DE-40 (DC&SE)

YEAR 2022

NUST COLLEGE OF

ELECTRICAL AND MECHANICAL ENGINEERING

Gait Assessment Device

A PROJECT REPORT

DE-40 (DC&SE)

Submitted by

GC RASHID ALI

NS SHAKEEL HASSAN

NS RAZA ALVI

NS MASOOM ALI

BACHELORS

IN

COMPUTER ENGINEERING

YEAR

2022

PROJECT SUPERVISOR

DR. AHSAN SHAHZAD

COLLEGE OF

ELECTRICAL AND MECHANICAL ENGINEERING

PESHAWAR ROAD, RAWALPINDI

Gait Assessment Device

A PROJECT REPORT

DEGREE 40

Submitted by

GC RASHID ALI

NS SHAKEEL HASSAN

NS RAZA ALVI

NS MASOOM ALI

BACHELORS

IN

COMPUTER ENGINEERING

Year

2022

PROJECT SUPERVISOR

DR. AHSAN SHAHZAD

COLLEGE OF

ELECTRICAL AND MECHANICAL ENGINEERING

PESHAWAR ROAD, RAWALPINDI

DECLARATION

We hereby declare that no portion of the work referred to in this Project Thesis has been submitted in support of an application for any other degree or qualification of this for any other university. If any act of plagiarism found, we are fully responsible for every disciplinary action taken against us depending upon the seriousness of the proven offence.

COPYRIGHT STATEMENT

ACKNOWLEDGEMENTS

First of all, Alhamdulillah, that our FYP is finally made and all Thanks to Allah for giving us the strength and moral to keep pushing forward and helping us on each and every step of the way.

Secondly, we would like to offer heartily thanks our supervisor, Dr. Ahsan Shahzad, who helped us a lot, tremendously, on each and every single issue, who’s help ad guidance became a source of strong determination for us. Thank You, sir`s you played a great role in our lives, one that we can never forget.

And lastly, we would like to thank our parents and friends, without whose unimaginable support and constant motivation, we might not have been able to complete our Final year project. They played an unparalleled role throughout our journey and we are eternally thankful to them. Their constant support, motivated us to do more than we ever realized and they inspired new hope in us, when we found none in ourselves.

ABSTRACT

The manner of walking style of a person is called gait. Gait analysis hasn't been taken serious, despite serious health issues are arising related to Gait. Gait measurement is carried out only in certain hospitals, by specific doctors and using specific machines, & only serious issues are tackled. Also the gait analysis is very expensive, that not everyone can get it tested. Already present gait measurement devices are heavy and big machines, or the doctors measure it manually.

We are going to give one solution to all of the above problems by making a Wearable Gait Device. User will wear it on his leg/shank. The device will analyse the gait of the person, do some signal processing, calculate gait parameters and give a real time feedback to the user through Bluetooth app. The data will be stored in a database. With the passage of time, if gait parameters (like step height, walking speed, leg spin angle, etc.) are getting worse or better, it will prompt a message to the user in the mobile app.

This report will give you an insight about the work we have done in order to make the Gait device. We will start with some introduction. The designing of such a device which is very important, but not available was our main motivation. Selection of hardware like nRF52832 SoC and MPU-6050 IMU was a good decision. They will be discussed in detail.

The current available techniques aren't affordable by everyone, so we have tried to make the gait analysis device available to everyone making it very economical and easy to use.

We first designed a firmware for our SoC. The firmware consisted of two main parts, i.e. one being reading data from IMU and process it, and second to use Bluetooth to send it to Mobile Phone Application. We finally designed our own hardware (Gait device) and a mobile app to display the gait parameters received from the device.

Table of Contents

Table of Figures

Chapter 1: Introduction

1.1 Introduction

The term "gait" refers to the locomotion of animals. In brief, the manner of walking of a person or any animal, or the movement of the limbs (legs or arms of an animal or a person) of an animal is called gait.

More generally, gait is a very intricate movement that heavily depends on certain, but very important parts of the body. Those parts include brain, muscles, sensory nerves, eyes, joints, backbone, and ears. In addition, gait relies on intricate interplay of the crucial parts of human biological systems, like cardiorespiratory system, musculoskeletal system and mainly nervous system.

Gait analysis can give us information about many crucial changes in the body. To be more specific, when a person is affected by any disease, it directly affects the walking manner of that person. Even a small headache can alter the walking style of the person or an animal. Moreover, some of the other severe diseases that affect gait are Dementia (Alzheimer's disease – Loss of memory), Parkinson's Diseases (PD – Unintended shaking or stiffness), Brain Hemorrhage (Bleeding in bran tissues), Spinal Cord Compression, Myopathy(Neuromuscular Disorder), Vertigo(Dizziness), Multiple Sclerosis (MS – Attacks on CNS). [1]

During the initial stages of these diseases, they have a mild effect on the walking pattern of a person. These mild changes in gait go unnoticed by the person. When the conditions becomes much worse, the gait gets affected severely. Now, the person can notice his walking manner has changed. But the disease has now become more severe, and can be hard to treat in this stage.

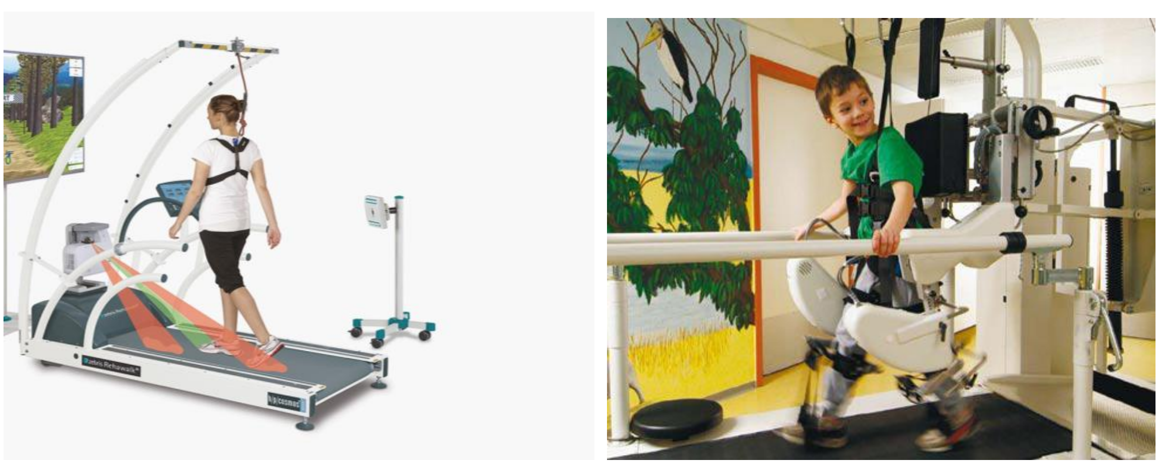


Figure 1 Rehawalk (left) and Locomat (Right)

Moreover, even in the later stages, if the person wants to confirm such diseases, he/she has to visit specific and very rare rehabilitation centers to analyze his/her gait. Certain devices are present in a few rehabilitation centers that are very heavy and non-portable. The person has to spend a lot of money to get his gait analysis in those devices as they are very expensive. Appointments has to be taken prior to visiting the center because the few devices may already be engaged for days. During initial stages, the operator/physician don't even entertain the patients. They only prefer latter stages patients. Some of the devices are Rehawalk [See Fig. 1 (left)] and Locomat [See Fig. 1 (right)].

After going through all these issues, we committed to give one solution to all. We have designed a device that will analyze the walking pattern of a person and give real time feedback. Gait features like gait speed, swing time, cadence and step time will be extracted and sent to the mobile app through Bluetooth.

A user friendly app is also designed that will receive gait features from the device through Bluetooth and plot them. From the plots, user can easily analyze the history of his gait feature records and compare them with present values.

1.2 Motivation

According to WHO, 684,000 people die from falls across the worldwide every year. All the diseases that affect gait are indirect causes of these deaths due to falls. Because those diseases alter the walking manner, and ultimately the person may lose balance and fall.

According to a survey conducted by CDC (Centre for Disease Control and Prevention), the death rates from falls are increasing each year. If the same increment continues, then by 2030, there will be seven deaths every hour from falls.

Following graph shows the statistics of the survey conducted by CDC: [2]

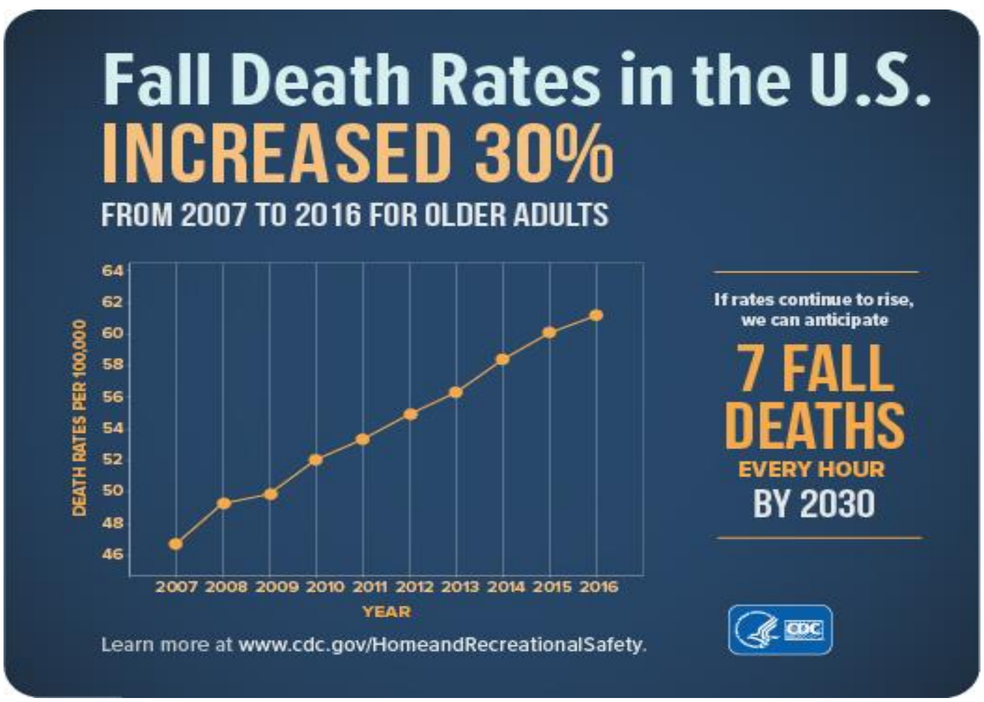


Figure 2 Survey conducted by CDC for fall rates in the US

Deaths from falls aren't taken serious and aren't studied in detail. These are mainly due to the alteration in normal gait parameters of a person. Despite there is plenty of research in the field of gait impairments, but till now no any device is designed that will be portable, cheap, user friendly and time saving.

Another major problem associated with the pre-mentioned diseases is that the tests of those diseases are very expensive, time consuming and complex. For instance, certain neurological tests are required to detect Dementia. In addition to being expensive, these tests are also not available in many health care centers. Only a few state-of-art hospitals offer these test. Also these tests are recorded in latter stages. It is very hard to diagnose such diseases in early stages. As we have already mentioned that these diseases indirectly affect the manner of walking, so we can indirectly test them by testing gait impairments.

So we got motivated to design a small (portable) device with all the required features. Along with it, we intended to design a user-friendly mobile app that will receive the processed data from the device and plot with respect to time. The user can now assess his gait parameters without the aid of any operator or a physician, and he or she can easily notice the mild changes arising in the walking manner.

1.3 Scope

The size of hardware is decreasing day by day as the technology is advancing tremendously. People prefer to use devices with minimum size hardware as possible, which are easy to carry and user-friendly. In this regard, we have designed a device consisting of all these features. The device is very small, wearable and gives real-time feedback to the user through the Mobile Application.

The device we have designed is consisting of an Inertial Motion Sensor (IMU) and a modern SoC (System on Chip), with a Low Energy Bluetooth enabled technology. So our device's functionality is mainly associated with the locomotion of the person. To get the readings, the person will attach the device with his shank and walk normally to get data from IMU that will be processed in the SoC and will be sent to mobile phone app for plotting.

Since the device is designed for analysis of gait features, so it must be used while walking normally. Acceleration values are taken for extracting gait features, which change with the change in walking manner. So, it must be used for assessment for normal gait.

The following goals can be used to determine the project's scope:

- IMU is used to get acceleration values from the shank of the user.
- SoC is used to receive the raw values from IMU.
- The received values are calibrated and converted in terms of gravitational acceleration.
- The acceleration values are further process to extract/calculate gait features
- The features are sent to Mobile Application using Bluetooth.
- Mobile app receives the data and stores the values in local database.
- Stored data in the database is plotted in the app.
- User can now assess his gait features from the plots.

1.4 Structure

Following is the structure of the report ahead:

- Chapter 2, the chapter is the most important one which will explain the literature used in the project. Different terms, protocols and other things will also be discussed
- Chapter 3, this chapter explains the current devices and techniques present in the market to assess gait parameters, and their cons.
- Chapter 4, it gives an insight about the hardware we have used, their details, working and pros
- Chapter 5, explains the firmware we have developed for the SoC for both IMU data acquisition and BLE Services. It also explains about the mobile app development
- Chapter 6, It briefly concludes our project and the future possible developments in the gait assessment field

Chapter 2: Literature Overview

There are many terms related to the literature of gait. Without knowing them, we won't be able to understand the project completely. Words and phrases associated with gait are defined and explained in this chapter.

2.1 Introduction

Gait:

The manner of walking of a person or any animal, or the movement of the limbs (legs or arms of an animal or a person) of an animal is called gait. [3]

Gait Assessment:

The process of extracting gait features from the walking of person is called Gait Assessment.

Important Terms:

Some of the important terms that will be used in this report are defined below: [4]

- **TO:** It means Toe-Off. The instance at which the user just lifts his foot from the ground [see fig. 4].
- **HS:** It mean Heal-Strike. It is the instance at which the user's heel just touches the ground after toe-off [see fig. 4].

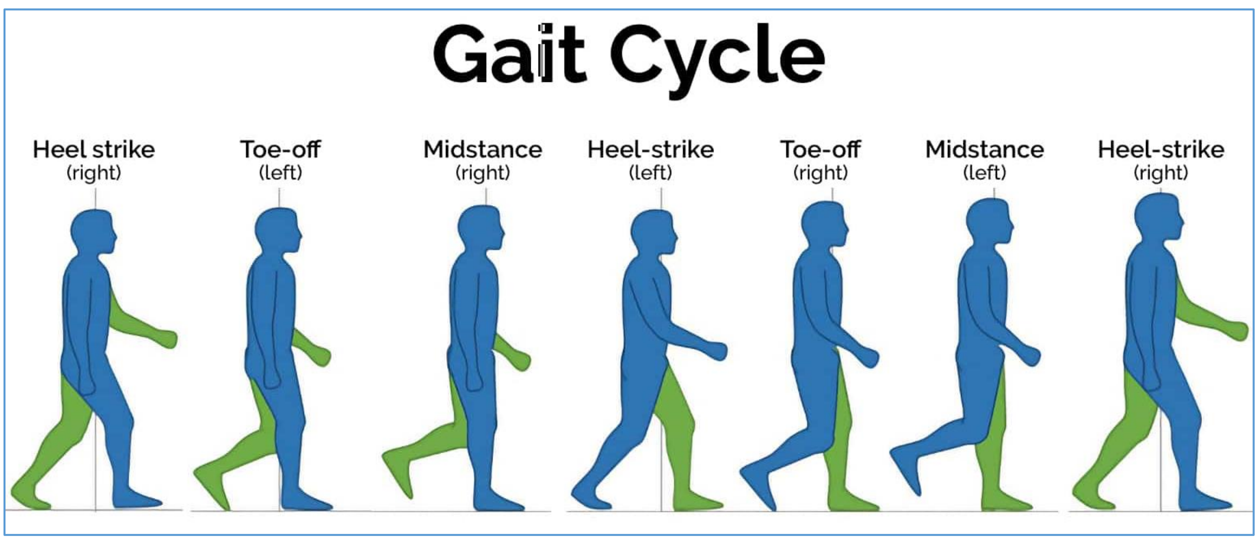


Figure 3 TO, HS and Gait Cycle

Gait Cycle:

It is the time elapsed from the HS of one foot to the next HS of the same foot. In other words, the time taken from placing one foot on the ground to placing the same leg on the ground in the next step is the gait cycle. Most simply, it is the time taken for two steps [see fig. 3].

Gait Features:

These are the spatiotemporal characteristics associated with the walking manner of a person. Some of the features are: [5]

- **Cadence:** It is the total number of steps taken in a minute.
- **Swing Time:** It is the time elapsed in one gait cycle from TO of one leg to HS of the same leg.
- **Stride Time:** The time elapsed from HS of one leg to the HS of the same leg in a gait cycle.
- **Gait Speed:** It is the average distance covered in one second. It can be calculated instantaneously or step-by-step.

2.2 Rehabilitation

Rehabilitation is a set of treatments intended to improve functioning and lowering disability in people with health issues as they interact with their environment.

In gait rehabilitation, patients have to visit rehabilitation centers. Either the patient’s walking style is assessed manually or some heavy machines like Rehawalk and Locomat are used. As the name refers, rehabilitation does not cure the disease. It just helps you enhance the functionality and train your brain and body to improve your functions.

Physical therapy is currently the primary form of gait rehabilitation, with robots being utilized only seldom. All of the physical therapies aim to improve effective ambulation, with ground gait training being the most important. Apart from the technique used, all techniques require carefully designed warm-up exercises. On a regular surface, direct interaction of the lower limbs position during gait, as well as assisted walking exercise on the ground. [6]

2.3 Importance of Our Device:

As we have seen many methods that can analyze the gait of any person. But all of those methods are very complex, expensive and non-portable. The device we have designed has many useful features that are explained below:

- **Early Stage Detection:** Since, other gait assessment methods are adopted once the condition becomes much worse or in the latter stages. The device we have designed will be used to detect any mild changes in the gait parameters in initial stages.

- **Portable:** Devices like Rehawalk and Locomat (discussed in chapter 3) are very big in size and consume a lot of space. Our device is very small, that can be attached on the shank and can be carried anywhere.

- **Economical:** Whereas other gait assessment techniques are very much expensive, our device is affordable for everyone. Anyone who wants to see his gait parameters can easily buy this device.

- **Real-time Feedback:** Using the low energy Bluetooth feature of the SoC used, gait parameters are continuously sent to mobile app, which the user can monitor in real-time with ease.

2.4 System-on-Chip (SoC)

A system-on-chip (also called System-on-a-Chip or just SoC) is a sort of Integrated Circuit (IC) that has a whole computing system or a whole electronic system on a single platform. The main components in an SoC are Central Processing Unit (CPU), input/output devices on chip, secondary-storage interfaces and peripherals. Other components on an SoC may be Graphic Processing Units (GPUs), radio-modems, Bluetooth, NFC, Wi-Fi and many other. [7]

High-performance SoCs are frequently combined with specialized and physically independent memory and secondary storage chips, which can either be stacked on top of the SoC in a package on package (PoP) arrangement or put close to the SoC. Separate wireless modems can also be used by SoCs.

SoCs are integrated circuits, as opposed to the typical motherboard-based PC design, which separates elements by function and connects them through a central interface circuit board. A motherboard houses and connects removable or replaceable components, whereas a SoC combines all of these elements onto a single ic. A SoC typically integrates a Central processing unit, graphics and memory interfaces, hard-disk and USB connectivity, RAM and ROM and secondary storage, and/or their controllers on a single circuit die, whilst these components would be attached to a motherboard as individual units or expansion cards.

Small, more complicated consumer electrical gadgets employ system-on-a-chip technology. Some of these gadgets have more processing power and memory than a conventional desktop computer from ten years ago. SoC-equipped Nano-robots (robots with minuscule dimensions) might one day operate as programmable antibodies to fight illnesses that were previously incurable. Blind people's brains may be implanted with SoC video devices, allowing them to see, while deaf people's brains could be implanted with SoC audio devices, allowing them to hear. Handheld computers equipped with tiny whip antennas may one day be able to access the Internet at megabit-per-second rates from anywhere on the planet's surface.

The following figures shows a generic picture of an SoC: [8]

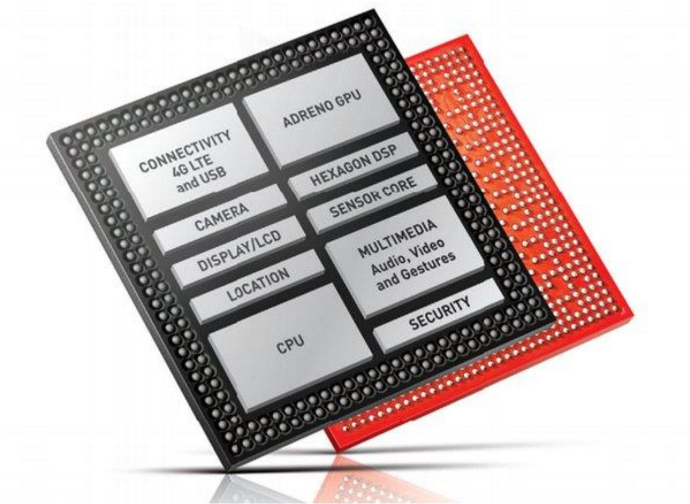


Figure 4: System-on-Chip

2.5 Inertial Measurement Unit (IMU)

Inertial Measurement Unit (IMU) is an electronic device that can measure the movement of an object in any direction and can report this movement as forms of electric signals. Mostly, an IMU is composed of an accelerometer (detects force/acceleration) and gyroscope (detects angular change), but sometime magnetometer (measures magnetic field around the system) can also be added to it. [9]

Accelerometers (one or more) and gyroscopes (one or more) monitor linear acceleration and rotational rate, respectively, in an inertial measurement unit. In some cases, a magnetometer, which is commonly used as a direction reference, is also incorporated.

Inertial Navigation Systems (INS), which use raw IMU signals to detect velocity, angular rates, altitudes and location relative to another reference frame, frequently use IMUs. Many military and commercial vehicles, such as piloted airplanes, ships, missiles, satellites and submarines, rely on the IMU-equipped INS for navigation and control. Unmanned systems such as UAVs, UUVs and UGVs, rely on IMUs for direction and control. Attitude and Heading Reference Systems, which are simpler versions of INSs, use IMUs to determine vehicle attitude and heading relative to magnetic north. Via data from the IMU's sensors, a computer may monitor the craft's position using a technique called dead reckoning.

There are mainly various types of IMUs present. For only acceleration, gyro, or magnetometer, it is 3-axis IMU. For any of the two features (let’s say acceleration and gyro), there is 6-axis IMU. For all these three features, there are 9-axis IMUs. The axis represent the directions. 3-Axis Accelerometer can calculate acceleration across all three axis (x, y and z). Same goes for Gyro and magnetometer. 6-axis IMU means 3-axis for acceleration and 3-axis for gyroscope. The following figure represents a generic IMU structure: [10]

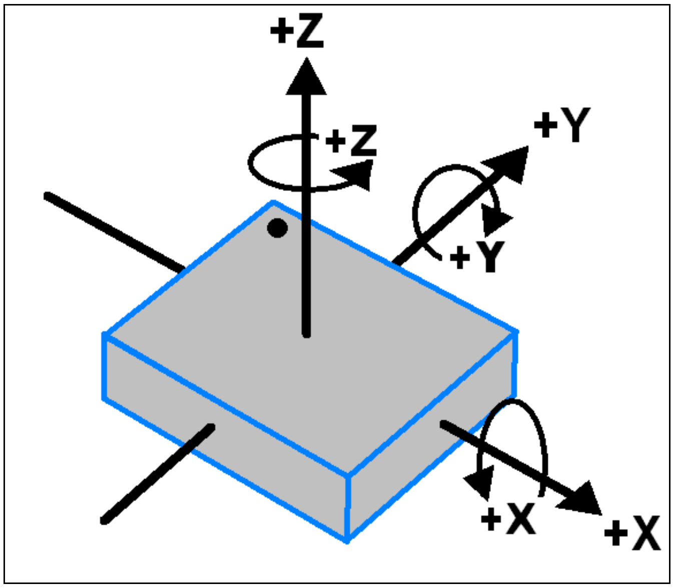


Figure 5: Inertial Measurement Unit

2.6 Development Kit (DK)

Whenever we talk about development kit (also known as DK), we think that it must be a Software Development Kit (SDK). But there are mainly two types of Development Kits that are Software DK and Hardware DK.

Hardware Development Kit

Hardware Development Kit (or just Development Kit or DK) is a single board platform designed to do flexible and easy programming into the any microprocessor or SoC. It gives a platform to easily interface with the microprocessor, for Bluetooth, ANT and other peripheral or central program development into the microprocessor. Development kits have a USB port to connect it with the PC or Laptop, some radio components (like NFC antenna, Bluetooth, etc.), proper connecting ports interfaced with the pins of microprocessor (or SoC), and many other user-friendly features. The board may also contain some buttons, LEDS, etc. so that user can test the working of his/her firmware using those components. The USB port is used to burn any program into the microprocessor, which is very easy. User can burn as many programs as he/she wish within few seconds through the USB port.

The Development Kits are mainly used to test various programs into the SoC. Using the DK, the programmer can observe changes needed in the code. So, he can make changes in the code, update it, and download it into the SoC or Microprocessor. After successful completion of the program, the user has can now download it into his own PCB's microprocessor. So, in short, DKs are recommended if you are starting to code the program from a scratch. Numerous tests are required at various stages that only can be detected using various components (LEDs, etc.) on the DK. [11]

Following figure shows the block diagram of a DK (Nordic's nRF52832 DK): [12]

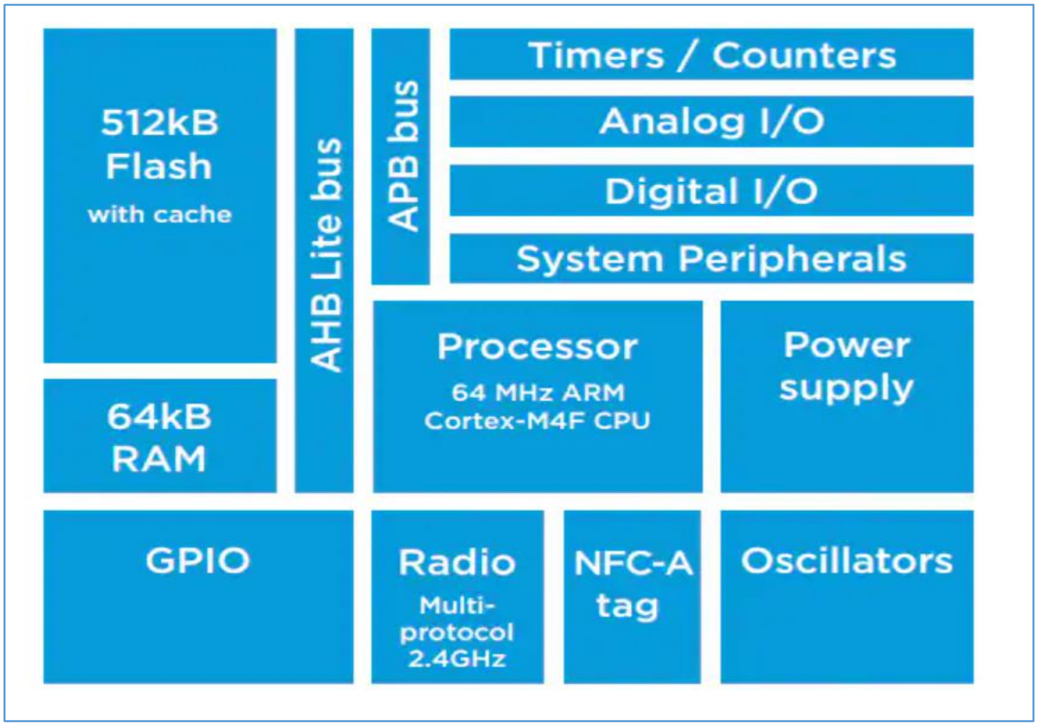


Figure 6: Nordic's nRF52832 DK Block Diagram

Software Development Kit (SDK)

A Software Development Kit (SDK) is combination of various tools for software development in a single installable file. It may include various examples, libraries, and various other tools. Mostly is specific for a specific hardware system, and sometimes it may also be specific for a specific Operating system. For example SDK designed for Windows may not work.

APIs are on-device libraries of reusable methods used to interact with a programming language, or machine-specific tools that can connect with a specific embedded system. In an IDE, debugging tools and other utilities are widespread . To help explain issues covered in the primary reference material, SDKs may include example software and/or technical notes, as well as reference and tutorials. [13]

There are various types of SDKs currently being used by different companies. For instance:

- .NET framework (Microsoft) SDK
- Android (Google) SDK
- AirTag's NFC SDK

- Nordic’s SDK
- iOS (Apple) SDK
- Xbox (Microsoft) DK

Following figure shows example of an SDK: [14]

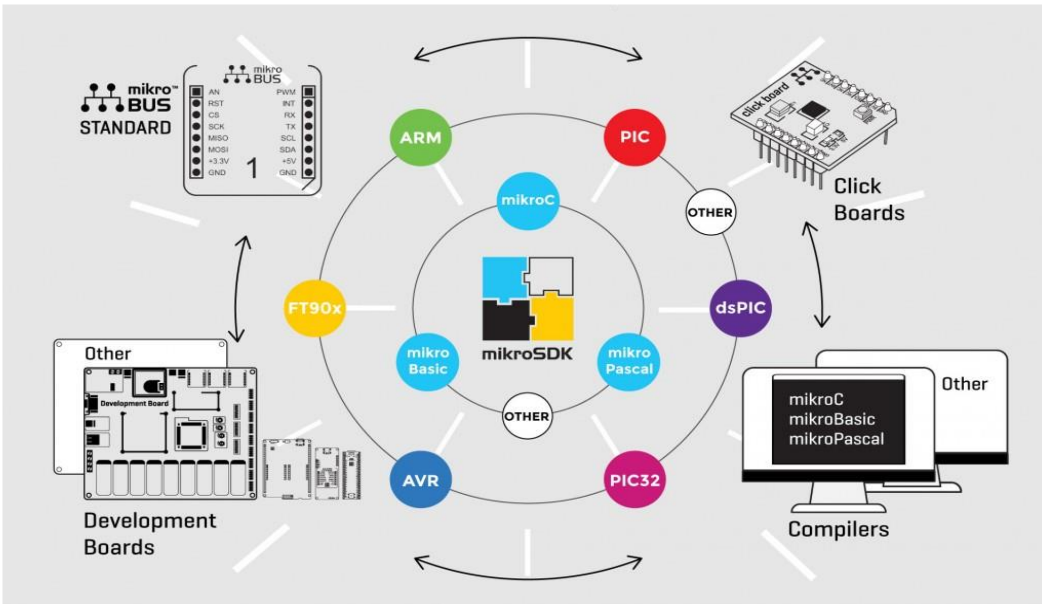


Figure 7: SDK (mikroBUS)

SDKs usually include licences that prohibit them from using it to generate software that is intended to be created under a licence that is incompatible. A proprietary SDK is incompatible with the production of free software for legal reasons, but a GPL-licensed SDK may be incompatible with proprietary software development. SDKs distributed under the GNU,LGPL are often suitable for proprietary development. In cases where the technology involved is innovative, SDKs may include hardware.

2.7 Important Terms

When dealing with the nRF52832 and related stuff, we need to know about many important terms. These will be used frequently in this project. Before using these terms, we will explain them one by one. The terms are related to the SoC, mainly how it works and its BLE part. Some of the important terms are described below:

Interrupts

Interrupts are methods that allow you to perform numerous activities at once and take action as necessary. An interrupt is an asynchronous event that causes the current code to be stopped based on its priority (The higher the priority of an interrupt, the more significant it is; this will cause a lower-priority interrupt to be suspended. **Interrupt Service Routine** is the code that handles the interrupt**.** [15]

To detect an interrupt we use either the falling or rising edge of the clock signal. The following figure shows a rising and a falling edge of the clock. [16]

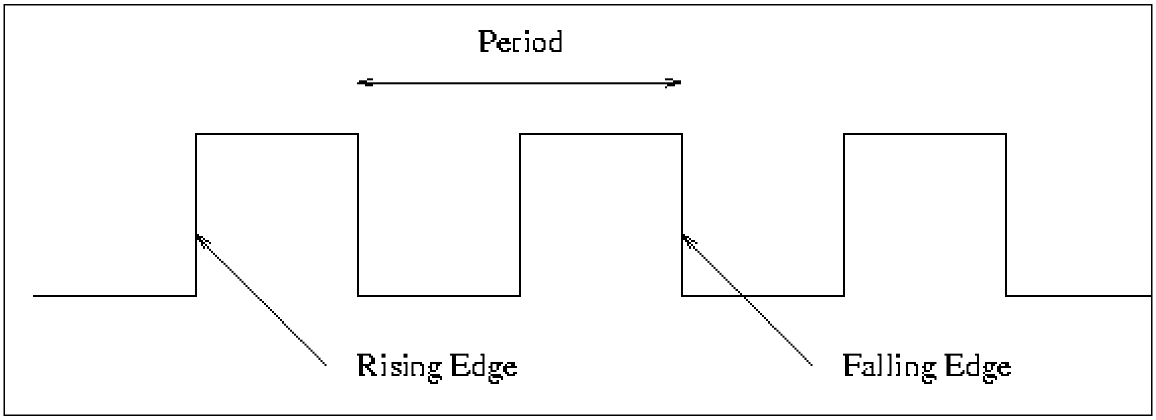


Figure 8: Rising and Falling edge of a clock

Interrupts are assigned priority at the place where they are generated. The priority is usually compared using the words LOWER priority and higher PRIORITY. Following points explain interrupts in various ways.

- Lower priority means the interrupt will be served later, if other interrupts are pending.
- Higher Priority means the interrupts will be served first.
- In terms of numbers, usually interrupts are prioritized from 1 to 7, where 1 is the highest priority and 7 is the lowest.
- When we use a Soft Device (Wireless protocol stack in nRF, e.g. Bluetooth), we only have three priorities commonly available which are 3, 6 and 7.

Timers

In embedded systems, there are a special sort of clocks that are used to measure the elapsed time. They are called timers. Usually they used for generation of time intervals. They start counting from zero each time they are used. [17]

In general, timers can be used in following two modes:

- Timer: This is used to measure the elapsed time.
- Counter: Counters are a sort of timers that are used to store the number of times a specific type of an even had occurred, and two display it.

In Nordic's SOCs (especially in nRF52), there are five timers available. First three timers have 4 CC registers, while other two have 6 CC registers.

To determine the frequency of timer (clock), we have the following formula:

$f_{TIMER} = 16\ MHz / 2^{PRESCALER}$

Peripheral Connection

Adding additional hardware to the SoC/DK for additional functionalities are referred as Peripheral Connections. MPU6050 is added in Peripheral with the DK in this project. There are some terms within this section that we need to know about: [18]

- **Tasks:** These are used to initiate an action in a portable device, such as starting a specific behavior. A peripheral can perform several tasks, each of which is represented by a different register in the task register group of that peripheral.

- **Events**: Events are used to inform peripherals and the CPU about events that have occurred, such as a change in a peripheral's state. A peripheral can generate many events, and each has its own register in the event register group of that peripheral.

- **Shortcuts**: Within the same peripheral, there is a direct link between an event and a task. When a shortcut is activated, the task associated with it is automatically initiated when the event connected with it occurs.

Master and Slave Device

A model of communication in which one device controls another device connected to it. There may be one or more devices. The central device, which is controlling all other devices is the MASTER DEVICE and the device or devices that are being controlled by the master device is/are SLAVE DEVICE(s). [19]

2.8 Inter-Integrate Circuit (I^2C or I2C) Protocol

The communication protocol we have used is the I2C. Whenever we are using a sensor like IMU with the SOC, we follow I2C communication protocol. The I2C bus was created in the early 1980s by Philips (now NXP) to provide efficient communication between elements on the same PCB board. [20]

TWI (Two-Wire-Interface) is similar to I2C and has all the basic compatibility with I2C. It was created to avoid copyright issues. It can support 100 kHz, 400 kHz, fast mode (1000 kHz), and high speed (3.4Mbits/s). nRF52832 devices support 100kHz and 400kHz modes only. [21]

I2C is a serial communication between two devices using two wires only. One wire transmits the clock signals and the other wire is used for transmission or receiving of data. It uses one wire to transmit or receive data so it’s a half-duplex communication. It is an open-drain communication so we need two pull-up resistors which will be connected on both signal wires to put the bus in a high state. The master is responsible for the communication on the bus. At a time only one master can initiate communication.

In order to start the communication between master and slave, we need some start or stop conditions. So, I2C protocol defines that master will always start the communication and it will send a start condition on the bus by transitioning from Logic High state to Logic Low state on SDA line, while the SCL line remains in Logic High state. [22]

Until the master puts a stop condition on the bus, the bus is said to be in busy state. In order to generate a stop condition the master will transition the signal on SDA line from Logic Low to Logic High. While SCL Line stays in Logic High. The following figure shows how it works. [23]

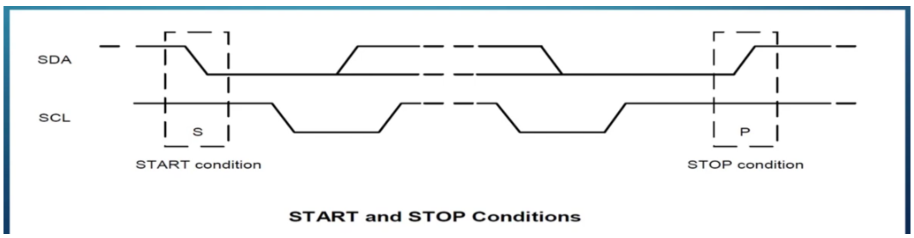


Figure 9: I2C Start and Stop Communication Conditions

<u>I2C Protocol Specs</u>

The length of I2C data bytes is defined as 8 bits. The number of bytes transferred every data transfer is unrestricted. An acknowledgement (ACK) signal must be sent after each byte sent. The true acknowledge signal is generated by the receiver bringing down SDA and keeping it low during the HIGH component of the acknowledge clock pulse.

Following figure illustrates it: [24]

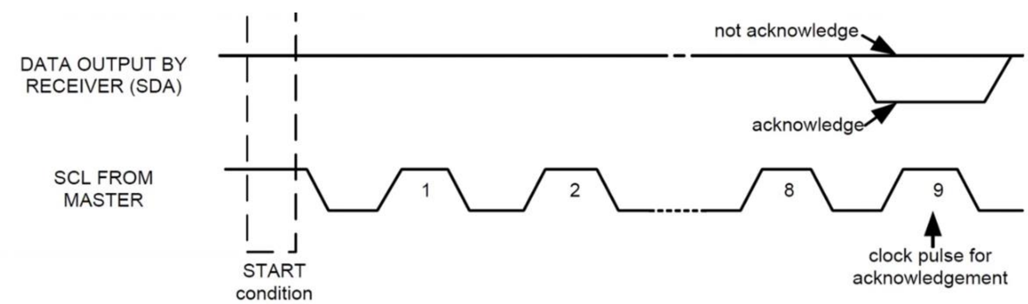


Figure 10: Acknowledgement on the I2C Bus

2.9 Bluetooth Low Energy (BLE)

BLE was added in Bluetooth specification 4.0. With the increasing demands of low energy portable devices, Bluetooth evolved to adopt for the new demands. BLE features an ultra low power protocol which uses less time on air and allows the devices to sleep for longer time periods while the connection is not being used. BLE devices can consume very less power and can even work for more than a year on just a single coin battery. It is highly secure with AES-

128 encryption. Using BLE, long transmission distance can be achieved. It gets connected with devices instantly. [25]

There are four basic roles that a device can work as: [26]

- **Broadcaster:** A device which advertises data only and cannot bond or connect with other devices.
- **Observer:** A device which scans for the advertised data and can receive the advertised data but unable to connect.
- **Central:** A master device Bluetooth which scans for the advertisement packets, it initiates a connection and is capable of connecting with other peripherals (slave devices), upon receiving advertisement packets. A Master (Central) Device can connect with multiple Peripherals (slaved) devices.
- **Peripheral:** A slave device which advertises its data and can connect with masters.

For the Bluetooth Architecture [27], a general representation is briefly explained here. The architecture can be divided into following three parts:

1. Controller:

Modulation happens in the **Physical Layer** and is done using Gaussian Frequency Shift Keying (GFSK). There are total 40 channels out of which 3 channels are fixed for broadcasting and rest 37 channels are used for communication purposes. Data transmission crate can be set to 1 Mbps or 2 Mbps.

The **Link Layer** is responsible for advertising, scanning, establishing and maintaining connections. It also makes sure the data packets are arranged in correct order and are transmitted correctly.

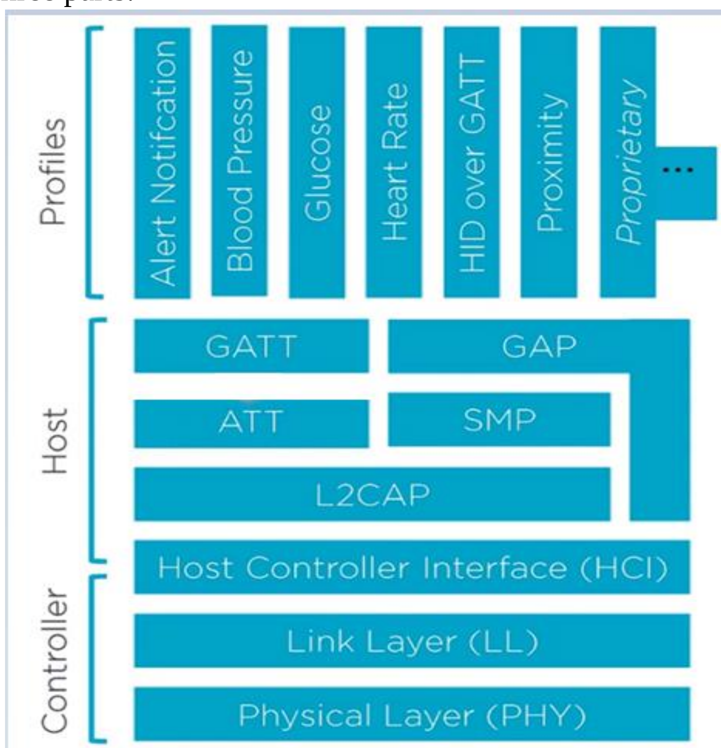


Figure 11: Bluetooth Architecture

There are five states as in the **Link State Machine** as follows: [28]

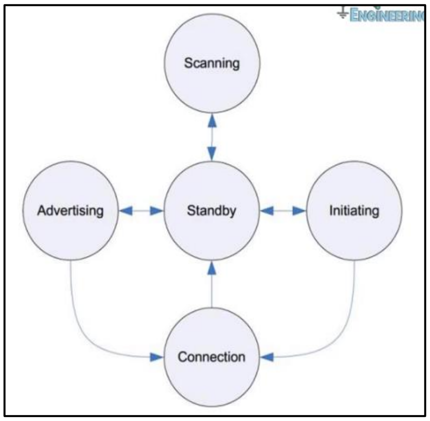


Figure 12: Link Layer State machine

- **Standby Layer:** When the device is powered on, it goes to standby state and waits until a command is received from the host and then it goes to the required state.
- **Advertising State:** Broadcast the advertising data and related responses.
- **Scanning State:** Listen to the advertising data on the specific channels.
- **Initiating State:** In this state, the connection ignition procedures are followed.
- **Connection State:** The devices are connected and now the data can be transmitted or received using the data channels.

HCI (Host Controller Interface) provides a standardized interface between Host layer and the upper layer, which can be interface by application programs API or use a hardware interface like UART, SPI or USB to control by sending commands and data. The controller passes HCI Send data and events to the host, the host passes HCI Send commands and data to the controller.

2. HOST

The L2CAP layer encapsulates data for the higher layer and enables end-to-end data transfer. It provides connection-oriented or connectionless data services that can be reused, the data packets can be broken into chunks and could be reorganized if required. In our project, we allow each L2CAP Channel flow control and retransmission.

Security Manager (SM) provides pairing and key distribution for securing the communication.

Attribute Protocol (ATT) specifies how to access data using a client server model. The data is stored in attributes which can be accessed by client. Attribute is composed of three basic element i.e. 16 bit handle, UUID (attribute type) and Value of a certain length.

Generic Attributes (GATT): All top-level LE profiles use Generic Attributes (GATT) as a basis profile. It specifies how a collection of ATT attributes is combined to form useful services.

Client Server Architecture: Two roles are defined i.e. Server and Client.

Generic Access Profile (GAP) is responsible for handling device access methods and processes, including device discovery, connection establishment, connection termination and bonding. [29]

3. Profiles

Profiles describe how two devices can discover each other and how they can communicate. Each profile can have multiple services. A profile will describe the overall functionality of the device.

Service is a collection of device characteristics and behaviors. A service might contain more than one characteristics. Moreover, the services are further divided into two categories. First category is the services that are defined by SIG. For example heart rate service and the second category of services are custom services.

Characteristics are the numeric values and are the actual data that user can access and use for different purposes. For example Temperature Value, Battery change, gait speed, etc.

UUID service or characteristics is an attribute and all the attributes have a Unique ID which is called UUID (Universally Unique Identifier). It is 128 bits long, so we have to transmit 16 bytes for each UUID. In order to improve the efficiency of the communication, SIG has defined some standardized services, these services have 16 bits of UUID so if we use these standard services, we will just set there standard UUIDs. The following figure illustrates the process: [30]

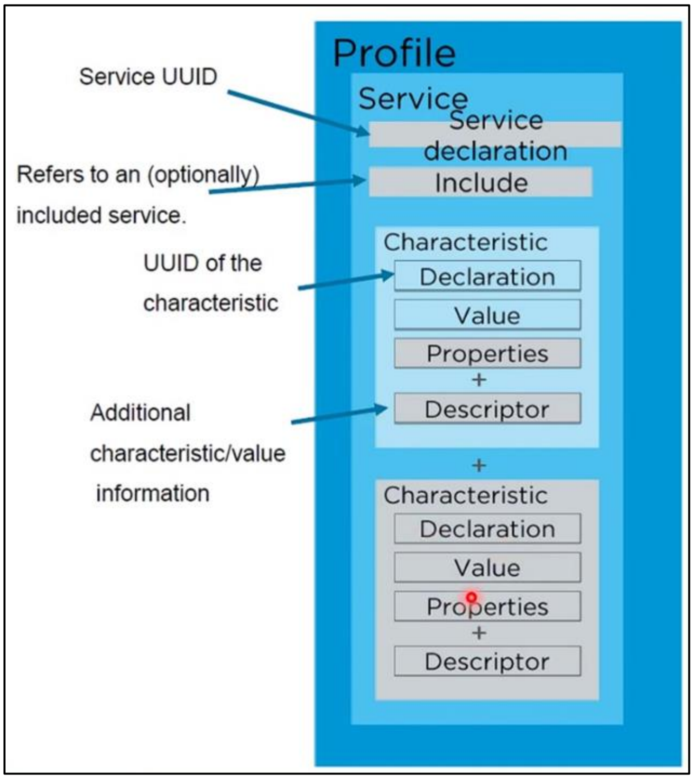


Figure 13: Profile Structure

For the Bluetooth Architecture, a general representation is briefly explained here. The architecture can be divided into three parts:

Soft Devices:

A soft device is a compiled binary of the BLE code which is developed by Nordic semiconductors. A soft device implements all the underlying specifications of the Bluetooth.

In order to use it we need to include soft device along with some header files necessary to call the internal functions of the compiled binary. Soft device helps the developer to focus more on the application code rather than writing all the BLE code. [31]

Chapter 3: Related Products and Therapies

3.1 Available Rehabilitation Techniques

For gait assessment, there are certain techniques present in the market. Some of them are discussed below: [32]

Neurophysiological Techniques:

- **Bobath:** In Europe, Bobath is the most well-known therapeutic concept. It posits a link between spasticity and movement, taking into account the muscular weakening caused by spastic antagonist resistance. This therapy combines passive mobilization with tactile and proprioceptive signals to reduce increasing muscle tone (spasticity). As a result, pathological synergies or reflex movements are not produced during exercise. First, the trunk, scapular, and pelvic waists are treated, then the more distal regions.

- **The Brunnström method:** It's also well-known, yet it's not used very often. This approach, unlike Bobath's, emphasises pathologic synergy in order to establish a normal movement pattern.

- **Proprioceptive neuromuscular facilitation (PNF):** Although it is well-known and widely utilised, it is relatively seldom employed for stroke rehabilitation. It is based on spiral and diagonal movement patterns, with a variety of inputs utilised to create normalised movements, increasing the recruitment of more motor units and maximising the required motor response.

- **The Vojta method:** This therapy was primarily created to treat children who had suffered brain injury as a result of birth. To support the formation of physiological movement patterns, the reference principle is to activate nerve endings at certain body key sites. This method is based on the activation of "innate, stored movement patterns" that are subsequently "exported" to trunk and extremity muscles as coordinated motions. The Vojta approach is well-suited to central pattern generator hypotheses for posture and gait regulation, and it is also used in adult stroke patients under the idea that brain injury suppresses but does not break stored movement patterns.

- **The Rood technique:** This method emphasises the use of peripheral input to promote movement and postural responses in a natural way.

- **The Johnstone method:** According to this approach, damaged reflex systems that create spasticity are the major cause of posture and movement problems. These abnormal responses can be controlled by positioning and splinting to prevent aberrant patterns and regulating tone to regain central control. In this domain, gross motor performance is addressed initially, followed by more complex motions.

Physical Techniques:

Certain devices are present in the market that are used to analyze the gait patterns of a human. Some of the devices are explained below:

- **RehaWalk®:** The Zebris Rehawalk® device is used in neurology, orthopedic, and geriatric rehabilitation to diagnose and treat gait issues. ln addition to a treadmill, RehaWalk® has a device for adaptive visual cueing by the presentation of gait parameters on the treading surface. [33]

Figure 14: Rehawalk ®

- LocoMat®: The Lokomat System incorporates high-quality computer-controlled motors (drives) at each hip and knee joint of the gait orthosis

Force transducers at the joints accurately measure the patient's interaction with the Lokomat. The treadmill's pace is perfectly in sync with the motors. This sensitive mechanism exactly matches the pace of the gait orthosis and the treadmill.

To develop a physiologically realistic walking pattern, software adjusts hips and knee joint angles in real time. [34]

Figure 15: Locomat ®

Cons of the above mentioned methods:

- These methods are highly expensive
- Need to get tested regularly
- Need to visit hospital more frequently
- Not everyone can avail it
- Gait assessment will not be continuous

Chapter 4: Hardware

In accordance with the need of our device, we went through the documentation and properties of various types of System-on-Chips (SoCs) and IMUs (Inertial Measurement Units). Among a variety of components, we tried our best to select hardware components that will meet our requirements and will be economical. Finally, we finalized Nordic's nRF52832 SoC, and Invensense's MPU6050 IMU.

4.1 Nordic's nRF52832 SoC

One of the key drivers behind the development of systems on a chip is the fact that, as we move forward in time, our primary aim is to decrease energy waste, save money, and reduce the amount of space occupied by massive systems. With an SoC, you may achieve all of these aims by condensing what are generally multichip architectures into a single processor that consumes far less power. These processors have enabled us to construct a wide range of portable devices that we can take with us everywhere we go without sacrificing their capabilities or performance.

Nordic Semiconductors is a famous and emerging company that mainly focuses on designing wireless technology for the Internet of Things (IoT). Nordic was chosen as a leader in the Bluetooth LE market. The majority of our significant partners were already tightly associated with Nordic, making them an easy strategic pick. We also discovered that, despite having some extremely large customers, Nordic gave us with a level of support and engagement that was really personable and engaged. We always expected high, and Nordic has always met them, providing us with early access to technical resources and assisting us in obtaining supplies while difficult times.

The Nordic nRF52832 highly developed multiprotocol SoC integrates a 64MHz, 32-bit Arm® Cortex® M4 processor with FPU with a 2.4GHz multiprotocol radio with 8dBm output power and -95dBm RX sensitivity (supporting Bluetooth 5.2, Long Range, Bluetooth mesh, Thread,ANTTM, Zigbee, IEEE 802.15.4, proprietary 2.4GHz RF protocol software and NFC). This Multiprotocol functionality is unusual in that it allows many protocols to communicate wirelessly at the same time. The SoC has a USB 2.0 interface with a speed of 12Mbps. [35]

WHY nRF52832?

Since a multitude of SoCs are available in the market. But we opted Nordic's nRF52832 because of its unique features and our requirements. We needed an SoC that could process incoming data from an IMU, do some calculations, and transfer the values to Mobile App using Bluetooth. So mainly, we need a Bluetooth, Memory, and a CPU. So we selected nRF52832. It has all the features we needed, along with some other useful characteristics. The following are some of the features of the selected SoC:

- 5.3 Bluetooth Low Energy (BLE)
    - 2Mbps speed

    BLE is a sort of wireless data transfer technology engineered for short-range communication. Low Energy means it consume a little amount of power. So, BLE is used when we prefer to save battery life over communicating in a faster data transfer rate.

    For our project, we needed a real-time feedback. We needed Bluetooth to transfer data from the SoC to Mobile App. We wanted to save our battery life, as we don't needed faster communication.

- 512 kB flash
  We needed a little memory to save the program in the SoC. The flash stores the coded program in the SoC

- 64 kB RAM
  RAM is required to store the incoming data from IMU, process the results, and store the data again temporarily before sending it to mobile app.

- 64 MHz Arm Cortex-M4 with FPU
  To read data from IMU, calibrate it, process it, convert it into useful gait features, we needed a lot of computational power.

- 1.7 – 3.6 V Supply Voltage Range
  We wanted to make our device as small as possible. In this regard, we needed the power voltage consumption as little as possible. The selected SoC met our requirement in this section also.

- 8 kB Cache
  For fast processing and sending the data, 8kB Cache is more than enough in our case.

- 6 x 6 mm dimensions
  As we needed to make our device portable, so we tried to make it as small as possible. The dimensions were perfect to make a small, portable device.

Other members in the series were also having similar features. But those features were extra redundant that we didn't need and were comparatively high power consuming. This finalized hardware had all the required features and is very less power consuming. Moreover, the dimensions are very impressive. The SoC is very small and can be operated in just 1.7 V voltage. Following is the figure of Nordic's nRF52832 SoC: [36]

Figure 16: Nordic's nRF52832 SoC

nRF52832 PINOUT

Figure [18] shows a brief pinout diagram of nRF52832.The pin details are: [37]

- Pins 1, 13, 31-33, 36 and 48 are for power supply
- Pins 2-12, 14-24, 27-29, 37-43 are general purpose I/O
- Pin 25: Serial wire debug clock
- Pin 26: Serial wire debug I/O
- Pin 30: Single-ended Radio antenna
- Pin 34,35: Connection for 32MHz
- Pin 44: No Connection
- Pin 45: Ground
- Pin 47: DC output

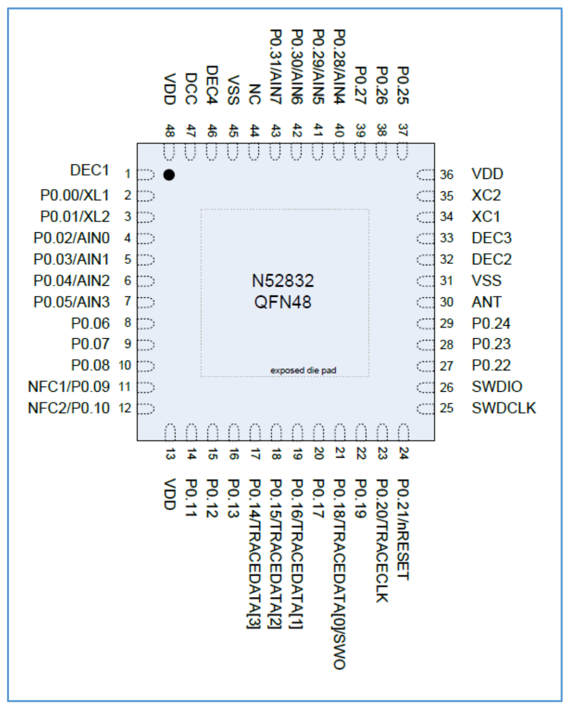


Figure 17: nRF52832 Pinout diagram

4.2 Nordic's nRF52832 Development Kit:

nRF52832 DK is a board for development with a coin cell, NFC antenna and some other extra stuff. Programming into the SoC using this DK is very easy. Micro-USB port is available on the board. We just have to connect it to the PC/Laptop and burn the firmware into it through the USB. This same USB can be used to monitor serial data from the board.

The DK board gives a user-friendly layout to access all the inputs/outputs and interface using connectors. The board is also equipped with four LEDs and four Buttons. These LEDs and Buttons are programmable. Fig 18 shows the picture nRF252832 Development Kit and Fig 6 shows the Block Diagram of the Development Kit. [38]

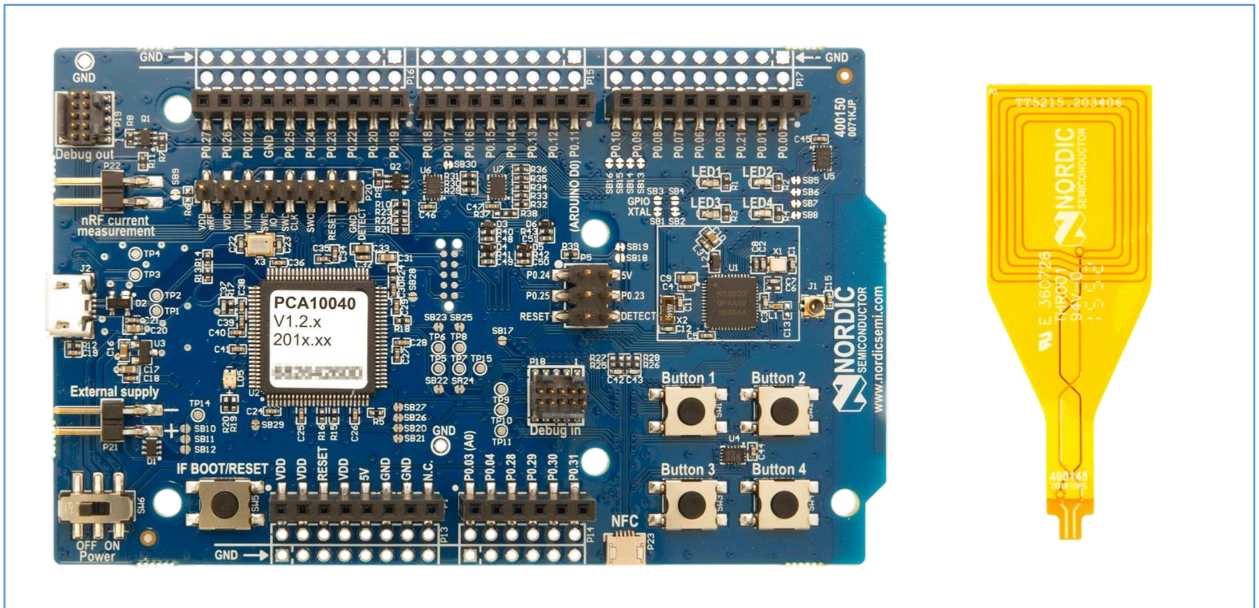


Figure 18: Nordic's nRF52832 Development Kit

In our project we have used a total of four pins i.e. Vdd, Gnd, Pin 26 and Pin 27. These four pins are interface with MPU6050.

4.3 Invensense's MPU6050 IMU

Since our project is based on the detection of the movement of shank/leg of a person. To detect that movement, we needed an IMU. After comparing a variety of IMUs, we finalized to use Invensense's MPU6050.

InvenSense is US company that manufactures motion detecting sensors for various applications. The sensor MPU6050 is one of the most widely used in a number of products. Because of its unique features, most developers prefer to use it. Characteristics of this incredible sensor are: [39]

- I2C Digital Output
- 4 x 4 x 0.9 mm dimensions
- 2.3 – 3.46 V Operating Voltage

Following figure shows the Invensense's MPU6050 IMU:

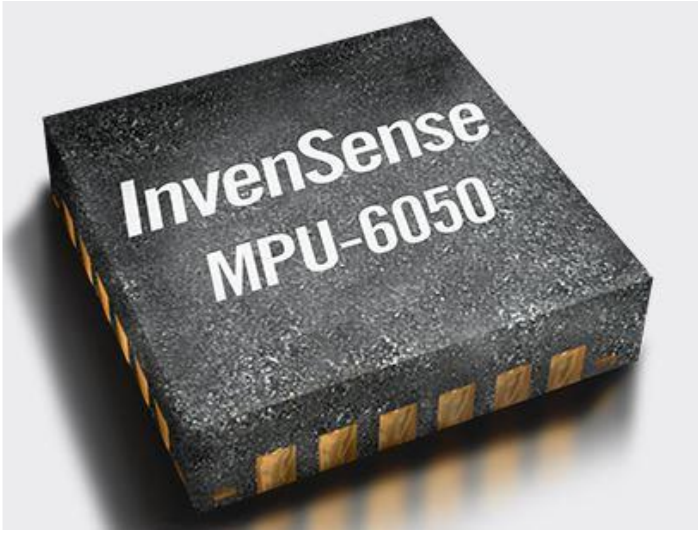


Figure 19: Invensense' MPU-60050

MPU-6050 is a 6-axis IMU, which gives us information about the acceleration and gyroscope in 3 directions (x, y, z).

MPU-6050 Module

To make the interfacing easier, we have the modules in market. It gives you an easy pin out diagram at which you can use your own pins of interest. A Digital Motion Processor (DMP) is also circuited on the board.

It has easy connecting pins for power, ground SCL and SDA. Wires can now be easily soldered on the module, which will easily connect you with the sensor without counting any addition things. Figure 20 shows the MPU6050 module:

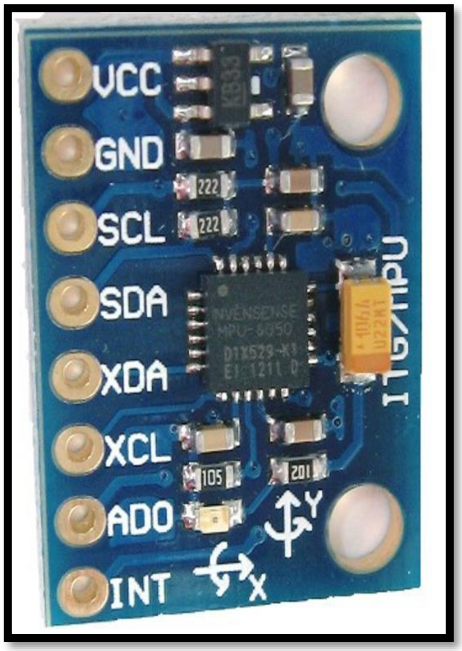


Figure 20: MPU-6050 Module

On the module, pin VCC is designed to supply power voltage. Usually it can work in 3.3-5.5V voltage. Pin GND is connected to ground. Pin SCL (Serial Clock Communication) is used to receive clock signals. Pin SDA (Serial Data Communication) is used to transmit the data (acceleration and gyro values) to the connected master device. SCL, SDA and other things related to communication were discussed in chapter 2.

The Gyroscope and Accelerometer are in fact micro-electromechanical systems.

Micro-Electromechanical System (MEMS)

Micro-Electromechanical Systems (MEMS) are the type of device which contain moveable electronic devices.

MEMS devices are typically 20 micrometers to a millimeter in size, and range in size from 1 to 100 micrometers, while components grouped in arrays can be over 1000 mm2. They generally comprise of a central unit that processes data (a microprocessor, for example) and a number of components that interact with the environment. Because MEMS have a large surface area to volume ratio, ambient electromagnetism and fluid dynamics are more relevant design issues than larger mechanical devices. [40]

The accelerometer and gyroscope in the MPU6050 are also MEMS. Following is the illustration how an IMU detects the motion.

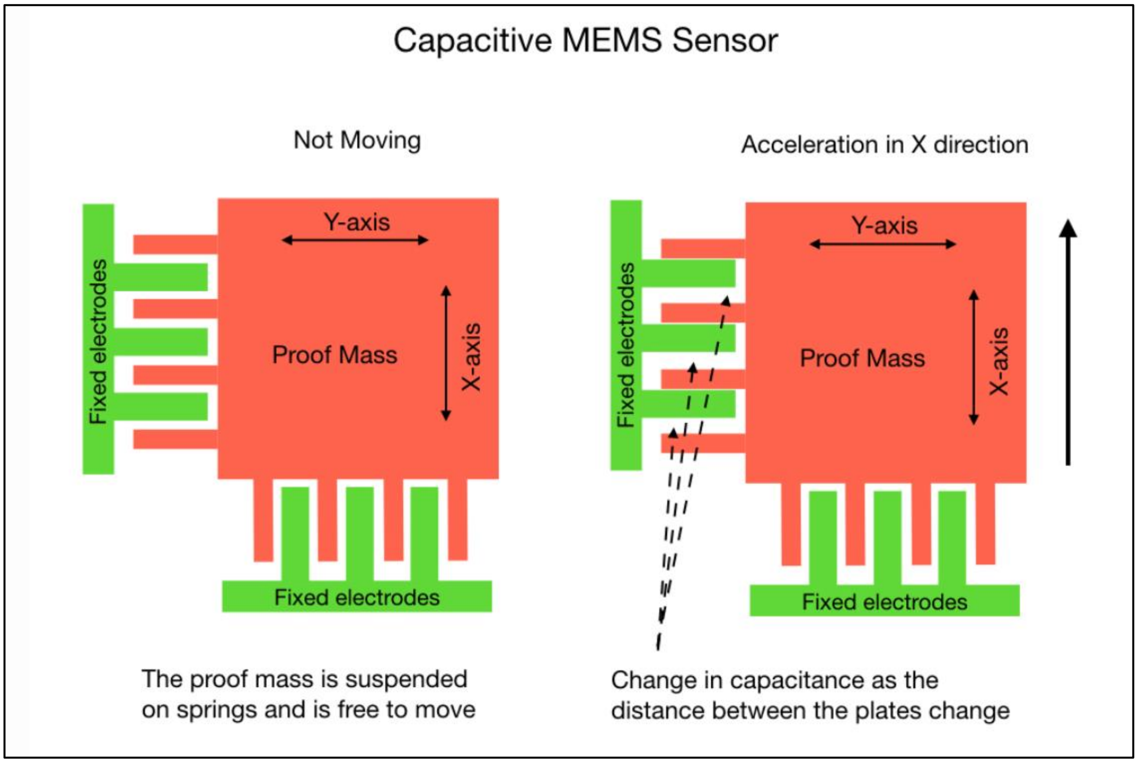


Figure 21: MEMS Sensor

The Proof Mass, which is hung on springs and free to move about when the gadget accelerates, is referred to as the Proof Mass. A capacitive action is created between the fixed comb of electrodes and the proof mass. An ADC records the change in capacitance as the device travels and converts it to a digital value between 0 and 32,750. The gyroscope operates in a similar fashion, except instead of acceleration, it uses the Coriolis Effect.

Interfacing nRF52832 DK with MPU-6050:

On the DK, pin number 27 is assigned for Serial Clock Communication (SCL) and pin number 28 is assigned for Serial Data Communication (SCL). We can also use any of the general I/O pins for SCL and SDA connection.

Pin VDD on the DK can supply 3.3V voltage. It will be connected with the VCC of MPU-6050 Module. Any of the GND pins on the DK can be connected with the GND pin of the module. We are connecting Pin 27 (SCL) with the SCL pin of MPU-6050 to transmit clock signals and Pin 26 (SDA) is connected with the SDA pin of the module to receive the data. In this case, DK is the master device and MPU6050 module is the slave device. The following figure shows the interface of nRF52832 DK with MPU6050:

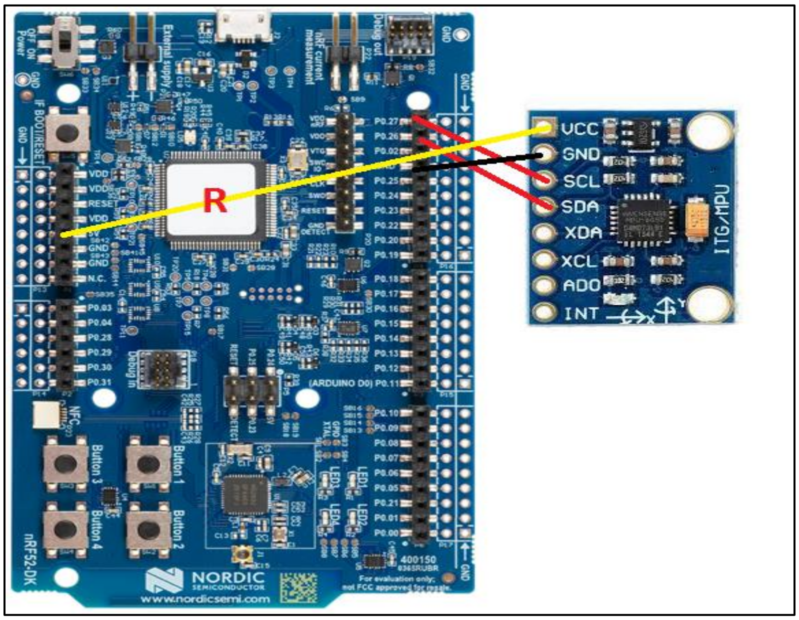


Figure 22: Interfacing MPU6050 with nRF52832 DK

Chapter 5: Software

Our project was based on both software and hardware. The software portion was comparatively larger than the hardware one. Because it included the firmware of the SoC, that we had started from a scratch. It also included the software for mobile application. Details are described in this chapter. We have used prototype based software development lifecycle model. For the development of firmware, we have used SEGGER Embedded Studio as an IDE. To monitor the serial data from the DK, we have used PuTTY software.

5.1 SEGGER Embedded Studio (IDE)

The all-in-one solution for managing, creating, testing, and deploying embedded programs is SEGGER Embedded Studio. Because of its extensive set of capabilities, this translates to easy and efficient development processes. The great project manager makes it possible to handle both large and small tasks. Automatic application deployment is made possible by version control features.

The MS Visual Studio (Microsoft VS)-like style of SEGGER Embedded Studio boosts both output and user experience. Embedded Engineers, Computer System Engineers or anyone related the field can now use it for the development of various software/hardware. All the credit goes to its simple, and versatile UI. [41]

It's been created to provide users everything they need to get started with professional embedded C programming and development: For every development environment, an all-in-one solution that provides stability and a consistent workflow. This C/C++ based IDE is very powerful for development of microcontroller boards.

The following first figure shows an interface of the SEGGER Embedded Studio and the second shows the menu of connecting J-Link (nRF52832 DK), and downloading a program in it.

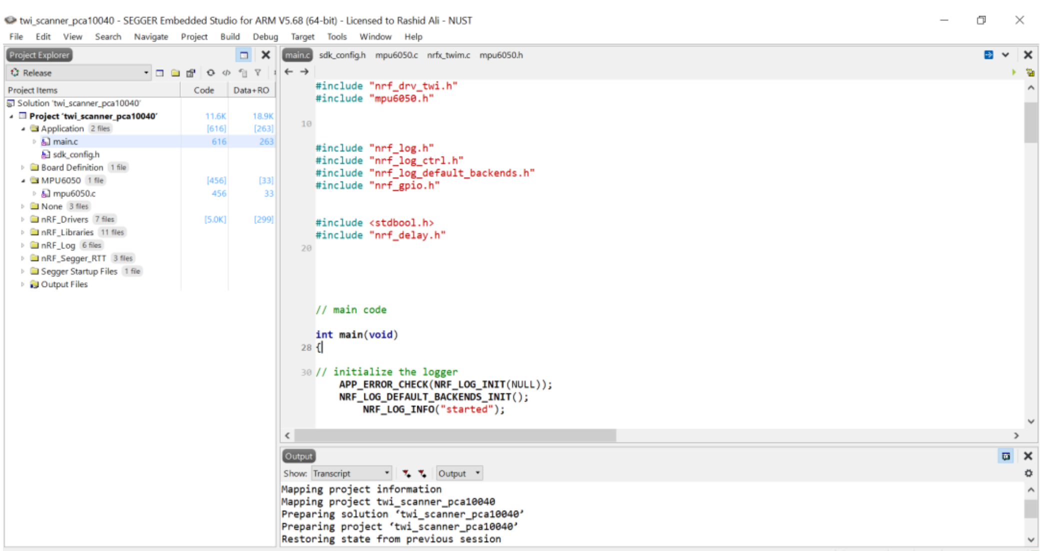


Figure 23: SEGGER Embedded Studio Interface

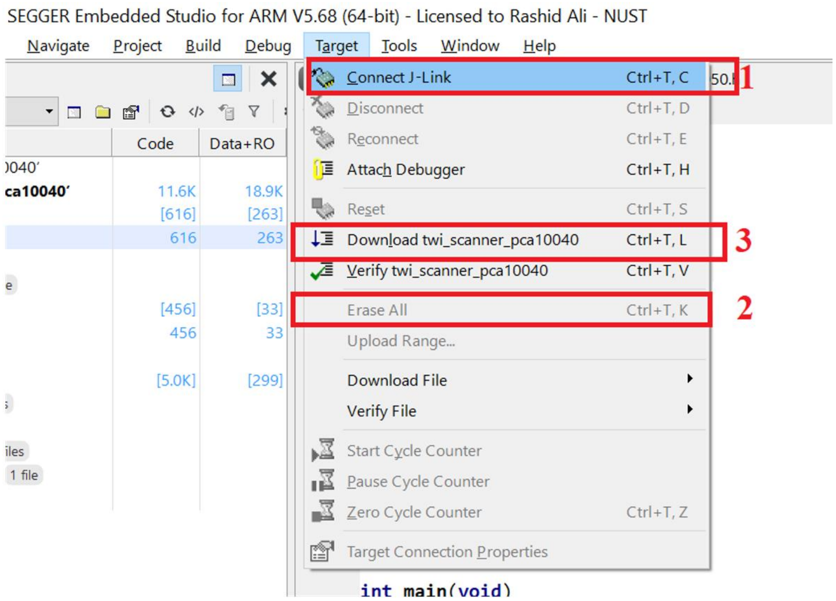


Figure 24: Connecting a device and downloading a program

5.2 Arduino

In the project, we have also used Arduino. The Arduino was mainly used to test the working of IMU, the calibration, and reading values. It was used just as a helping tool. If the IMU worked good in Arduino, we then taken it to our DK.

5.3 PuTTY

There are various software that are used to monitor the serial data. The COM (USB Port) to which a device is connected, is selected and the data communication can be monitored using those software. For example Arduino has its own serial monitor. Similarly, PuTTY is a software used to monitor serial data.

PuTTY is a free, open-source software that can be used as serial console, file transfer and terminal emulator. Figure 25 shows an interface of the PuTTY.

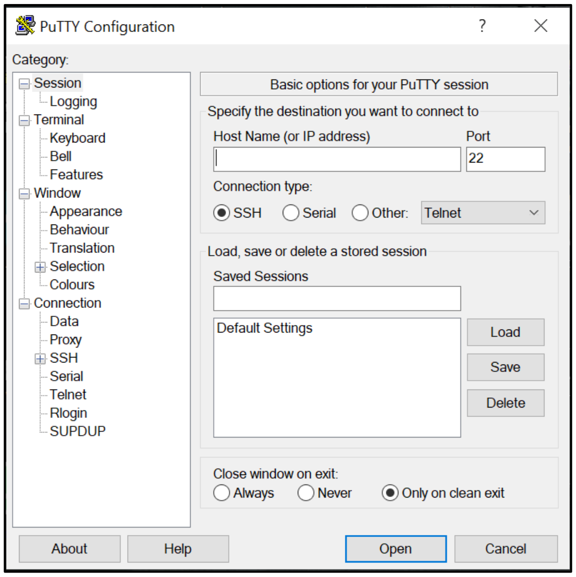


Figure 25: PuTTY interface

5.4 Firmware

Typically, firmware is a low level software program that gives access to a low-level control of any specified hardware. In other words, we can say that we burn a firmware directly into a hardware based system, which that hardware uses in order to perform its functions. It is a program to interact with the hardware. [42]

As we used an SoC, which needed a firmware to do all the functionalities. Most of our project was based on it, the mobile app part was just to display the results. So we coded our own firmware in accordance with our needs.

The main goals to achieve in our firmware were:

- to acquire data from MPU6050
- calibrate the received data
- convert it into useful information
- calculate gait parameters
- enable Bluetooth
- create custom BLE functions
- Send the data to mobile app using BLE

These goals are explained below:

<u>Data Acquisition</u>

The raw data we received from the sensor is shown in the following diagram:

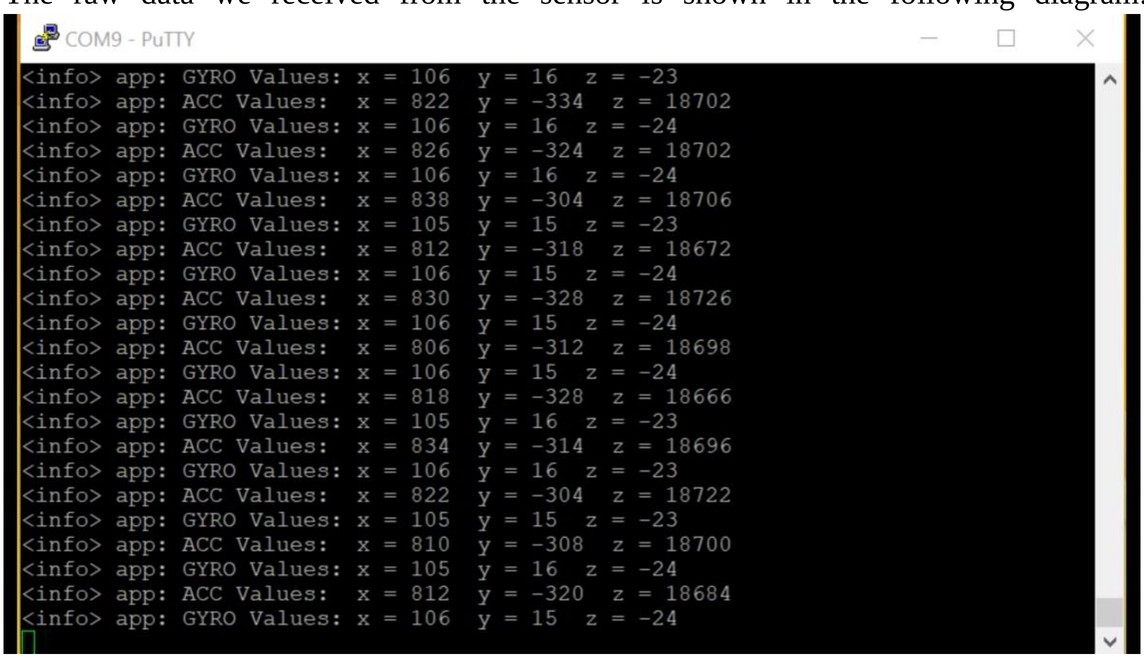


Figure 26: MPU6050 Raw Data

Reading data from the IMU was the first, but challenging goal. We had to initiate various services for MPU6050. Each device attached to the DK has a specific address. Similar, MPU6050 has its own address. If we want to read data from the MPU6050, the address will be 0x68 and for writing the data, it would be 0x69. After sending clock signal to the peripheral device, we start receiving the data. When SoC receives a data, it sends an acknowledgment signal, so that the MPU6050 would send the next data. This process runs in a loop.

Data is received after every 100 milliseconds (according to our firmware). We are continuously receiving three values for acceleration and three values of gyro (in the direction x, y, z for each). This data was received when the device was static. But is it true? The answer is NO. Because when our sensor is static, acceleration and gyro values should be zero except the acceleration downwards. The downward acceleration should be 9.8 m/s^2.

So, we need to calibrate our data now.

Data Calibration

Data Calibration is the process of adjusting the parameters of an acquired data so the data matches with the actual data as closely as possible. As discussed earlier, when static, the MPU6050 sensor should give zeros in all direction except downwards.

In order to calibrate the received data, we have to apply some offsets. But how can we know how much offset we need to apply? As we discussed in chapter 4, the analogue voltage read from the MEMS sensors is transformed to a digital signal in the range of 0 to 32750 values, as previously stated. The gyroscope and accelerometer measuring units are made up of these numbers. To represent meaningful information, the measuring units must be divided apart. [43]

So, we use 32750 to calibrate the accelerometer and gyro values. For MPU6050, the default value of accelerometer is 2g and for the gyro it is 250^o. To calculate the sensitivity, we need to divide 32750 with the default values.

For Acceleration: 32750/2g = 16384 sensitivity

So, now we need to set an offset in way that the raw acceleration data when divided with 16384 gives 0, except downward. The downward acceleration should be 1. These are the actual values of acceleration we received in the unit of g. To convert it into m/s2, we need to multiply it with 9.8. To calculate the offset, we read the data from IMU, discard first 100 values, then take an average of next 1000 values.

Acceleration (x_axis) = ((x-axis value+offst1)/173384) g

Acceleration (y_axis) = ((x-axis value+offst2)/173384) g

Acceleration (z_axis) = ((x-axis value+offst3)/173384) g

For Gyro: 32750/251 = 131 sensitivity

At static position, gyro values in all directions should be zero. So we adjust the offsets so that dividing the values by 131 will give zero. Same procedure as in acceleration is used to calculate the offsets for gyro as well.

Gyro (x_axis) = ((x-axis value+offst1)/131)^{o}

Gyro (y_axis) = ((x-axis value+offst2)/131)^{o}

Gyro (z_axis) = ((x-axis value+offst3)/131)^{o}

The following figure [Fig 27] shows calibrated values of Gyro and Accelerometer. When static, all the values are zero, except the gravitational acceleration. We have converted the acceleration's unit to cm/s2. So 9.8m/s2 becomes 980cm/s2. In the second figure [Fig 28], the sensor was lifted and moved forth (in the direction of x-axis). Observe the values:

COM9 - PuTTY

```
<info> app: Reading Values from ACC & GYRO
<info> app: ACC Values:  x = 0  y = 980  z = 0
<info> app: ACC Values:  x = 0  y = 980  z = 0
<info> app: ACC Values:  x = 0  y = 980  z = 0
<info> app: ACC Values:  x = 0  y = 980  z = 0
<info> app: ACC Values:  x = 0  y = 980  z = 0
<info> app: ACC Values:  x = 0  y = 980  z = 0
<info> app: ACC Values:  x = 0  y = 980  z = 0
<info> app: ACC Values:  x = 0  y = 980  z = 0
<info> app: ACC Values:  x = 0  y = 980  z = 0
<info> app: ACC Values:  x = 0  y = 980  z = 0
<info> app: ACC Values:  x = 0  y = 980  z = 0
```

Figure 27: Calibrated acceleration values with IMU at rest

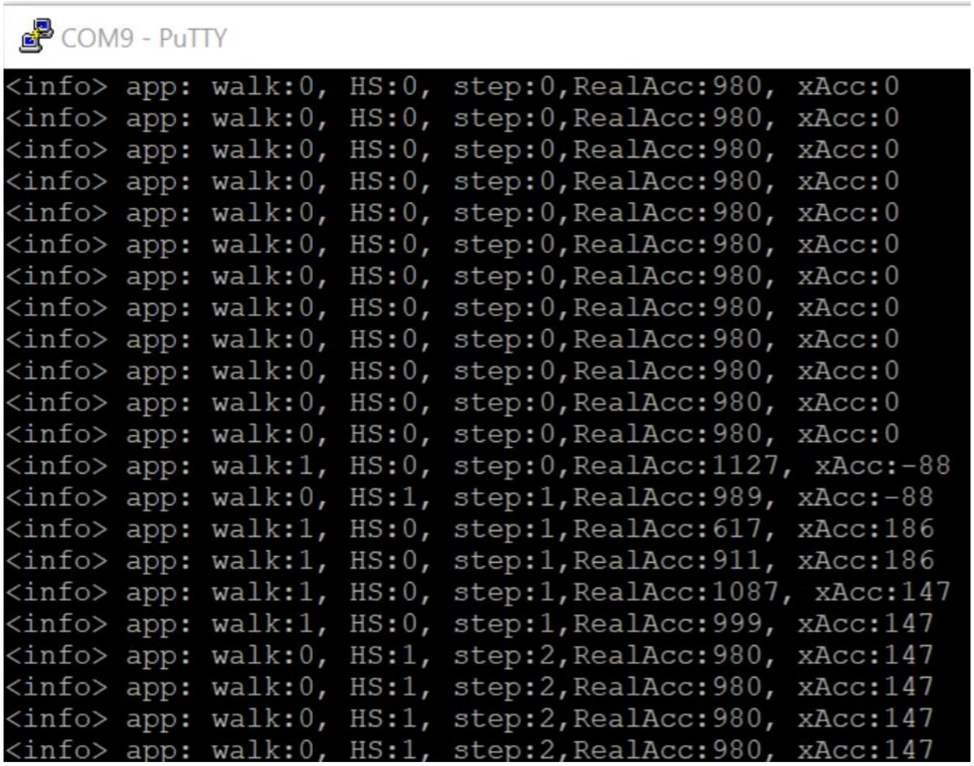
```
<info> app: ACC Values:  x = 800  y = 700  z = 30
<info> app: ACC Values:  x = 120  y = 650  z = 30
<info> app: ACC Values:  x = 130  y = 550  z = 4
<info> app: ACC Values:  x = 150  y = 400  z = 3
<info> app: ACC Values:  x = 150  y = 250  z = 5
<info> app: ACC Values:  x = 160  y = 280  z = 3
<info> app: ACC Values:  x = 140  y = 280  z = 3
<info> app: ACC Values:  x = 120  y = 480  z = 2
<info> app: ACC Values:  x = 100  y = 580  z = 3
<info> app: ACC Values:  x = 90  y = 700  z = 2
<info> app: ACC Values:  x = 800  y = 700  z = 30
<info> app: ACC Values:  x = 120  y = 650  z = 30
<info> app: ACC Values:  x = 130  y = 550  z = 4
<info> app: ACC Values:  x = 150  y = 400  z = 3
<info> app: ACC Values:  x = 150  y = 250  z = 5
<info> app: ACC Values:  x = 160  y = 280  z = 3
<info> app: ACC Values:  x = 140  y = 280  z = 3
<info> app: ACC Values:  x = 120  y = 480  z = 2
<info> app: ACC Values:  x = 100  y = 580  z = 3
<info> app: ACC Values:  x = 90  y = 700  z = 2
<info> app: ACC Values:  x = 800  y = 700  z = 30
<info> app: ACC Values:  x = 120  y = 650  z = 30
```

Figure 28: Calibrated acceleration values with IMU at motion

Gait Features Extraction

After getting the calibrated values from the IMU sensor, we can now calculate the gait features. We coded an algorithm that works to calculate the gait features from the acceleration values. We have fixed the sensor in a way that the y-axis is downward, x-axis is directed forward and z axis is towards right. In this way, the acceleration across y-axis shows the gravitation acceleration, y-axis shows the forth-back acceleration and z-axis shows left-right acceleration.

At rest, y-axis acceleration will be 9.8 and the rest two values will be zero. When the user lifts his foot, the y-axis and x-axis acceleration values will change. In the algorithm, a time counter will start once the y-axis acceleration goes away from 9.8. This is the instance of Toe-Off (TO). During the same instance, x-axis acceleration will be accumulated in a place. When y-axis acceleration comes back to normal, it is the Heal-Strike (HS) point. The time counter will stop. At this instance the user has completed a step. A step counter will be incremented. Now the accumulated x-axis acceleration is divided by total time taken in order to calculate average acceleration across x-axis. This acceleration value can now be integrated with time elapsed to find speed. Integrating the speed again will give distance value. The figure below shows successful calculation of average acceleration and step counts. It also shows HS and Walk (TO).

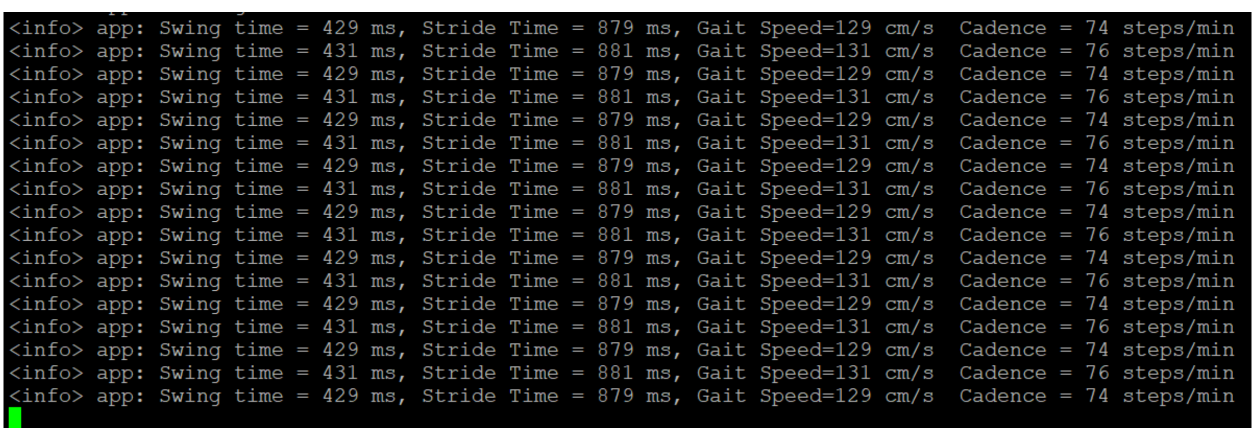


```
COM9 - PuTTY
<info> app: walk:0, HS:0, step:0,RealAcc:980, xAcc:0
<info> app: walk:0, HS:0, step:0,RealAcc:980, xAcc:0
<info> app: walk:0, HS:0, step:0,RealAcc:980, xAcc:0
<info> app: walk:0, HS:0, step:0,RealAcc:980, xAcc:0
<info> app: walk:0, HS:0, step:0,RealAcc:980, xAcc:0
<info> app: walk:0, HS:0, step:0,RealAcc:980, xAcc:0
<info> app: walk:0, HS:0, step:0,RealAcc:980, xAcc:0
<info> app: walk:0, HS:0, step:0,RealAcc:980, xAcc:0
<info> app: walk:0, HS:0, step:0,RealAcc:980, xAcc:0
<info> app: walk:0, HS:0, step:0,RealAcc:980, xAcc:0
<info> app: walk:0, HS:0, step:0,RealAcc:980, xAcc:0
<info> app: walk:0, HS:0, step:0,RealAcc:980, xAcc:0
<info> app: walk:1, HS:0, step:0,RealAcc:1127, xAcc:-88
<info> app: walk:0, HS:1, step:1,RealAcc:989, xAcc:-88
<info> app: walk:1, HS:0, step:1,RealAcc:617, xAcc:186
<info> app: walk:1, HS:0, step:1,RealAcc:911, xAcc:186
<info> app: walk:1, HS:0, step:1,RealAcc:1087, xAcc:147
<info> app: walk:1, HS:0, step:1,RealAcc:999, xAcc:147
<info> app: walk:0, HS:1, step:2,RealAcc:980, xAcc:147
<info> app: walk:0, HS:1, step:2,RealAcc:980, xAcc:147
<info> app: walk:0, HS:1, step:2,RealAcc:980, xAcc:147
<info> app: walk:0, HS:1, step:2,RealAcc:980, xAcc:147
```

Figure 29: TO, HS, step counts and acceleration

At this stage, we now have enough information. The step counter at the end of one minute will divide the total steps by 60 get **cadence**. **Gait speed** is already calculated by integrating the acceleration with time. The time counter will tell us the time taken from TO till HS, which is **Swing Time**. Finally, data from two steps will be taken and **Stride Time** will be calculated.

Now we have extracted all the required gait features. [Figure 30] shows the extracted features.



```
<info> app: Swing time = 429 ms, Stride Time = 879 ms, Gait Speed=129 cm/s  Cadence = 74 steps/min
<info> app: Swing time = 431 ms, Stride Time = 881 ms, Gait Speed=131 cm/s  Cadence = 76 steps/min
<info> app: Swing time = 429 ms, Stride Time = 879 ms, Gait Speed=129 cm/s  Cadence = 74 steps/min
<info> app: Swing time = 431 ms, Stride Time = 881 ms, Gait Speed=131 cm/s  Cadence = 76 steps/min
<info> app: Swing time = 429 ms, Stride Time = 879 ms, Gait Speed=129 cm/s  Cadence = 74 steps/min
<info> app: Swing time = 431 ms, Stride Time = 881 ms, Gait Speed=131 cm/s  Cadence = 76 steps/min
<info> app: Swing time = 429 ms, Stride Time = 879 ms, Gait Speed=129 cm/s  Cadence = 74 steps/min
<info> app: Swing time = 431 ms, Stride Time = 881 ms, Gait Speed=131 cm/s  Cadence = 76 steps/min
<info> app: Swing time = 429 ms, Stride Time = 879 ms, Gait Speed=129 cm/s  Cadence = 74 steps/min
<info> app: Swing time = 431 ms, Stride Time = 881 ms, Gait Speed=131 cm/s  Cadence = 76 steps/min
<info> app: Swing time = 429 ms, Stride Time = 879 ms, Gait Speed=129 cm/s  Cadence = 74 steps/min
<info> app: Swing time = 431 ms, Stride Time = 881 ms, Gait Speed=131 cm/s  Cadence = 76 steps/min
<info> app: Swing time = 429 ms, Stride Time = 879 ms, Gait Speed=129 cm/s  Cadence = 74 steps/min
<info> app: Swing time = 431 ms, Stride Time = 881 ms, Gait Speed=131 cm/s  Cadence = 76 steps/min
<info> app: Swing time = 429 ms, Stride Time = 879 ms, Gait Speed=129 cm/s  Cadence = 74 steps/min
<info> app: Swing time = 431 ms, Stride Time = 881 ms, Gait Speed=131 cm/s  Cadence = 76 steps/min
<info> app: Swing time = 429 ms, Stride Time = 879 ms, Gait Speed=129 cm/s  Cadence = 74 steps/min
```

Figure 30: Gait parameters

We have completed the 1st major part of signal processing. Now the most challenging task is still remaining, i.e. the BLE part. It is also a part of firmware, but due to a lot of functions and complexity, we will explain it in a whole section.

5.5 Data Transfer

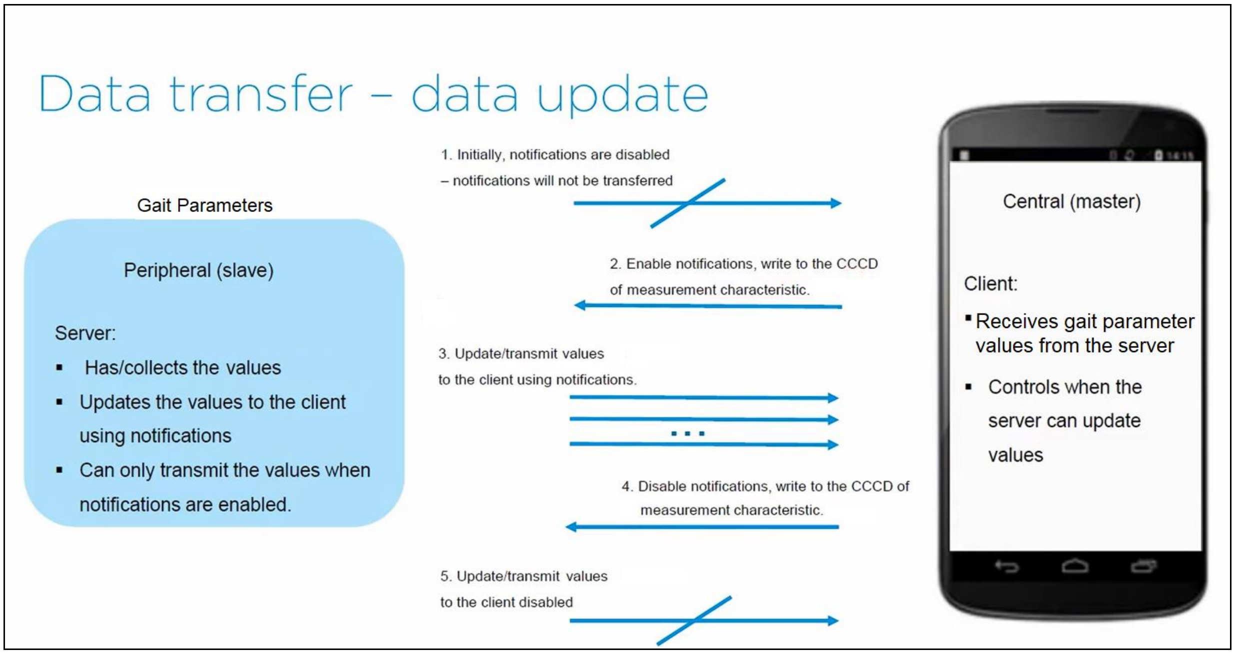


Figure 31: Data Transfer and Update

So let's see a simple example: Two Bluetooth devices want to communicate and send data over Bluetooth. Let's consider our Mobile is the Central Device, so in Client Server Model this will be the client. Let's consider our sensor device which in our case is nrf52832 or nrf52840 microcontroller, it will work as our server and it's going to read the sensor values and update them over Bluetooth communication. [44]

How Server updates data over Bluetooth or transmits the data?

BLE standard defines two ways to transmit data from server to client: Notifications & Indications.

Notifications: Are faster way to transmit data, they transmit without acknowledgement signal from client.

Indications: In indications we need an acknowledgement signal after data transfer, so we cannot send two indications together as the client has to send an acknowledgement signal before a new transfer.

The Bluetooth standard define data payload size 27 to 251 Bytes for each transmission.

At the start of communication notifications and indications are disabled, so the client has to enable them to receive the data updates.

In terms of Bluetooth connection with the DK, Mobile App is the Master Device while the SoC/DK is the slave device. Because we are controlling the DK/SoC using our Mobile app.

The above figure [Fig 31] illustrates how Data is Transferred and Updated in our project:

5.6 Device Scanning and Connection Maintenance

On each scan event the device turns on its receiver and starts listening to the advertising device. It has two parameters:

- **Scan Window:** it's the time period for one scan.
- **Time Interval:** it's the time interval between each scan.

Here the question arises. How a connection is maintained between two bonded devices?

After a connection is established, we set some connection parameters, these parameters will also effect the power consumption as well. We set the connection interval time which can be between 7.5 Milliseconds to 4 Seconds, data transfers are also done in this event so shorter interval time means faster throughput and more power consumption.

The following figure shows the Application Structure for using BLE services: [45]

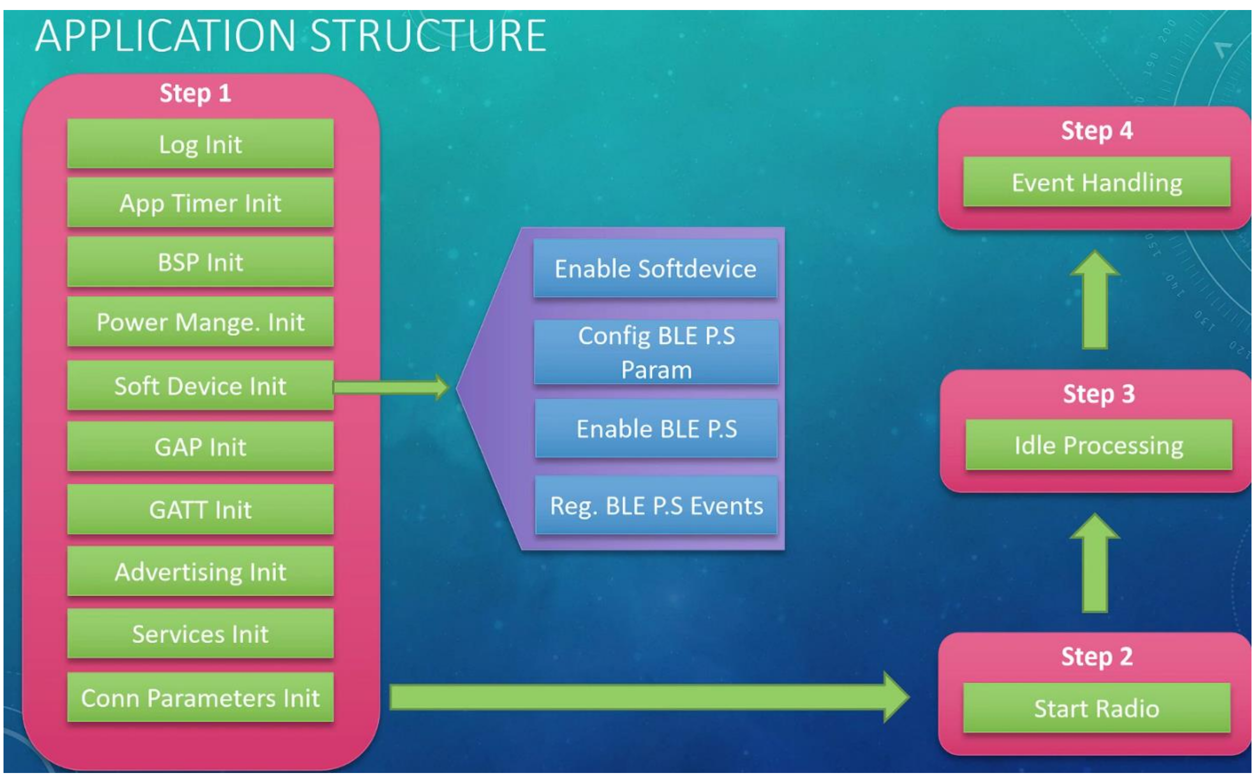


Figure 32: Data Transfer and Update

Slave delay: refers to a specific number of times the slave device can skip the connection event to keep itself in low power mode.

Supervision Timeout: refers to the time interval between two successful connection events, if the slave device does not respond within this time then the connection is lost and the device comes back to unconnected state. It can be b/w 100 ms to 32 seconds.

Pairing is authenticating another device by (ESTABLISHES CONNECTION) using shared keys which can be used to encrypt links.

Bonding pairing followed by distribution of keys which can encrypt the links in future reconnections.

5.7 Mobile Application

After the hardware part focus of our project is to build a suitable user friendly APP. Which will help user to under the complicated data in easy words, numbers, figures and graphs. The app will notify the User with according to their settings if user wants a data of a week The APP will notify him/her after one week to check the results compiled. The App will also provide monthly, Quarterly and yearly Data. App will have notification system to any latest update so that the user do not miss anything important.

A normal user of our device won't know about the firmware or what is happening at the backend. He will just know that he has to use a mobile application to assess his gait parameters. So we had to design a mobile app, with Bluetooth enable functions to receive store and display the gait parameters.

We have used the famous IDE Android Studio for the development of our application. It is one of the fastest tools to build state-of-the mobile applications. It is widely used and famous IDE.

The platform we have used in Android Studio is Flutter. It is a famous and emerging software development kit that can be used to build cross-platform applications. It means same source code can be used in Flutter to build an Android app, iOS app, and may be a Windows app at the same time, using the same source code. [46]

For the development of our mobile application we have used Java programming language. Java is the most famous programming language for development purposes. It is Object-Oriented-Based and a high level programming language.

User Interface

We have designed an effective user interface with mainly three pages. When you open the app, it displays STARTUP page when you open the application. There is an attractive graphics of walking men and a signal. One pushbutton is also there captioned "Get Started" as shown in the figure below:

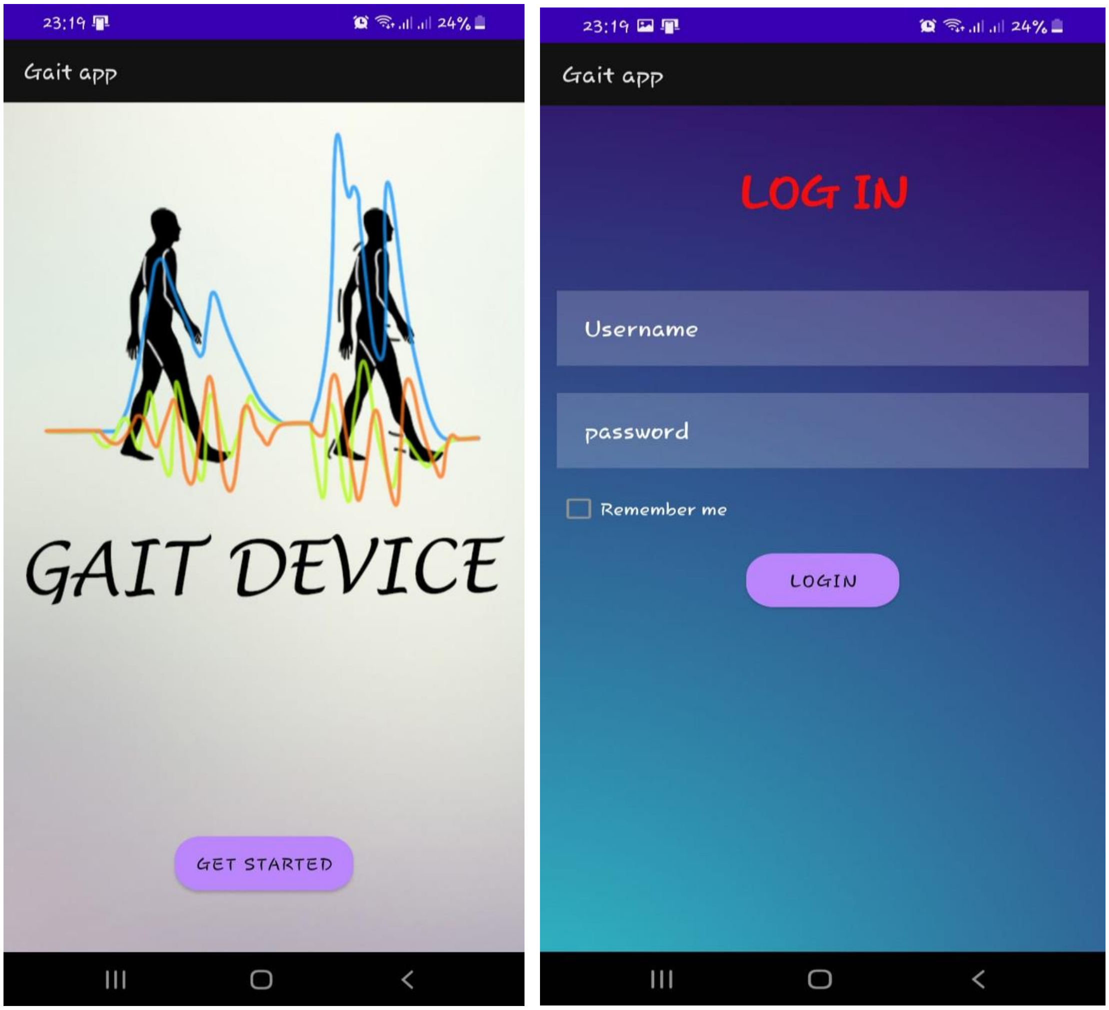


Figure 33: App Startup Page (left), LOGIN Page (right)

After pressing the only button, you will be directed to a login page. If a user wants his data to be accessed by only himself, he can set a password. If not, then he can choose the option of saving credentials. On this page, there is one text field for "USERNAME" and one password field for "PASSWORD". A checkbox is also there. After marking it, you won't be asked to login again and again. The pushbutton named "LOGIN" verifies your credentials gives access to the data as shown in the above figure.

After logging in successfully, a menu will be displayed to connect your gait device with your mobile application. The name of the device is "Gait Device". Connect to it, as show in the figure below:

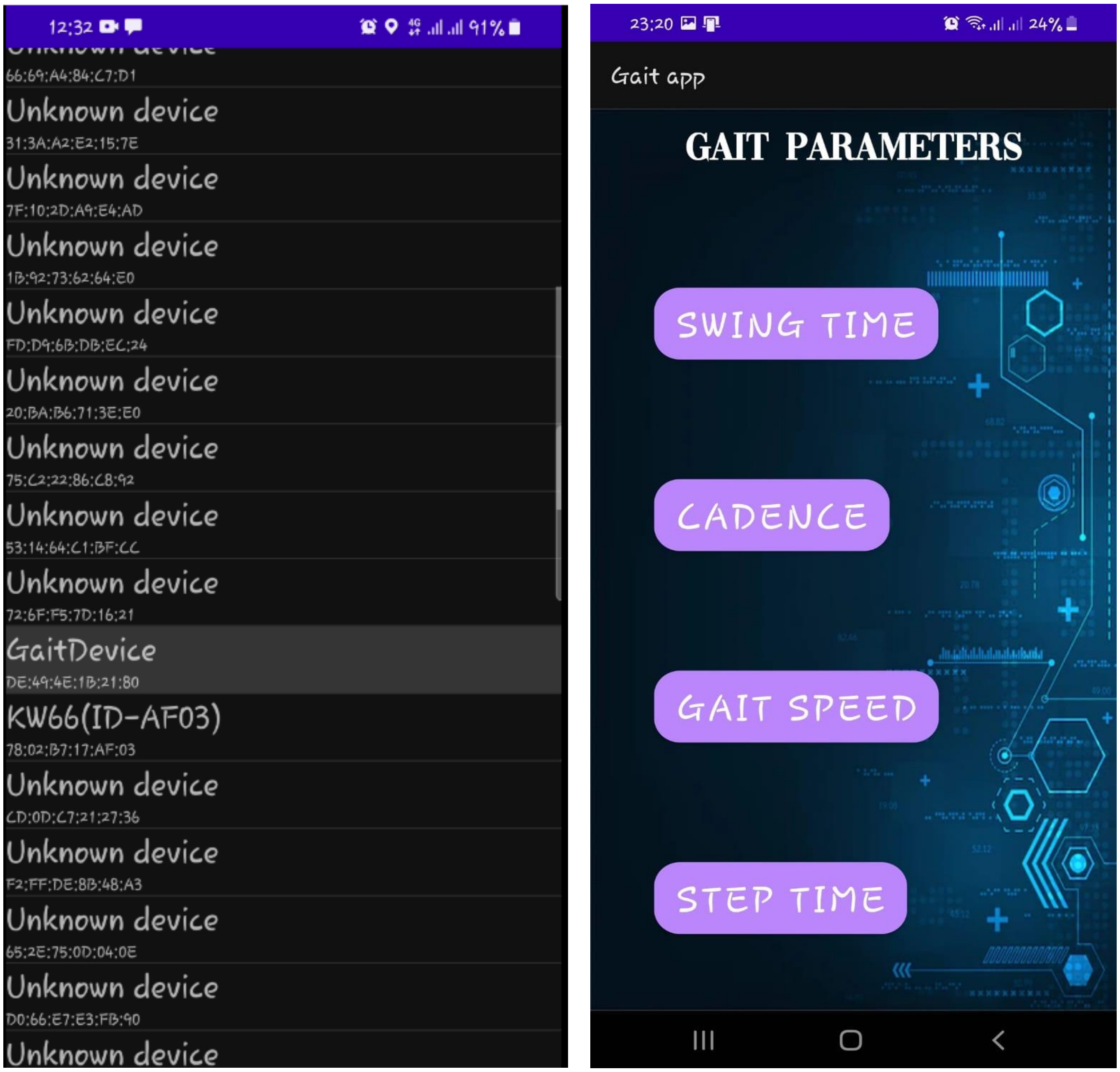


Figure 34: Bluetooth scanner (left), Main Menu (right)

Upon successful connection, the main menu will be displayed. There are four buttons on the main menu. The button names are SWING TIME, CADENCE, GAIT SPEED and STEP TIME, as shown in the above figure.

When you press any of the buttons, the corresponding data from the Gait Device will be started to get received in the mobile application. Then it will be stored in local storage and finally, the local data storage will be fetched and plotted on a line graph. Swing time scaled by 10 is plotted in the following figure.

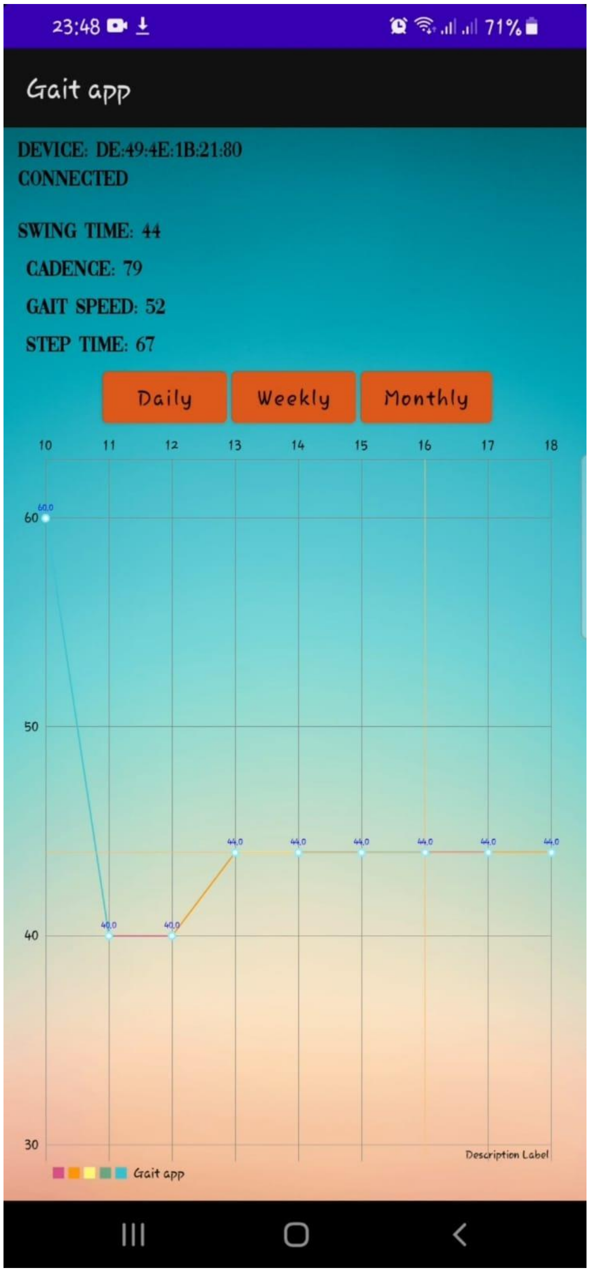


Figure 35: Plotted data on Mobile App

As we mentioned earlier that, the data received from the device is stored in local storage and then it is plotted. For the purpose of storing data in local storage, we have used SQL Lite.

SQL Lite or SQLite:

SQL Lite, written as SQLite, is database engine developed in c language. SQLite is neither an application nor any software. Rather it is a whole library which can be added into other programs to store data. SQLite is used especially for fast, high-reliable, fully featured, but a small data has to be dealt with. [47]

Chapter 6: Conclusion and Future Prospects

6.1 Conclusion

The project is designed to extract the gait features of any person. Gait features tells us a plenty of information about the body. There are numerous gait features, out of which we have calculated four – Cadence, Swing Time, Gait Speed and Step Time.

With the passage of time, if one's gait parameters are getting worse, then he can predict many information from it. With the passage of time, if gait features get worse, he can consult a physician.

Our FYP was mainly base on two parts:

- Hardware & Firmware: Connecting necessary components and coding a firmware according to our needs.
- Software part: It mainly focused on the development of the mobile application, Bluetooth services, etc.

The following figure is the hardware combination of our nRF52832 Development Kit and MPU6050 Module:

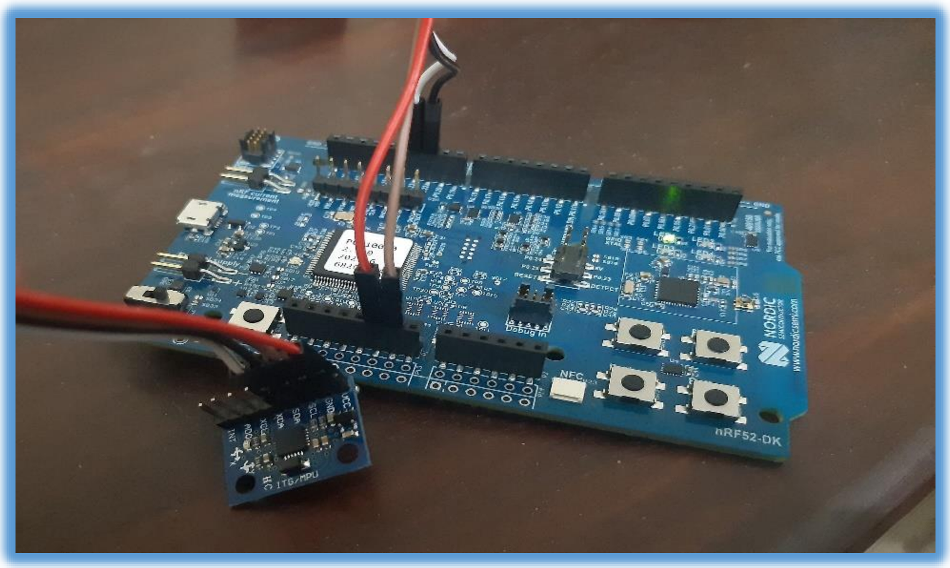


Figure 36: Complete Hardware

Wearing it on the shank looks like:

Figure 37: Wearing the device on shank

6.2 Future Prospects

In the developing countries, gait isn't given that much importance to assess. But in many developed countries like USA, work is being done to extract as much information from the gait as possible.

To be more specific to our project, as we have done a lot of work in making a firmware and making our own BLE services. In future, this project can be taken for next iterations and can be developed more efficiently. For instance, a PCB board can be designed with the combination of nRF52832 SoC and MPU6050 IMU. This hardware can be as big as a wrist watch. We already have designed a firmware for extraction of gait features, BLE services, and transfer the final values to Mobile application using BLE. So, the future iterations can just burn/download this firmware into the SoC and start improving it to a next level. Further they can do adaptations in the firmware/source code according to their needs.

Commercial production of such devices can be highly beneficial. As it would be very much economical (a few dollars), easy to use, and very light weight to carry it.

In general, the gait related issues are being researched in detail.

According to BrightFocus, a minimum of 50,000,000 are having various sort of dementias (Alzheimer's Disease.) [48]

With the increasing number of patients that have abnormalities such as Alzheimer's disease, Dementia, Parkinson's disease etc. the need to early detect and rehabilitate such conditions is extremely important. This is where the importance of portable, cheap, easy-to-use wearable gait analysis devices could be seen in the future of medical diagnosis.

Gait Analysis has great potential in enhancing healthcare practices with the help of data analysis. Daily monitoring of the data provided by such wearable devices will help physicians in detecting health problems, especially for early diagnosis. Such devices provide long-term motion tracking at many different conditions which can help physical therapists in understanding the walking patterns of a patient, which can help in making appropriate modifications to enhance therapy results. Also, most current wearable gait assessment devices are not accurate enough, but it is predicted that, in the future, such devices would achieve great accuracy, even higher than motion-tracking platforms.

Another point is that full Gait Analysis can be performed on only a few number of patients in specialized centers or hospitals. This is because gait analysis and rehabilitation devices currently in the market are expensive and need specialized people to operate. Many physicians have emphasized that gait analysis should be applied to all patients with degenerative diseases and especially those which require long-term rehabilitation. So, in the future, the demand for Gait Analysis will definitely increase. Other than using wearable sensor devices there are also many different ways in which Gait Analysis can be measured, such as using smartphone sensors, sensor fabrics etc. All these methods have a great prospect in the future of clinical practices. Such intelligent analysis formed from data collected from sensor devices will improve clinical procedures and biomechanical research.

In the future, such devices can be easily used in performing real-time computation based on kinematics and other physiological parameters and provide desired feedback on detecting gait anomalies. Such devices would engage the patients to continuous and real-time rehabilitation. As such wearable devices are lightweight and compact, this will help the patient to experience and learn in realistic scenarios. [49]

Appendix A

Hardware

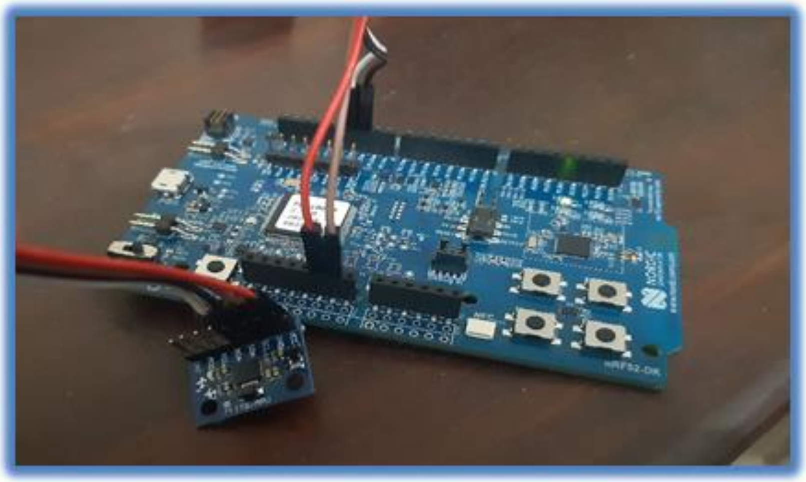


Figure 38: Hardware

Figure 39: Finalized Device Hardware

References

[1] [Online]. Available: my.clevelandclinic.org/health/symptoms/21092-gait-disorders.

[2] [Online]. Available: https://www.cdc.gov/Homeandrecreationalsafety.

[3] [Online]. Available: https://www.physio-pedia.com/Gait.

[4] D. A. D. D. K. K. Dr. Ahsan Shahzad, "Automated Prescreening of Mild Cognitive Impairment Using Shank-Mounted Inertial," 2022.

[5] S. K. S. L. J.-A. L. a. K. K. Ahsan Shahzad, "Quantitative Assessment of Balance Impairment for Fall-Risk Estimation Using Wearable".

[6] [Online]. Available: https://www.physio-pedia.com/Gait_Training_in_Stroke.

[7] [Online]. Available: https://en.wikipedia.org/wiki/System_on_a_chip.

[8] [Online]. Available: https://www.allaboutcircuits.com/news/sip-soc-som-comwhats-the-difference/.

[9] [Online]. Available: https://www.vectornav.com/resources/inertial-navigation-articles/what-is-an-inertial-measurement-unit-imu.

[10] [Online]. Available: https://core.ac.uk/download/pdf/323305998.pdf.

[11] [Online]. Available: https://www.abaco.com/hdk.

[12] [Online]. Available: https://www.beaconzone.co.uk/blog/category/nordic/page/3/.

[13] [Online]. Available: https://www.techtarget.com/whatis/definition/software-developers-kit-SDK.

[14] [Online]. Available: https://www.mikroe.com/blog.

[15] [Online]. Available: https://www.geeksforgeeks.org/interrupts/.

[16] [Online]. Available: https://www.geeksforgeeks.org/interrupts/.

[17] [Online]. Available: https://www.tutorialspoint.com/embedded_systems.

[18] [Online]. Available: https://www.wilcom.com/help/e4/es/en/MainHelp/Setup/hardware.

[19] [Online]. Available: https://en.wikipedia.org/wiki/Master/slave_(technology).

[20] [Online]. Available: https://www.geeksforgeeks.org/i2c-communication-protocol.

[21] [Online]. Available: http://web.engr.oregonstate.edu/~traylor/ece473/lectures/twi.pdf.

[22] [Online]. Available: https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/.

[23] [Online]. Available: https://www.mikroe.com/blog/i2c-everything-need-know.

[24] [Online]. Available: https://i2c.info/i2c-bus-specification.

[25] [Online]. Available: https://www.avsystem.com/blog/bluetooth-low-energy-ble/.

[26] [Online]. Available: https://microchipdeveloper.com/wireless:ble-gap-roles.

[27] [Online]. Available: https://www.tutorialspoint.com/the-bluetooth-protocol-architecture.

[28] [Online]. Available: https://subscription.packtpub.com/book/iot_&_hardware/9781788399449.

[29] [Online]. Available: https://pcng.medium.com/ble-protocol-stack-host-controller-interface-hci-44dd5697bd8.

[30] [Online]. Available: https://randomnerdtutorials.com/esp32-ble-server-client/.

[31] [Online]. Available: https://infocenter.nordicsemi.com.

[32] [Online]. Available: https://www.physio-pedia.com/Gait_Training_in_Stroke.

[33] [Online]. Available: https://www.medicalexpo.com/prod/zebris-medical.

[34] [Online]. Available: https://exoskeletonreport.com/product/lokomat/.

[35] [Online]. Available: https://www.nordicsemi.com/products/nrf52832.

[36] [Online]. Available: https://www.asmag.com/showpost/26611.aspx.

[37] [Online]. Available: https://infocenter.nordicsemi.com/pdf/nRF52832_PS_v1.4.pdf.

[38] [Online]. Available: https://www.nordicsemi.com/Products/Development-hardware/nrf52-dk.

[39] [Online]. Available: https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf.

[40] [Online]. Available: https://engineeringproductdesign.com/mems-micro-electro-mechanical-system/.

[41] [Online]. Available: https://www.segger.com/products/development-tools/embedded-studio/.

[42] [Online]. Available: https://www.andplus.com/blog.

[43] [Online]. Available: https://mjwhite8119.github.io/Robots/.

[44] [Online]. Available: https://www.youtube.com/c/SumairsEmbeddedEngineering.

[45] [Online]. Available: https://www.youtube.com/c/SumairsEmbeddedEngineering.

[46] [Online]. Available: https://flutter.dev/.

[47] [Online]. Available: https://www.sqlite.org/.

[48] [Online]. Available: https://www.brightfocus.org.

[49] [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnins.2022.859298/full.

[50] [Online]. Available: https://devzone.nordicsemi.com/guides/short-range-guides.