Low-Budget Baropodometer Prototype Using ESP-WROOM-32 Modules

Gabriel Domenici Mozzer Arantes [a]

[a] Faculty of Electrical Engineering Emphasis on Robotics and Industrial Automation, Federal University of Juiz de Fora, Brazil, mozzer.gabriel@engenharia.ufjf.br

Abstract. This paper introduces a novel low-cost baropodometer device designed for foot pressure distribution analysis across various applications including clinical diagnostics, sports biomechanics, and rehabilitation. Emphasizing accessibility and accuracy, the device offers promising solutions for assessing gait abnormalities, foot posture, and biomechanical imbalances. The study meticulously details the prototype creation process, encompassing component selection and software development, while also addressing challenges encountered during its development. Following device testing, data analysis and diagnoses were conducted by a professional physiotherapist on two female patients. The project demonstrates effectiveness in its objectives, marking a advancement in the field of foot pressure analysis.

Keywords. ESP-WROOM-32, Low Budget, Baropodometer, Force Sensitive Resistor, FSR-402,Prototype.

1. Introduction

As the demand for foot plantar analysis rises across clinical, sports, and rehabilitation sectors, confronting the significant costs linked to the necessary equipment becomes paramount. While the total expense for the Project amounted to R$360,00, the average price of a baropodometer far exceeds this, reaching values that can exceed R$ 20.000. Elevated plantar pressures, as illuminated by research such as that by Menz et al (1) and Zammit et al. (2), can precipitate a range of foot ailments, from pain and stress fractures to calluses and neuropathic ulcers. Nonetheless, the accessibility of this crucial examination remains constrained by the prohibitive equipment costs, leaving treatment beyond the reach of many individuals.

In light of this, the primary objective of this paper is to develop and validate the efficacy of a low-cost baropodometer. By conducting tests on different patient demographics, with the expertise of a professional physiotherapist (DDM) aiding in data interpretation, the aim is to address the gap in affordable foot plantar analysis solutions.

2. Methodology

2.1 Materials and Methods

To design the prototype, an EVA insole was crafted to accommodate four FSR-402 sensors, each with a 10k resistance capable of withstanding up to 100 Newtons (N) of pressure. These sensors were strategically placed on specific areas of the feet (Figure 2), following recommendations from Tiago M. Jardim's work(3) and guidance from DDM. Data from the sensors was transmitted and interpreted using two ESP-WROOM-32 modules, which communicated via ESP-NOW protocol. One module was positioned above the EVA insole, which is housed within the Crocs footwear., while the other was connected to a computer to receive and interpret data collected by the four sensors.

The prototype (Figure 1) was developed to demonstrate the feasibility of creating an affordable and functional baropodometry device. To validate this concept, 2 individuals with identical shoe sizes with different heights and weights underwent a brief walk using the device. Following the walk, the sensor readings were depicted in a graph, and subsequently interpreted by the professional physiotherapist. Upon completion of the testing, the physiotherapist was able to diagnose whether the patients had or did not have a plantar problem.

Fig. 1 – Prototype image. **Fig.2–**Sensor placement. Available on reference [3].

2.2 FSR-402 Sensor

The main component of the project is the FSR-402 sensor. In short terms, it senses the pressure applied to it and varies its internal resistance depending on the magnitude of the force applied. Then, the voltage drop in the 10kohm resistance is converted into a PWM (Pulse-Width Modulation) value that the ESP32 reads and converts into a number that varies between 0 to 4095.

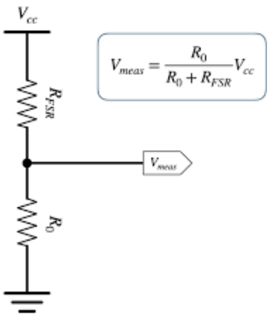

Fig. 3 – Graphic representation of a FSR-402 circuit. Available on: [Force Sensitive Resistors (FSRs) with Arduino — Maker Portal (makersportal.com)]

The equation provided in Figure 3 was obtained using fundamental circuit analysis techniques, with R0 set at 10kohms and Vcc at 5 volts. The output Vmeas represents the signal sent to the ESP32 port, returning the PWM value. Throughout this project, I've operated under the assumption of a linear relationship between the PWM value and the force applied to the sensor. However, for enhanced precision, alternative methods can be employed to precisely estimate the applied force. Our findings validate this approximation. For those seeking even more precise and detailed results, the graphic below, illustrating the correlation between the applied force and the Rfsr value, can be referenced.

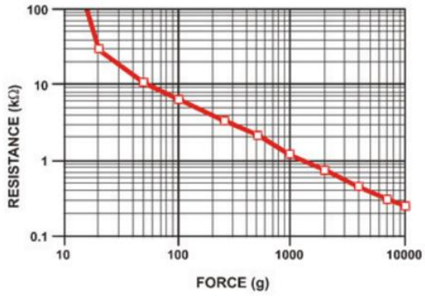

Fig. 4 – Graph of pressure-resistance correlation. Available on: https://cdn-learn.adafruit.com/downloads/pdf/force-sensitive-resistor-fsr.pdf

Utilizing this graphic represented on Figure 4, Lady Ada from the official Adafruit website created a code that calculates with more precision the pressure applied to the sensor.

2.3 The Software

The project's software is structured into two main components: communication between the ESP32 modules and data analysis performed through Python code. The ESP-NOW protocol was employed to facilitate device communication. This protocol was selected due to its simplicity in establishing communication through coding and its extensive communication range. Inspiration for its implementation was drawn from the Random Nerd Tutorials webpage (Available on: https://randomnerdtutorials.com/esp-now-esp32-arduino-ide/). Below, images of the ESP modules' codes are provided, with detailed comments outlining the steps for both sending and receiving functionalities respectively.

```
ODP_MANDAR.ino
#include <esp_now.h>
#include <WiFi.h>
// REPLACE WITH YOUR RECEIVER MAC Address
uint8_t broadcastAddress[] = {0xB0, 0xA7, 0x32, 0xF2, 0x93, 0x98};
// Structure example to send data
// Must match the receiver structure
typedef struct struct_message {
  int a,b,c,d;
} struct_message;
// Create a struct_message called myData
struct_message myData;
esp_now_peer_info_t peerInfo;
// callback when data is sent
void OnDataSent(const uint8_t *mac_addr, esp_now_send_status_t status) {
  Serial.print("\r\nLast Packet Send Status:\t");
  Serial.println(status == ESP_NOW_SEND_SUCCESS ? "Delivery Success" : "
}
void setup() {
  // Init Serial Monitor
  Serial.begin(115200);
  // Set device as a Wi-Fi Station
  WiFi.mode(WIFI_STA);
  // Init ESP-NOW
  if (esp_now_init() != ESP_OK) {
    Serial.println("Error initializing ESP-NOW");
    return;
  }
  // Once ESPNow is successfully Init, we will register for Send CB to
  // get the status of Trasnmitted packet
  esp_now_register_send_cb(OnDataSent);
  // Register peer
  memcpy(peerInfo.peer_addr, broadcastAddress, 6);
  peerInfo.channel = 0;
  peerInfo.encrypt = false;
  // Add peer
  if (esp_now_add_peer(&peerInfo) != ESP_OK){
    Serial.println("Failed to add peer");
    return;
  }
}
void loop() {
  // Set values to send
  myData.a = map(analogRead(33),0,4095,0,100);
  myData.b = map(analogRead(32),0,4095,0,100);
  myData.c = map(analogRead(34),0,4095,0,100);
  myData.d = map(analogRead(35),0,4095,0,100);
  Serial.println(myData.b);
  // Send message via ESP-NOW
  esp_err_t result = esp_now_send(broadcastAddress, (uint8_t *) &myData,
  if (result == ESP_OK) {
    Serial.println("Sent with success");
  }
  else {
    Serial.println("Error sending the data");
  }
  delay(2000);
}
```

Fig. 5 – ESP-WROOM-32 Sender code.

```
ODP_MANDAR.ino
#include <esp_now.h>
#include <WiFi.h>
// REPLACE WITH YOUR RECEIVER MAC Address
uint8_t broadcastAddress[] = {0xB0, 0xA7, 0x32, 0xF2, 0x93, 0x98};
// Structure example to send data
// Must match the receiver structure
typedef struct struct_message {
  int a,b,c,d;
} struct_message;
// Create a struct_message called myData
struct_message myData;
esp_now_peer_info_t peerInfo;
// callback when data is sent
void OnDataSent(const uint8_t *mac_addr, esp_now_send_status_t status) {
  Serial.print("\r\nLast Packet Send Status:\t");
  Serial.println(status == ESP_NOW_SEND_SUCCESS ? "Delivery Success" : "Delivery Fail");
}
void setup() {
  // Init Serial Monitor
  Serial.begin(115200);
  // Set device as a Wi-Fi Station
  WiFi.mode(WIFI_STA);
  // Init ESP-NOW
  if (esp_now_init() != ESP_OK) {
    Serial.println("Error initializing ESP-NOW");
    return;
  }
  // Once ESPNow is successfully Init, we will register for Send CB to
  // get the status of Trasnmitted packet
  esp_now_register_send_cb(OnDataSent);
  // Register peer
  memcpy(peerInfo.peer_addr, broadcastAddress, 6);
  peerInfo.channel = 0;
  peerInfo.encrypt = false;
  // Add peer
  if (esp_now_add_peer(&peerInfo) != ESP_OK){
    Serial.println("Failed to add peer");
    return;
  }
}
void loop() {
  // Set values to send
  myData.a = map(analogRead(33),0,4095,0,100);
  myData.b = map(analogRead(32),0,4095,0,100);
  myData.c = map(analogRead(34),0,4095,0,100);
  myData.d = map(analogRead(35),0,4095,0,100);
  Serial.println(myData.b);
  // Send message via ESP-NOW
  esp_err_t result = esp_now_send(broadcastAddress, (uint8_t *) &myData, sizeof(myData));
  if (result == ESP_OK) {
    Serial.println("Sent with success");
  }
  else {
    Serial.println("Error sending the data");
  }
  delay(2000);
}
```

Fig. 6 – ESP-WROOM-32 Reciever code.

After the receiver module captures the information, it will then transfer the data to a Python code responsible for plotting a graph correlating the pressure applied on each sensor with the specific time the same pressure was applied. The code can be seen down below

Recieve.py

Receber data > Recieve.py > ...

```python
import time
import serial
import numpy as np
import matplotlib.pyplot as plt
#Declaring the vectors that will be used in the sensor dat
leitura1 = []
leitura2 = []
leitura3 = []
leitura4 = []
def media(leitura):
    return np.mean(leitura)
#Declaring in wich port the ESP32 board is connected
sensores = serial.Serial('COM3', 115200)
#Creating the figure where the data will be shown
fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
#Creating a Loop that will gather the ESP32 information in
while True:
    #If the program doesnt recieve a string the code will
    while(sensores.inWaiting()==0):
        pass
    #Capting the ESP32 string and converting it into a lis
    for t in range(10):
        leitura = sensores.readline()
        leitura = str(leitura,'utf-8')
        leitura = leitura.strip('\r\n')
        valores = leitura.split(',')
        for i in range(4):
            valores[i]=float(valores[i])
        print(valores)
        leitura1.append(valores[0])
        leitura2.append(valores[1])
        leitura3.append(valores[2])
        leitura4.append(valores[3])
        time.sleep(1)
    #plots the graph
    media_leitura1 = media(leitura1)
    media_leitura2 = media(leitura2)
    media_leitura3 = media(leitura3)
    media_leitura4 = media(leitura4)
    print(f"Calcanhar:{media_leitura1}")
    print(f"Metatarso da direita:{media_leitura2}")
    print(f"Metatarso da esquerda:{media_leitura3}")
    print(f"Sensor meio do pé:{media_leitura4}")
    tempo=np.arange(1,11,1)
    ax.set_xlabel('Tempo')
    ax.set_ylabel('Leitura dos sensores')
    plt.plot(tempo, leitura1, color='r', label='Calcanhar'
    plt.plot(tempo, leitura2, color='g', label='Metatarso
    plt.plot(tempo, leitura3, color='b', label='Metatarso
    plt.plot(tempo, leitura4, color='m', label='Sensor mei
    plt.legend()
    plt.show()
    break
```

Fig. 7 – Python data processing code.

2.4 Other Studies

In their study, Jor et al(4) developed a low-cost prototype for foot plantar measurement, utilizing an FSR sensor. Their research aimed to assess the impact of various types of heels, each with differing heights. To achieve this, they recruited 8 female participants with typical characteristics: an average age of 21 years, height of 159 cm, and weight of 56 kg.

Following the tests, an increase in plantar pressure was noted with the rise in heel height. This variation was subsequently compared with findings from other studies by Cong et al (5), Yung-Hui et al(6), and Gu et al(7). It was concluded that the results closely resembled those obtained using more expensive baropodometry devices in prior research.

Grenez et al (8), have collaborated on the development of a prototype aimed at creating a technological solution to evaluate the walking capabilities of individuals affected by neurological disorders that impact their gait. These disorders may include Parkinson's disease, multiple sclerosis (MS), or cerebral palsy.

To create the device they used an Arduino mini of 3.3 volts, and to do the data anylis they used 3 flexi-force 100 lbs sensors to registrate the preassure points on the feet and 2 bending sensors to know the angle formed by the ankle and the metatarsals.

Their analysis unfolds across three distinct steps. Initially, participants maintain a stationary position, offering a baseline for pressure sensors to measure the pressure exerted solely by their weight. Following this, participants take six steps, ensuring three complete steps with the left foot, scrutinizing sensor functionality and capturing a sample of typical walking patterns. Finally, participants embark on a 7.5-meter walk, followed by a return journey of the same distance, instructed to walk naturally while timing the three stages of the test: the initial 7.5 meters, the turnaround duration, and the subsequent 7.5 meters. After this, they have realized the same test with the patients dragging their feet.

The researchers have determined that the results indicate the feasibility of utilizing the standard deviation as a metric to differentiate individuals exhibiting foot drag. This parameter was selected due to its consistent pattern across all participants: heightened standard deviation during normal walking compared to foot dragging. Additionally, the study observed an increase in the walking speed of individuals during instances of foot dragging; however, a universal threshold for differentiating users based on speed was not identified.

2.5 Difficulties

Despite the successful development of the prototype, several challenges arose regarding both the software and hardware components of the baropodometer. While the software exhibited functionality, achieving precision in identifying pressure applied on the FSR402 sensors remained elusive. Attempts were made to implement code referenced in the FSR datasheet provided by Adafruit Industries, yet without success. Additionally, efforts to generate a heatmap of the feet using the software proved daunting. Although a Python library called Seaborn, known for its capabilities in creating heatmaps with collected data, was identified, integrating this functionality into a foot diagram posed difficulties.

The principal challenge encountered with the prototype stemmed from its hardware configuration. Employing protoboards and solder jumpers led to instability during testing, as some wires became disconnected. Consequently, only two out of the three selected subjects underwent effective analysis during the trial runs. Upon the initiation of the third trial, a wire connecting the sensor to the ESP32, responsible for data transmission, became dislodged. This resulted in erratic readings, rendering accurate foot plantar analysis unattainable. A potential resolution involves the development of a printed circuit board (PCB) to minimize the number of wires utilized in the project. My development team, embarked on a PCB prototype for implementation in the project; however, due to time constraints, the prototype remains unfinished.

Another issue arose due to the utilization of sensors from different manufacturers. Despite having identical specifications, it was observed that sensors from one manufacturer exhibited greater sensitivity compared to those from another manufacturer.

3. Results and Discussion

3.1 How the Test Works

The purpose of the paper is to showcase the effectiveness of a cost-friendly baropodometer in analyzing foot plantar gait. To achieve this objective, a Python program was developed to process data transmitted by the sensors at 10-second intervals. During these intervals, the subject is instructed to walk normally while the data is harnessed for later analysis. The program generates a graphical representation that correlates sensor readings with their respective timestamps. Once the graph is generated, the program displays the average measurements for each sensor reading. The complete Python program is outlined in section 2.3, focusing on the software aspect. After data collection, a qualified physiotherapist analyzes the results. In this study, a professional physiotherapist was responsible for performing the professional evaluation.

Two female individuals (FSS, IRCT) were selected to perform the tests. Their weight, height and shoe size of them are displayed in the table below.

Tab. 1 – Subjects information.

Name	Age	Height	Weight	Brazilian Shoe Size
FSS	35	1,61m	74,3Kg	37
IRCT	30	1,71m	124,3Kg	37

3.2 Data Analysis and Discussion

Following the testing phase, the data presented in the table below was collected, along with corresponding graph images illustrating the pressure applied on the sensors during the 10-second intervals.

Tab. 2 – Average pressure readings from each sensor during the test.

Patient	Calcaneus Bone	First Metatarsal	Fourth Metatarsal	Midfoot Region
FSS	28,3 N	2,0 N	38,7 N	49,6 N
IRCT	42,5 N	3,6 N	10,6 N	76,6 N

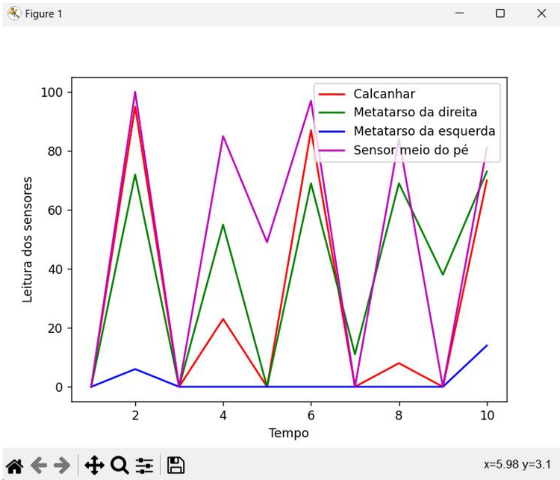

Fig. 8- FSS sensor readings.

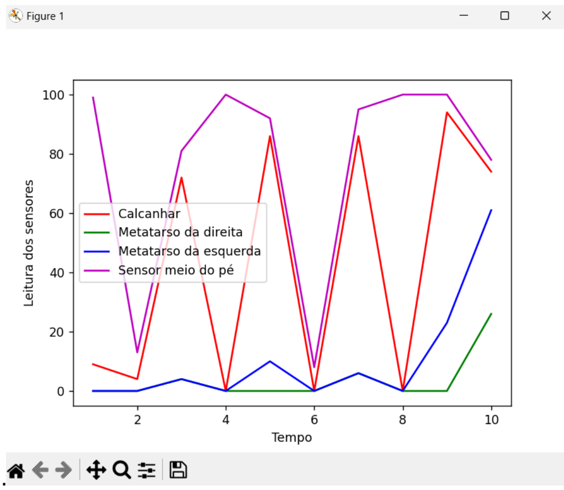

Fig. 9 – IRCT sensor readings .

It's important to note that in the images, the names of the sensor positions and graph labels are in Portuguese. The translations to English are as follows: "Calcanhar" corresponds to "Calcaneus Bone", "Metatarso da direita" translates to "Right Metatarsal", "Metatarso da esquerda" translates to "Left Metatarsal", and "Sensor meio do pé" corresponds to "Midfoot Region".

After analyzing the data, the physiotherapist (DDM) concluded that the first patient, FSS, exhibits a tendency towards a more supinated foot strike, accompanied by reduced activation of the left metatarsal sensor during the contact phases of the plantar region. This finding suggests a predisposition to ankle valgus stress, consequently altering pressure distribution during the gait phases. Following the diagnosis of the second patient, who presents a collapsing pattern of the plantar arch, commonly known as flatfoot or fallen arches, it was observed that greater pressure is concentrated on the midfoot electrode during gait phases, while the sensors on the left and right metatarsals show minimal activation. This observation supports the notion of decreased arch support, which in turn justifies the heightened pressure recorded by the midfoot sensor.

Following the tests, the physiotherapist (DDM) concluded that the prototype demonstrated its efficacy. It enables the identification of various pressure zones on the feet during walking, aiding in biomechanical correction, and the prevention or treatment of foot or ankle injuries caused by misalignment.

4. Conclusion

This paper elaborates on the creation and functionality of a low-cost baropodometer, highlighting its development process, encountered challenges, and the effectiveness demonstrated by the prototype. The device measures pressure distribution and conducts gait analysis during walking or standing, employing 2 ESP-WROOM-32 modules and 4 FSR-402 sensors. Collaborating with professional physiotherapist (DDM) it was concluded that despite inherent limitations such as hardware issues, the prototype has proven useful for diagnosing patients.

5. Acknowledgment

The author extends sincere gratitude to their colleagues from PET Elétrica UFJF, particularly their tutor Danilo Pereira Pinto, whose encouragement and support were instrumental in the pursuit and completion of this project. Special appreciation is also extended to colleagues Gabriel da Silva Ribeiro, Mariana Luciano Saar, Julia Almeida da Silva Pereira, and Wylker de Oliveira Alves for their valuable contributions.

Daniel Domenici Mozzer (DDM), the professional physiotherapist, played a crucial role in the project by supplying the patients for the testing phase and contributing to the data interpretation presented in the study.

Furthermore, heartfelt thanks are expressed to the author's parents, Consuelo Domenici Mozzer Pinto and Sergio Paulo dos Santos Pinto, for providing the opportunity to enroll in the INCBAC Unigou remote program.

6. References

1. Menz HB, Morris ME. Clinical determinants of plantar forces and pressures during walking in older people. Gait Posture. 2006;24(2):229-36.
2. Zammit GV, Menz HB, Munteanu SE. Reliability of the TekScan MatScan(R) system for the measurement of plantar forces and pressures during barefoot level walking in healthy adults. J Foot Ankle Res. 2010;3:11.
3. Jardim TM FA, Affeldt D, Júnior JSD, Amaral EMH. Projeto e desenvolvimento de um prototipo de baropodômetro. 2023. Available in: https://doi.org/10.5753/sbcas_estendido.2023.229657. Accessed on 12/04/2024.
4. A. Jor SD, A. S. Bappy, A. Rahman. Foot Plantar Pressure Measurement Using Low Cost Force Sensitive Resistor (FSR): Feasibility Study. Journal of Scientific Research. 2019;11:311-9.
5. Cong Y, Cheung JT, Leung AK, Zhang M. Effect of heel height on in-shoe localized triaxial stresses. J Biomech. 2011;44(12):2267-72.
6. Yung-Hui L, Wei-Hsien H. Effects of shoe inserts and heel height on foot pressure, impact force, and perceived comfort during walking. Appl Ergon. 2005;36(3):355-62.
7. Gu Y RM, Ruan G. The Outsole Pressure Distribution Character during Highheeled Walking. Procedia Environmental Sciences. 2011;8:464-8.
8. Grenez F, Viqueira Villarejo M, Garcia Zapirain B, Mendez Zorrilla A. Wireless prototype based on pressure and bending sensors for measuring gait [corrected] quality. Sensors (Basel). 2013;13(8):9679-703.