NeeFlex: A wearable device for measuring knee flexion angles in rehabilitating patients.

Antony M. GITAU[1,2] and Ciira wa MAINA[2]

[1] *Kenyatta University, BSc Biomedical Engineering, Department of Electrical and Electronic Engineering.*
[2] *Dedan Kimathi University of Technology, Centre for Data Science and Artificial Intelligence (DSAIL), Private Bag-10143, Nyeri, Kenya.*

Email: antony.gitau@students.ku.ac.ke
Email: ciira.maina@dkut.ac.ke

DSAIL Technical Report
DSAIL-2022-001

April 2022

Abstract

In this project, we develop a knee wearable device for accurately measuring, recording, and displaying a patient's knee flexion angles during flexion and extension. This device aims to solve the challenges orthopedics surgeons and physiotherapists experience while using mechanical goniometers which are highly inaccurate, very invasive, and difficult to use. We used an Arduino Nano 33 BLE Sense microcontroller board to record knee flexion angle measured by a flex sensor. The recorded values were then stored in a time-series database and displayed on a real-time gauge plot hosted as a dash application. On making on-bench and on-human testing, we were able to correctly measure the flexion angles and observe them on the gauge plot as they changed. This project is part of the continuing work of rehabilitation monitoring of orthopedics patients and our next step is to test the prototype with a rehabilitating patient.

Keywords: Flexion angles, Bluetooth Low Energy, Knee rehabilitation, Wearable device

Table of contents

1. Introduction

The knee bears 80% of the weight of a human body when standing still [1]. This makes it prone to many healthcare conditions. Osteoarthritis is a common condition characterized by knee pain. According to Johns Hopkins Medicine[2], major causes of knee pain include; obesity, injury, aging, and repeated strain on the joint and ligament protecting the end of the tibia and femur. Treatment varies depending on the cause of the knee pain. However, surgery is a common treatment method. Total knee replacement (TKR) is an effective procedure done on osteoarthritis patients. However, TKR patients suffer from knee stiffness and reduced knee flexion of less than 90° therefore they need physiotherapy sessions. Restricted postoperative knee flexion is actually the most frequent complication after TKR procedures and it is also the main cause of patient dissatisfaction.[3].

Currently, surgeons use goniometers to obtain the postoperative knee ROM during routine ward inspections or clinic visits for outpatients. This system is slow, highly prone to errors, and considered invasive as it involves too much contact with the patient.

In this project, we developed a knee wearable device using an Arduino Nano 33 BLE Sense [4], and a flex sensor [5]. The device was able to instantly and simultaneously record, and transmit flexion angles to a computer and display the measured flexion angles on a dashboard. This project made use of Bluetooth Low Energy technology [6] to connect the Arduino (hardware) and the computer (software) for data transfer. This device was designed for rehabilitation purposes. The goal is to allow physiotherapists and/or orthopedic surgeons to use it on their patients to track knee flexion angles digitally.

2. Objectives

The main goals of this project are;

1. To design and develop hardware and software that can measure, record, and display flexion angles of knee rehabilitating patients.
2. To test the device with potential users.

3. Methodology

To achieve our objectives, we started by building a system architecture as shown in figure 1. The implementation is made up of hardware and software as described in sections 3.1 and 3.2. The patient can wear the wearable device, flex their recovering knee and see the flexion angles on a gauge plot as demonstrated by the system architecture.

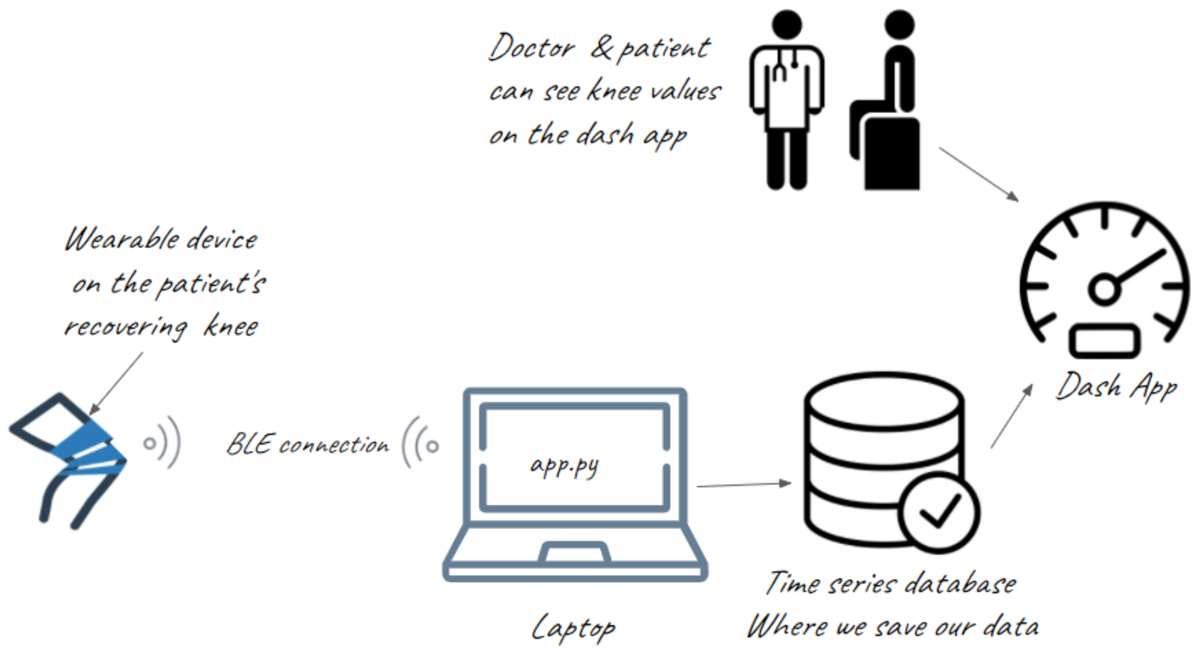

Fig. 1 System Architecture.

3.1 Hardware description.

The smart wearable knee device is made up of five main hardware components. They include; Arduino Nano 33 BLE Sense, 4.5" flex sensor, 3.7v Lithium-ion polymer battery, a switch, and a 10kΩ resistor. The intention is to use the flex sensor to measure the changes in the flexion angle of a recovering knee. The flexion angle data is collected by Arduino Nano 33 BLE Sense and transmitted using Bluetooth Low Energy (BLE) to a computer that stores the data in a time-series database.

The hardware design was done on KiCAD software. The circuit design is shown in figure 2. The design shows a battery and voltage divider circuit.

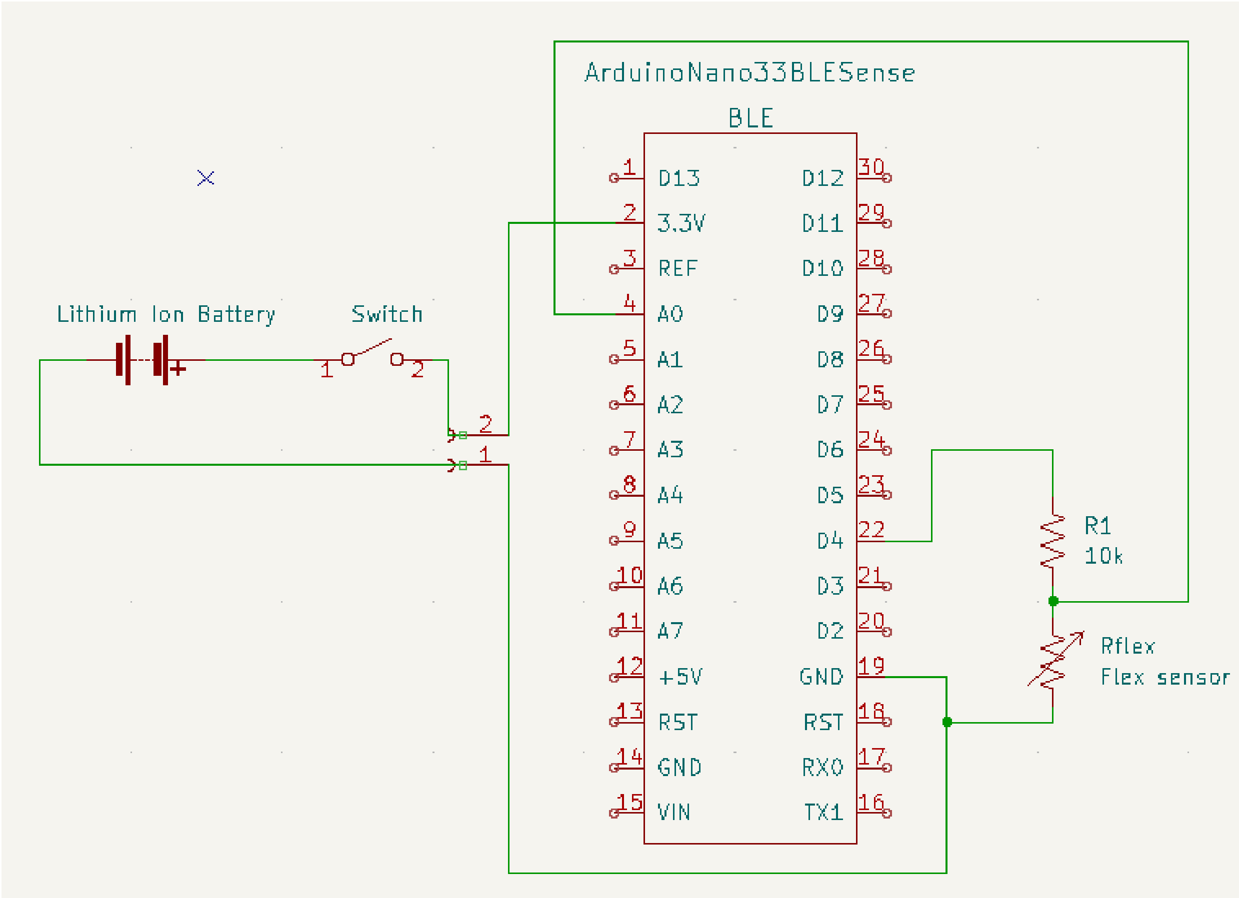

Fig 2. Circuit Design

Testing of the circuit first happened on a breadboard, then proceeded to print a PCB. Figure 3 shows the PCB design of the circuit.

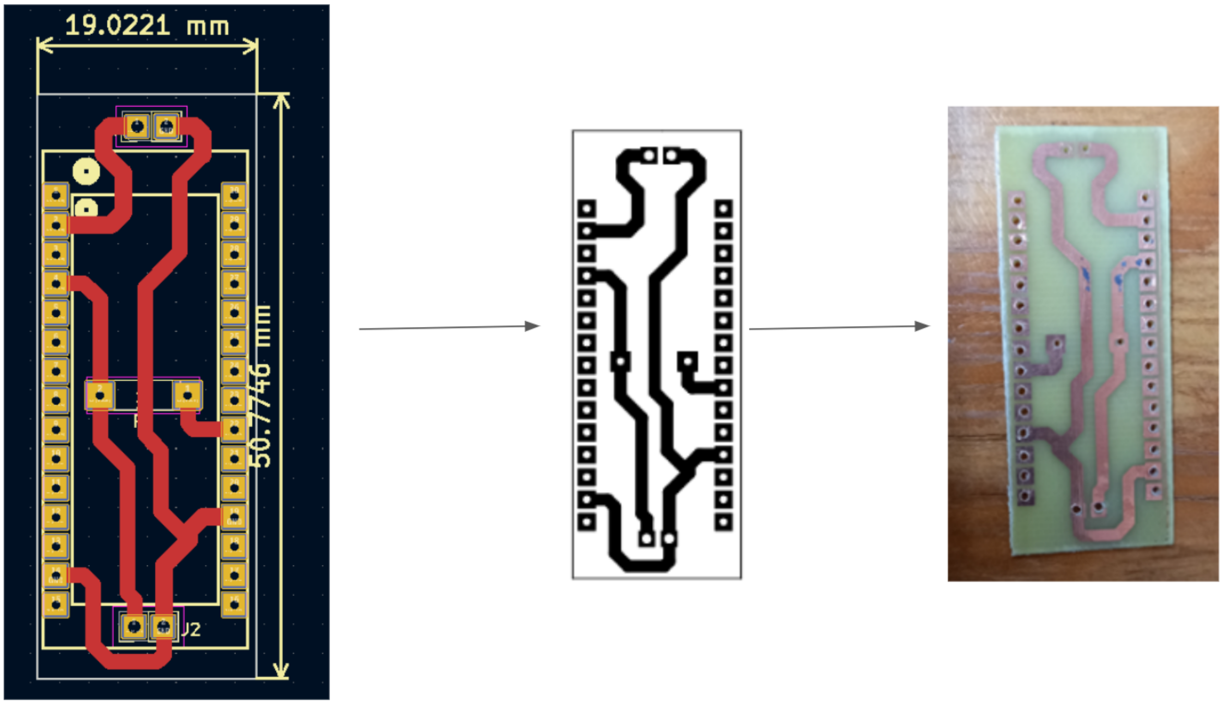

Fig.3 PCB Design

The hardware is knitted on a knee brace as shown in figure 4. A recovering knee surgery patient can wear the brace and as they flex their knee, the flexion angle is measured, transmitted to a computer, and displayed on a real-time gauge plot.

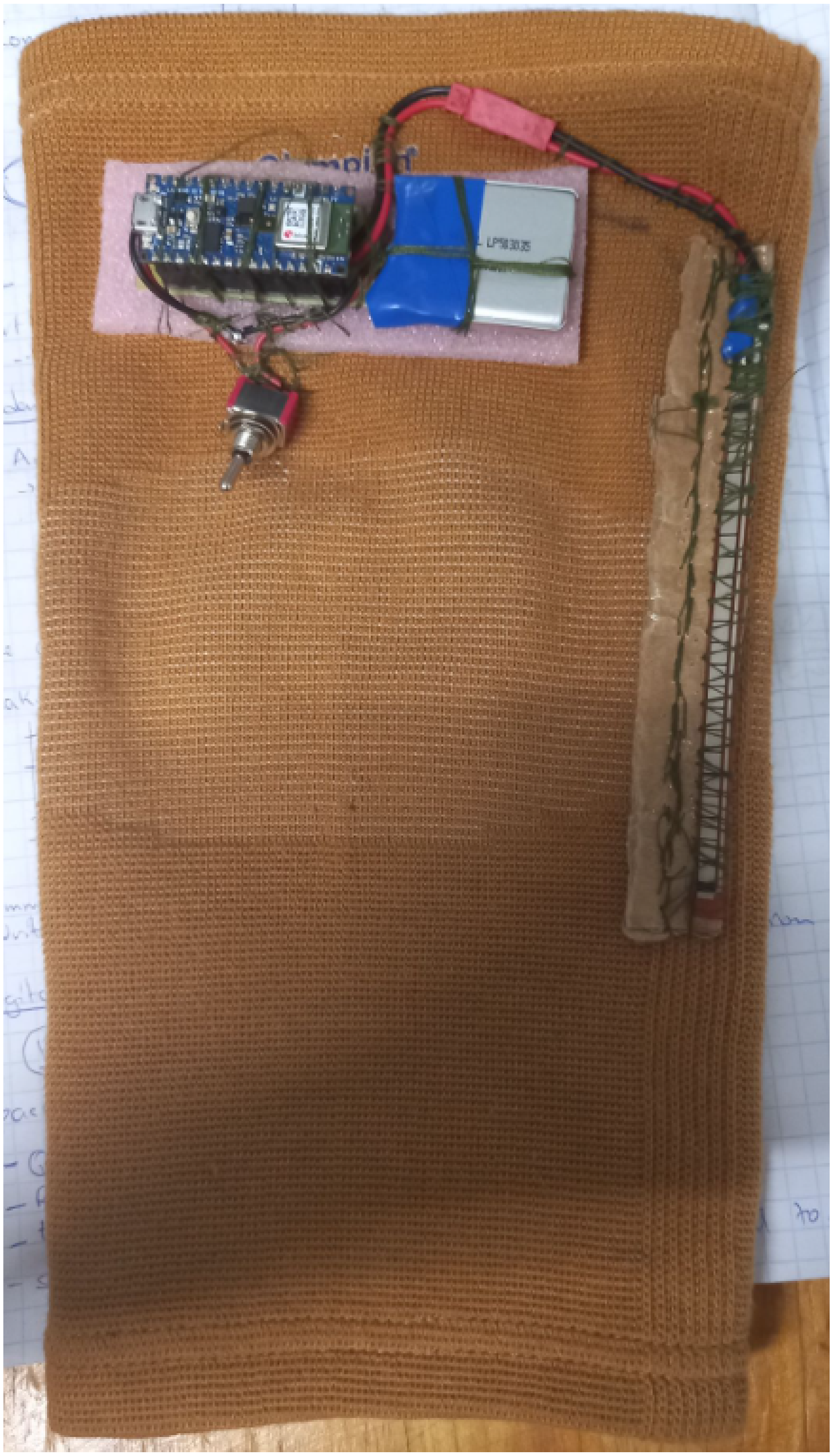

Fig 4. Hardware on a Knee brace

3.2 Software description.

The software is made up of three code components; an Arduino code, and two python scripts. One that writes and another that reads the data. The Arduino code gets the knee flexion angle from the patient's knee using a flex sensor. The writing script stores the data in influxdb, a time series database while the reading script reads from that database and displays the flexion angles on a gauge plot. We refer to the Arduino code as the peripheral and it acts as the server since it has the flexion angle data to be sent to the computer. The peripheral runs on the Arduino Nano 33 BLE Sense. The writing script is the central code. It acts as the client as it requests flexion angle data and it runs on the computer. The reading script is the dash app code that displays the read values on a gauge plot. It is written in python and also runs on a computer.

The flex sensor acts as a variable resistor. As the flex sensor bends, its resistance increases. This characteristic allows us to map that resistance change with flexion angle change. The mapping is made possible by creating a voltage divider circuit. The voltage divider is made up of 10kΩ, and the flex sensor whose resistance ranges between 11.5kΩ (when straight) to 21.0kΩ (when bent). The goal is to use the reading from A_0 to calculate the resistance of the flex sensor and use Arduino's map function to get the corresponding angle from 0º-140º. The voltage divider formula, equation 2.1 is used to calculate the resistance of the flex sensor as rearranged in equation 2.3.

$$V_o = (R_{flex} / (R_1 + R_{flex})) * V_{cc} \tag{2.1}$$

$$V_{flex} = (A_0 / 4095) * V_{cc} \tag{2.2}$$

$$R_{flex} = R_1 * (V_{flex} / (V_{cc} - V_{flex}) \tag{2.3}$$

The mapped value is added to the BLE characteristic. BLE characteristic contains the unique identifier to be used for data exchange. Refer to this book [7] for more on BLE. Figure 5 shows the flowchart of the peripheral code.

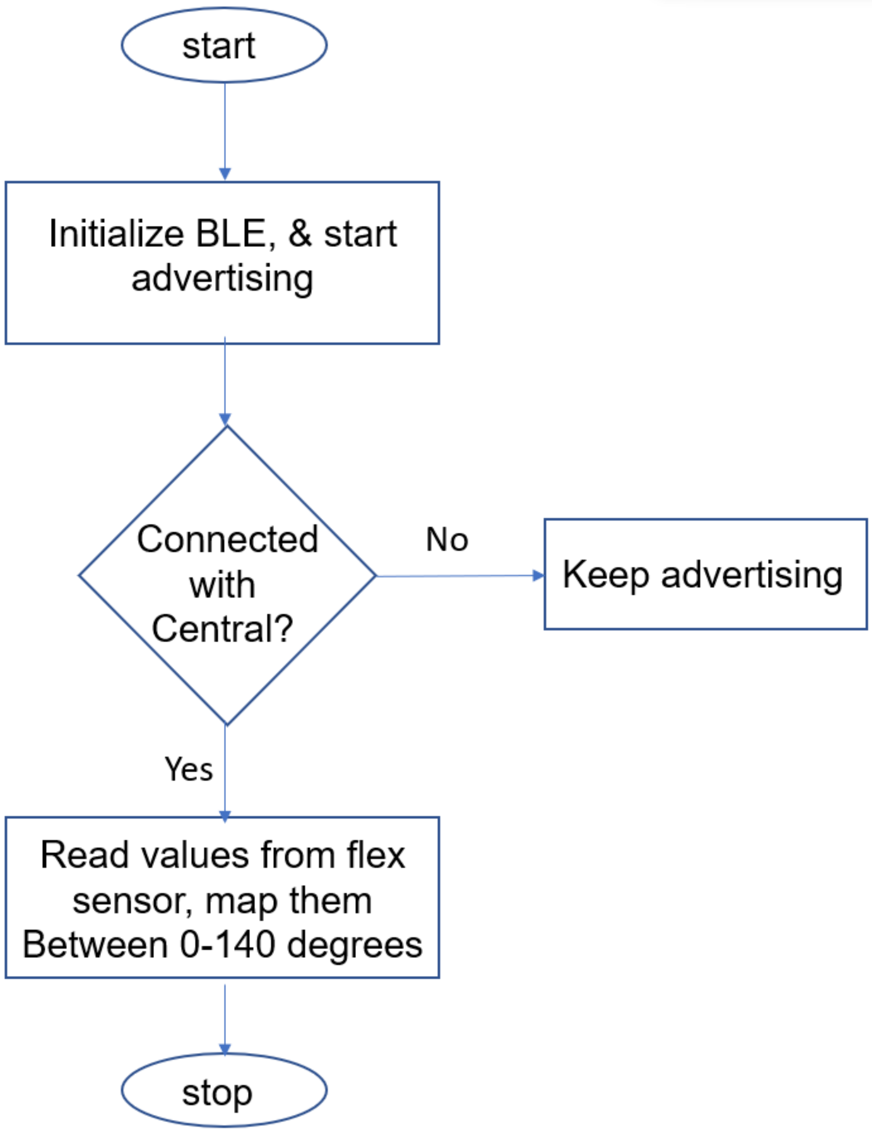

Fig 5. Flowchart of the peripheral code

To receive the data on our computer, we used Bluetooth Low Energy platform Agnostic Klient (BLEAK) [8] a generic attributes (GATT) client software, capable of connecting to BLE devices. Universally unique identifiers(UUID) [9] are used to connect the central and the peripheral code. The UUID is coded on both codes, the peripheral and central together with the address of the Arduino Nano 33 BLE Sense for the connection and data transfer to take place. The central code scans looking for peripheral devices. The central code connects to Arduino Nano 33 BLE Sense using its address.

Once that connection is made, the flexion angles we added to the BLE characteristic in the peripheral code are transmitted to the central code. As long as the connection is maintained, data transmission keeps happening. Since our project does not require the peripheral and the central to be too distant, we take our readings in the same room. The peripheral and central are approximately within 1-2 meters of each other. The readings are stored in a time-series database, influxdb[10] as integer values. Figure 6 shows the flowchart of the central code.

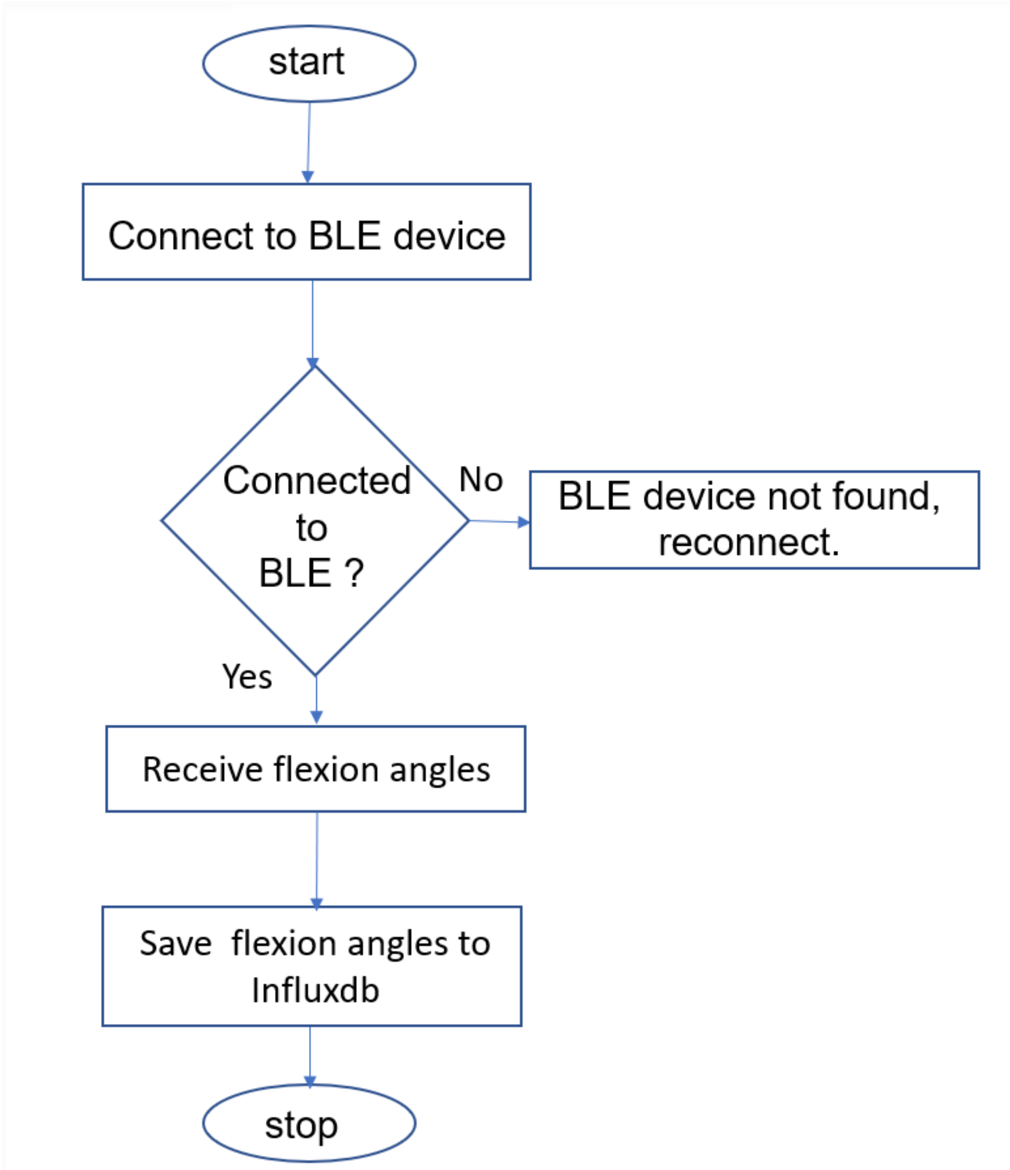

Fig 6. Flowchart of the central code

To achieve a real-time display of the stored data, we use a dash application[11]. In the dash application code, we extract the five most recent stored values and display their mode on the gauge plot. The code updates every second, making it possible to read the knee flexion changes in real-time. Figure 7 shows the flowchart of the dash app code.

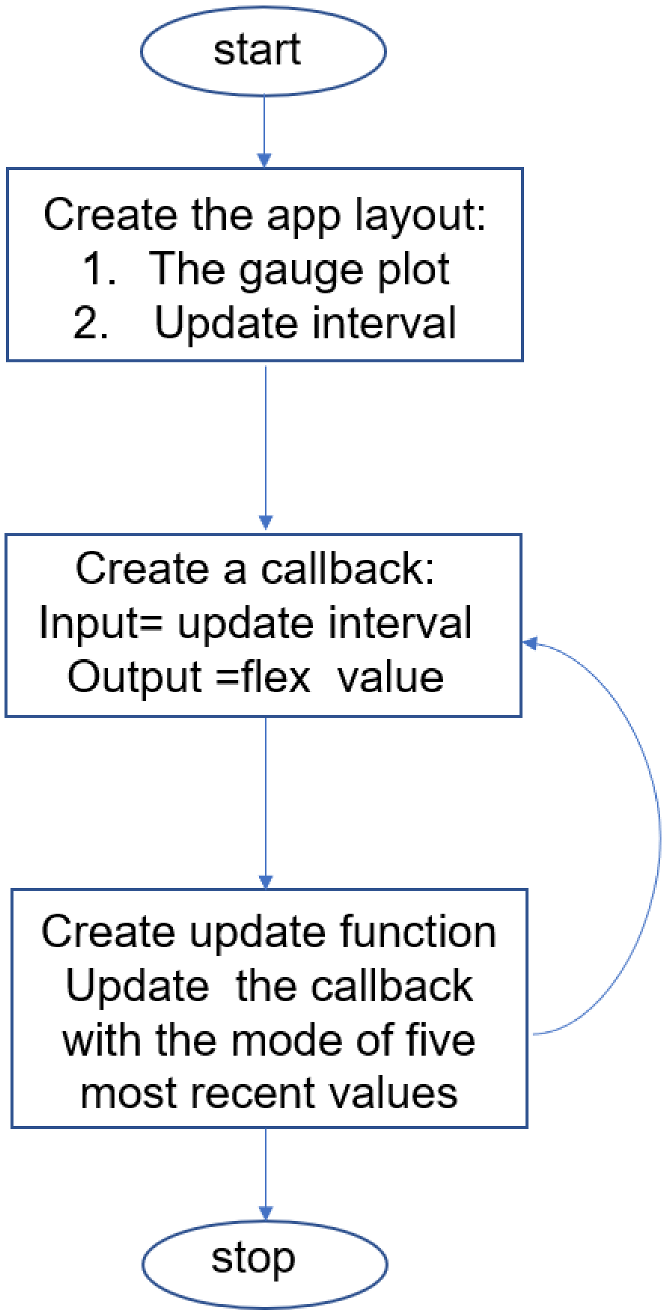

Fig 7. Flowchart of the dash app code

4. Results and Discussion

To confirm the efficacy of the device, we did multiple tests on our device. One involved testing on the workbench, without anyone actually wearing the device, and another on a healthy human's knee.
On the bench, the test involved bending the knee brace to various angles as we confirmed the angles on the gauge plot. We then had a healthy volunteer test our prototype to validate our workbench test as shown in figures 8 to 10.

In a standing position, the knee is fully extended. This means that getting a zero degree flexion degree is the expected correct reading. An observation made during multiple testing is that the measured angle could range up to five degrees. Such an offset was experienced after the volunteers sat, then stood. This offset was corrected by straightening the knee brace.

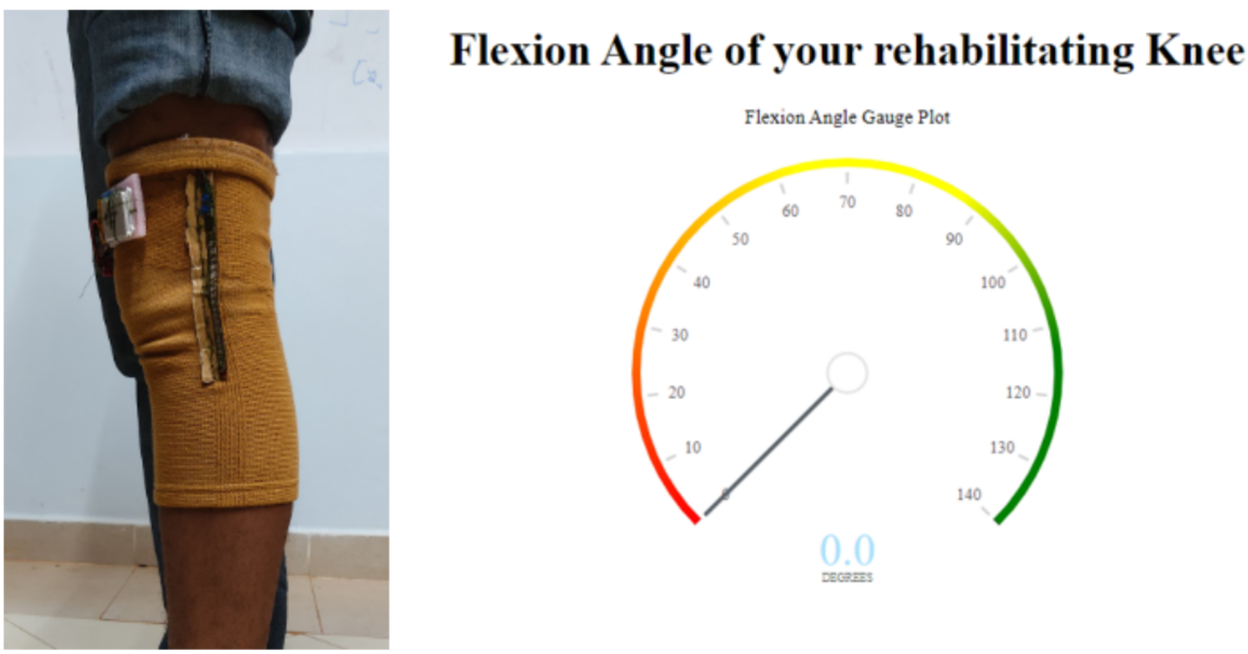

Fig. 8 Test of the prototype when standing

When sitting, the angle measured was 86. Multiple measurements produced values in the range of 80-90 degrees.

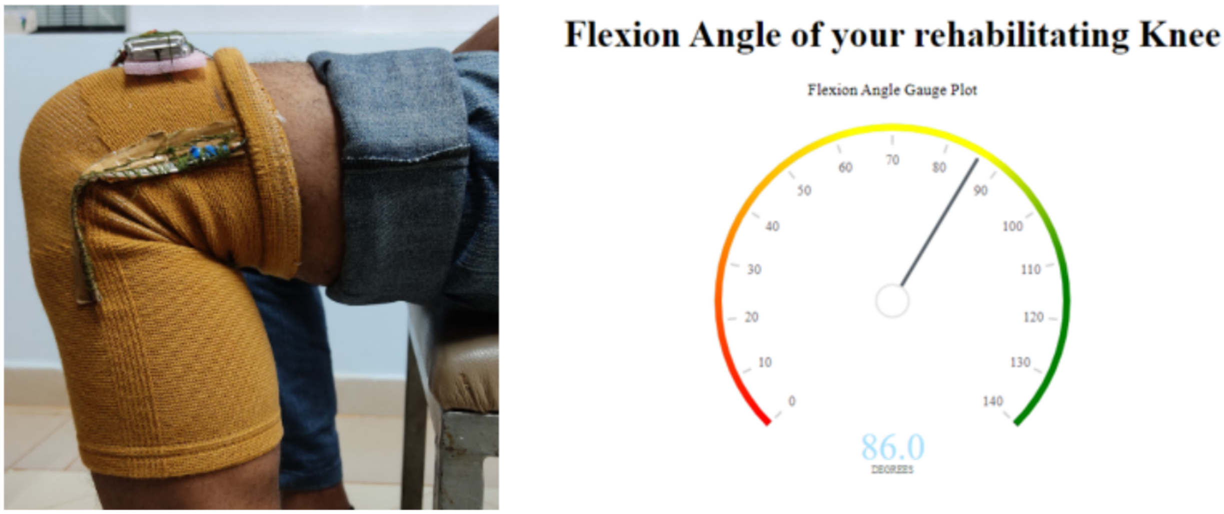

Fig. 9 test of the prototype when seated

While squatting, the readings were above 120 degrees. Multiple readings produced readings in the range of 120- 135 degrees.

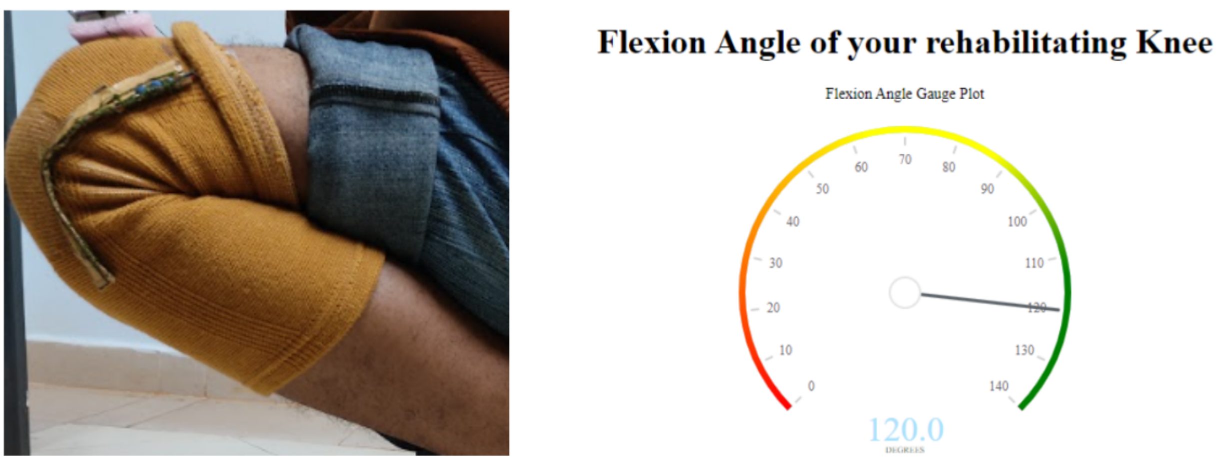

Fig. 10 test of the prototype when squatting.

From the above results, we were able to measure, record, and display the flexion angles of a healthy human's knee at different positions. However, we observed slight differences in the angles measured from multiple tests that we did. A common source of that variation in measurement was inconsistent bending of the flex sensor.

5. Future of work

Having developed the first version of the smart wearable knee device that can measure, record, and display flexion angle data in a real-time gauge plot, future work involves integrating more functionalities to get the gait data of a recovering patient. Gait data will give relevant information about the biomechanics of the recovering patient, informing on the most appropriate therapeutic interventions. Additionally, using this data to draw descriptive analytics using machine learning models to help doctors make more informed treatment decisions is also part of the future plan. We also plan to develop a mobile application that will substitute the python scripts.

6. Conclusion

We have developed a wearable device that can measure the knee flexion angle, save that data, and display it on a dash application. This project contributes to the continuing work of promoting affordable, lightweight, wearable devices that can monitor patients during their rehabilitation. The data collected by this device is important as a metric for measuring recovery from knee joint surgeries. With this system, it is already possible to store flexion angle data digitally in a database. The storage of this data is especially useful to researchers, hospitals, biomedical firms developing prostheses, or institutions promoting the advancement of remote monitoring of orthopedic patients.

Acknowledgment

This work has been possible because of the unrelenting support and mentorship offered by the Center for Data Science and Artificial Intelligence (DSAIL). Deep appreciation to the director of DSAIL, Dr. Ciira Maina for giving me the opportunity to pursue my attachment in the organization and for the continuous support, mentorship, and feedback throughout the device development. I am also grateful to DSAIL interns, and undergraduate research assistants who offered me an exciting working environment abundant in concern and care. Appreciation as well to our affiliated orthopedics expert, Dr. Gachathi Wanjema, the Chief Medical Officer at DeKUT and Orthopedics surgeon, for his expert opinion and user feedback during the project development.

References

[1] Frysh, P. (2020, September 1). *Body Weight And Knee Pain*. WebMD. Retrieved April 25, 2022, from https://www.webmd.com/pain-management/knee-pain/body-weight-knee-pain

[2] *Knee Pain and Problems*. (n.d.). Johns Hopkins Medicine. Retrieved April 25, 2022, from https://www.hopkinsmedicine.org/health/conditions-and-diseases/knee-pain-and-problems

[3] Chiang, C.-Y.; Chen, K.-H.; Liu, K.-C.; Hsu, S.J.-P.; Chan, C.-T. Data Collection and Analysis Using Wearable Sensors for Monitoring Knee Range of Motion after Total Knee Arthroplasty. *Sensors* 2017, *17*, 418. Sensors | Free Full-Text | Data Collection and Analysis Using Wearable Sensors for Monitoring Knee Range of Motion after Total Knee Arthroplasty

[4] *Arduino Pinout*. (2021, 11 3). Arduino. Retrieved April 24, 2022, from https://content.arduino.cc/assets/Pinout-NANOsense_latest.pdf (*Arduino Pinout*, 2021)

[5] Sparkfun. (n.d.). *Flex sensor*. https://www.sparkfun.com/datasheets/Sensors/Flex/flex22.pdf

[6] Bagur, J. (2022, April 11). *Connecting Nano 33 BLE Devices over Bluetooth® | Arduino Documentation*. Arduino Documentation. Retrieved April 24, 2022, from https://docs.arduino.cc/tutorials/nano-33-ble-sense/ble-device-to-device

[7] Davidson, R., Akiba, Cufí, C., & Townsend, K. (2014). *Getting Started with Bluetooth Low Energy*. O'Reilly.

[8] *BLEAK Documentation*. (2022, 01 24). bleak — bleak 0.14.2 documentation. Retrieved April 24, 2022, from https://bleak.readthedocs.io/en/latest/

[9] *Assigned Numbers | Bluetooth® Technology Website*. (2022). Bluetooth SIG. Retrieved April 24, 2022, from https://www.bluetooth.com/specifications/assigned-numbers/

[10]*Influxdb Documentation*. (n.d.). InfluxDB: Open Source Time Series Database | InfluxData. Retrieved April 24, 2022, from https://www.influxdata.com/

[11] *Dash Docs*. (n.d.). Plotly: The front end for ML and data science models. Retrieved April 24, 2022, from https://plotly.com/