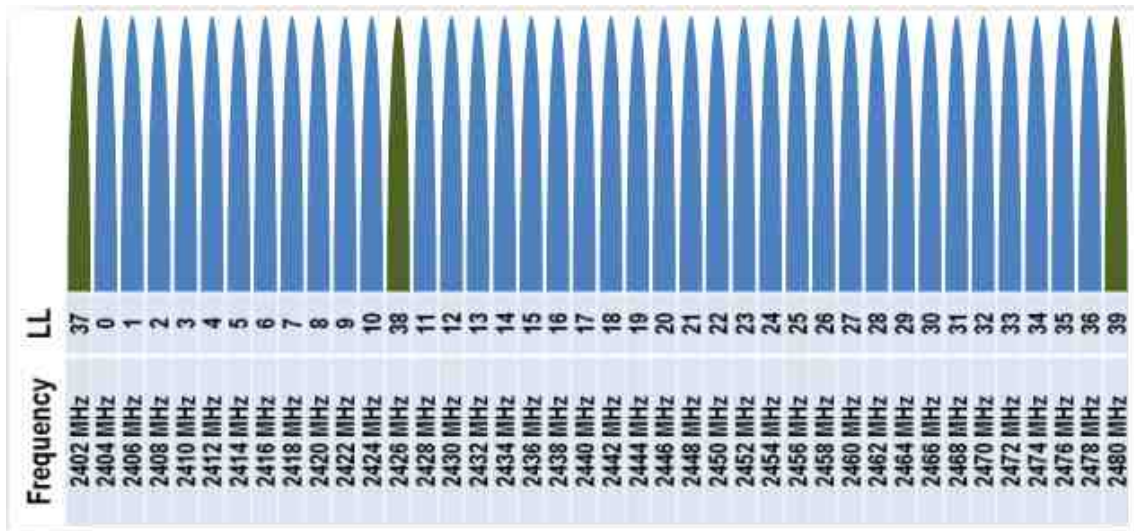


Fig. 4.12. nRF52 BLE advertising channels

When the advertising is done via channels 37, 38 and 39, a receiver period exists. This receiver period waits for the requests from the scanner, such as request for additional information. After a certain time interval, the master and slave communication starts with slave switching on to receive packets from master. If the slave receives it, it responds with another packet confirming an established connection.

4.5.3 Basic Setup

For BLE to work, we need to add BLE driver to the project. Name of the device, declaration of the board being peripheral, device name and the Nordic UART service UUID type has been defined. Nordic UART service, also called as nus is used to send

UART information to the BLE. Advertising intervals and timeout can also be defined here. Also, minimum and maximum connection interval is also defined. Generic Access Profile helps in declaring connection parameters. There is a function for it which is predefined. The function is also used in the main code. There are other functions that help in connections. These are the `on_conn_params_evt()`. Here, when the connection has failed, it informs the board that it is disconnected. If there is any error in the connection, it can be detected via `conn_params_error_handler`. `On_adv_evt()` is used to advertise about the device to nearby devices.

`On_ble_evt()` is used to provide information based on the SoftDevice on how to respond to the BLE event occurrence when device is connected or disconnected. This function also checks if the pairing is supported or if implementation of the BLE event is needed or not. `Ble_evt_dispatch()` is a function for dispatching a SoftDevice event to all modules with a SoftDevice event handler. This function is called from the SoftDevice event interrupt handler after a SoftDevice event has been received. There is also a function used to define usage of SoftDevice handler.

Function for handling `app_uart` events is that this function will receive a single character from the `app_uart` module and append it to a string. The string will be sent over BLE when the last character received was a 'new line' i.e. `'\n'` (hex `0x0D`) or if the string has reached a length of `NUS_MAX_DATA_LENGTH`.

The declaration of `err_code=ble_nus_string_send(ble_nus_t * p_nus, uint8_t * p_string, uint16_t length)` helps in sending the required information via BLE. There are two ways to use this.

1. For sending characters: After declaring `ble_nus_t`s variable, in this case, `m_nus`, the following command can be used to send the required string of data.

```
err_code=ble_nus_string_send(&m_nus, "monoplegic gait", 15);
```

where,

The last parameter, 15, is the number of characters present in the string, inclusive of spaces and any other punctuations.

2. For sending integer or float values: A new integer is created, `buffer`, which stores the information in it and keeps sending it out to BLE in FIFO order.

```
err_code=ble_nus_string_send(&m_nus, (uint8_t*)buffer, strlen(buffer));
```

where,

the last parameter is the size of the buffer being declared.

In the main function, the discussed parameters are usually called to initiate those functions. Also, it is important to inform the board to start advertising via using the following command.

```
err_code = ble_advertising_start(BLE_ADV_MODE_FAST);
```

Finally, all the codes and commands are available in a single project. There are a few important things to do before executing the code.

1. Open project->options for the target `nrf52832_xxaa`. In the C/C++ tool bar, all the paths of the drivers, libraries and any file that it called on the project, must be included.
2. Optimization level in the code must be of Level 0. Higher the optimization level, larger the code size. Using level 0 optimization not saves the originality of the source code but also helps in reducing the space of the code. This can be done at Open project->options for the target `nrf52832_xxaa`->C/C++ tool bar.
3. Pre-processor symbol definitions play a major role in getting the output desired. It helps in making the preprocessor understand what is supposed to be done when the program runs. The definitions include declaration of UART, declaring the presence of SoftDevice, disabling single wire interface(SWI), the type of MPU used and the type of the board used for running the program. This is discussed in detail in results section.
4. `Nrf_drv_config.h` is an important header file used to inform the board about the drivers the user wants to be active. Drivers such as `CLOCK`, `GPIOTE`,

TIMER0, TIMER3, RTC, UART0, TWI0 and SAADC are enabled. 1 is used for enabling a driver while 0 is used to disable it. The code used in it is present in the results section.

4.5.4 Flowchart

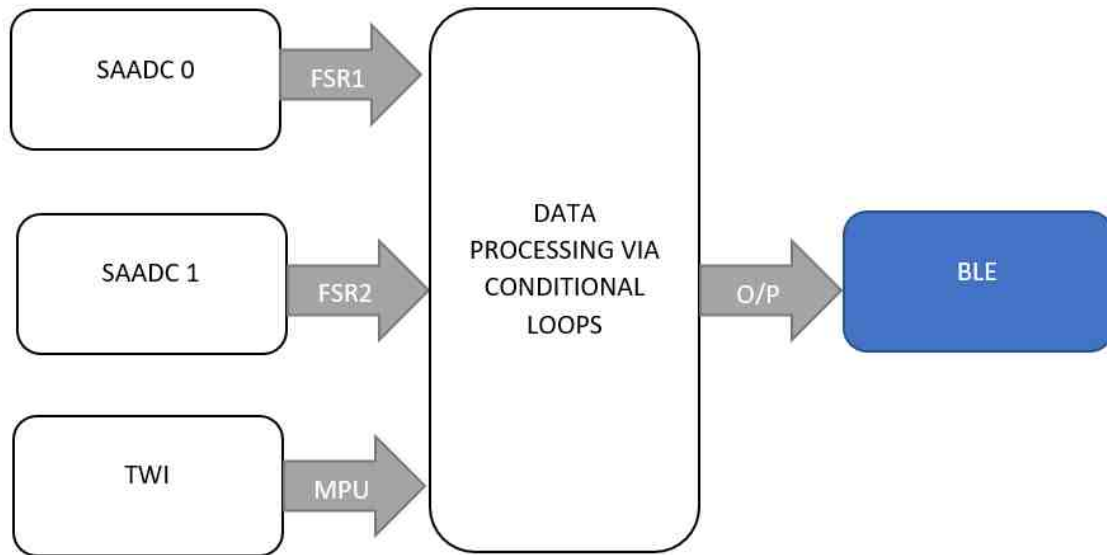


Fig. 4.13. Flowchart for BLE data processing

Figure 4.13 shows the flow chart about the data transmission in nRF52. Figure 4.14 explains the base station used to obtain the required FSR values and Motion sensor information. The values obtained from these sensor readings are the used to detect different types of gaits.

4.6 Power Consumption

The power consumption of nRF52 development board can be calculated using an online power consumption calculator for nRF52832 micro controller [78]. For calculation of power, the voltage was selected as 3.3V. This is because the voltage

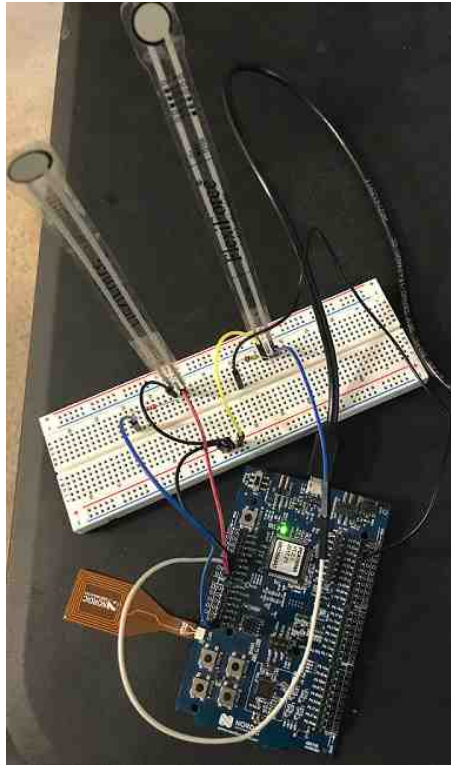


Fig. 4.14. Smart shoe prototype design using nRF52 development kit as the base station

to the micro controller is at least 3.3V. The BLE interval is selected to be done at every 75ms and TX payload is selected as 20 bytes, as the number of bytes sent by the development kit is 20. Finally, the default TX power of the board is 0 dB/m. The power consumption was calculated for two events:

4.6.1 Power Consumption during Advertising

For an advertising event, the estimated values are as follows:

BLE event total charge: $9.5 \mu\text{C}$

BLE event total length: 3.9 ms

Average BLE event interval: 105.0 ms

Total average current: $92 \mu\text{A}$

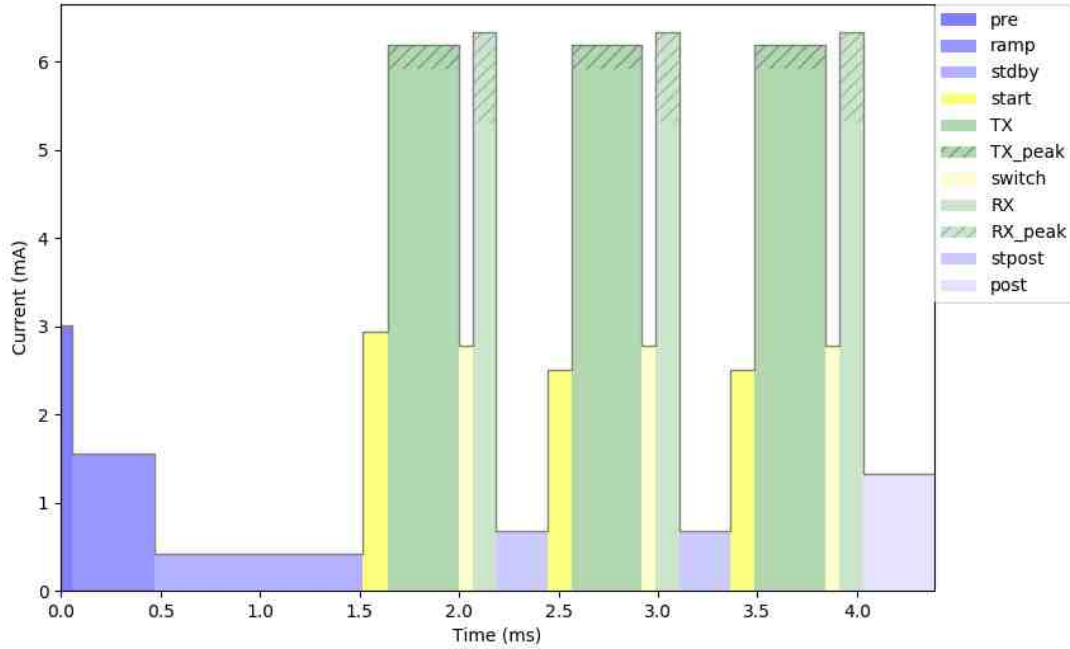


Fig. 4.15. Power consumption during BLE advertising event

Figure 4.15 shows the current ratings of the nrf52 development board during BLE advertising event. The maximum current consumption is during transmission stage of the BLE advertising. The current consumption is about 6mA. However, at sleep mode, it consumes a mere 1.9 μ A.

CR2450 is a coin cell battery that has 3V 620mAh of power rating [79]. The total time taken to discharge this coin cell battery at the peak current of 6mA can be calculated as follows [80]:

$$BatteryLife = BatteryCapacityinmAh / LoadCurrentinmA * 0.70 \quad (4.8)$$

Where,

0.70 is the allowances considered for external factors which can affect battery life.

With battery capacity being 620mAh and load current being 6mA, the total battery life is calculated to be 72hours.

4.6.2 Power Consumption during Peripheral Connection

For an peripheral connection event, the estimated values are as follows:

BLE event total charge: $5.5 \mu\text{C}$

BLE event total length: 2.4 ms

Average BLE event interval: 75.0 ms

Total average current: $75 \mu\text{A}$

Total average current: $92 \mu\text{A}$

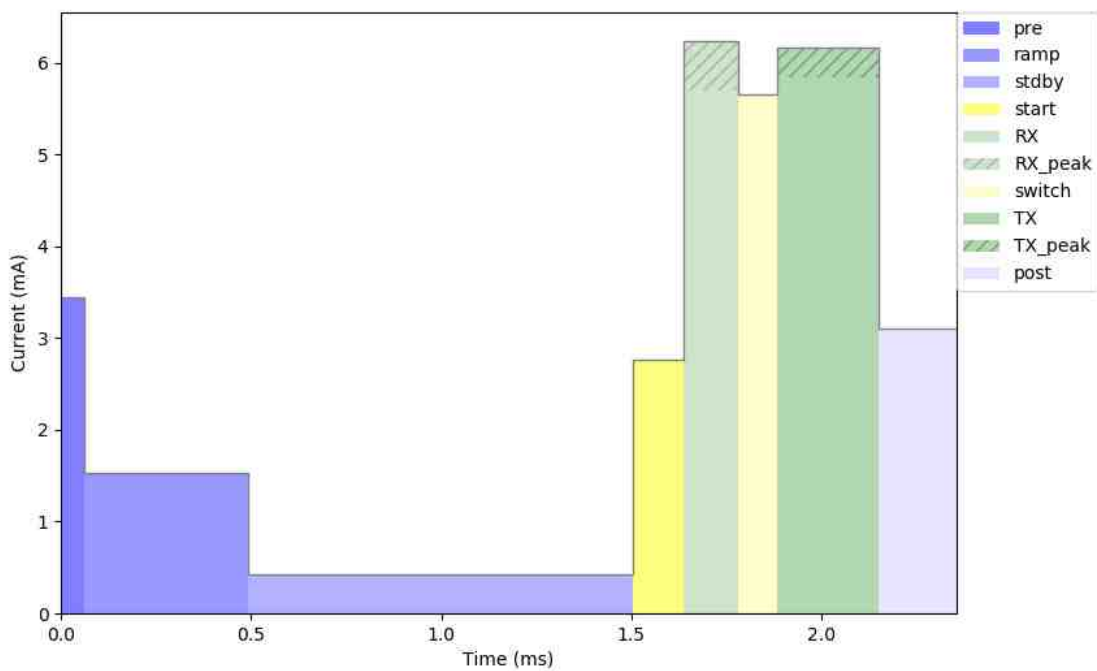


Fig. 4.16. Power consumption during BLE peripheral connection event

Figure 4.16 shows the current ratings of the nrf52 development board during BLE Peripheral connection event. Here, the peripheral is the mobile device used to connect the BLE to. The maximum current consumption is during transmission stage of the BLE connection establishment. The current consumption is about 6mA. However, at sleep mode, it consumes a mere $2\mu\text{A}$.

The CR2450 battery ratings are as follows:

$$BatteryLife = BatteryCapacityinmAh / LoadCurrentinmA * 0.70 \quad (4.9)$$

Where,

0.70 is the allowances considered for external factors which can affect battery life.

With battery capacity being 620mAh and load current being 6mA, the total battery life is calculated to be 72 hours which is similar to that of during advertising event.

However, it should be noted that at IDLE state, the current drawn is as low as $100\mu A$. This means that when the device is not advertising or connecting to a peripheral device, it lasts for almost an year.

Chapter 5 discusses about the Design of the Custom Printed circuit board (PCB) and parameters considered while designing.

5. HARDWARE PROTOTYPING AND DESIGN

5.1 Introduction

The main aim for the thesis is to design a custom PCB that can be placed inside a shoe sole without any external wires in order to monitor the body walk cycle. Also, it needs to be as compact as possible. Another condition to be satisfied is that the components should be only on the top layer for proper sensing. The dimensions of the board should be appropriate to the shoe size. A normal shoe is at least 5.5 inches in width [81]. The length of the board is not a concern for the design but it is preferable to have as small as possible in order to avoid mechanical stress from the body weight.

To design the PCB the software used was EAGLE by Autodesk. This is an Electronic Design Automation (EDA) application that can perform many tasks including PCB layout editing, Auto-routing and Computer Aided Manufacturing (CAM). This premium version allows usage of up to 999 schematic sheets, 16 layered PCB designs with friendly user software when compared with other EDAs such as Fritzing and Altium. Also, due to its free access, there is a wide variety of device and component libraries available online. Let us discuss a few of the important concepts related to PCB design:

Traces are the interconnection between the pins of various components which are connected via copper ribbons [82]. For any simple circuit board, these traces are usually placed on the top surface, meaning that these are single layered PCBs. However, as the connection complexity increases, these connections tend to go through many layers. To facilitate a single connection to move between layers, holes are drilled and plated through the board. These holes are generally called as vias, through-holes or feed throughs. Interconnect density is the number of connections required on the board. Greater the complexity of the board, higher the interconnect den-

sity. The size of the traces impacts the size of the PCB. Smaller the trace width size, smaller will be the PCB design. However, there are two tracing requirements, namely, Manufacturing and electrical. The smaller traces prove to be costlier during the manufacturing and are quite expensive to design. The nominal width size of these traces are about 10 mils or 0.010in. The lowest possible trace width in eagle is about 0.003in. However, these prove to be very expensive for the design. Second limitation is related to the current carrying capacity of a trace. Consider the formula,

$$R = \rho L/A \quad (5.1)$$

Where,

R= end to end track resistance in Ohms

ρ = Resistivity of the track material in Ohm meters

L= Length of the track in Meters

A= track cross sectional area in meter squared.

From equation 5.1, the smaller the cross-sectional area, the higher the end to end track resistance would be. This means that relatively low current passes through the traces with low widths. The general current carrying capacity of a normal circuit should be around 0.1 A or less. Cross talk is another issue that needs to be considered. When the trace width is small, the designer tends to bring the traces closer. This might lead to crosstalk between the traces due to the fact that different traces have different signal-current characteristics leading to one influencing the other. To overcome these issues, designers generally opt for multilayer PCB designs.

There are several types of packaging available for electronic components. These are widely divided as Dual Inline Package (DIP), Single Inline Package (SIP), Through Hole or Surface Mount Device (SMD). SMD are used for PCB design mounting as they are easier to install, cost effective, and more appropriate for the design. EAGLE, was used to design the PCB. The flowchart of the design is given in Figure 5.1. The steps of the design are:

Schematic Entry:

The primary step to be performed in EAGLE is Schematic entry. Here, the required libraries are loaded, and electronic components were added. The selection of components is dependent on the data sheet of the ICs or MCUs selected. The size of the passive electronic components and preferred locations were all detailed in the datasheet. The alternative passive components, if required, must have the power and voltage ratings as that of the replaced component. After the selection of the components, the basic connections are made using wire command. These provide the EDA with the information about the pins connected and those which are not. Schematic also provides information about all the components and parts used. This is obtained via NetList.

Netlist Extraction:

The Netlists are broadly divided into two types, Connection list and parts list. Connection list gives the information about a particular net on the schematic and the connections it has with the other pins. Parts list gives information about components and the signal names attached with the various pins of these components. Apart from this, there are many types of schematic information extraction lists available. These can be executed using User Language Program (ULP). These are programs given by EAGLE to get the required information such as Bill Of Material (BOM) which gives the list of all components available on the schematic page. Apart from this, there are many other programs which are executable after completion of trace routing in PCB simulation.

PCB Simulation:

The designer can enter the PCB simulation mode, shown as brd on the toolbar of EAGLE. Here, the air wires, the untraced wires of the PCB, and components are

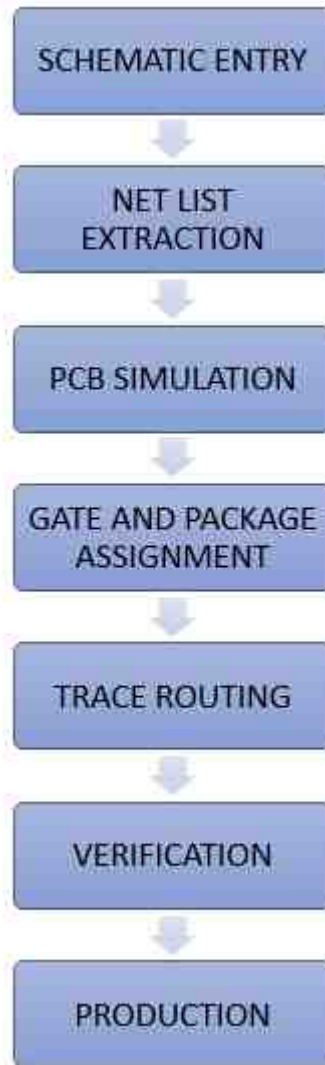


Fig. 5.1. Flowchart of PCB design flow

readily imported to the simulation mode to design the required board. The number of layers were decided here before the placement of the components.

Gate and Package Placement:

The placement of the components is relative to the designer. In this step, all the components are placed within the design area and are set at the locations the

designer wants. The gate or components needs to be correctly oriented and the packages having heat sinks underneath is preferred. Most of the packages have their own necessary components. These components are placed in accordance with the datasheet. Packages must be placed in a location where there is space for connections.

Trace Routing:

This is the most important part of the entire PCB design. This can be done in two ways: manual routing or auto routing. Manual routing is more laborious but is the most accurate method of routing. Auto routing on the other hand, is an EDA specific program which uses algorithms to track down the closest and most accurate paths. The best approach for routing is fusion of both Manual and Auto routing. Manual routing is usually done for traces moving in or out of the ICs. This is done to get better traces as auto routing tends to bring the traces near the ICs very close, thus putting the circuit at the risk of crosstalk. However, auto routing can be used once the completion percentage of the routing reaches around 50%. This is because the leftover connections are usually long traces which the computer can take care of. The trace width was selected to be between 0.006in to 0.008in. The resistance of these wires should be minimal. It was calculated that the maximum resistance the lines can create are about 0.01020Ω as shown in Figure 5.2.

Verification:

The final verification before sending it to the production is done using the Design Rule Checker (DRC). This is a computer driven verification check which notifies of any errors or warnings. Warnings are not impacting the system function, rather they directing the user to missing listing of components. These can be ignored but errors must be verified before the device is sent to the production. Gerber files are then generated after meeting the design specifications. These are the crucial Gerber files required for PCB production [84].

Inputs

Trace Width	<input type="text" value="0.006"/>	<input type="text" value="inch"/>
Trace Length	<input type="text" value="0.14"/>	<input type="text" value="inch"/>
Trace Thickness	<input type="text" value="1"/>	<input type="text" value="oz/ft^2"/>
Temperature	<input type="text" value="40"/>	<input type="text" value="°C"/>

Output

Resistance (R):	<input type="text" value="0.0120"/>	Ohms
-----------------	-------------------------------------	------

Fig. 5.2. Track Resistance calculator [83]

1. Component side: extension is .cmp. This is used to provide the information about the pads and vias available on the top layer.
2. Solder side: extension is .sol. This is used to provide information about the pads and vias available in the bottom layer.
3. Silk screen: Extension is .plc. This is used to print the dimensions, information about the place and names of components placed in the top layer.
4. Solder stop mask: Extensions are .stc and .sts. These are used to denote where not to have a solder mask on top and bottom layers respectively.
5. Solder paste: Extension is .crc. This is used to show the location of placement of the cream on the top layer of PCB for SMD packs. If SMDs are on both layers, tcream and bcream are selected from layers section.
6. Drill data: The extension is .drd. This is used to show the location of the drills and holes on the PCB.

Apart from this, Pick and place, and BOM ULPs results must also be sent to the PCB fabricator.

Production:

This is the final stage before receiving the PCB. The files discussed above are sent as a zip file to the company, here, Advanced assembly. Sometimes, the vendor sends the Gerber file program and the output of these files must be extracted and sent.

There are several PCB models designed for this project. The advantages and disadvantages of each product is discussed.

5.2 Model 1

5.2.1 Schematic

This was the first custom PCB. Here, there are several ICs added, nRF52832, ATSAM21G-M, MCP6004-I/SL and MPU6000. nRF 52 development kit has ATSAM3U2CAAU as the interface MCU. This MCU helps nRF 52 to run as SEGGER JLINK OB or mbed OB interface firmware. This is an ARM Cortex M3 processor. ATSAM21G-M is a ARM cortex M4F processor which is much faster than ARM cortex M3 [85]. Both the processors are almost identical but ARM cortex M4 has DSP instruction add-ons. This would help the BLE module in filtering or compress the analog signals it might receive. There is MCP6004 op amp in addition to MPU6000. MCP6004 is an op amp that has 4 inputs and ground pairs with 4 output terminals, hence the number 4 in the name. This is the opamp selected by the FlexiForce to be used to amplify the output of the sensor. MPU6000 is the motion sensor used for Custom PCB. This sensor is sufficient to get the required output and is comparable to MPU9250 which was used in the software implementation. The components around the ATSAM21G-M is selected based on the configuration specified in the datasheet as shown in the figure 5.3. Similarly, the components around nrf52832 were also

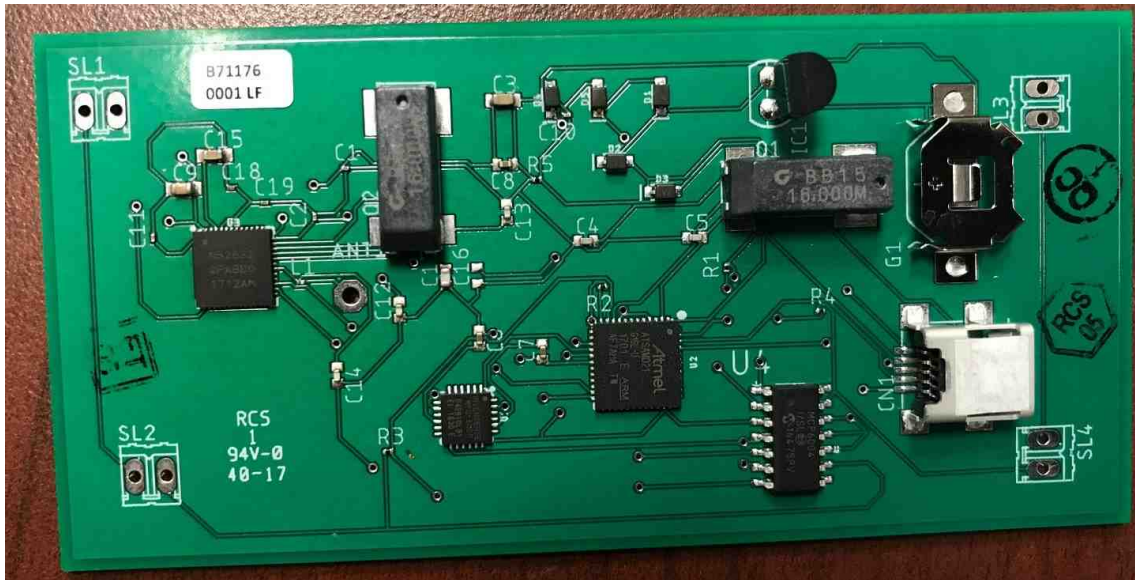


Fig. 5.4. Fabricated Custom PCB based on Model 1

and the number of SMDs used. Also, PCB design guidelines given in the nrf52832 datasheet which were followed. This includes placement of the components and the distance between them. Figure 5.4 shows the PCB with the various components. The various parameters of the board were shown in Figure 5.5.

5.2.3 Drawbacks

Usage of an interface MCU is preferred but not required. Presence of this MCU not only consumes space but also complicates the circuit. The absence of antenna and NFC in this design can be considered as another drawback. MCP6004 covers more space than required. Heat sinks when first ignored within the board, the high power components did not last long. The board is then redesigned and the heat sink metal was added to protect the devices from the thermal heating.

Fabrication Parameters:	
Number of layers:	2
Board name:	new thesis schematic.brd
Board width (dimension X):	3.499213 in
Board length (dimension Y):	1.687008 in
Board thickness:	0.061811 in
Copper thickness outer layers:	0.001378 in
Copper thickness inner layers:	undefined
Solder sides:	Both Sides
Silkscreen sides:	Top Side
Number of SMD pads on top:	219
Number of SMD pads on bottom:	0
Number of blind or buried hole types:	0
Minimum trace width (track width):	0.006000 in
Minimum SMD pitch:	0.006000 in
Minimum hole size:	0.013780 in
Assembly Parameters:	
Number of different packages:	17
Number of BGAs:	0
Number of QFNs:	3
Number of fine pitch packages:	6
Number of other SMDs:	30
Number of thru hole packages:	5
SMDs on both sides:	No

Fig. 5.5. Model 1 board parameters

5.3 Model 2

5.3.1 Schematic

The new design is based on the Sparkfun's nrf52832 breakout board. This design removes the usage of interface MCU and uses MCP6002. MCP 6002 is an op amp that has only two input and ground pairs with two output signals. Also, as seen in the specifications, the device is very compact, smaller than the previous design. It has an antenna added as per the nRF52 requirements. Apart from this, there is also a pin for NFC. Two buttons were provided to help in resetting and bootloading.

5.3.2 PCB layout

This has the similar Design rule check with only the change in minimum drill size which is 0.6 mil in this design. Apart from this, adding an antenna and NFC can be considered as a major change for this design.

Fabrication Parameters:	
Number of layers:	2
Board name:	NRFSELFMADE.brd
Board width (dimension X):	1.849213 in
Board length (dimension Y):	1.677008 in
Board thickness:	0.061811 in
Copper thickness outer layers:	0.001378 in
Copper thickness inner layers:	undefined
Solder sides:	Both Sides
Silkscreen sides:	Top Side
Number of SMD pads on top:	224
Number of SMD pads on bottom:	0
Number of blind or buried hole types:	0
Minimum trace width (track width):	0.005000 in
Minimum SMD pitch:	0.006000 in
Minimum hole size:	0.006000 in
Assembly Parameters:	
Number of different packages:	31
Number of BGAs:	0
Number of QFNs:	2
Number of fine pitch packages:	0
Number of other SMDs:	55
Number of thru hole packages:	5
SMDs on both sides:	No

Fig. 5.6. Model 2 board parameters

5.3.3 Drawbacks

The major drawback associated with this small size is that the antenna requires drills or holes placed at every 10mm in x and y directions for reducing heat radiation.

In order to avoid thermal effect on the board components, the space around the was void (not utilized). As it can been in Figure 5.6, the number of components on the board increased when compared to the previous model. Hence, this model was not favorably selected for the PCB fabrication.

5.4 Model 3

5.4.1 Schematic

This is the final PCB design model that was used. The drawbacks of the previous designs were eliminated by placing drills every 10 mm on the board. A proper NFC pin (FPC) was installed to facilitate the connection of the NFC antenna. This reduces the requirement of the design with antenna, with minor drawback regarding the use of a battery cell at the bottom layer in order to balance out the space used by antenna side of the board.



Fig. 5.7. Fabricated Custom PCB based on Model 3

Figure 5.7 shows the fabricated final prototype. Though it is not as small as model 2, it is much compact when compared to model 1. Another major design change is the elimination of the op amp. Using the gain in the software, the required output can be obtained from the FSRs. This design was done by taking Nordic semiconductors development support to eliminate any possibility of errors. It is known that the voltage supplied by the USB is about 5V. A voltage regulator was used in all the three models to bring it down to 3.3V so that the power consumption will not impact the nrf52832. The voltage for FSRs and motion sensors are connected to this 3.3V line. Finally, a number of holes or drills are provided under the nrf52832 to act an heat sink to eliminate the possibility of damage due to heat. The dimensions and the number of components can be seen in Figure 5.8.

5.5 Effect of Relative Humidity and Temperature in Shoe Sole

Deng et al., [86] studied the relative humidity and temperature inside the shoe with respect to time was clearly discussed. Regular shoes were used in this experiment and results were as follows:

1. As the duration of wearing of the shoe progresses, the temperature in the shoe increased gradually. The normal temperature inside a shoe is about 21°C. As the time progresses, this temperature reached about 27°C in 100 minutes.
2. The relative humidity inside the shoe is about 30% at the nominal conditions. As the time progresses, the humidity inside the shoe also increases. The relative humidity increases to 90% in 90 minutes. However, the temperature and relative humidity tends to stabilize in 30 minutes and 10 minutes while walking. Considering these values, all the components selected for the Custom PCB were within this limit.

Most of the components works without an issue between temperatures, -55°C to 125°C. Moisture sensitivity level is the electronic standard for an IC which is exposed to ambient room conditions for any given time [87]. Most of the components used

Fabrication Parameters:	
Number of layers:	2
Board name:	nRF52832_MCU.brd
Board width (dimension X):	2.191339 in
Board length (dimension Y):	1.033504 in
Board thickness:	0.061811 in
Copper thickness outer layers:	0.001378 in
Copper thickness inner layers:	undefined
Solder sides:	Both Sides
Silkscreen sides:	Both Sides
Number of SMD pads on top:	197
Number of SMD pads on bottom:	1
Number of blind or buried hole types:	0
Minimum trace width (track width):	0.005906 in
Minimum SMD pitch:	0.005906 in
Minimum hole size:	0.015748 in
Assembly Parameters:	
Number of different packages:	19
Number of BGAs:	0
Number of QFNs:	1
Number of fine pitch packages:	1
Number of other SMDs:	46
Number of thru hole packages:	8
SMDs on both sides:	Yes

Fig. 5.8. Model 3 board parameters

has the relative humidity of 85% at a temperature of 30°C. These components are categorized as MSL level-1.

6. PROTOTYPING OF THE CUSTOM PCB

6.1 Testing and Debugging

During the testing and debugging phase of the custom PCB design, the issues that might be introduced and probable solution opted is discussed below:

1. There might be issues with the design proposed or in the schematics. The design was based on the nordic semiconductors' reference design. The designed schematic was verified again with respect to datasheet to eliminate the possibility of errors.
2. Careful inspection of the Custom PCB has been done with components and circuit connections.
3. There are two open connections that jumpers were used to close the circuit.
4. The orientation and the placement of the components need to be verified versus the schematics.
5. Proper firmware needs to be dumped into the custom PCB. The way in which this is done is relative to the processor. JLINK debugger is used to dump the firmware into the custom PCB. SoftDevice is later dumped into it using nrfjprog.

6.2 Verification of Working of the PCB

The oscilloscope and voltmeters were used to verify the working functions of the custom PCB based on the flow of the voltage throughout the board. Initially, it was observed that the voltage from the USB is not being supplied to the nrf52832 MCU.

This was because of an error during fabrication. A jumper was connected to make it a closed circuit. Three holes on both the sides of the USB pin was provided to verify the voltage across the device. The proper output voltage with levels of 5V and 3.3V were verified as specified in the design.

Figure 6.1 shows the results obtained from the oscilloscope. This shows that the voltage across the USB and the ground of the PCB is uniform and cyclic. The



Fig. 6.1. Readings taken from oscilloscope

current ratings of the board was compared with the Development kit. Between 5V and ground, the resistance was about $11\text{M}\Omega$ in the development kit. This was the same for the custom PCB as well. Also, the resistance between 3.3V and ground were about $7\mu\text{A}$. From [67], the minimum current rating is about $0.4\mu\text{A}$ when the system is off and $1.6\mu\text{A}$ when system is on. From Ohm's law, it can be seen that the current satisfies the condition mentioned.

The issue was with the analog channel output. The resistance at those pins were 370Ω . For a FSR to work, a high resistance at the pins were required. To eliminate that issue, an external resistors were added in series to the FSR. Since the basic conditions were all met, the code is then dumped to verify it is working. Before that, board initiation was done.

6.3 Initiation of the Board

The board needs to be detected by the system to dump the code. This can be done either by writing a driver for the board or by using debuggers. For this thesis, debuggers were used to dump the code. There are two different debuggers that can be used to dump the code.

Using JLINK Debugger

SWD pins were used to debug the PCB. The debug in of the custom board is connected to debug out of the JLINK debugger. Figure 6.2 shows the connection of the debugger with the PCB. When the commands shown in Figure 6.3 were entered in the JLINK command prompt, the custom PCB is then detected. Now, the previously used JLINK bin file was again dumped into this PCB. It can also be seen that when the reset button on the PCB was pressed, the red light blinks on the JLINK debugger. However, no further programming can be done on the PCB without loading the SoftDevice in it. To accomplish that, nrfjprog commander is used along with the nRF52 development kit acting as the debugger.

Using Development Board as a Debugger

The development board was connected to the PCB as shown in the Figure 6.4. The debug in of the PCB is connected to the debug out of the nRF52 development kit. The dumping of the SoftDevice via nrfjprog in command prompt. Now, the preexisting



Fig. 6.2. Bootloading with JLINK Debugger

```

J-Link Commander
Connecting to J-Link via USB...O.K.
Firmware: J-Link V9 compiled Feb 23 2018 11:28:35
Hardware version: V9.40
S/N: 269402382
License(s): FlashBP, GDB
DEM: SEGGER-EDU
Vtref = 3.360V

Type "connect" to establish a target connection, '?' for help
J-Link>connect
Please specify device / core. <Default>: NRF52832_XXAA
Type '?' for selection dialog
Device>
Please specify target interface:
  3) JTAG (Default)
  5) SWD
JTAG>
Specify target interface speed [KHz]. <Default>: 4000 KHz
Speed>
Device "NRF52832_XXAA" selected.

Connecting to target via SWD
Found SW-DP with ID 0x28A01477
Found SW-DP with ID 0x28A01477
Scanning AP map to find all available APs
AP[2]: Stopped AP scan as end of AP map has been reached
AP[0]: AHB-AP (IDR: 0x24778011)
AP[1]: JTAG-AP (IDR: 0x82880000)
Iterating through AP map to find AHB-AP to use
AP[0]: Core found
AP[0]: AHB-AP ROM base: 0xE00FF000
CPUID registers: 0x410FC241, Implementer code: 0x41 (ARM)
Found Cortex-M4 r0p1, little endian.
FPUnit: 6 code (BP) slots and 2 literal slots
CoreSight components:
ROMTbl[0] @ E00FF000
ROMTbl[0][0]: E000E000, CID: B105E00D, PID: 000BB00C SCS-M7
ROMTbl[0][1]: E0001000, CID: B105E00D, PID: 003BB002 DWT
ROMTbl[0][2]: E0002000, CID: B105E00D, PID: 002BB003 FPB
ROMTbl[0][3]: E0000000, CID: B105E00D, PID: 003BB001 ITM
ROMTbl[0][4]: E0040000, CID: B105900D, PID: 000BB0A1 TPIU
ROMTbl[0][5]: E0041000, CID: B105900D, PID: 000BB025 ETM
Cortex-M4 identified.

```

Fig. 6.3. Output received at JLINK Commander

code is dumped into the custom PCB via Keil. The only variation from the previous setup is the usage of level 3 optimization. The changes in the pressure sensor is noted at UART. Due to the lack of drivers in the custom PCB, the output from the

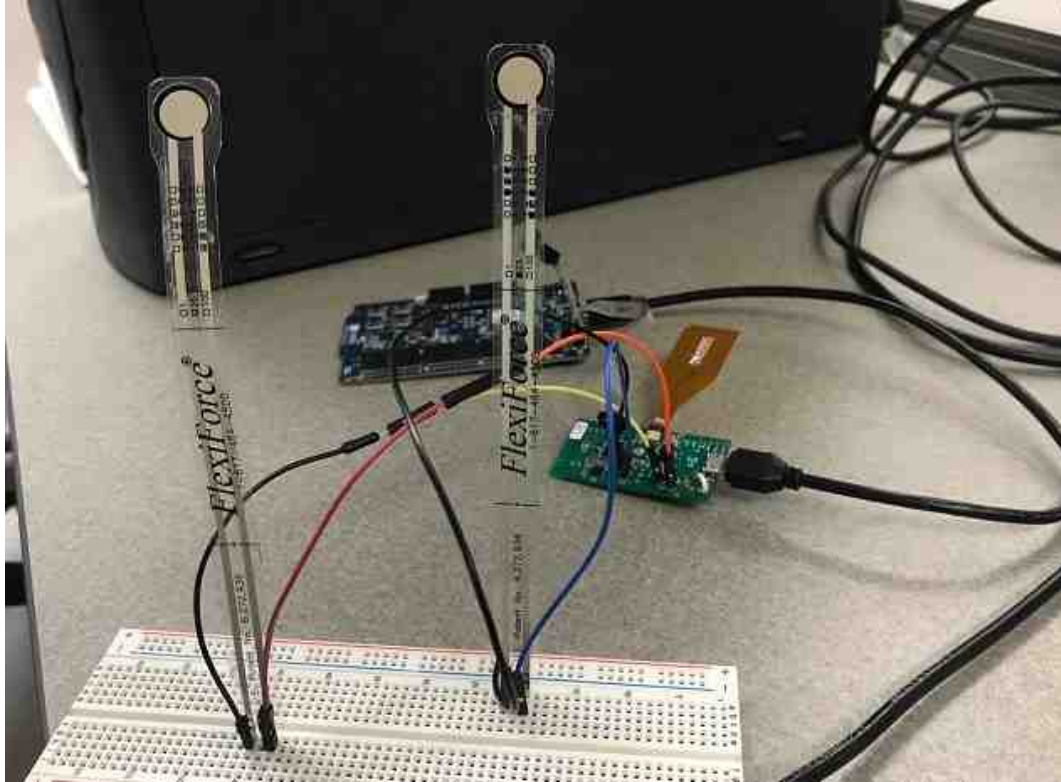


Fig. 6.4. Working set up of custom PCB with nRF52 Development board

analog signals cannot be seen directly on the screen. The device uses the UART driver available on the nRF52 development board to get the output on TeraTerm. This drawback can be eliminated when the drivers for it can be written which was considered as a future work. Chapter 7 discusses about the results obtained in this thesis and the impact of the results in the field of gait research.

7. RESULTS

The setup used to detect a human gait with nRF52 development kit as the base station with motion sensor and FSRs embed in it. Figure 7.1 was the smart shoe prototype used to get the desired outputs from the system.

7.1 Output from FSRs

It was previously discussed about gains in SAADC analog channels. Two FSRs were placed, one at heel and one at toe area respectively. Two walk cycles were considered while recording the values obtained. The outputs obtained from the sensor for different gains are as shown below.

Figure 7.2 shows the output of the two FSRs with gain =1/6. Here, It can be seen that the maximum peak pressure value is about 4KPa. This makes it difficult to differentiate between the no pressure condition and peak pressure condition. Hence, this gain was not used to get the desired output.

Figure 7.3 shows the output of the two FSRs with gain =1/5. Here, the maximum peak pressure value is about 6KPa. Just like with gain-1/6, this is also difficult to differentiate between the no pressure condition and peak pressure condition. Hence, this gain was also not used to get the desired output.

Figure 7.4 shows the output of the two FSRs with gain =1/4. Here, the maximum peak pressure value is about 6.5KPa. Just like with the previous conditions, this is also difficult to differentiate between the no pressure condition and peak pressure condition.

Similarly, Figure 7.5 and Figure 7.6 gives peak pressure value as 12. Though these are higher when compared with previous gains, the difference is not big enough to be used for all cases.

Figure 7.7 shows a decent difference between the minimal pressure value and the peak. This may be selected for to be the required gain value. However, Figure 7.8 of Gain=2 shows a much better difference when compared with Gain=1.

When differences in the pressure is to be only considered, then Gain =4 gives the best possible result as shown in Figure 7.9. However, gain=4 poses another issue. With increase in pressure, the values change drastically. To eliminate such possibility, Gain=2 was the best out of all the given gains. Hence, it was selected for the design.

7.1.1 Flexibility and Sensitivity

Flexibility of the FSR was verified by manually applying pressure to bend the sensor area inward as shown in Figure 7.10(a) and outward as shown in Figure 7.10(b). It was observed that the bending effects the sensors leading to abnormal analog readings. To eliminate this issue, thin tapes were used to fix the FSRs from moving or bending. Sensitivity of the FSRs was tested using COMSOL Multiphysics. 350KPa pressure was applied and the results can be seen in Figure 7.11. The output obtained shows that the sensors work as expected even at high pressures. The displacement on the sensor is very negligible. The output for two FSRs placed at a distance of 10 cm, distance between heel to toe, is given in Figure 7.12.

7.2 Output from the Motion Sensor

The accelerometer of the MPU9250 has been varied between 2G and 16G. The results are given in Figure 7.13 for 2G and Figure 7.14 for 16G. Accelerometer with 16G sensitivity is used for this work. The comparison between various filters are given in the Figure 7.13 and Figure 7.14. The accelerometer values are sent through the low pass filter, and gyroscope values are sent through high pass filter for the complementary filter. Because Kalman filter gives a smoother and better output, it was selected for the filtering method.

7.3 BLE Outputs

Figure 7.15 shows the advertising of the device when the code is downloaded into it. Figure 7.16 shows the log data when the device is connected to a mobile device via BLE. Figure 7.18 shows the output displayed on the phone based on the data received from analog values. The sensor values can be also be seen on the mobile device if required as shown in Figure 7.17.

7.4 Improper Gait: Analysis and Diagnosis

Normal Gait

The walk cycle of a Normal Gait can be seen in Figure 7.19. (1) shows the heel strike,(2) shows foot flat to mid stance transition, end of (2) shows initiation of heel off, end of (3) initiates toe off. (4) is the mid-swing to swing transition. End of (4) shows the initiation of heel strike, thus completing a walk cycle.

Normal gait is detected when all the stages (except swing) are detected within a specified time and output is sent via BLE which can be seen in Figure 7.20.

The output for the Normal gait can also be shown as per the Figure 7.21. This shows the amount of readings taken from each of the stages in a walk cycle. It is known that the 60% of an entire walk cycle comprises of the selected foot being in contact with the ground. The obtained result shows the percentage of each stage in a walk cycle when a part of the foot is in contact with the floor. When the cumulative readings of Heel strike, foot flat and toe off conditions cover over 60%, then the gait is considered to be normal.

Parkinson's Gait

The walk cycle of Parkinson's Gait can be seen in Figure 7.22 . As discussed, the walk cycle of a person suffering from Parkinsonian disease has smaller walk cycle and mostly contains foot flat condition.

Parkinson's Gait is detected when most of the readings are within $((30 < \text{heel} < 80) \&\& 5 < \text{toe} < 15))$ for a period of time (40 readings). The output is sent via BLE which can be seen in Figure 7.23.

The output for the Parkinson's Gait can also be shown as per the Figure 7.24. This shows the amount of readings taken from each of the stages in a walk cycle. From the previous discussion, it can be seen that the readings from foot flat occupies most of the output. When the readings of the foot flat condition cover over 60%, then the gait is considered to be Parkinson's Gait.

Injured Gait

The walk cycle of Injured Gait can be seen in Figure 7.25. As discussed, the walk cycle of a person suffering from an injury at toe has smaller walk cycle and mostly contains heel strike condition.

Injury Gait is detected when it satisfies $(\text{heel} > 50 \&\& \text{toe} < 5)$ for a period of time. The output is sent via BLE can be seen in Figure 7.26.

The output for the Injury Gait can also be shown as per the Figure 7.27. This shows the amount of readings taken from each of the stages in a walk cycle. From the previous discussion, it can be seen that the readings from heel strike occupies most of the output. When the readings of the heel strike condition cover over 60%, then the gait is considered to be Injury Gait.

Hemiplegic Gait

Hemiplegic Gait output was also obtained from the motion sensor which is given in Figure 7.28. It shows a noticeable variation in angle, by -10° , when compared with normal readings. This is used to detect this abnormal gait. It is detected when it satisfies $(\text{heel} < 7 \&\& \text{toe} < 7)$ and motion sensor readings show values less than -7° . The output is sent via BLE can be seen in Figure 7.29.

The output for the Hemiplegic Gait can also be shown as per the Figure 7.30. The Figure shows that the output of the sensors can also be sent to the mobile device along with the notification of abnormal gait. For this prototype, only one set of sensor output for roll and pitch was shown but multiple results from sensors can be obtained on the phone which can be saved as a data file. This data file can be viewed later for comparison and further diagnosis.

7.5 Flowchart

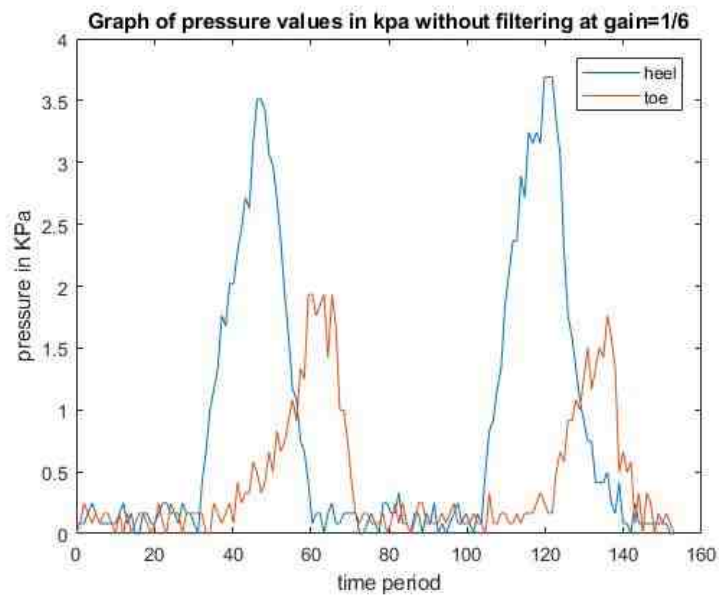
Figure 7.31 shows the flowchart of the code written. Pressure values and the motion sensor thresholds were taken from the previous results. The output of the code can be seen on the mobile phone as seen in the Figure.

7.6 PCB Layouts

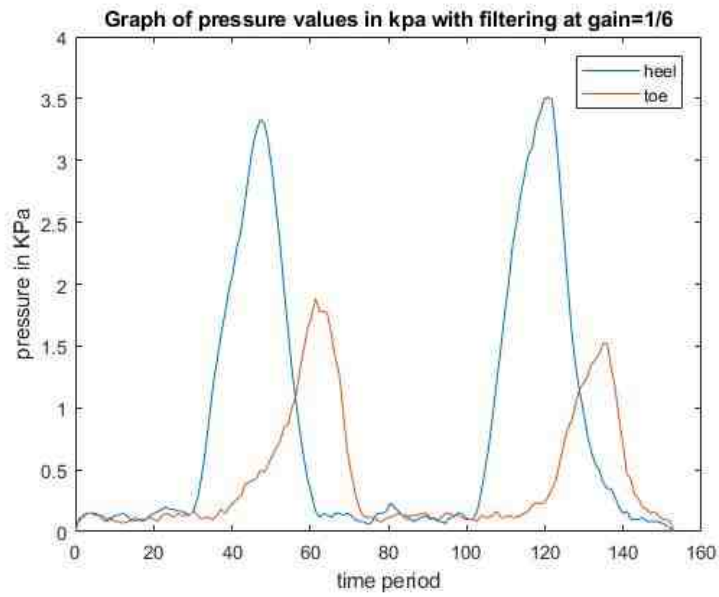
The schematics and PCB layout of the three models are given in Figure 7.32 to Figure 7.39. Final output from all the sensors embed are given in Figure 7.40.



Fig. 7.1. Prototype of the smart shoe

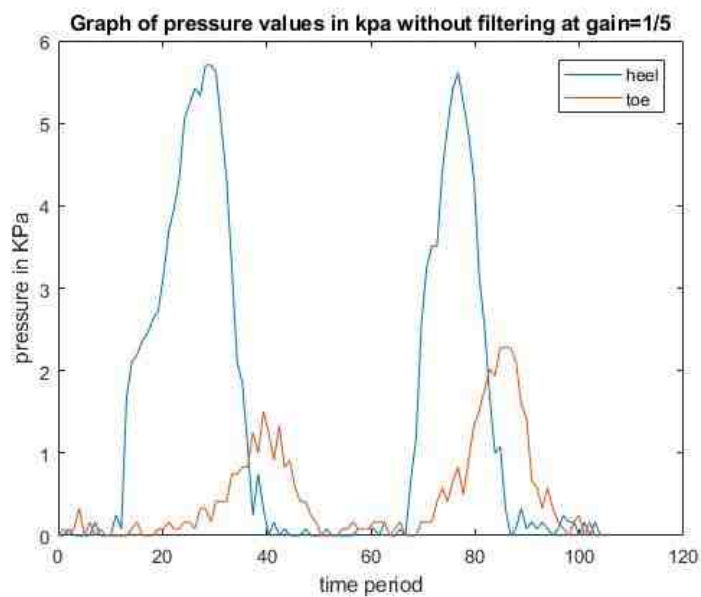


(a) Output of FSRs without filtering for gain=1/6

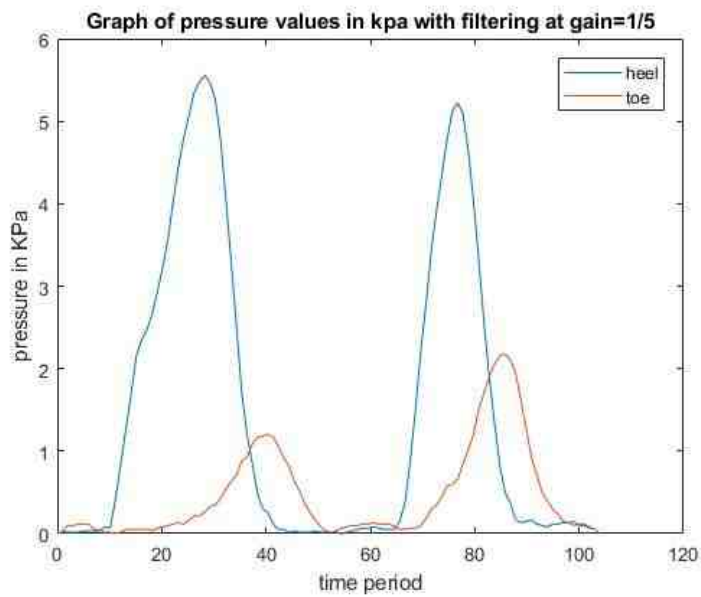


(b) Output of FSRs with smooth filtering for gain=1/6

Fig. 7.2. Outputs obtained from FSRs with Gain=1/6.

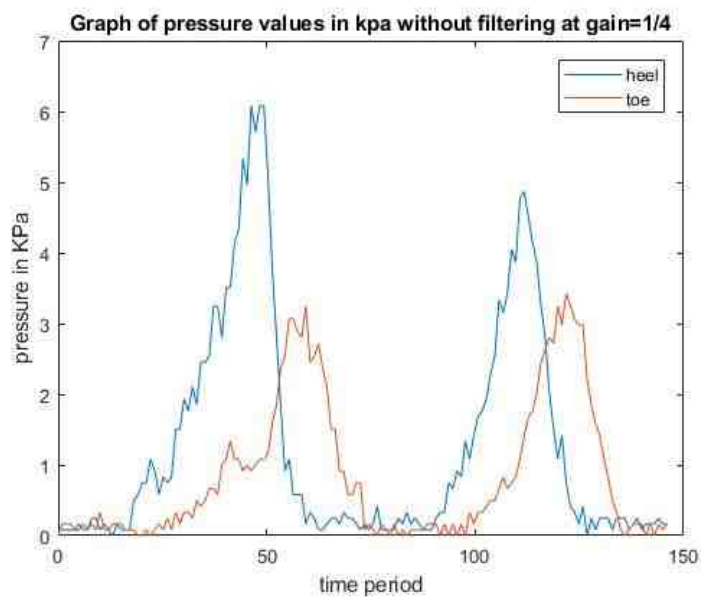


(a) Output of FSRs without filtering for gain=1/5

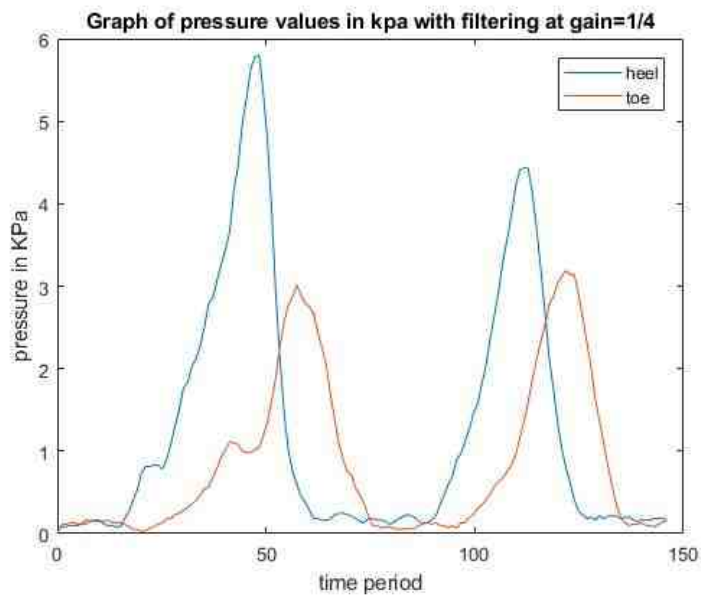


(b) Output of FSRs with smooth filtering for gain=1/5

Fig. 7.3. Outputs obtained from FSRs with Gain=1/5.

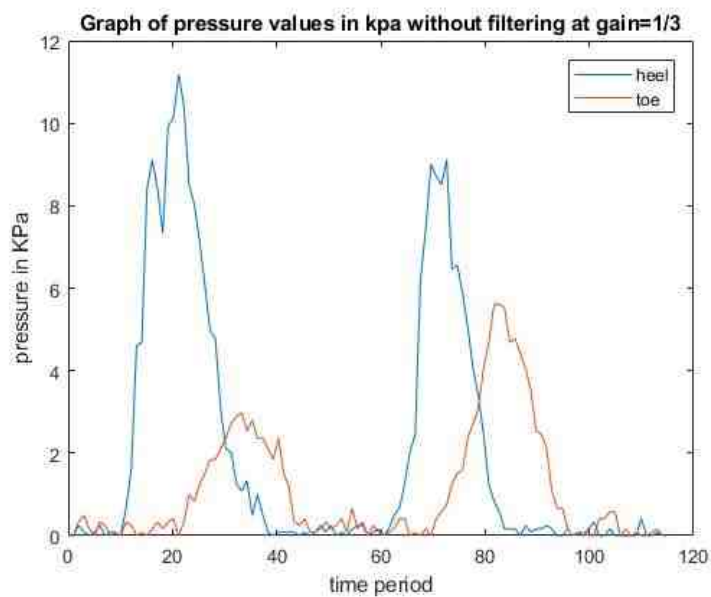


(a) Output of FSRs without filtering for gain=1/4

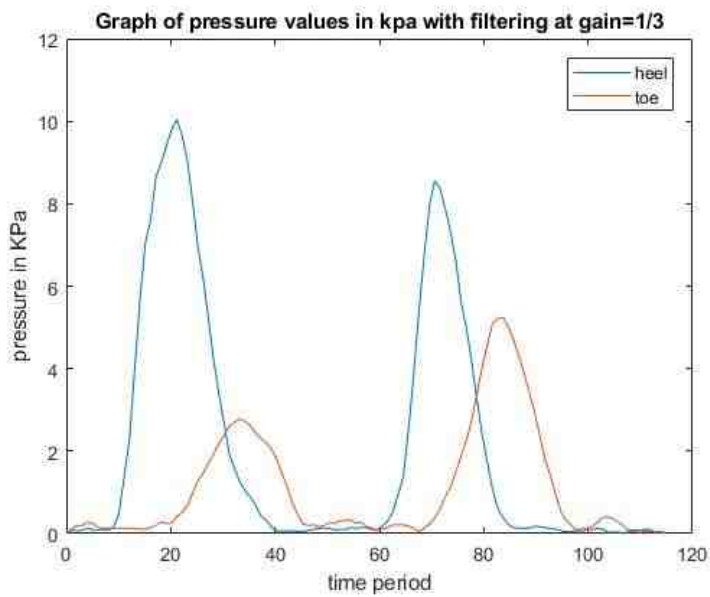


(b) Output of FSRs with smooth filtering for gain=1/4

Fig. 7.4. Outputs obtained from FSRs with Gain=1/4.

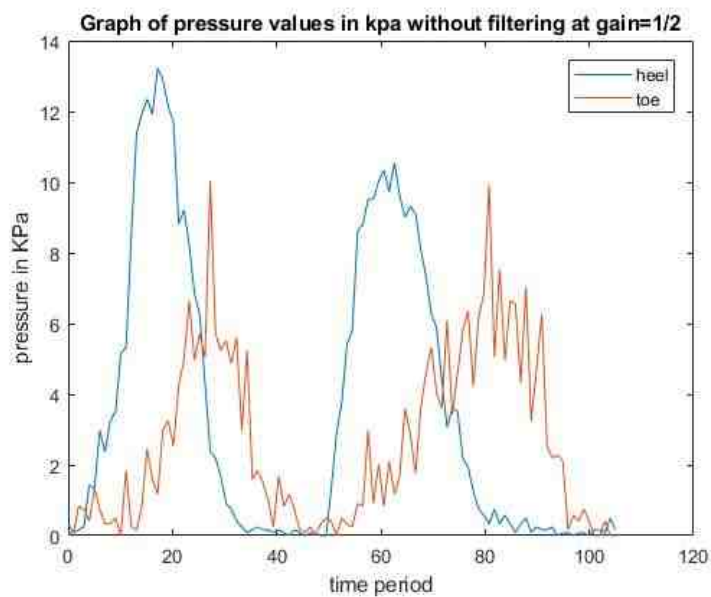


(a) Output of FSRs without filtering for gain=1/3

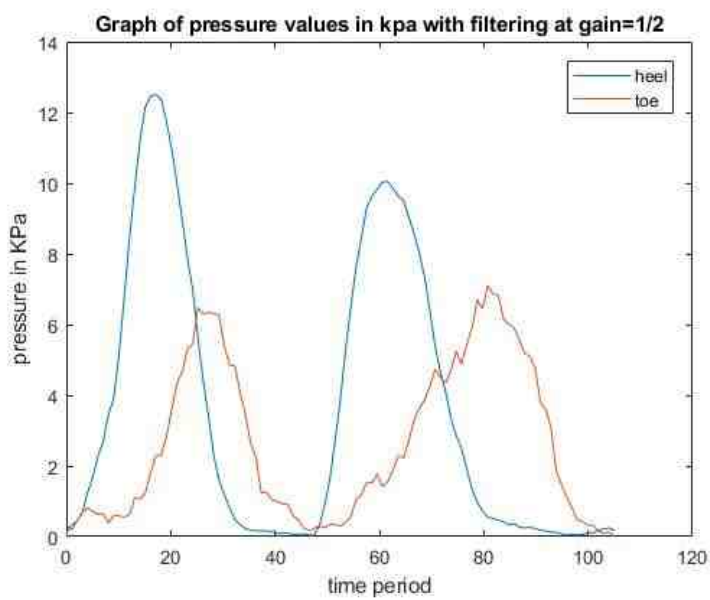


(b) Output of FSRs with smooth filtering for gain=1/3

Fig. 7.5. Outputs obtained from FSRs with Gain=1/3.

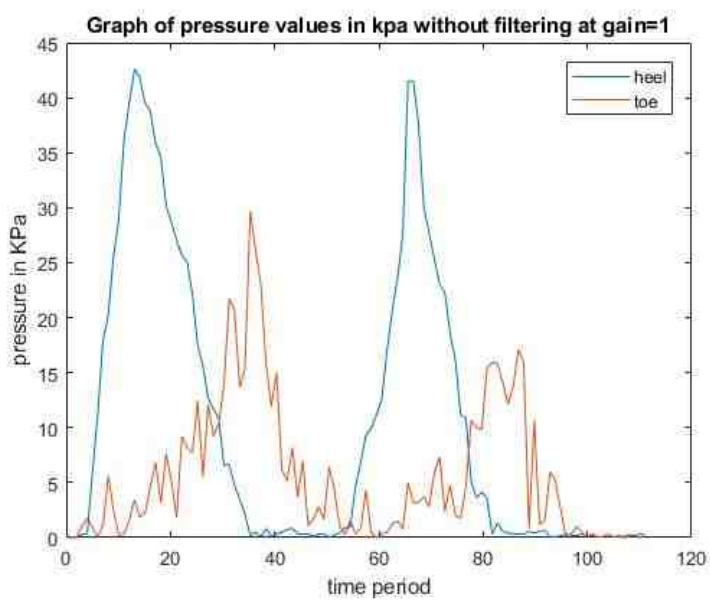


(a) Output of FSRs without filtering for gain=1/2

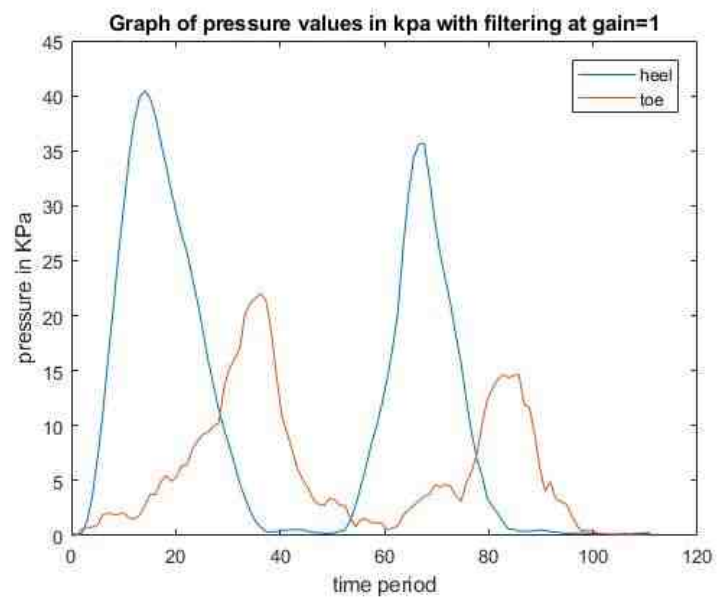


(b) Output of FSRs with smooth filtering for gain=1/2

Fig. 7.6. Outputs obtained from FSRs with Gain=1/2.

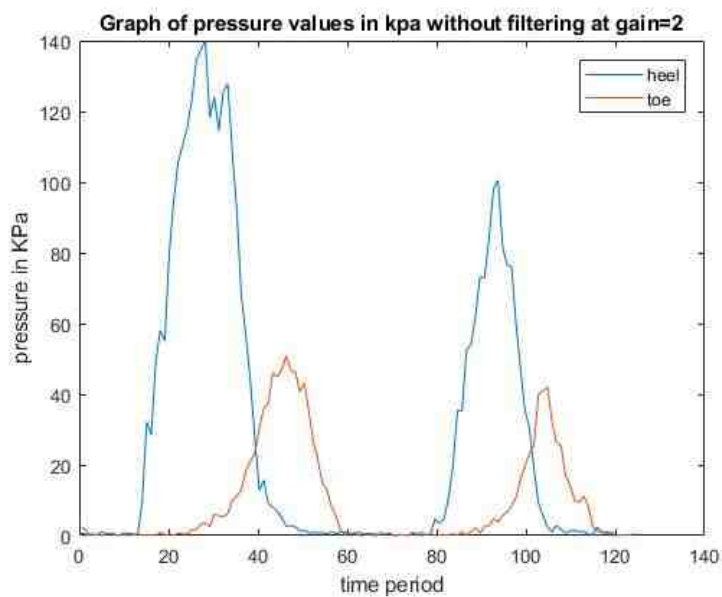


(a) Output of FSRs without filtering for gain=1

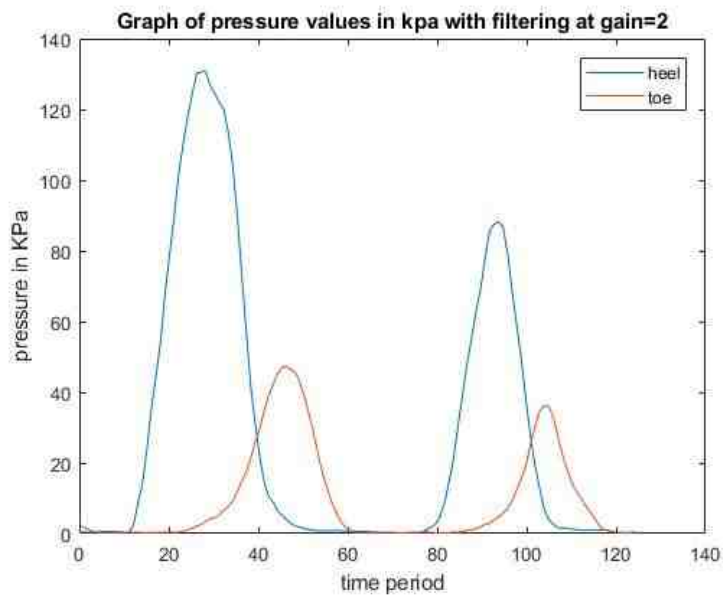


(b) Output of FSRs with smooth filtering for gain=1

Fig. 7.7. Outputs obtained from FSRs with Gain=1.

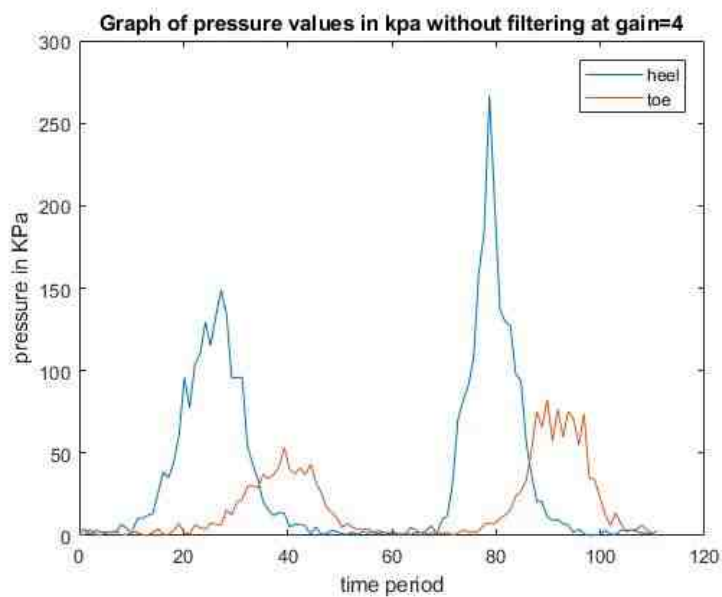


(a) Output of FSRs without filtering for gain=2

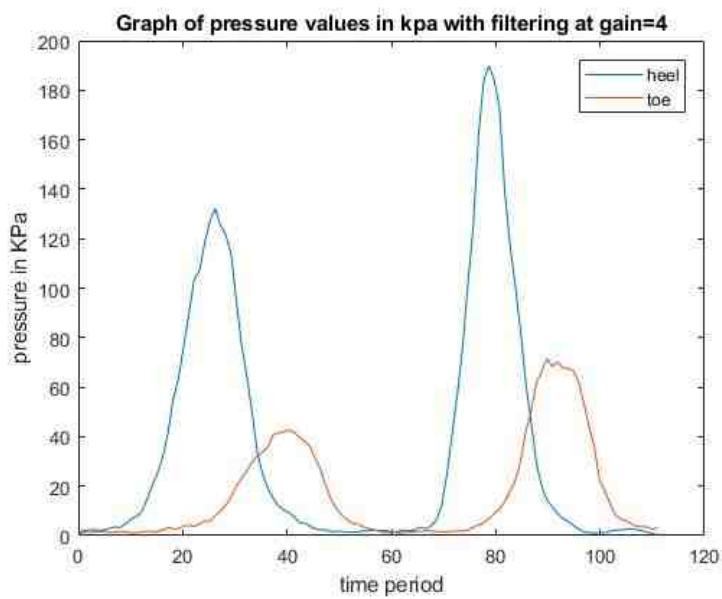


(b) Output of FSRs with smooth filtering for gain=2

Fig. 7.8. Outputs obtained from FSRs with Gain=2.

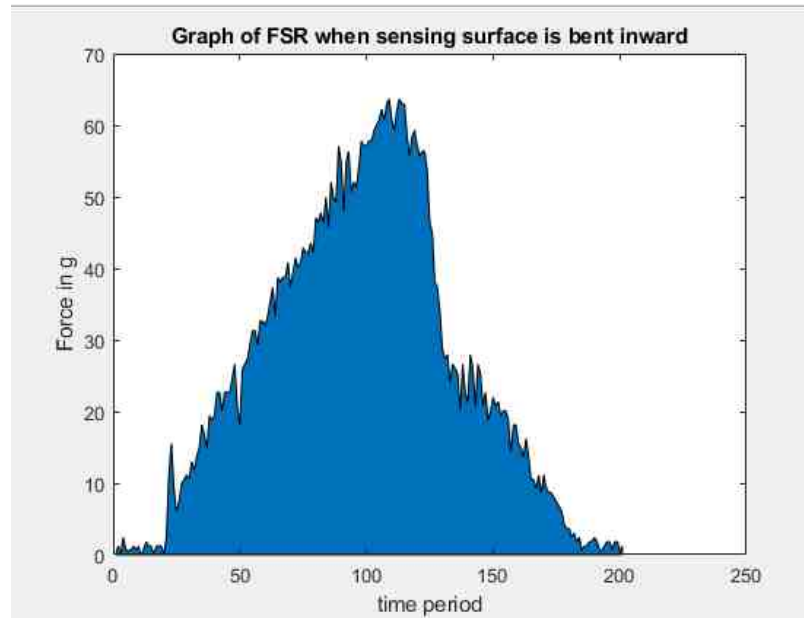


(a) Output of FSRs without filtering for gain=4

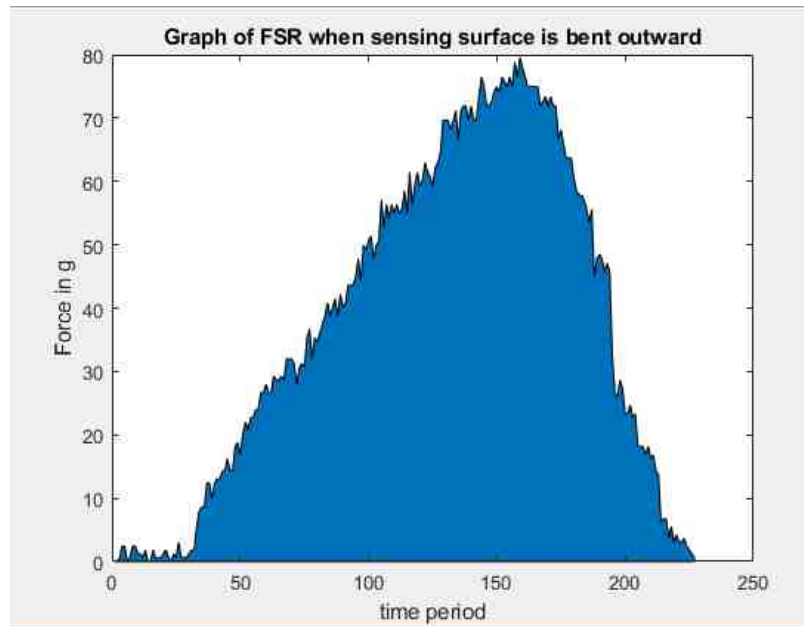


(b) Output of FSRs with smooth filtering for gain=4

Fig. 7.9. Outputs obtained from FSRs with Gain=4.



(a) Output of FSRs when bent inward



(b) Output of FSRs when bent outward

Fig. 7.10. Flexibility test of the FSRs.

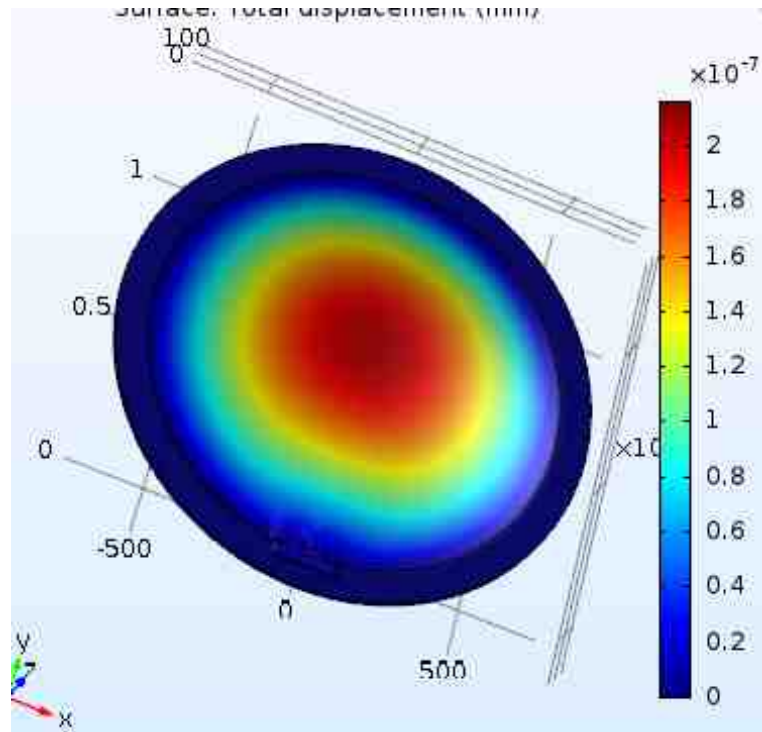


Fig. 7.11. COMSOL simulation of single FSR at 350KPa

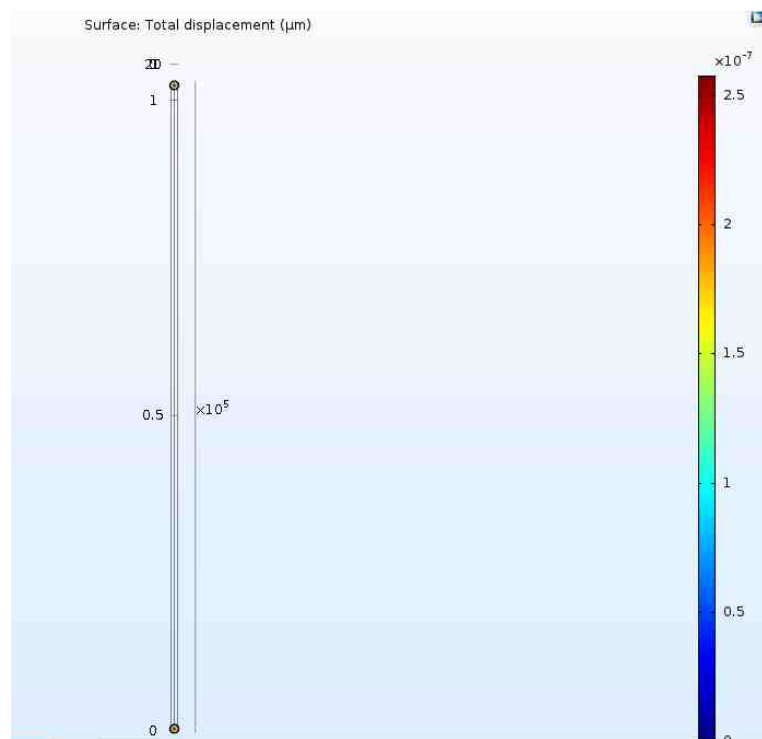
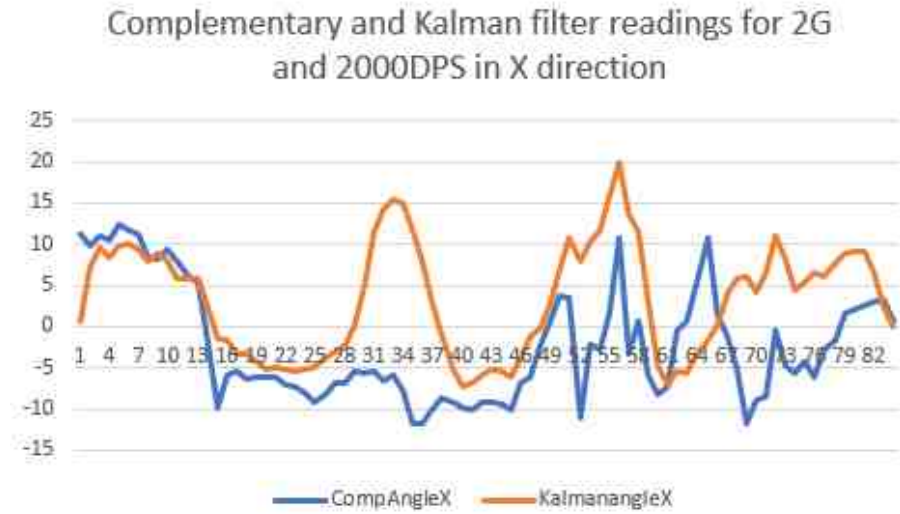
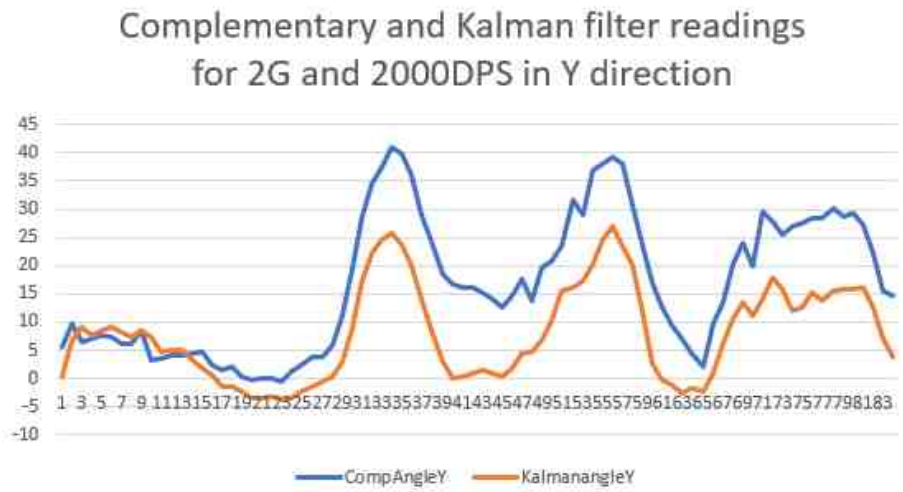


Fig. 7.12. COMSOL simulation of two FSRs at 350KPa

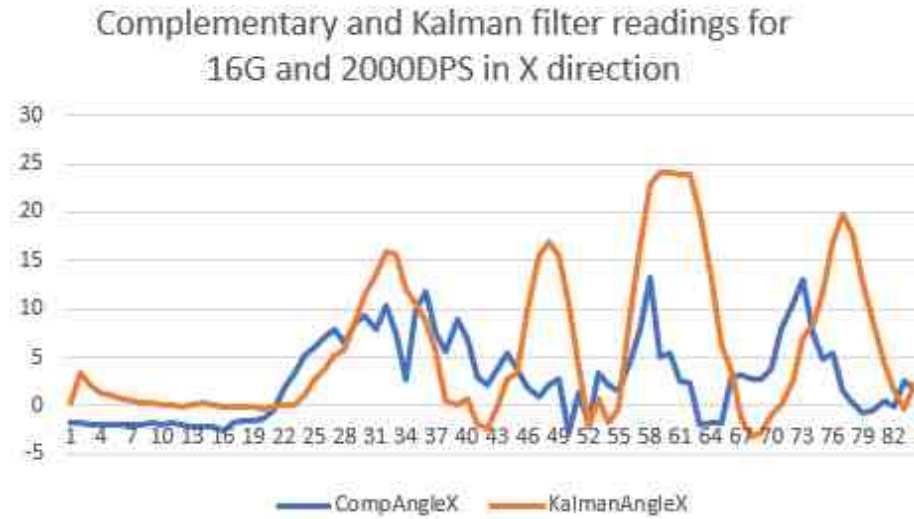


(a) Output of pitch for 2G accelerometer and 2000DPS gyroscope

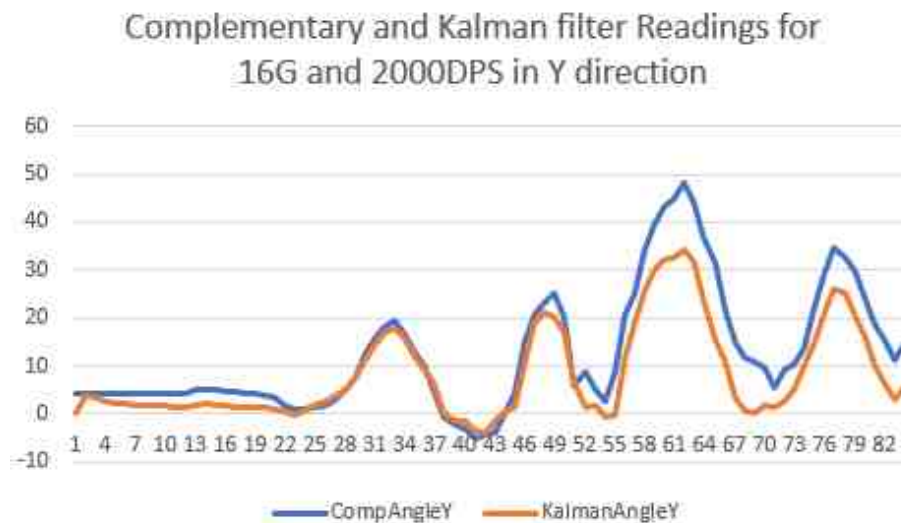


(b) Output of roll for 2G accelerometer and 2000DPS gyroscope

Fig. 7.13. Roll and Pitch values of 2G accelerometer and 2000 DPS Gyroscope



(a) Output of pitch for 16G accelerometer and 2000DPS gyroscope



(b) Output of roll for 16G accelerometer and 2000DPS gyroscope

Fig. 7.14. Roll and Pitch values of 16G accelerometer and 2000 DPS Gyroscope

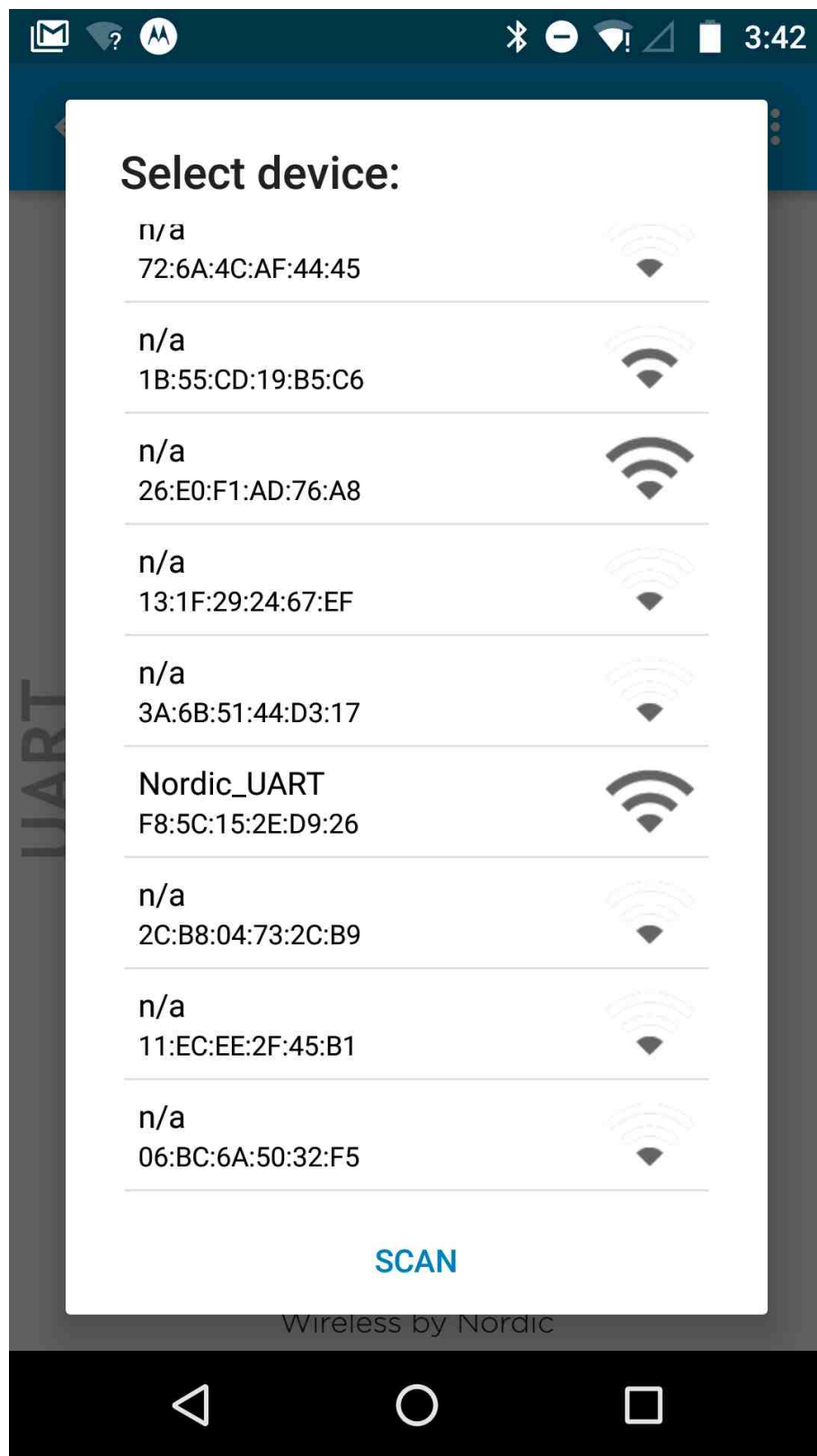


Fig. 7.15. Output from nRF toolbox Application when the device is advertising.

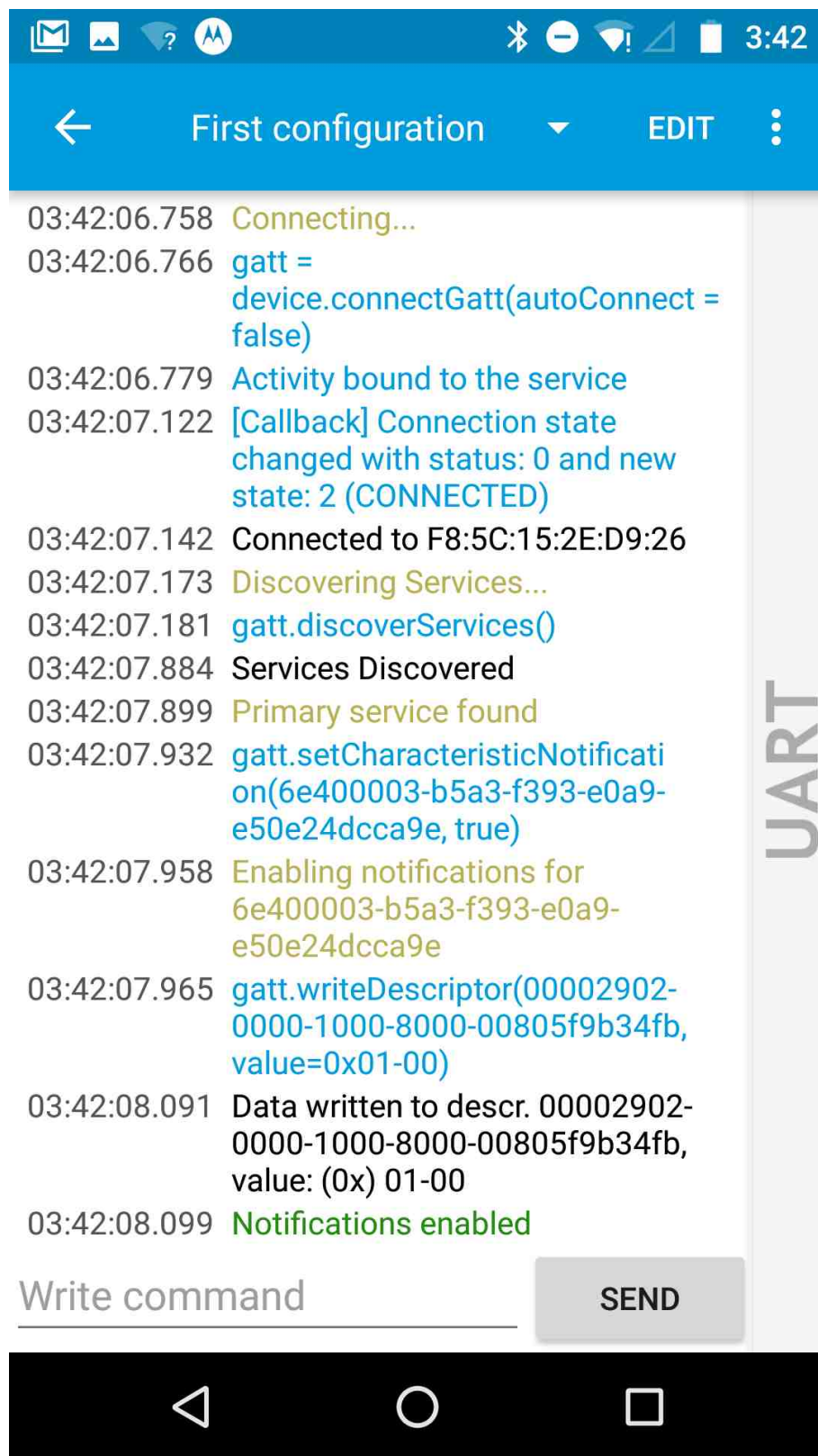


Fig. 7.16. Output from nRF toolbox Application-UART module when the device is connected.

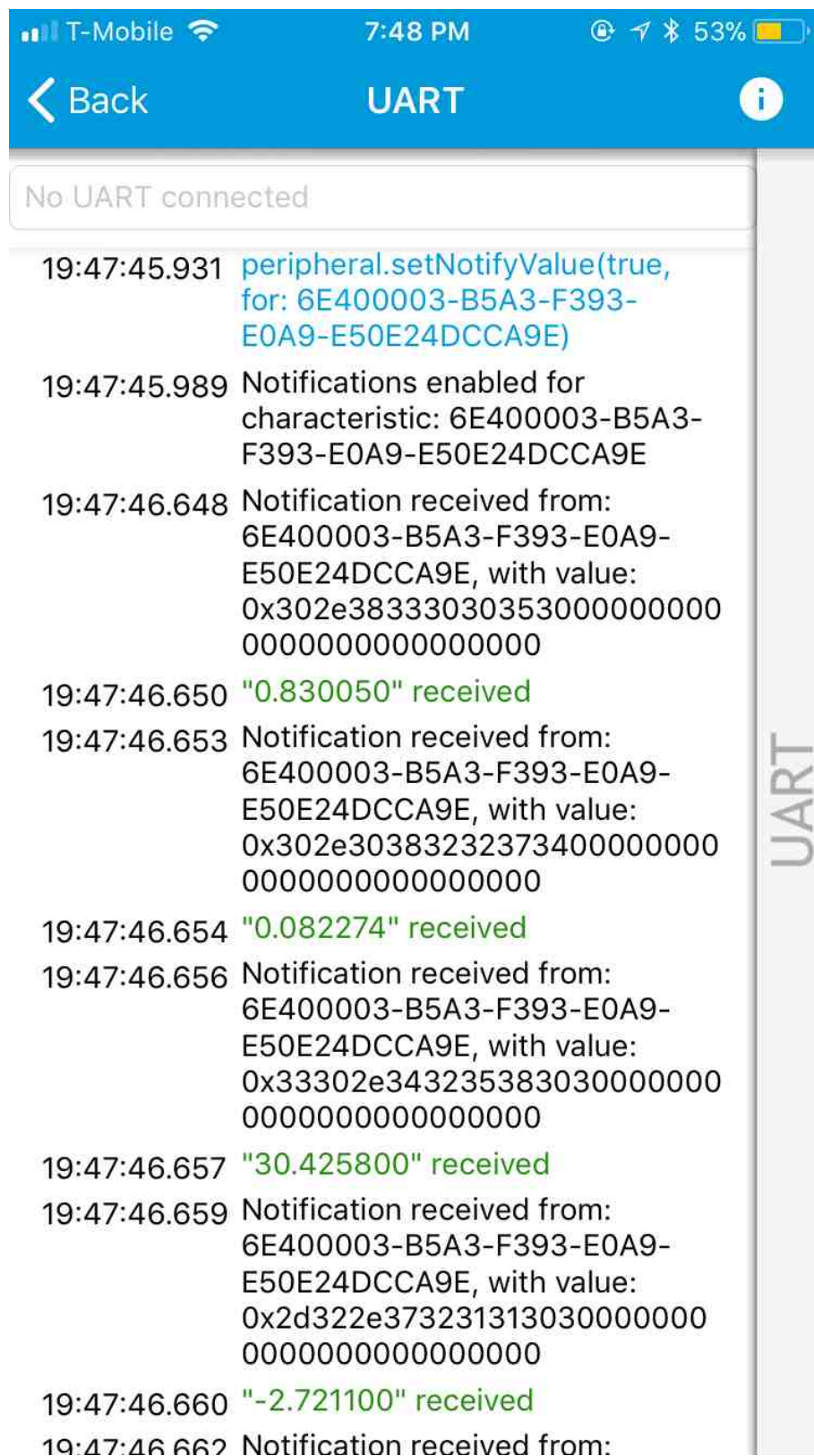


Fig. 7.17. Output from nRF toolbox Application- sensor values received via BLE.

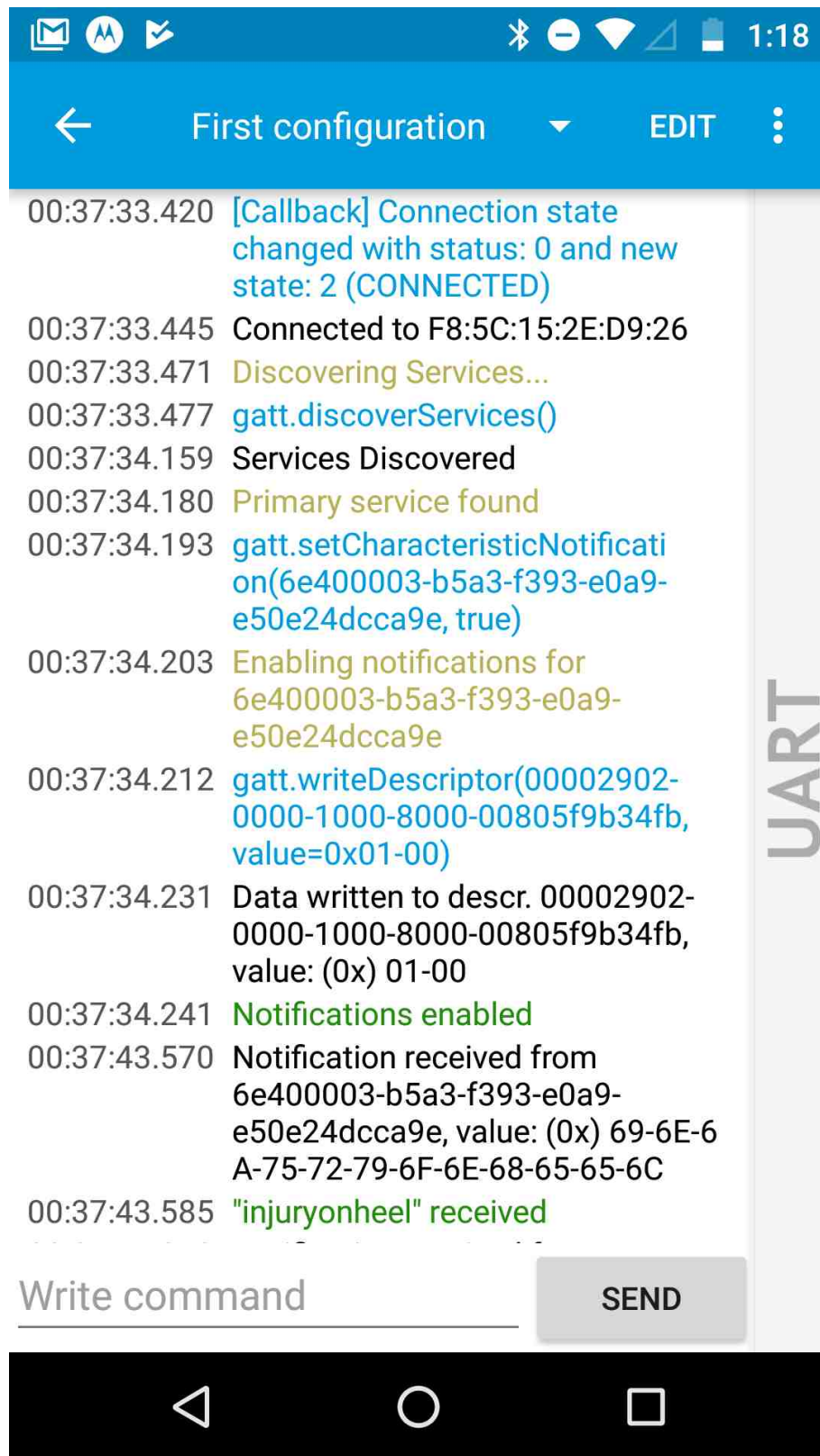


Fig. 7.18. Output from nRF toolbox Application-UART module when the device is connected.

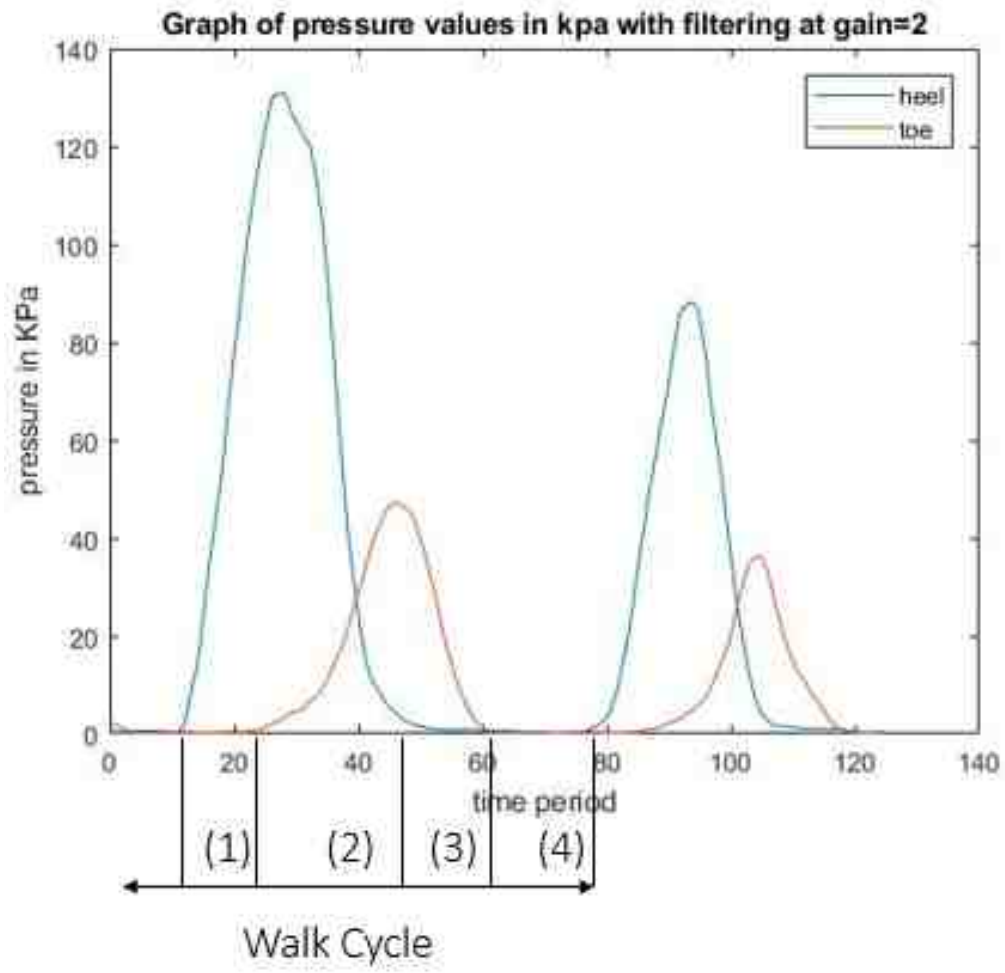


Fig. 7.19. Output of a Normal Gait

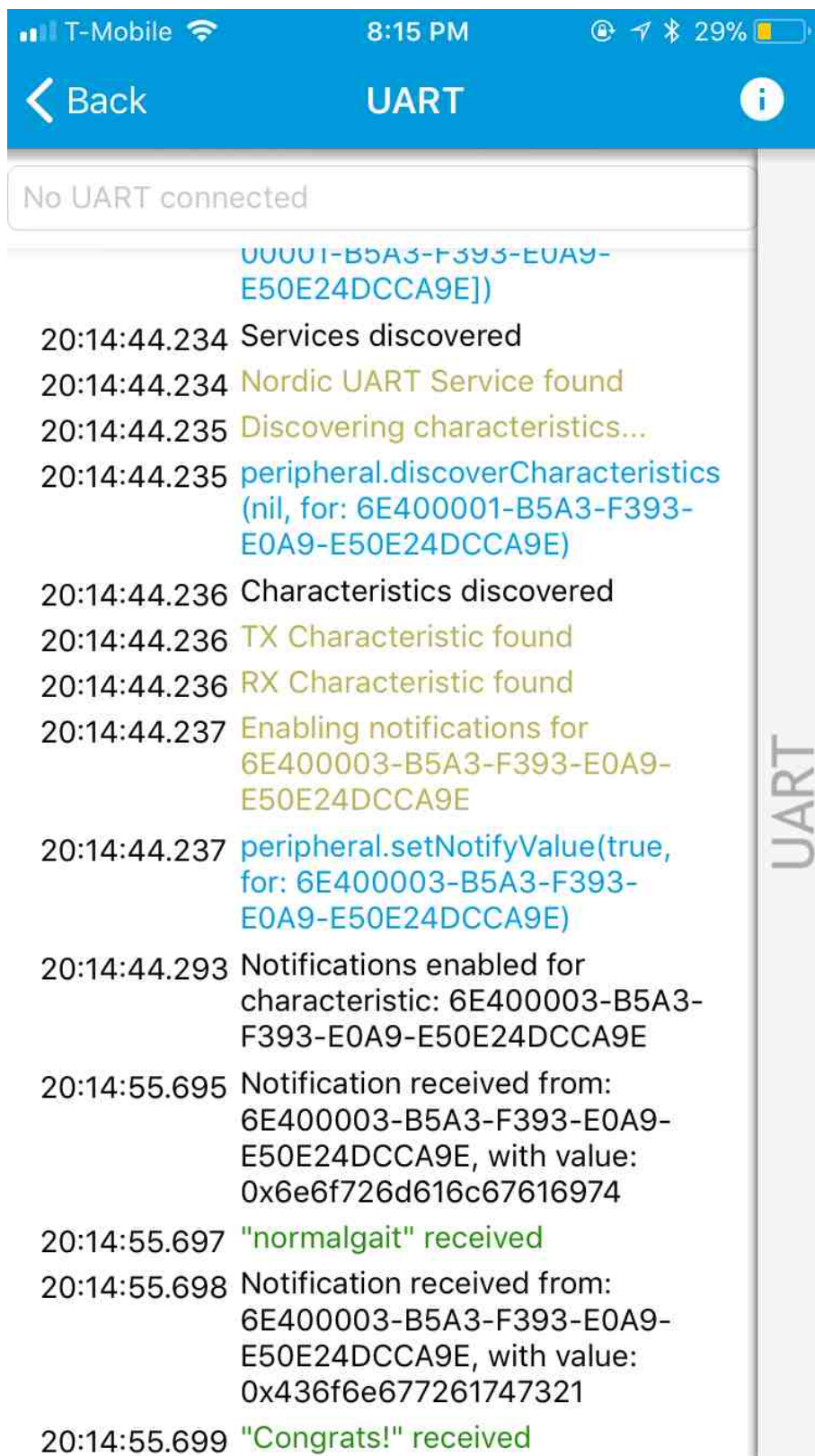
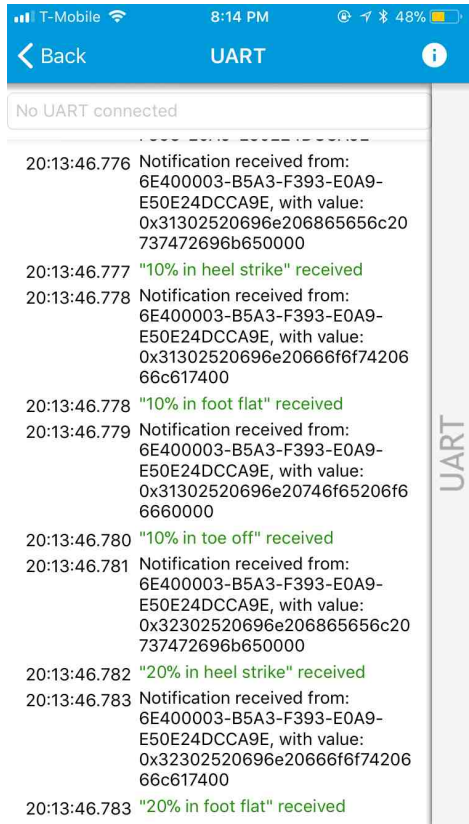
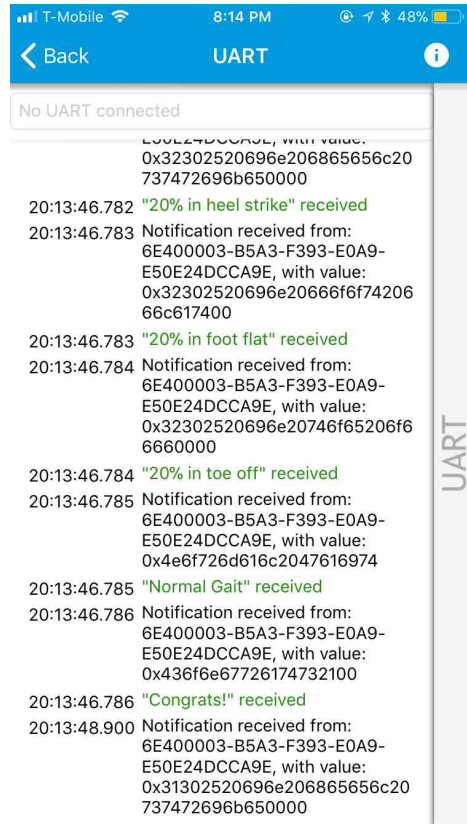


Fig. 7.20. Output of a Normal Gait in BLE



(a) Normal Gait Readings-1



(b) Normal Gait Readings-2

Fig. 7.21. BLE results for Normal Gait Diagnosis

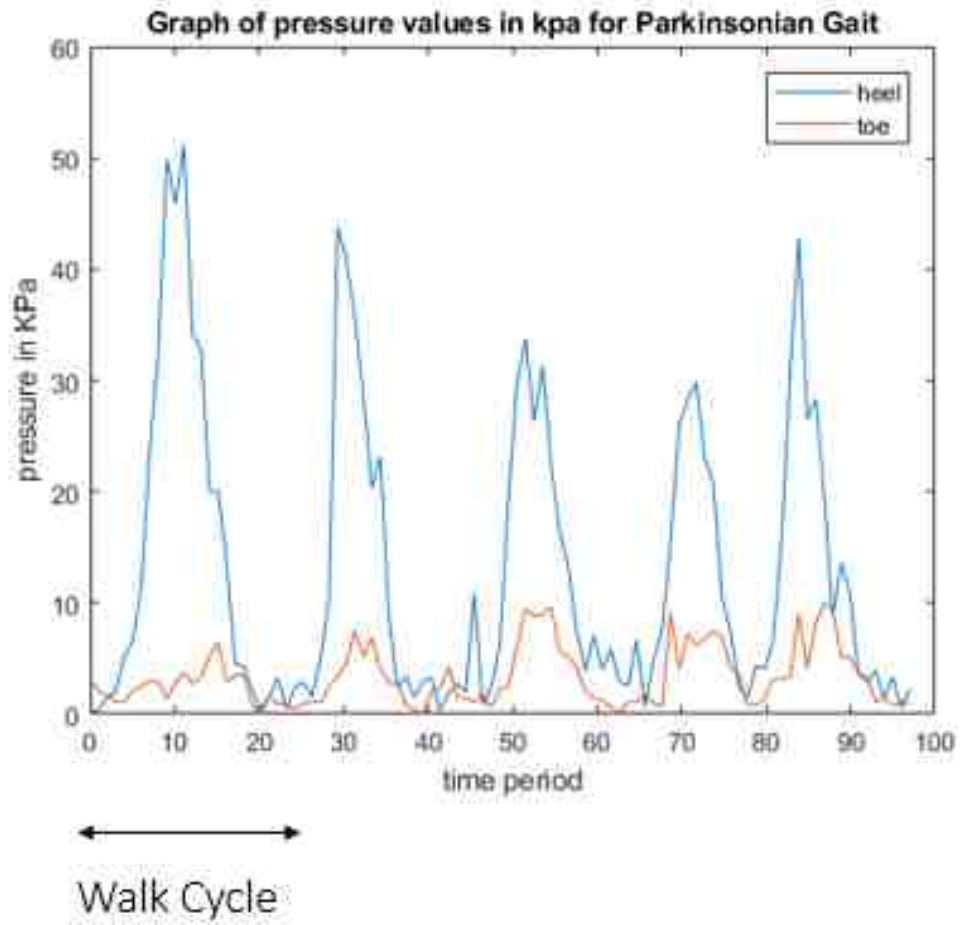


Fig. 7.22. Output of a Parkinson's Gait

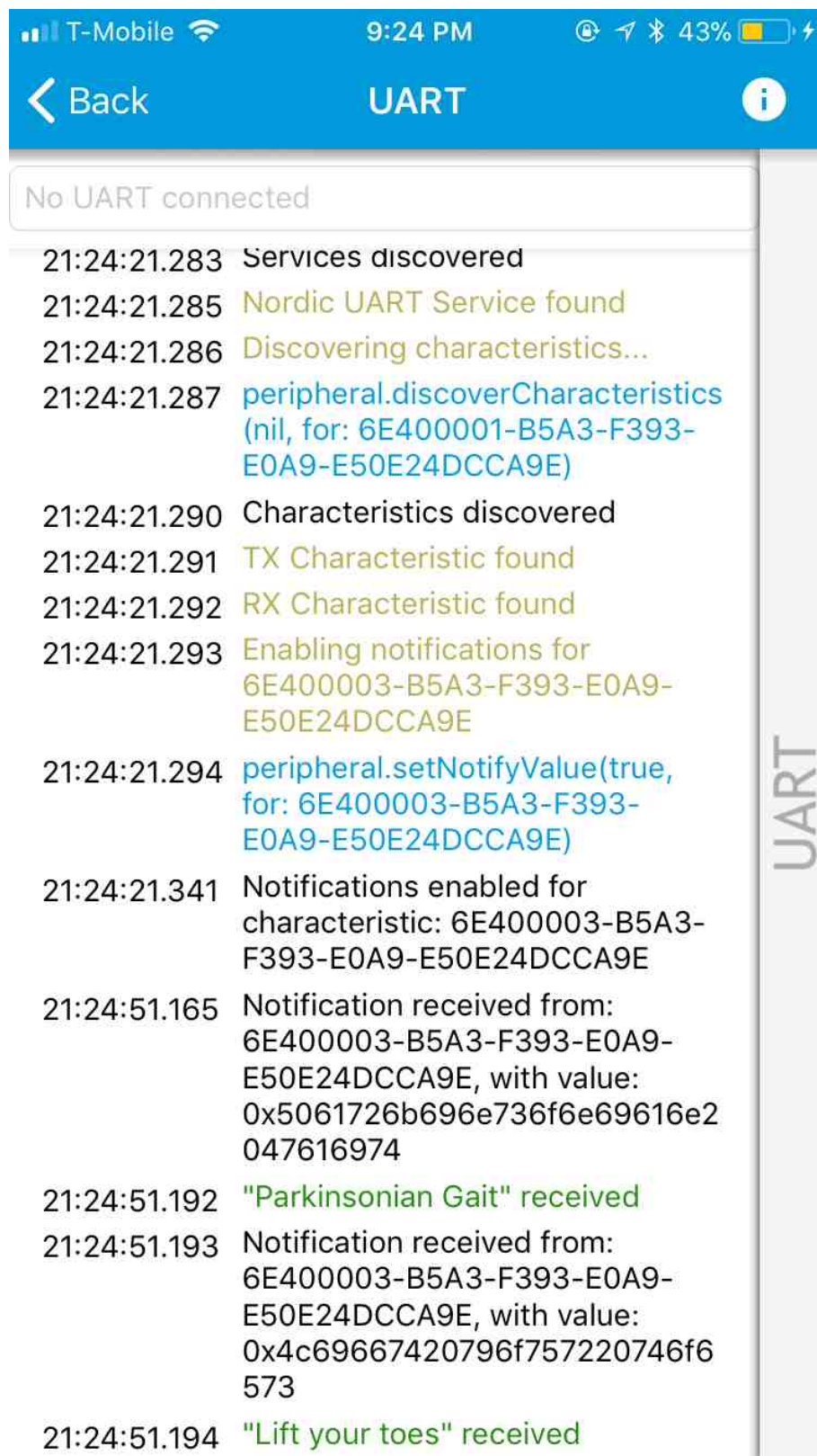
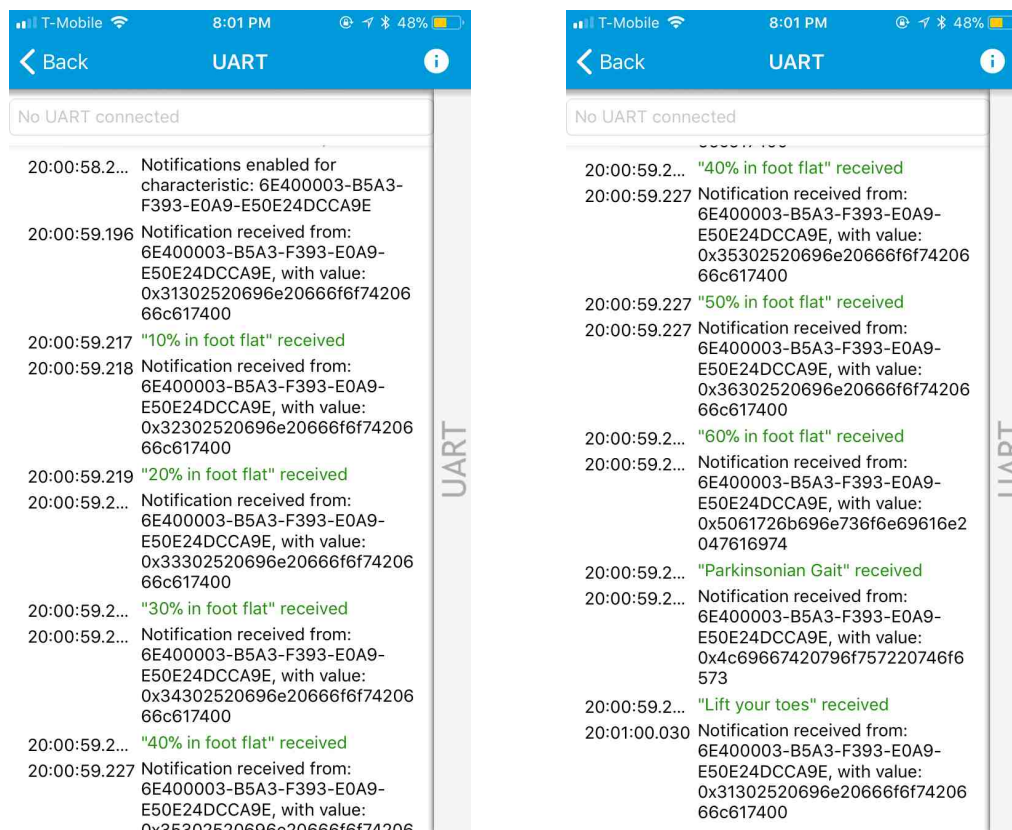


Fig. 7.23. Output of a Parkinson's Gait in BLE



(a) Parkinson's Gait Readings-1

(b) Parkinson's Gait Readings-2

Fig. 7.24. BLE results for Parkinson's Gait Diagnosis

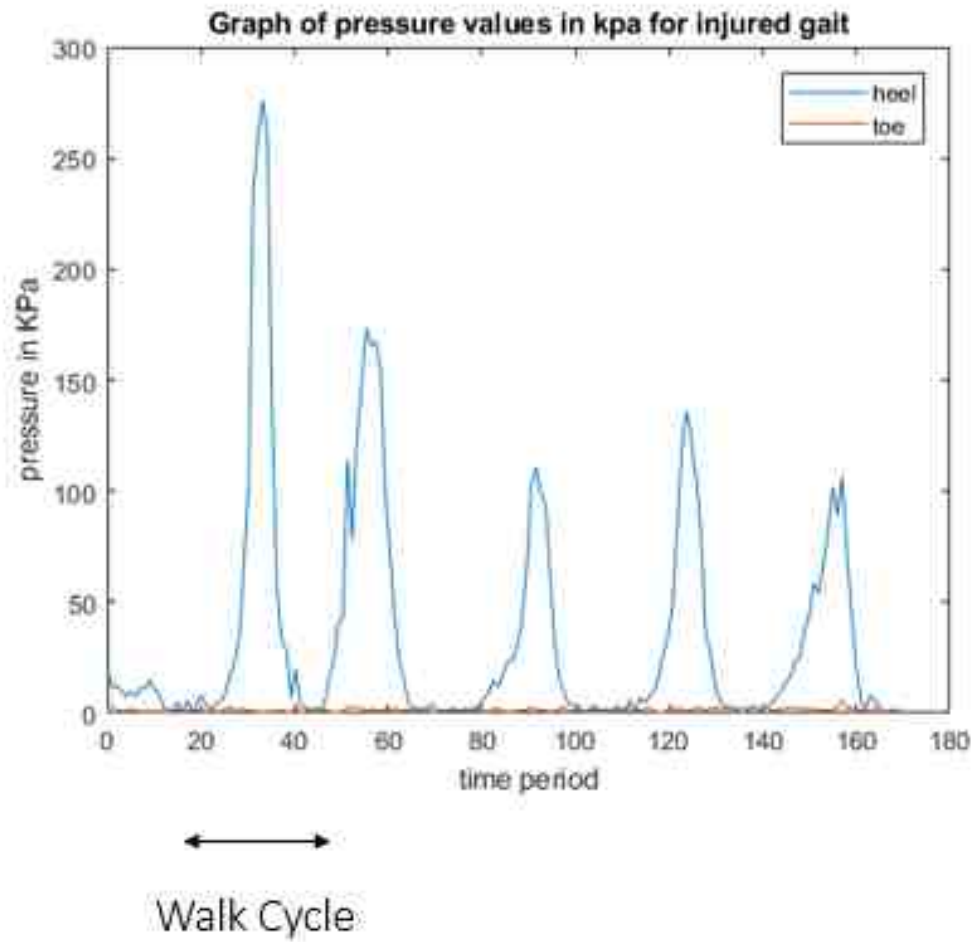


Fig. 7.25. Output of an Injured Gait

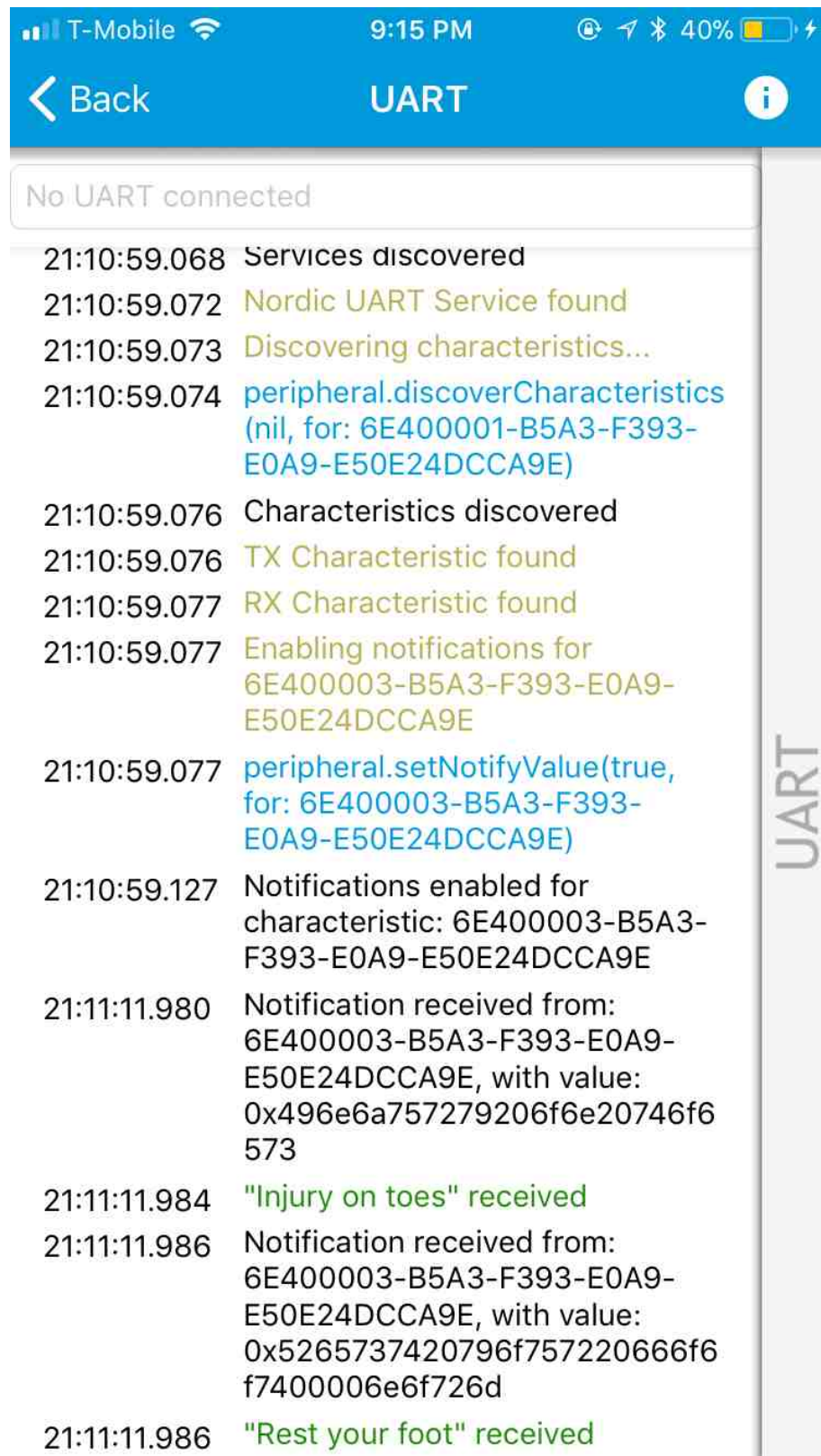
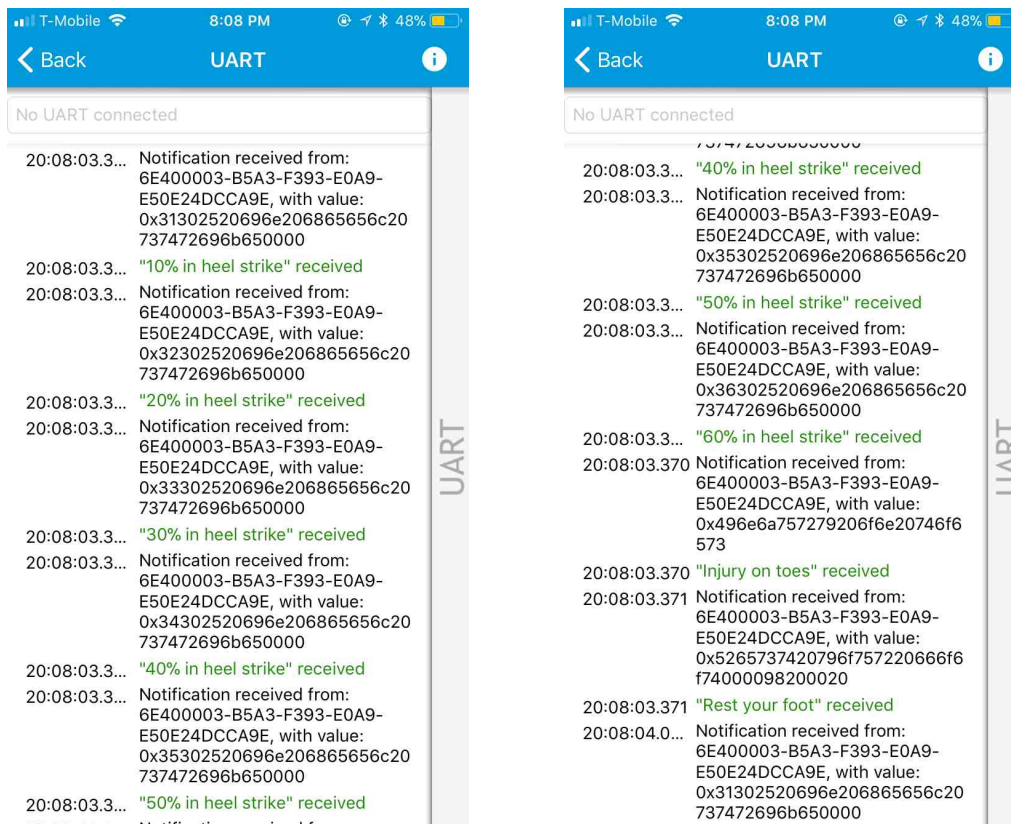


Fig. 7.26. Output of an Injured Gait in BLE



(a) Injury Gait Readings-1

(b) Injury Gait Readings-2

Fig. 7.27. BLE results for Parkinson’s Gait Diagnosis

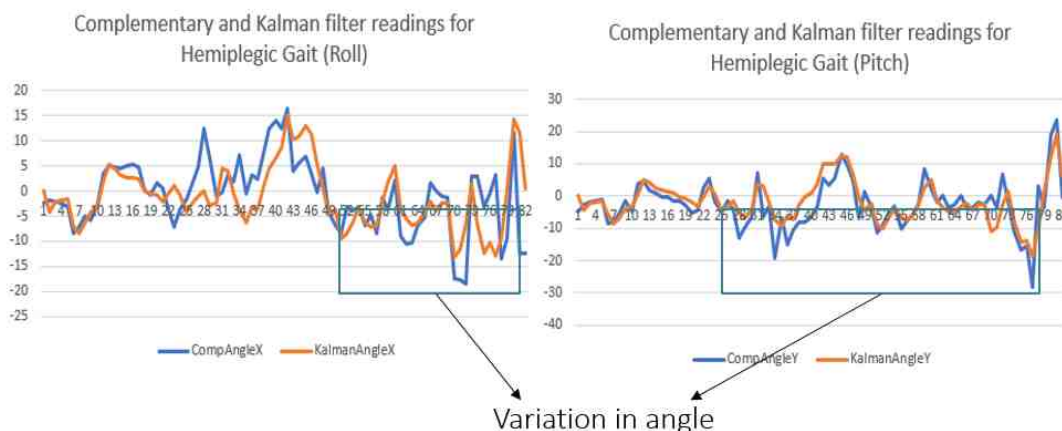


Fig. 7.28. Output of a Hemiplegic Gait

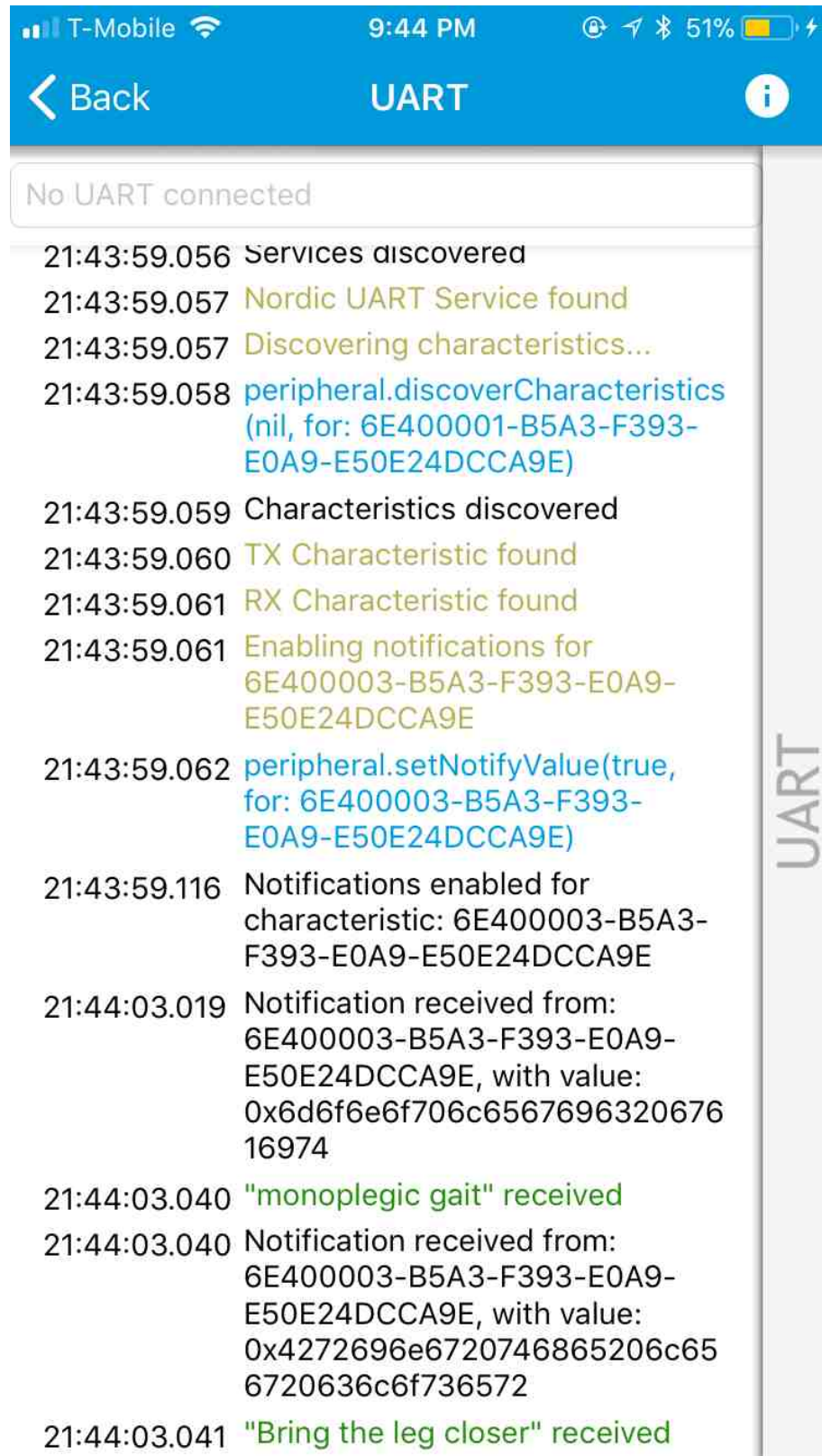


Fig. 7.29. Output of an Hemiplegic Gait in BLE

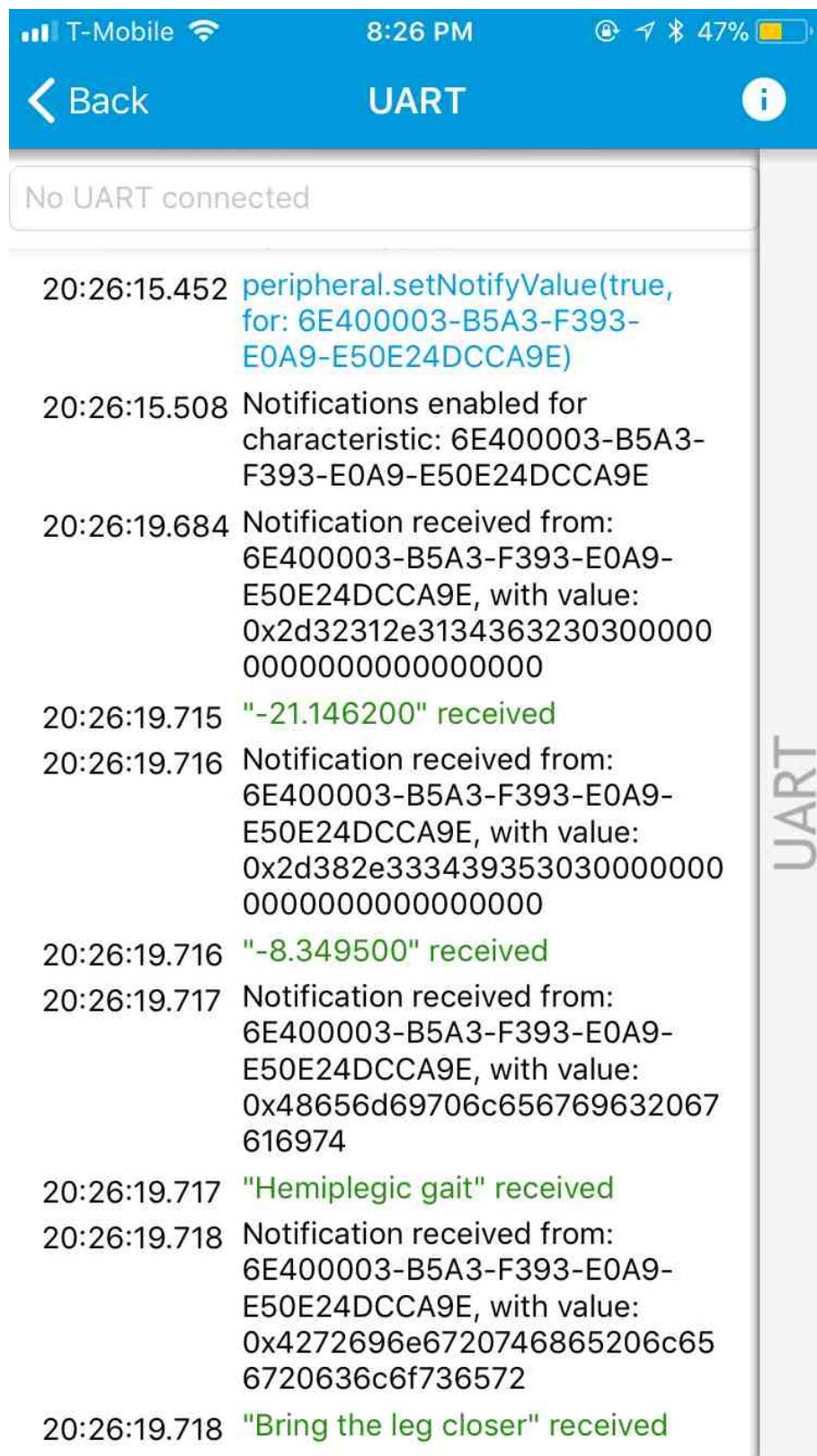


Fig. 7.30. BLE results for Hemiplegic Gait Diagnosis

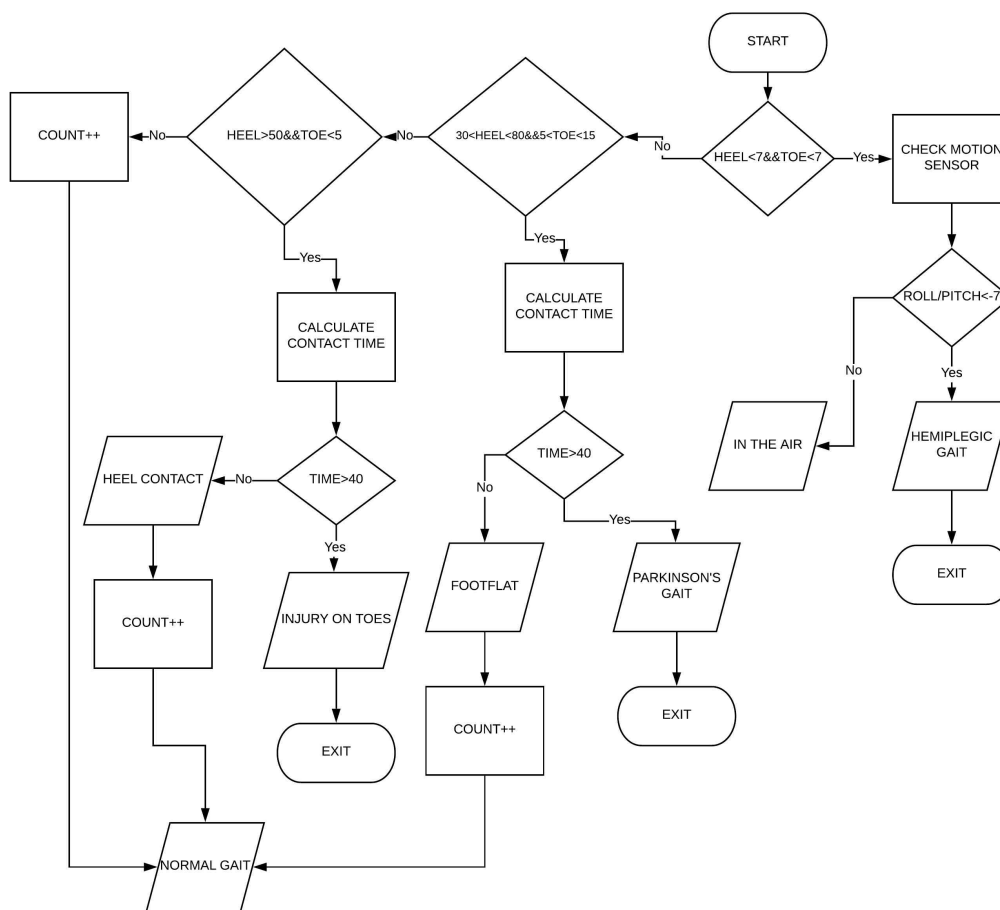
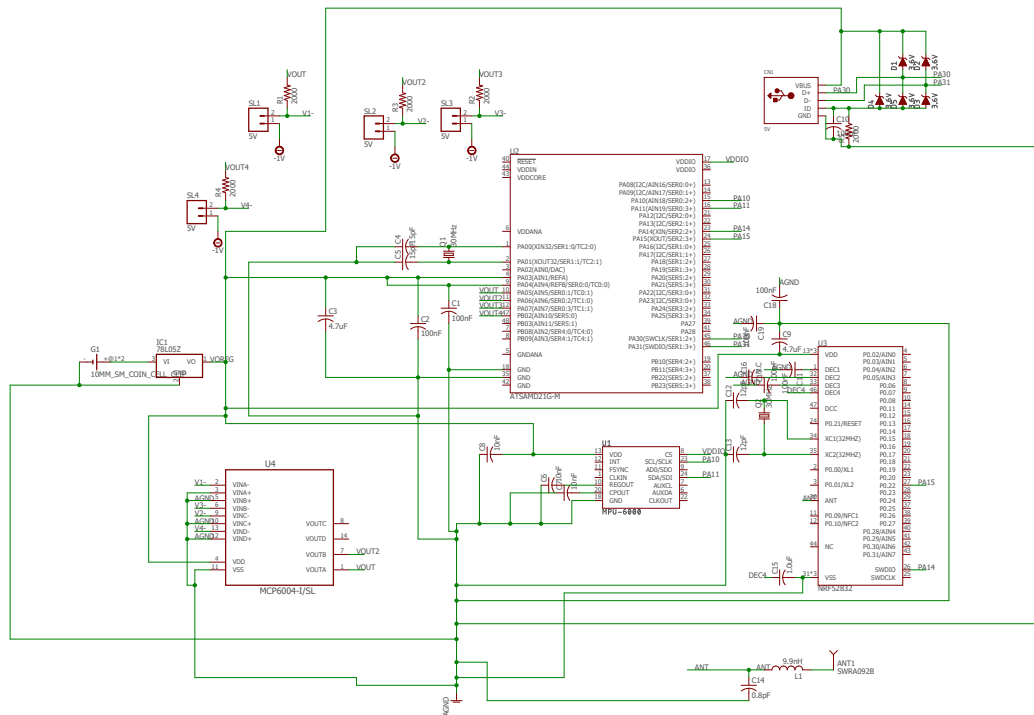


Fig. 7.31. Flowchart of the code



3/4/2018 8:49 PM f=0.80 C:\EAGLE 8.4.0\new thesis schematic.sch (Sheet: 1/1)

Fig. 7.32. Schematic of Board1

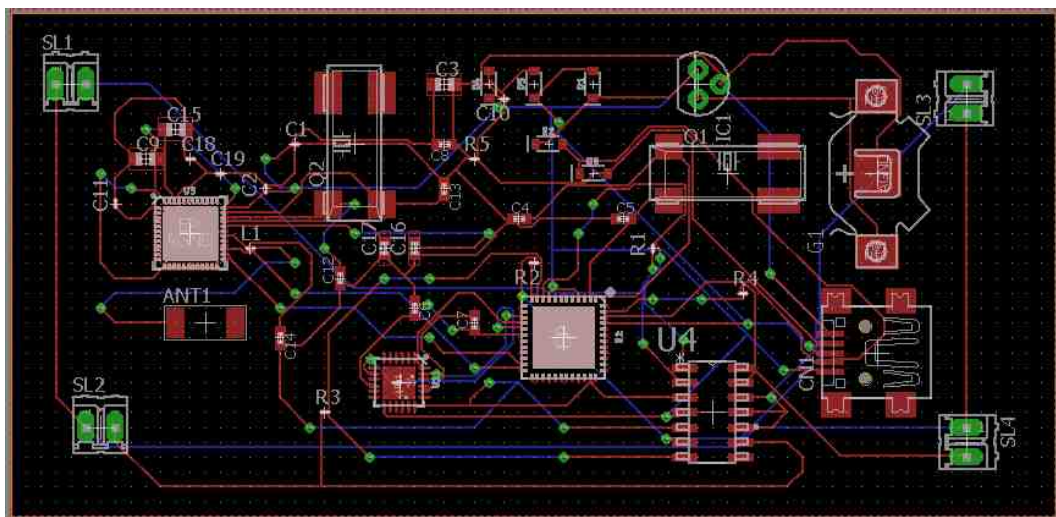
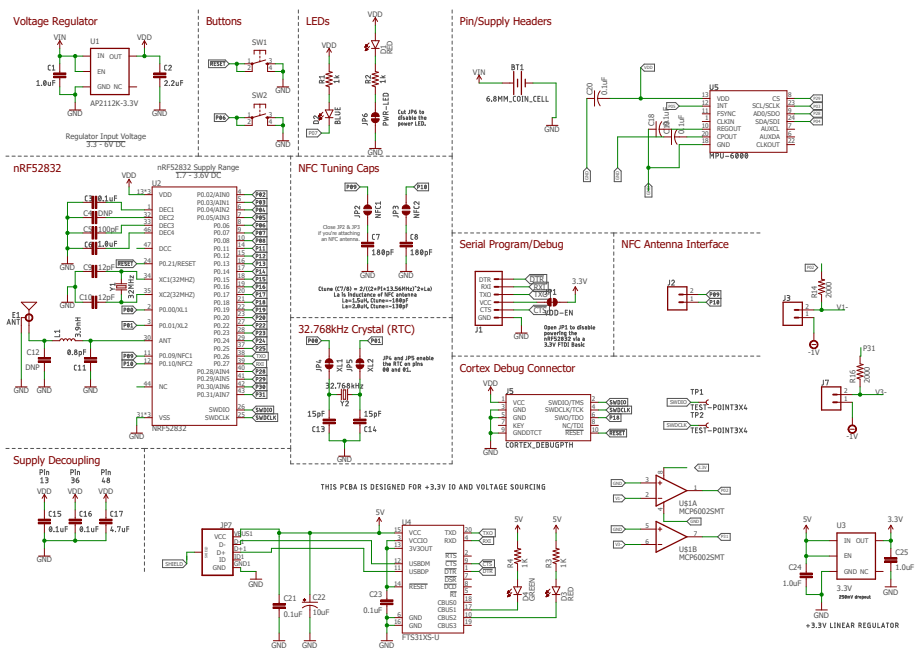


Fig. 7.33. Layout of Board1



3/4/2018 8:52 PM f=0.80 C:\EAGLE 8.4.0\lbr\NRFSELFMADE.sch (Sheet: 1/1)

Fig. 7.34. Schematic of Board2

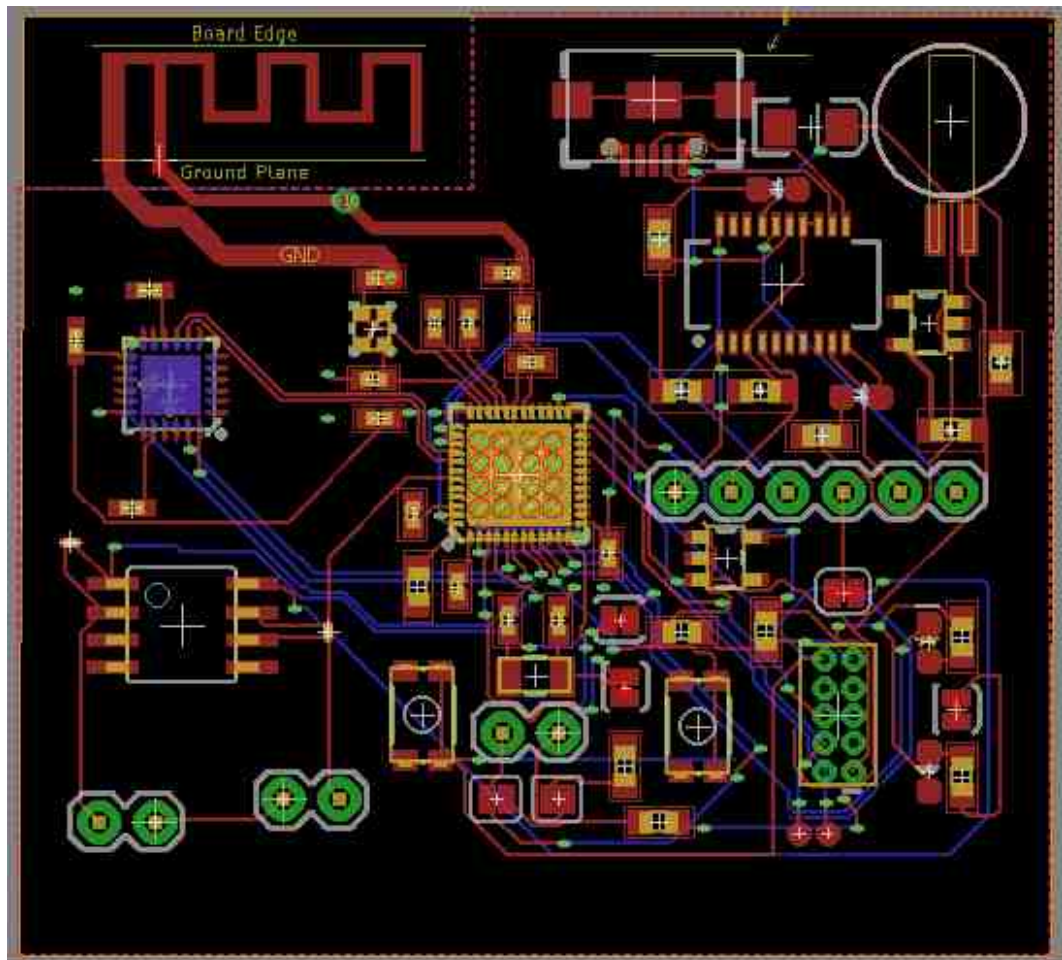
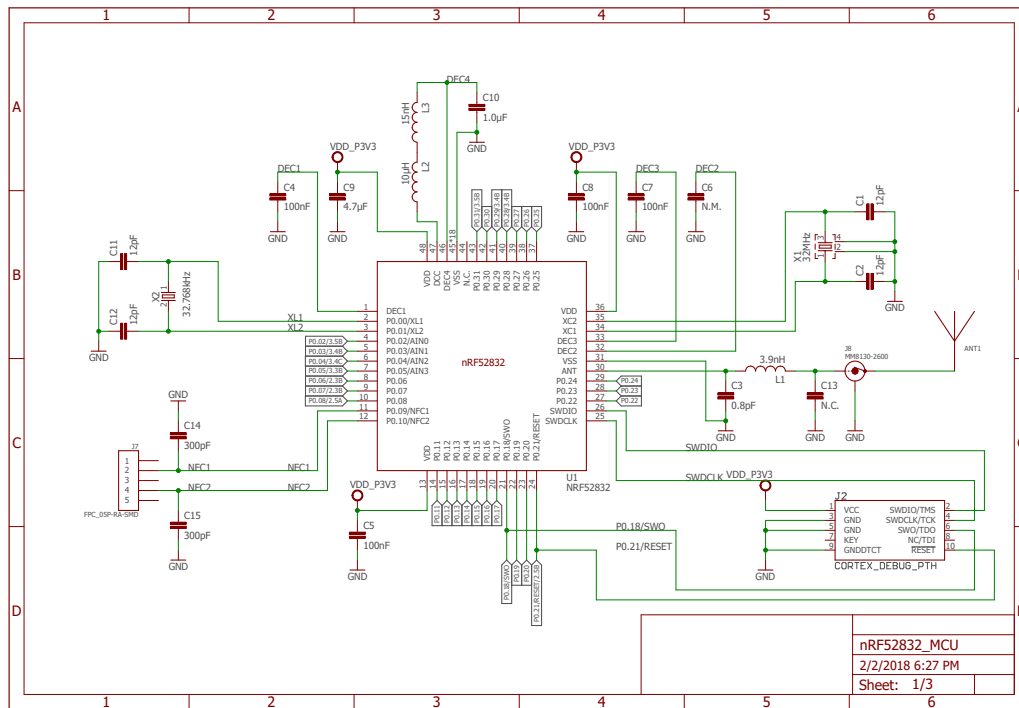


Fig. 7.35. Layout of Board2



3/4/2018 8:54 PM f=1.10 C:\Users\Vamsi Krishna\Downloads\nrf5-eagle-reference-design-master\nrf5-eagle-reference-design-master\nRF52832_MCU_Board\nRF52832_MCU.sch (Sheet: 1)

Fig. 7.36. Schematic of Board3 Part 1

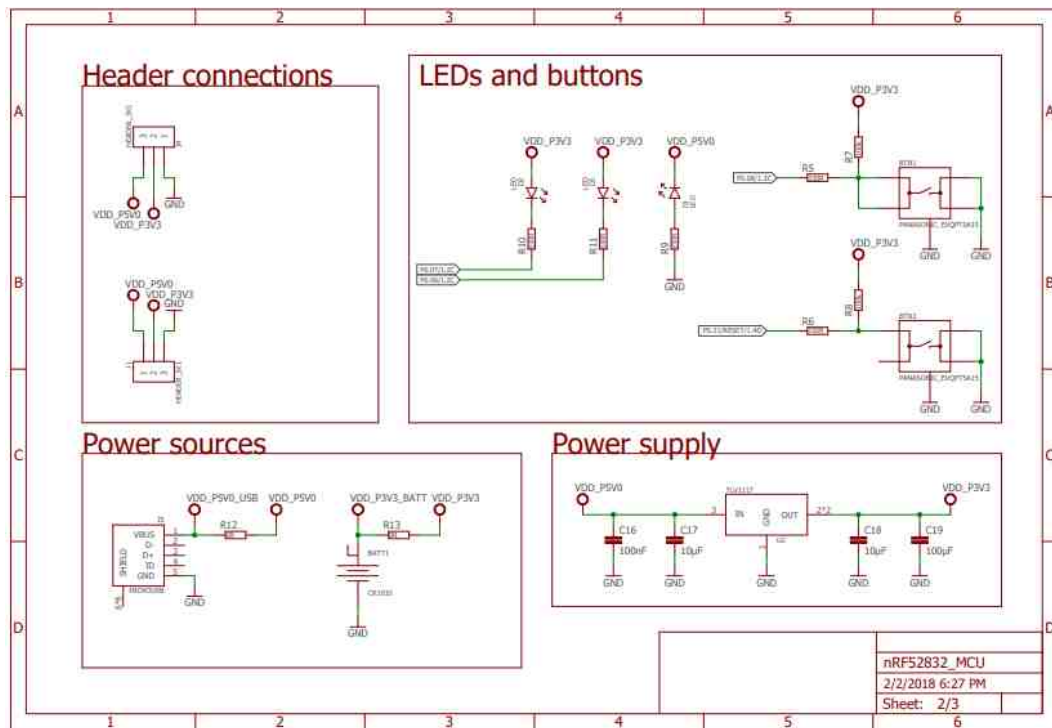


Fig. 7.37. Schematic of Board3 Part2

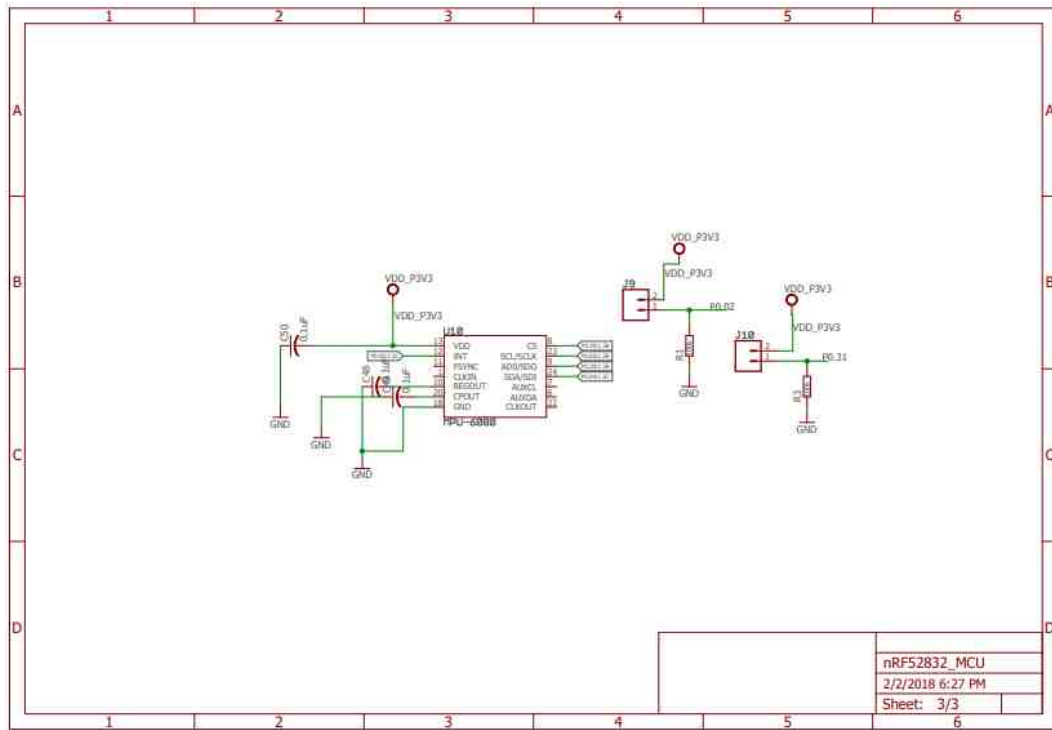


Fig. 7.38. Schematic of Board3 Part3

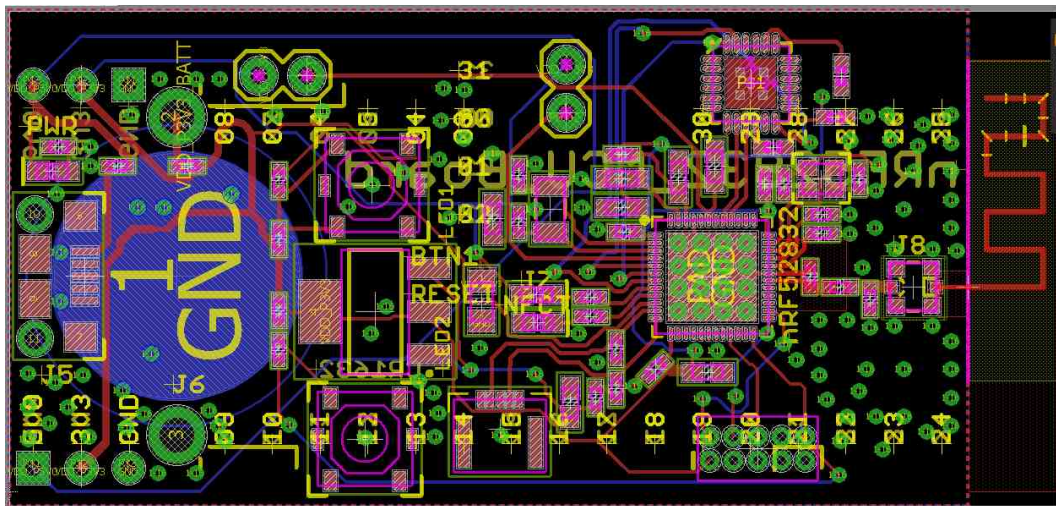


Fig. 7.39. Layout of Board3

```

COM3 - Tera Term VT
File Edit Setup Control Window Help
0.648074, 0.482119, -0.829819, -62.217842
0.248798, 0.168375, -0.282869, -62.477836
0.482119, 0.321379, -0.197361, -62.344896
0.248798, 0.483017, -0.255188, -62.484172
0.248798, 0.321379, -0.281102, -62.448118
0.321379, 0.248798, -0.285429, -62.517888
0.564874, 0.645291, -0.188617, -62.486764
0.482119, 0.483017, -0.192376, -62.297324
0.168375, 0.248798, -0.888858, -62.195683
0.482119, 0.321379, -0.091873, -62.456898
0.248798, 0.248798, -0.812548, -62.488318
0.168375, 0.482119, -0.142454, -62.247387
0.292665, 0.168375, -0.191718, -62.387649
0.321379, 0.483017, -0.199174, -62.365949
0.645291, 0.248798, -0.851188, -62.272484
0.482119, 0.482119, -0.174845, -62.228227
0.482119, 0.564874, -0.187274, -62.272856
0.321379, 0.248798, -0.195855, -62.332266
0.483017, 0.483017, -0.197571, -62.271574
0.168375, 0.248798, -0.838257, -62.357488
0.248798, 0.482119, -0.192715, -62.315287
0.248798, 0.248798, -0.863367, -62.197335
0.482119, 0.321379, -0.188735, -62.242681
0.482119, 0.321379, -0.288447, -62.265183
0.248798, 0.248798, -0.882887, -62.395640
0.321379, 0.168375, -0.173146, -62.121858
0.483017, 0.483017, -0.831899, -62.229777
0.483017, 0.321379, -0.185399, -62.293778
0.248798, 0.645291, -0.197576, -62.384892
0.168375, 0.248798, -0.189149, -62.338398
0.321379, 0.482119, -0.184718, -62.287771
0.482119, 0.564874, -0.181853, -62.245645
0.248798, 0.168375, -0.181879, -62.224617
0.321379, 0.168375, -0.173378, -62.117889
0.482119, 0.248798, -0.194788, -62.368672
0.888189, 0.482119, -0.125582, -62.399859
0.321379, 0.888888, -0.186798, -62.212435
0.248798, 0.482119, -0.188618, -62.288937
0.248798, 0.168375, -0.185318, -62.178161
0.483017, 0.888189, -0.182384, -62.286798
0.483017, 0.483017, -0.288931, -62.115212
0.321379, 0.483017, -0.192867, -62.397782
0.482119, 0.482119, -0.288937, -62.328832
0.482119, 0.248798, -0.188375, -62.384228
0.321379, 0.248798, -0.288128, -62.435246
0.321379, 0.168375, -0.172857, -62.153288
0.888888, 0.248798, -0.168827, -62.827683
0.248798, 0.483017, -0.281876, -62.392816
0.248798, 0.248798, -0.192576, -62.335586
0.248798, 0.482119, -0.181181, -62.455188
0.483017, 0.168375, -0.185547, -62.324683
0.321379, 0.888888, -0.188551, -62.188225
0.482119, 0.321379, -0.193233, -62.378768
0.321379, 0.168375, -0.286882, -62.431398
0.321379, 0.482119, -0.194354, -62.398148
0.248798, 0.482119, -0.198446, -62.338178
0.321379, 0.248798, -0.181151, -62.338528
0.564874, 0.321379, -0.196888, -62.344334

```

Fig. 7.40. Outputs from two FSRs and Motion sensor

8. DISCUSSION AND FUTURE WORK

Integrated sensor system

In general, the study here integrates a gait diagnostic system via integrated sensor system. This combines the sensor components, the DSP with filtering schemes. A second objective in this research work is to provide a working code which can detect various gaits and diagnose some medical issues.

Distinguished product from existing technology

Our approach in this work was distinguished from others in the sense that it provides an integrated hardware and software system that can provide accurate diagnosis as compared to existing systems. For instance, Pacilli et.al., [29] shows the recognition of a gait but not analyzing it. The work done provides the analysis and detection of various gaits, such as Hemiplegic gait, Parkinsons gait and an Injury gait. Different types of improper gaits were discussed and the possibility of detection the diagnosis via sensors were detailed.

Optimum power consumption

The power consumption is another important contribution of this thesis. The sleep mode turns off the BLE advertising so that the device lasts longer was proper approach to minimize the power consumption of the portable system. The components were selected to work under temperature and relative humidity present inside the shoe.

Custom PCB design

The secondary aim of this thesis is to design an integrated system via a working custom PCB. Custom PCB was designed and working function was verified. The BLE of the custom PCB advertises itself, which confirms it is working. The output can be seen via UART but it requires the development kit for the results to be visible. Additional research needs to be done on utilizing the maximum performance of the custom PCB.

8.1 Future Work

There is a scope of adding another PCB to the sole of the second shoe to detect other gaits as well. More FSRs can be added to the shoe to get a more refined output. Furthermore, flexible PCB can be designed to make it more user friendly. Usage of rechargeable batteries increases the productivity of the shoe. Sending the data to the cloud and storing it for future examination can be added at later stages of the research.

9. CONCLUSIONS

In this study we have demonstrated a working model with the approach for the software and hardware design of an integrated sensor system that will serve applications in medical sciences. The software approach may fit into other applications related to filtering and DSP embedded and IoT applications.

The future work that may need to build on this work may include adding another PCB to the sole of the second shoe to detect other gaits as well. More FSRs can be added to the shoe to get a more refined output. Furthermore, flexible PCB can be designed to make it more user friendly. Sending the data to the cloud and storing it for future examination can be added at later stages of the research.

In order to have portable system, a low power consumption using other approaches within low power chips may be pursued. Energy harvesting for energy saving power may be also considered. The practical model demonstrated in this work may need more testing verification with varieties of patient cases.

REFERENCES

REFERENCES

- [1] K. Watanabe and M. Hokari, "Kinematical analysis and measurement of sports form," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 36, no. 3, pp. 549–557, May 2006.
- [2] M. Stork, P. Weissar, K. Kosturik, J. Novak, and V. Zeman, "Use of accelerometer for walk-run or shot analysis for sport and rehabilitation purposes," in *2016 International Conference on Applied Electronics (AE)*, September 2016, pp. 261–264.
- [3] G. Worsley, "Home based gait training following knee and hip arthroplasty." last accessed 16 September 2016. [Online]. Available: https://www.kingsfund.org.uk/sites/default/files/media/W2B_GrahamWorsley.pdf
- [4] D. Hodgins and I. Mccarthy, "How measuring an older person's walking pattern can help keep them mobile personalised healthcare for mobility," August 2015.
- [5] P. K. Larsen, E. B. Simonsen, and N. Lynnerup, "Gait analysis in forensic medicine*," *Journal of Forensic Sciences*, vol. 53, no. 5, pp. 1149–1153, last accessed 16 September 2016. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1556-4029.2008.00807.x>
- [6] I. Bouchrika, M. Goffredo, J. Carter, and M. Nixon, "On using gait in forensic biometrics," *Journal of Forensic Sciences*, vol. 56, no. 4, pp. 882–889, 2011, last accessed 20 September 2016. [Online]. Available: <http://dx.doi.org/10.1111/j.1556-4029.2011.01793.x>
- [7] H. Lee, L. Guan, and J. A. Burne, "Human gait and posture analysis for diagnosing neurological disorders," in *Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101)*, vol. 2, September 2000, pp. 435–438 vol.2.
- [8] J. Cho, J. Choi, and J. Park, "A system design of gait training device for monoplegia patients," in *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, August 2008, pp. 4266–4269.
- [9] V. Kyriazis and C. Rigas, "A new telemetry system designed for use with a conductive walkway: results on children suffering from diplegia," in *Proceedings of the 22nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (Cat. No.00CH37143)*, vol. 1, 2000, pp. 567–569 vol.1.
- [10] P. K. Larsen, E. B. Simonsen, and N. Lynnerup, "Use of photogrammetry and biomechanical gait analysis to identify individuals," in *2010 18th European Signal Processing Conference*, August 2010, pp. 1660–1664.

- [11] S. Williams, M. and Brown, *Neurologic Differential Diagnosis: A Case-Based Approach*. Cambridge University Press: A. Ettinger and D. Weisbrot, 2014.
- [12] Physiopedia, “Physiopedia walk cycles,” 2018, last accessed 24 September 2016. [Online]. Available: <https://www.physio-pedia.com/images/b/b0/Figure2.jpg>
- [13] W. H. Organization, “Who global report on falls prevention in older age,” 2017, last accessed 1 October 2016. [Online]. Available: http://www.who.int/ageing/publications/Falls_prevention7March.pdf
- [14] G. R. Cutter, M. L. Baier, R. A. Rudick, D. L. Cookfair, J. S. Fischer, J. Petkau, K. Syndulko, B. G. Weinschenker, J. P. Antel, C. Confavreux, G. W. Ellison, F. Lublin, A. E. Miller, S. M. Rao, S. Reingold, A. Thompson, and E. Willoughby, “Development of a multiple sclerosis functional composite as a clinical trial outcome measure,” *Brain*, vol. 122, no. 5, pp. 871–882, 1999, last accessed 1 October 2016. [Online]. Available: + <http://dx.doi.org/10.1093/brain/122.5.871>
- [15] S. Mathias, U. S. L. Nayak, and B. Isaacs, “Balance in elderly patients: The get-up and go test,” vol. 34. *Arch Phys Med Rehabil*, 06 1986, pp. 119–126.
- [16] W. LI, R. Whipple, P. Amerman, and J. Tobin, “Gait assessment in the elderly: A gait abnormality rating scale and its relation to falls,” in *Journal of gerontology*, vol. 45, February 1990, pp. M12–9.
- [17] D. Marcin, “Human gait recognition based on ground reaction forces in case of sport shoes and high heels,” in *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, July 2017, pp. 247–252.
- [18] Y. Hirata, A. Hara, and K. Kosuge, “Passive-type intelligent walking support system ”rt walker”,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 4, September 2004, pp. 3871–3876 vol.4.
- [19] S. Dubowsky, F. Genot, S. Godding, H. Kozono, A. Skwersky, H. Yu, and L. S. Yu, “Pamm - a robotic aid to the elderly for mobility assistance and monitoring: a ldquo;helping-hand rdquo; for the elderly,” in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 1, 2000, pp. 570–576 vol.1.
- [20] S. Shibutani, H. Tamura, and K. Tanno, “Human tracking and estimation under occlusion using laser range scanner and accelerometer,” in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, October 2013, pp. 4196–4200.
- [21] Z. Lu, Y. Xu, Z. Dai, and B. Ma, “A gait recognition based on link model of infrared thermal imaging,” in *2016 2nd International Conference on Control Science and Systems Engineering (ICCSSE)*, July 2016, pp. 165–168.
- [22] K. Leyden, M. Koller, M. Niemier, J. Schmiedeler, and G. Cserey, “Kinect image processing by cnn algorithm for gait recognition,” in *CNNA 2016; 15th International Workshop on Cellular Nanoscale Networks and their Applications*, August 2016, pp. 1–2.

- [23] Y. Wahab and N. A. Bakar, "Gait analysis measurement for sport application based on ultrasonic system," in *2011 IEEE 15th International Symposium on Consumer Electronics (ISCE)*, June 2011, pp. 20–24.
- [24] Q. Sun, Z. Zhou, J. Jiang, and D. Hu, "Gait cadence detection based on surface electromyography (semg) of lower limb muscles," in *2014 International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI)*, September 2014, pp. 1–4.
- [25] M. Tsuruoka, R. Shibasaki, Y. Yasuoka, Y. Tsuruoka, and S. Murai, "Analysis of impulse response on walking stability by a synchronized system of two accelerometers," in *Proceedings 14th IEEE Symposium on Computer-Based Medical Systems. CBMS 2001*, 2001, pp. 348–353.
- [26] S. K. A. Kork, I. Gowthami, X. Savatier, T. Beyrouthy, J. A. Korbane, and S. Roshdi, "Biometric database for human gait recognition using wearable sensors and a smartphone," in *2017 2nd International Conference on Bio-engineering for Smart Technologies (BioSMART)*, August 2017, pp. 1–4.
- [27] S. Piriyaikulkit, Y. Hirata, and H. Ozawa, "Real-time gait event recognition for wearable assistive device using an imu on thigh," in *2017 IEEE International Conference on Cyborg and Bionic Systems (CBS)*, Oct 2017, pp. 314–318.
- [28] N. Hegde, T. Zhang, G. Uswatte, E. Taub, J. Barman, S. McKay, A. Taylor, D. M. Morris, A. Griffin, and E. S. Sazonov, "The pediatric smartshoe: Wearable sensor system for ambulatory monitoring of physical activity and gait," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 2, pp. 477–486, February 2018.
- [29] A. Pacilli, I. Mileti, M. Germanotta, E. D. Sipio, I. Imbimbo, I. Aprile, L. Padua, S. Rossi, E. Palermo, and P. Cappa, "A wearable setup for auditory cued gait analysis in patients with parkinson's disease," in *2016 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, May 2016, pp. 1–6.
- [30] N. P. Foundation, "Tip sheet parkinsons disease vs. parkinsonism," 2018, last accessed 1 January 2017. [Online]. Available: <http://www.parkinson.org/sites/default/files/Parkinsonism.pdf>
- [31] X. C. K. A. Benoit Mariani, Hossein Rouhani, "Quantitative estimation of foot-flat and stance phase of gait using foot-worn inertial sensors," *Gait & posture*, vol. 37, pp. 229–234, January 2013.
- [32] what-when-how In Depth Tutorials and Information, "Nervous system disorders (adult care nursing) part 5," 2018, last accessed 17 October 2016. [Online]. Available: <http://what-when-how.com/nursing/nervous-system-disorders-adult-care-nursing-part-5/>
- [33] R. Bachwchmidt, G. F. Harris, J. Ackman, S. Hassani, M. Carter, A. Caudill, K. Reiners, W. Olson, P. Smith, and J. Klein, "Quantitative study of walker-assisted gait in children with cerebral palsy: anterior versus posterior walkers," in *Pediatric Gait: A New Millennium in Clinical Care and Motion Analysis Technology*, 2000, pp. 217–223.

- [34] N. A. Foundation, “What is ataxia?” 2018, last accessed 30 February 2018. [Online]. Available: <https://ataxia.org/what-is-ataxia/>
- [35] Curetick, “Loss of co-ordination,” 2018, last accessed 4 March 2018. [Online]. Available: <http://curetick.com/vertigo-symptoms/ataxia/>
- [36] N. I. of Neurological Disorders and Stroke, “Myopathy information page,” 2017, last accessed 15 October 2017. [Online]. Available: <https://www.ninds.nih.gov/Disorders/All-Disorders/Myopathy-Information-Page>
- [37] D. I. J. Harrington, “Symptoms in the opposite or uninjured leg,” 2005. [Online]. Available: http://www.wsiat.on.ca/english/mlo/symptoms_leg.htm
- [38] M. Cameron and P. Hill, “Hyperkinetic disorder: assessment and treatment,” *Advances in Psychiatric Treatment*, vol. 2, no. 3, p. 94102, 1996.
- [39] Y. P. Huang, W. H. Liu, S. Y. Chen, and F. E. Sandnes, “Using type-2 fuzzy models to detect fall incidents and abnormal gaits among elderly,” pp. 3441–3446, October 2013.
- [40] M. Pura, B. Tambunan, and A. A. Iskandar, “Design of shoe-integrated running gait analysis device for normal people case study,” pp. 324–328, November 2013.
- [41] W. Srivises, I. Nilkhamhang, and K. Tungpimolrut, “Design of a smart shoe for reliable gait analysis using fuzzy logic,” pp. 834–838, August 2012.
- [42] Sensormag, “Types of pressure sensors,” last accessed 30 January 2018. [Online]. Available: <https://www.sensormag.com/components/fundamentals-pressure-sensor-technology>
- [43] M. F. Shaikh, Z. Salcic, and K. Wang, “Analysis and selection of the force sensitive resistors for gait characterisation,” pp. 370–375, February 2015.
- [44] A. Martinez-Nova, J. C. Cuevas-Garca, J. Pascual-Huerta, and R. Snchez-Rodriguez, “Biofoot in-shoe system: Normal values and assessment of the reliability and repeatability,” *The Foot*, vol. 17, no. 4, pp. 190 – 196, 2007.
- [45] Tekscan, “Flexiforce a201 datasheet,” last accessed 3 January 2017. [Online]. Available: <https://www.tekscan.com/sites/default/files/resources/FLX-A201-D.pdf>
- [46] W. Srivises, I. Nilkhamhang, and K. Tungpimolrut, “Design of a smart shoe for reliable gait analysis using state transition theory,” pp. 1–4, May 2012.
- [47] Y. Zhuang, J. Gong, D. C. Kerrigan, B. C. Bennett, J. Lach, and S. Russell, “Gait tracker shoe for accurate step-by-step determination of gait parameters,” in *2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, June 2016, pp. 13–18.
- [48] Invensense, “Mpu6000 and mpu6050 datasheet,” last accessed 5 January 2017. [Online]. Available: https://store.invensense.com/datasheets/invensense/MPU-6050_DataSheet_V3%204.pdf

- [49] Dronetrest, “Inertial sensor comparison mpu6000 vs mpu6050 vs mpu6500 vs icm20602,” 2016, last accessed 5 January 2017. [Online]. Available: <https://blog.dronetrest.com/inertial-sensor-comparison-mpu6000-vs-mpu6050-vs-mpu6500-vs-icm20602/>
- [50] Invensense, “Mpu9250 datasheet,” last accessed 5 January 2017. [Online]. Available: <http://www.invensense.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf>
- [51] AsahiKasei, “Ak8975/ak8975c 3-axis electronic compass,” January 2016, last accessed 5 January 2017. [Online]. Available: <https://www.akm.com/akm/en/file/datasheet/AK8975.pdf>
- [52] Invensense, “Mpu9150 datasheet,” last accessed 5 January 2017. [Online]. Available: <https://www.invensense.com/wp-content/uploads/2015/02/MPU-9150-Datasheet.pdf>
- [53] AsahiKasei, “Ak8963c 3-axis electronic compass,” January 2016, last accessed 5 January 2017. [Online]. Available: <https://www.akm.com/akm/en/file/datasheet/AK8963C.pdf>
- [54] K. Kong and M. Tomizuka, “Smooth and continuous human gait phase detection based on foot pressure patterns,” pp. 3678–3683, May 2008.
- [55] Y. Tao, X. Wu, Y. Yang, H. Qian, C. Zhang, and Y. Xu, “An intelligent shoe system: Evaluation of weight load approaches by gait analysis,” pp. 145–151, December 2012.
- [56] N. Pinkam and I. Nilkhamhang, “Wireless smart shoe for gait analysis with automated thresholding using pso,” pp. 1–6, May 2013.
- [57] I. Mileti, M. Germanotta, S. Alcaro, A. Pacilli, I. Imbimbo, M. Petracca, C. Erra, E. D. Sipio, I. Aprile, S. Rossi, A. R. Bentivoglio, L. Padua, and E. Palermo, “Gait partitioning methods in parkinson’s disease patients with motor fluctuations: A comparative analysis,” pp. 402–407, May 2017.
- [58] Digikey, “Comparing low-power wireless technologies (part 1),” 2017, last accessed 29 October 2017. [Online]. Available: <https://www.digikey.com/en/articles/techzone/2017/oct/comparing-low-power-wireless-technologies>
- [59] RFworld, “Nfc vs rfid vs bluetooth vs wifi-difference between nfc,rfid,bluetooth and wifi,” last accessed 29 October 2017. [Online]. Available: <http://www.rfwireless-world.com/Terminology/NFC-vs-RFID-vs-Bluetooth-vs-wifi.html>
- [60] Raytac, “Nfc vs rfid vs bluetooth vs wifi-difference between nfc,rfid,bluetooth and wifi,” last accessed 29 December 2017. [Online]. Available: <https://raytac.blog/2017/12/21/nordic-nrf51822-vs-nrf52832-vs-nrf52810-ble-solution-specification-comparison-chart/>
- [61] DevZoneNordic, “New features in nrf52,” last accessed 29 December 2017. [Online]. Available: <https://devzone.nordicsemi.com/b/blog/posts/new-features-in-nrf52>

- [62] TexasInstruments, “2.4-ghz bluetooth low energy system-on-chip,” last accessed 29 December 2017. [Online]. Available: <http://www.ti.com/lit/ds/symlink/cc2540.pdf>
- [63] DevZoneNordic, “Setting up keil mdk for nrf52832 example,” last accessed 29 October 2017. [Online]. Available: <https://devzone.nordicsemi.com/f/nordic-qa/17404/setting-up-keil-mdk-for-nrf52832-example>
- [64] NordicSemiconductors, “Softdevices,” 2017, last accessed 30 august 2017. [Online]. Available: <http://infocenter.nordicsemi.com/index.jsp?topic=%2Fcom.nordic.infocenter.softdevices52%2F dita%2Fnrf52%2Fsoftdevices.html>
- [65] InfocenterNordic, “nrf5x command line tools v1.0,” last accessed 30 October 2017. [Online]. Available: http://infocenter.nordicsemi.com/pdf/nRF5x_Command_Line_Tools_v1.0.pdf
- [66] Superuser, “Path environment variables,” last accessed 1 March 2018. [Online]. Available: <https://superuser.com/questions/284342/what-are-path-and-other-environment-variables-and-how-can-i-set-or-use-them>
- [67] NordicSemiconductors, “nrf52832 - product specification v1.0,” 2016. [Online]. Available: http://infocenter.nordicsemi.com/pdf/nRF52832_PS_v1.0.pdf
- [68] InfocenterNordicSemi, “Saadc successive approximation analog-to-digital converter,” last accessed 2018. [Online]. Available: <http://infocenter.nordicsemi.com/index.jsp?topic=%2Fcom.nordic.infocenter.nrf52832.ps.v1.1%2Fsaadc.html>
- [69] InfocenterNordic, “Peripheral drivers,” last accessed 4 November 2017. [Online]. Available: https://infocenter.nordicsemi.com/index.jsp?topic=%2Fcom.nordic.infocenter.sdk52.v0.9.2%2Fgroup__nrf__drivers.html
- [70] Adafruit, “Using an fsr,” last accessed 7 November 2017. [Online]. Available: <https://learn.adafruit.com/force-sensitive-resistor-fsr/using-an-fsr>
- [71] X. Wu, X. Chen, Y. Duan, S. Xu, N. Cheng, and N. An, “A study on gait-based parkinson’s disease detection using a force sensitive platform,” pp. 2330–2332, November 2017.
- [72] A. M. Howell, T. Kobayashi, H. A. Hayes, K. B. Foreman, and S. J. M. Bamberg, “Kinetic gait analysis using a low-cost insole,” *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 12, pp. 3284–3290, December 2013.
- [73] K. D. Das, A. J. Saji, and C. S. Kumar, “Frequency analysis of gait signals for detection of neurodegenerative diseases,” pp. 1–6, April 2017.
- [74] Github, “nrf5-mpu-examples,” last accessed 7 November 2017. [Online]. Available: <https://github.com/Martinsbl/nrf5-mpu-examples>
- [75] PieterJan, “Reading a imu without kalman: The complementary filter,” last accessed 14 December 2017. [Online]. Available: <http://www.pieterjan.com/node/11>

- [76] Bilgin'sBlog, "Kalman filter for dummies a mathematically challenged man's search for scientific wisdom," last accessed 14 December 2017. [Online]. Available: <http://bilgin.esme.org/BitsAndBytes/KalmanFilterforDummies>
- [77] DevZoneNordic, "Introduction to bluetooth low energy and bluetooth low energy development (video tutorial)," last accessed 1 January 2018. [Online]. Available: <https://devzone.nordicsemi.com/tutorials/b/getting-started/posts/introduction-to-bluetooth-low-energy-and-ble-devel>
- [78] NordicSemiconductors, "Power consumption calculator," last accessed 30 March 2018. [Online]. Available: <https://devzone.nordicsemi.com/power/>
- [79] Mouser, "Cr2450 coin cell battery," last accessed 30 March 2018. [Online]. Available: <https://www.mouser.com/datasheet/2/315/AAA4000C337-947662.pdf>
- [80] Digikey, "Battery life calculator," last accessed 30 March 2018. [Online]. Available: <https://www.digikey.com/en/resources/conversion-calculators/conversion-calculator-battery-life>
- [81] Blitzresults, "Shoe size conversion: This easy tool helps you through the size-chaos," last accessed 1 January 2018. [Online]. Available: <https://www.blitzresults.com/en/shoe-size/>
- [82] AutoDesk, "The top 10 pcb routing tips for beginners," last accessed 5 February 2018. [Online]. Available: <https://www.autodesk.com/products/eagle/blog/top-10-pcb-routing-tips-beginners/>
- [83] A. A. Circuits, "Trace resistance calculator," last accessed 30 March 2018. [Online]. Available: <https://www.allaboutcircuits.com/tools/trace-resistance-calculator/>
- [84] SierraCircuits, "Gerber file format and odb++ file format," last accessed 5 February 2018. [Online]. Available: https://www.protoexpress.com/content/gerber_extensions.jsp
- [85] MicroChip, "Atsamd21 datasheet," last accessed 24 August 2017. [Online]. Available: <http://ww1.microchip.com/downloads/en/DeviceDoc/40001882A.pdf>
- [86] F. Deng, T. Gong, H. Jin, and S. Du, "Research on factors influencing temperature and relative humidity inside of shoes," January 2013.
- [87] Elgertagroup, "Moisture/reflow sensitivity classification for non-hermetic solid state surface mount devices," last accessed 14 March 2018. [Online]. Available: http://www.elgertagroup.com/files/tinytmc/files/Standart_IPC_JEDEC%20J-STD-020C.pdf