Microprocessors & Interfacing

Course Code: CSE2006

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

PEDOMETER USING ARDUINO

Team Members	Ohm Patel (19BCE0151) Mounvi Podapati (19BCE0396) Srishti Gohain (19BCE0435) V.S.R.P.V.K. Varun (19BCE0401)
Slot	L47 + L48
Faculty	Prof. Karthikeyan M

Pedometer using Arduino

Ohm Patel (19BCE0151), Mounvi Podapati (19BCE0396), Srishti Gohain (19BCE0435), V.S.R.P.V.K. Varun (19BCE0401)

Abstract

The incessant need to cater to the general public's health and welfare has only skyrocketed in the past few years. Especially with the advent of the pandemic and the unprecedented events that have been making headlines in the last few months, it is a well known fact that individual initiatives to maintain health and strength are the only possible way to tackle the lifestyle changes that have been imposed upon the larger part of the population. Health monitoring systems have long been in the business, however it is high time to reform and redesign the rigid and inflexible architectures of existing systems to make them better, accessible and accurate for mass usage. In this report we explored a cost-effective, reliable and extremely responsive approach to the pedometer problem. Our research has shown us that accelerometers are the best contenders for the 'step-count' situation and we have also simulated the same with the help of affordable, cost-effective options. This ensures that a gadget as fundamental as a pedometer is inclusive and is not influenced by age, stride length, angle with respect to the earth, or any other such parameter that conceptually should not interfere with pedometric calculations in the first place.

I. Introduction

Step counter is an automatic detection of walking steps taken by a human. It is a part of human activity recognition or better walking detection. Step counting is becoming a very important role in the healthcare system to monitor the human's daily routine. In the current age of technology, step counting technique using built-in sensors in a smartphone has been becoming progressive and important area due to coming with many embedded sensors such as accelerometer, gyroscope, magnetometer which provide valuable information of user's contexts and activities. Many researchers have approached the step counting techniques by built-in a variety of sensors in mobile phones with different points of views. However, there are many challenges to motivate a new step counter technique under more realistic conditions like performing multiple tasks while walking at the same time, the senros receiving a lot of noisy data, and changing the sensor position during motion. As well as step counting is very important to monitor human health, it can provide many applications such as many medical fields, fitness tracking centers, human activity recognition systems and so on. In the past, more reliable and electronic devices called pedometers were utilized, especially in medical applications. In recent years, pedometers are used as the electronic step counting devices, which are attached to waist, foot, or arms. Currently, step counting has become an increasing research area with the powerful capacities of smartphone sensors, including gyroscope and accelerometer. Typically, the accelerometer and gyroscope sensors are highly effective for step counting. Both sensors are suitable for motion analysis, but accelerometer is more popular as its ability detects the rate of change in motion activity.

This project involves the designing of a fundamental and competitive arduino based pedometer (using an accelerometer) for observing, assessing and making reports for measures of physical activity. The main motivation for this idea comes from the fact that during the Covid 19 pandemic, a large section of the population has been living a very sedentary lifestyle. It is crucial to recognize physical activity as a means to reduce health risks and future diseases. This device will help the user monitor various data such as step count and the lost calories. This data can be analyzed in a detailed manner to understand one’s body and metabolism and improve upon one’s daily habits.

II. Literature Survey

In their paper, Brian Y Lim stated that the use of a step counter has enhanced the participant being more active than the ones that are in other control groups who were not using their device, Pediluma. This can implicate the impact that using a step counter has on an individual using it. It also shows the positive effect on an individual using it and also it gives us a path to confirm that this can be a device that can be used by people to track their health and fitness everyday [8].

In the paper [1], Fontana G has shared that there have been hundreds of smartphone pedometer mobile applications in the market that are available, but only very few of those applications actually deliver accurate results and those are in a very small number. By this, it can be assumed that the pedometer that is a separate device and has only that specific purpose is needed to obtain an accurate result as the aim is to monitor people’s activity and turn to monitor their health in an exact and error free manner.

Wong R.C.P and their co-authors, in their paper analyzed through a survey, the activity of the people who use pedometers and pedometer apps, and the non-users, and the results obtained show the user’s perspective. It can be seen in the results that the people using the pedometers and pedometer apps tend to choose daily routes that have scope to walk more in order to be fit and the non-users tend to take a routine less distanced route [2]. So it is also seen that there is a large section of people who are already promoting the usage of the pedometers and also advocating the medical and health benefits that they get through using this device, indirectly, and this strongly shows the requirement of a cost effective and a reliable wearable pedometer for common people to benefit from using it.

Apurva Karandikar along with other authors wrote about the importance of the pedometers for athletes in the gymnasiums where keeping a track of their physical fitness is very essential. The authors developed a fitness tracking system that interacts with the pedometer and the data is recorded, and then analysed for future predictions [9].

Manish B. Nair discussed in their paper about their prototype of the pedometer that also has advanced features like GPS tracker and bluetooth along with a heart rate monitoring system, which alerts the person’s relatives in text messages by sending the user's location [4]. Though this is an advanced version of a pedometer, it would be very costly to be produced. The heart rate monitoring is useful but inducing it into the device makes it very expensive. As the main purpose of our device is to be affordable and cost effective, and more common people need to benefit from it, our proposed device will be more useful and user friendly.

III. Existing Approaches and Systems

Many existing pedometer systems make use of various approaches - Tilt Sensors, real time GPS tracking etc.

- Using a tilt sensor is not very accurate with respect to pedestrian movement calculation and monitoring.
- GPS tracking methods have measurement constraints and the steps counted during jogging can't be measured.
- Inclinometers measure the angular velocity but cannot keep track of movements along the three different axes at the same time.
- Gyroscopes are expensive motion sensors. It does not measure linear motion in any direction, or any static angle of orientation. It can only be applicable in measuring slow paced walking and movements.

IV. Disadvantages of Existing Systems

In general, existing pedometer systems and other health-monitoring gadgets have a number of issues in common. These can be listed out as:

- Needs to be charged very often as it drains a lot of battery
- No consideration on length of your individual walking and running stride
- High dependence on speed of walking, hence less accuracy during a slow walk.
- Most pedometers don’t work accurately while running or jogging
- Not very helpful for comparing people of different ages

Going into detail about the shortcomings of different principle components that are often seen as the ‘heart’ of pedometer systems’ architecture:

Tilt Sensors and Switches

Fig. 1(a). Mercury based Tilt Sensor

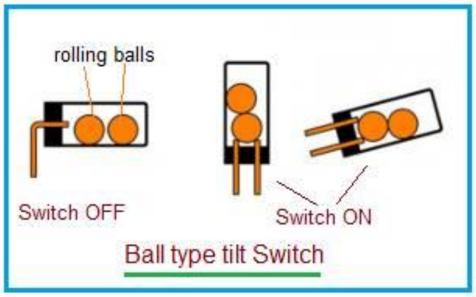


Fig. 1(b). Ball type tilt sensor

Mercury based tilt sensor:

- Mercury is hazardous due to its toxic nature.

Ball type tilt sensor:

- When this tilt switch is exposed to current beyond its rated specification, arcing may occur, causing its contacts to degrade and becoming less reliable.

- As the balls vibrate from one place to the next, irregular or unpredictable signals are produced.
- They can't function without gravity (i.e. to roll a ball or to move a blob of mercury). As a result, these switches will not function in low gravity, reversed gravity, or zero gravity.
- Tilt switch performance in a gadget that accelerates or decelerates, like in our case, pedometer, quickly also turns out to be unreliable.
- When there is a lot of vibration, tilt switches give erroneous results.

Inclinometer

Fig. 2. Inclinometer

Studies have frequently shown that inclinometers do not give reliable and accurate results in terms of simple parameters like ‘step-count’ in normal living situations. This one again plays into the fact that such devices are not ‘age’ and ‘demographic’ inclusive. The limits of agreement about pedometers based on accelerometers and inclinometers are huge. M.Oliver et al’s [6] study showed that the accuracy and agreement rates for the chosen accelerometers and inclinometers were 123% and 92% respectively. Inclinometers by design are used to measure the angle of a subject with respect to gravity. It clearly misses to record a significant amount of ‘real steps’ because it is actually looking for the angular velocities. As a result, it cannot keep track of the acceleration along the three axes.

Gyroscopes

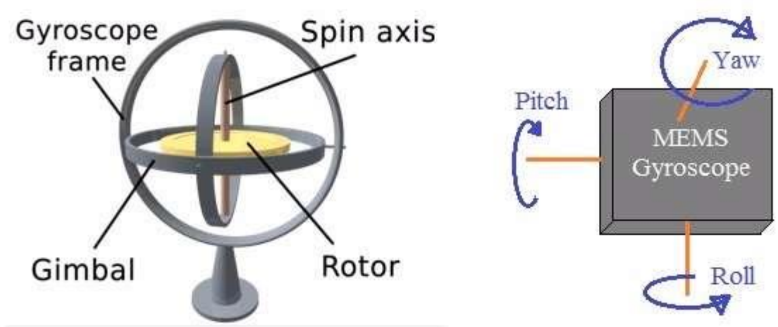


Fig. 3. Gyroscope Model

Gyro sensors, also known as angular rate sensors or angular velocity sensors, are devices that sense angular velocity.
- They are more expensive alternatives to navigation and tilt sensing applications.

- Free moving gyroscope type is always dependent on rotation of the Earth. Hence fast moving objects moving on trajectory from the east to the west cannot use gyroscopes for navigation purposes.
- It does not measure linear motion in any direction, or any static angle of orientation.
- It is subjected to relative azimuth drift unlike compass.
- Jayalath et al's study [7] showed that gyroscopes only showed significant results in terms of slow-paced walking and low movements.

V. Key Components:

A. ADXL335 Accelerometer

The ADXL335 is a small, thin, low power, complete 3-axis accelerometer with signal conditioned voltage outputs. The product measures acceleration with a minimum full-scale range of ±3 g. It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration. The user selects the bandwidth of the accelerometer using the CX, CY, and CZ capacitors at the XOUT, YOUT, and ZOUT pins. Bandwidths can be selected to suit the application, with a range of 0.5 Hz to 1600 Hz for the X and Y axes, and a range of 0.5 Hz to 550 Hz for the Z axis.

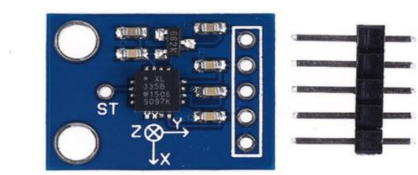


Fig. 4. ADXL335

B. LCD screen (16x2)

An LCD (Liquid Crystal Display) screen is an electronic display module and has a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. The 16 x 2 intelligent alphanumeric dot matrix display is capable of displaying 224 different characters and symbols. Uses like Alphanumeric output, Information Display, Process status, in short we can keep eye on every move of our microcontroller.

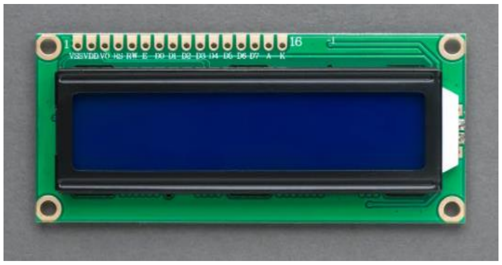


Fig. 5. 16x2 LCD

C. LCD I2C Module

I2C Module has an inbuilt PCF8574 I2C chip that converts I2C serial data to parallel data for the LCD display. With this I2C interface LCD module, you only need 2 lines (I2C) to display the information.

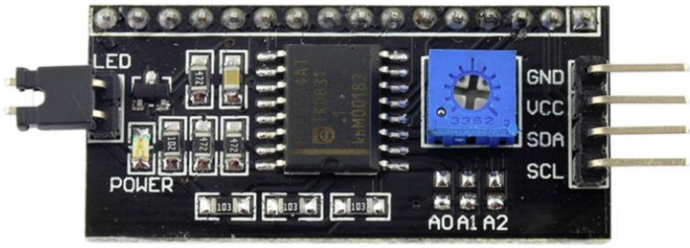


Fig. 6. LCD I2C Module

D. Arduino NANO

Arduino Nano is a small, complete, flexible and breadboard-friendly Microcontroller board, based on ATmega328p, developed by Arduino.cc, contains 30 male I/O headers, configured in a DIP30 style. Arduino Nano Pinout contains 14 digital pins, 8 analog Pins, 2 Reset Pins & 6 Power Pins. It is programmed using Arduino IDE. Arduino Nano is simply a smaller version of Arduino UNO, thus both have almost the same functionalities. It comes with an operating voltage of 5V, and however, the input voltage can vary from 7 to 12V.

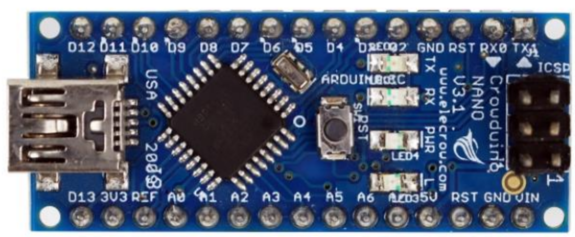


Fig. 7. Arduino Nano

E. Jumper Wires:

Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed

Fig. 8. Male to Male Jumper wires

F. Breadboard:

A thin plastic board used to hold electronic components (transistors, resistors, chips, etc.) that are wired together. Used to develop prototypes of electronic circuits, breadboards can be reused for future jobs. The components and jump wires (assorted wire lengths with pins at both ends) are plugged into the clips to create the circuit patterns.

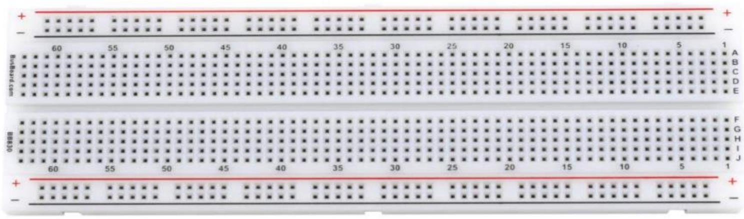


Fig. 9. Breadboard

VI. Proposed Architecture / Methodology

This project requires the usage of a microcontroller such as Arduino NANO as well as the Arduino IDE to write its code.
To detect the motion of stops and count the number, we have used an accelerometer, the ADXL335 Accelerometer.
After counting the number of steps we have used that data to find the calories burned through simple formulae.
Lastly, all the data will be displayed on the serial monitor as well as a web application.
We are using the ADXL335 Accelerometer which continuously updates the maximum and minimum values of the 3-axis acceleration after every defined no. of samples.
The average value of these 3-axis (Max + Min)/2, is called the dynamic threshold level, and this threshold value is used to decide whether the step is taken or not.
While running, the pedometer can be in any orientation, so the pedometer calculates the steps using the axis whose acceleration change is the largest.

- For the web-app we have used nodeJS and expressJS for server side scripting.
- We have styled the document using standard CSS and built it using HTML.
- We have fetched the data that is being printed to the arduino serial monitor and displayed it in the webapp.
- Additional details like calories burnt and daily progress are calculated through the webapp itself.

VII. Working Algorithm:

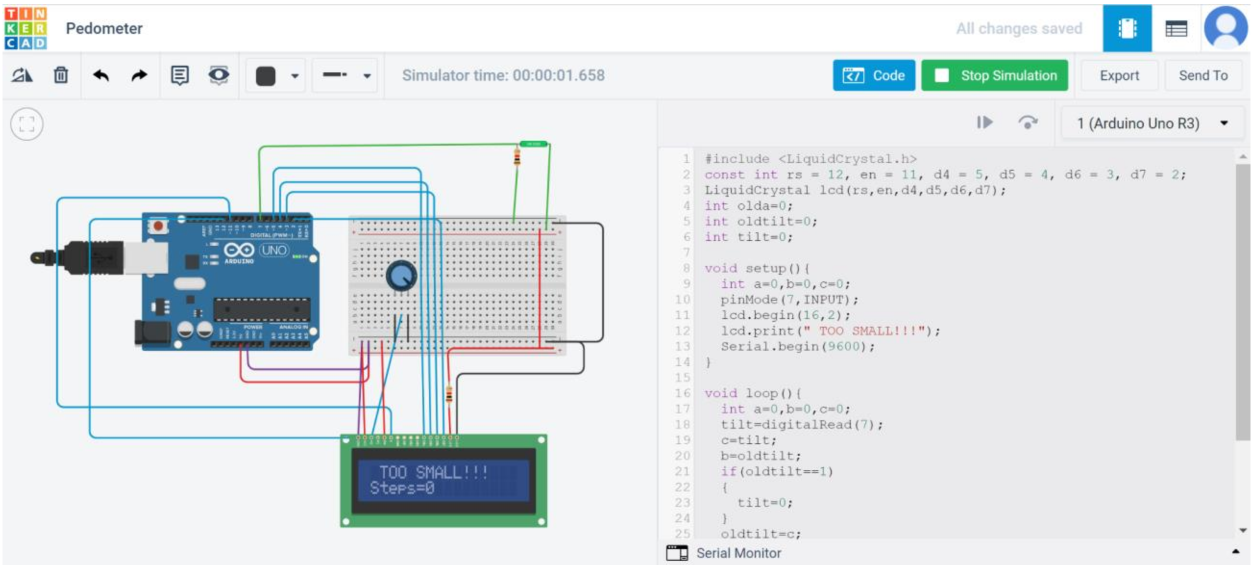
1. Firstly the pedometer starts the calibration as soon as it gets powered.
2. Then in the void loop function, it continuously gets the data from X, Y, and Z-axis.
3. After that, it calculates the total acceleration vector from the starting point.

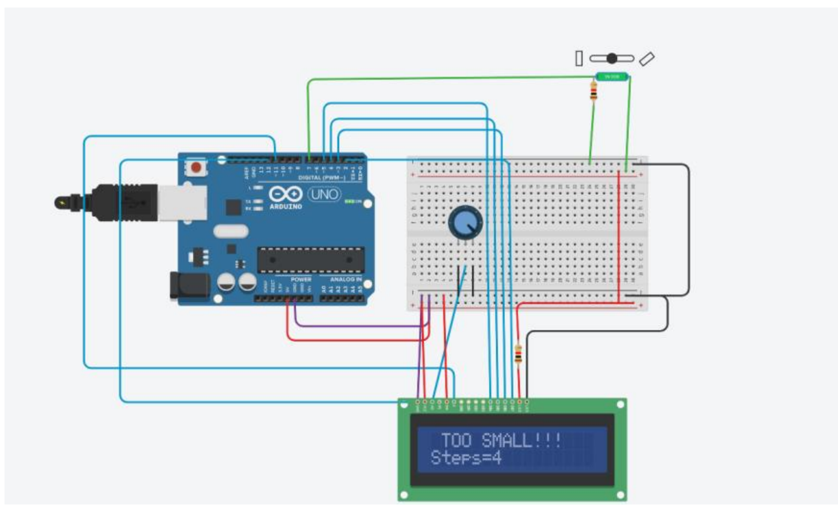
4. Acceleration vector is the square root (x^2+y^2+z^2) of the X, Y, and Z-axis values.
5. Then it compares the average acceleration values with the threshold values to count the step number.
6. If the acceleration vector crosses the threshold value, then it increases the step count; otherwise, it discards the invalid vibrations.
7. Sends the data to the Node Application to visualise the number of steps.

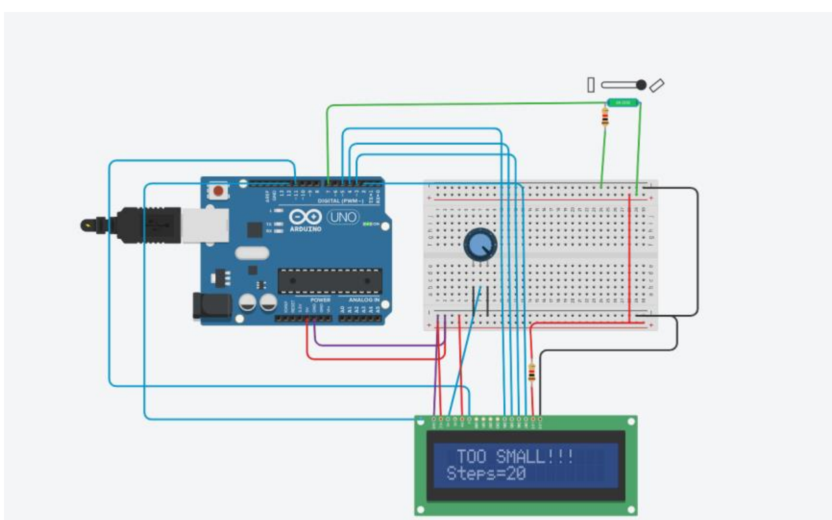
VIII. Software Simulation:

A. Tinkercad Code

```
#include <LiquidCrystal.h>
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs,en,d4,d5,d6,d7);
int olda=0;
int oldtilt=0;
int tilt=0;
void setup(){
  int a=0,b=0,c=0;
  pinMode(7,INPUT);
  lcd.begin(16,2);
  lcd.print(" TOO SMALL!!!");
  Serial.begin(9600);
}
void loop(){
  int a=0,b=0,c=0;
  tilt=digitalRead(7);
  c=tilt;
  b=oldtilt;
  if(oldtilt==1)
  {
    tilt=0;
  }
  oldtilt=c;
  delay(1);
  a=olda+tilt;
  lcd.setCursor(0,1);
  lcd.print("Steps=");
  lcd.print(a);
  olda=a;
}
```


Pedometer
All changes saved
Simulator time: 00:00:01.658
Code
Stop Simulation
Export
Send To
1 (Arduino Uno R3)
#include <LiquidCrystal.h>
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs,en,d4,d5,d6,d7);
int olda=0;
int oldtilt=0;
int tilt=0;
void setup(){
int a=0,b=0,c=0;
pinMode(7,INPUT);
lcd.begin(16,2);
lcd.print(" TOO SMALL!!!");
Serial.begin(9600);
}
void loop(){
int a=0,b=0,c=0;
tilt=digitalRead(7);
c=tilt;
b=oldtilt;
if(oldtilt==1)
{
tilt=0;
}
oldtilt=c;
Serial Monitor
TOO SMALL!!!
Steps=0


TOO SMALL!!!
Steps=4


TOO SMALL!!!
Steps=20

We have simulated a pedometer using Tinkercad with the following available sensors and components:

- **Tilt sensor**
- **Potentiometer**
- **Arduino Uno**
- **Resistors**
- **LCD Screen**

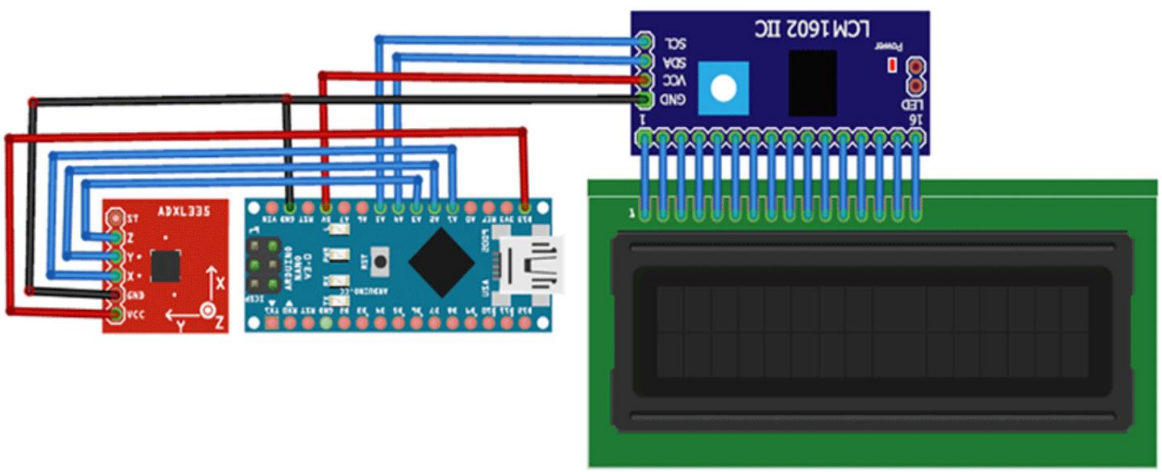
We have deduced the following from the above simulation:

- Steps keep increasing on just tilting the senor,
- The tilt sensor will still increase the number of steps if kept in a vehicle that accelerates or decelerates.
- The tilt sensor delivers erroneous results under significant vibrations conditions and sometimes counts steps under insignificant conditions as well.
- Few experiments also required titling the sensor more than 85 degrees in order for the step count to be increased.

This indicates that a step counter application using tilt sensors is very inaccurate and unreliable.

IX. Hardware Simulation:

A. Circuit Diagram:



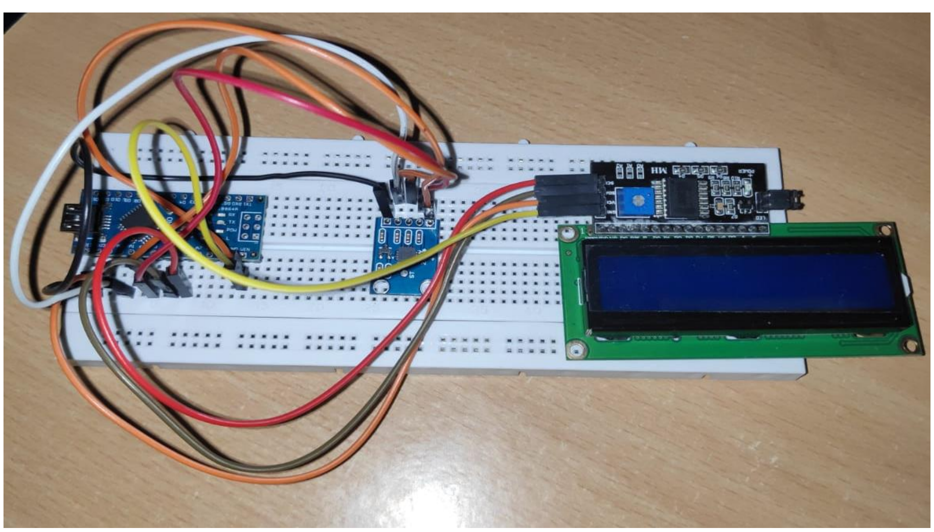
B. Arduino Code:

```
int xpin=A1;
int ypin=A2;
int zpin=A3;
float threshhold=220.0;
float xval[100]={0};
float yval[100]={0};
```

```
 float zval[100]={0};
 float xavg;
 float yavg;
 float zavg;
 int steps = 0;
 int state=0;
 void setup()
 {
 Serial.begin(9600);
 calibrate();
 }void loop()
 {
 int acc=0;
 float totvect[100]={0};
 float totave[100]={0};
 float xaccl[100]={0};
 float yaccl[100]={0};
 float zaccl[100]={0};
 for (int i=0;i<100;i++){
 xaccl[i]=float(analogRead(xpin));
 yaccl[i]=float(analogRead(ypin));
 zaccl[i]=float(analogRead(zpin));
totvect[i] = sqrt(((xaccl[i]-xavg)* (xaccl[i]-xavg))+ ((yaccl[i] - yavg)*(yaccl[i] - yavg)) + ((zval[i] -
zavg)*(zval[i] -zavg)));
totave[i] = (totvect[i] + totvect[i-1]) / 2 ;
//  Serial.println(totave[i]);
 delay(1000);
 //calculating number of steps
 if (totave[i]>threshhold && state==0){
 steps=steps+1;
 state=1;
 }
 if (totave[i] <threshhold  && state==1)
 {state=0;}
//  Serial.println('\n');
//  Serial.print("steps=");
 Serial.println(steps);
 };
 delay(1000);
 }
 void calibrate(){
 float sum=0;
```

```
  float sum1=0;
  float sum2=0;
  for (int i=0;i<100;i++){
  xval[i]=float(analogRead(xpin));
  sum=xval[i]+sum;
  }
  delay(100);
  xavg=sum/100.0;
//  Serial.println(xavg);
  for (int j=0;j<100;j++){
  xval[j]=float(analogRead(xpin));
  sum1=xval[j]+sum1;
  }
  yavg=sum1/100.0;
//  Serial.println(yavg);
  delay(100);
  for (int i=0;i<100;i++){
  zval[i]=float(analogRead(zpin));
  sum2=zval[i]+sum2;
  }
  zavg=sum2/100.0;
  delay(100);
//  Serial.println(zavg);
  }
```

C. Hardware Setup Snapshot:



X. Web Application

A. Node App Code:

(1) Index.js

```
const express = require("express");
const app = express();
var SerialPort = require("serialport");
var arduinoCOMPort = "COM6";
var data = 0;
const port = 5000;
app.set('view engine', 'ejs');
app.use(express.static("public"))
var arduinoSerialPort = new SerialPort(arduinoCOMPort, {
  baudRate: 9600,
});
arduinoSerialPort.on("open", function () {
  console.log("Serial Port " + arduinoCOMPort + " is opened.");
});
function read() {
  data = (arduinoSerialPort.read());
  // console.log(decodeURIComponent(data));
}
app.get("/", function (request, response) {
  console.log(data);
  response.render("pedo",{ steps : data })
});
app.listen(port, () => {
  console.log(`Example app listening at http://localhost:${port}`);
});
setInterval(read, 1000);
```

(2) Pedo.ejs

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="css/pedoStyles.css">
    <title>Document</title>
```

```html
</head>

<body>
  <nav>
    <h2>Pedometer</h2>
    <img src="pedo.png" alt="pedo" />
  </nav>
  <div class="body">
    <div class="main">
      <div class="progress-circle p<%=steps / 100 * 100 %>">
        <span>
          <%= (steps / 100 * 100).toFixed(0) %>%
        </span>
        <div class="left-half-clipper">
          <div class="first50-bar"></div>
          <div class="value-bar"></div>
        </div>
      </div>
      <p class="steps">Steps Today</p>
      <div class="bg">
        <%= steps %>
      </div>
      <p class="calories">
        <%= (steps * 0.063).toFixed(3) %> Calories
      </p>
    </div>
  </div>
  <footer>
    <p>CSE2006 J Component </p>
  </footer>
</body>

</html>
```

(3) Pedostyles.css

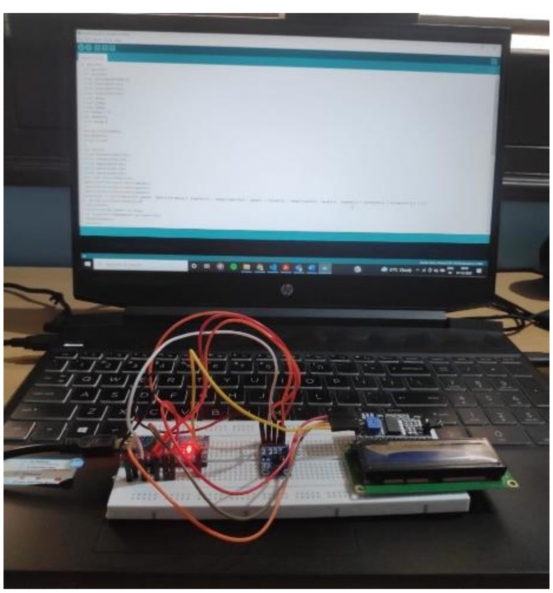
```css
@import url('https://fonts.googleapis.com/css2?family=Poppins&display=swap');
body{
  height:100vh;
  box-sizing: border-box;
  margin:0;
  padding:0;
```

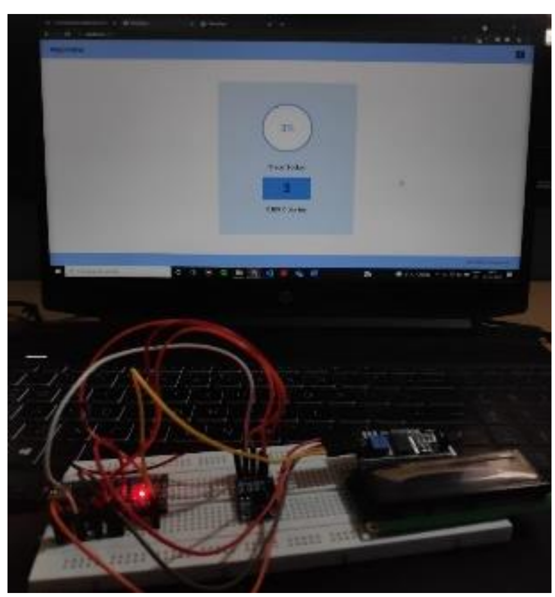
```
    font-family: 'Poppins', sans-serif;
    display:flex;
    flex-direction: column;
    justify-content: space-between;
}
nav{
    display:flex;
    justify-content: space-between;
    align-items: center;
    padding:0% 2%;
    background:#98BAE7;
}
nav img{
    height:2rem;
}
.body{
    width:100vw;
    display:flex;
    justify-content: center;
    align-items: center;
}
.main{
    background: rgba(152, 186, 231, 0.4);
    width:34rem;
    height:39.6rem;
    margin:2%;
    display:flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
}
.main p:first-child{
    font-size: 2em;
}
.steps{
    font-size: 1.5em;
}
.bg{
    background:#5396EF;
    padding:2% 15%;
    text-align: center;
    font-size: 3em;
```

```
}
.calories{
   font-size:1.5em;
}
footer{
   background:#98BAE7;
   display:flex;
   justify-content: flex-end;
   width:100%;
}
footer p{
   margin-right:2%;
}
```

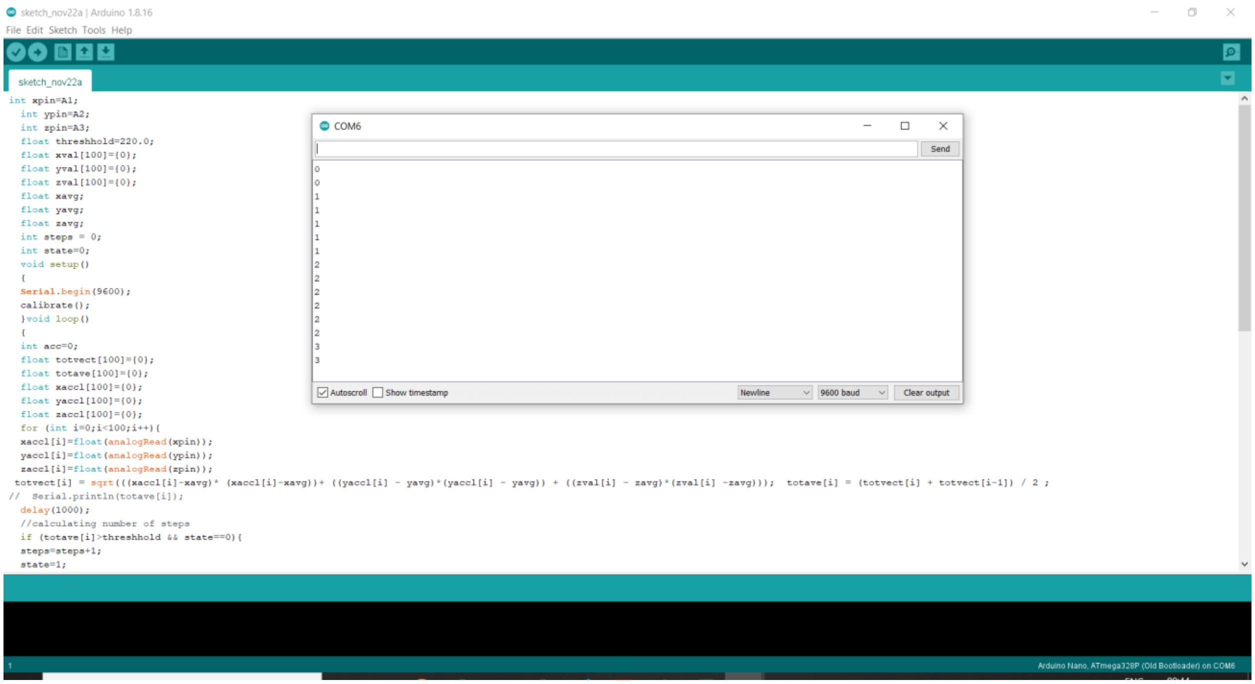
B. Web Application View

XI. Results Snapshots:

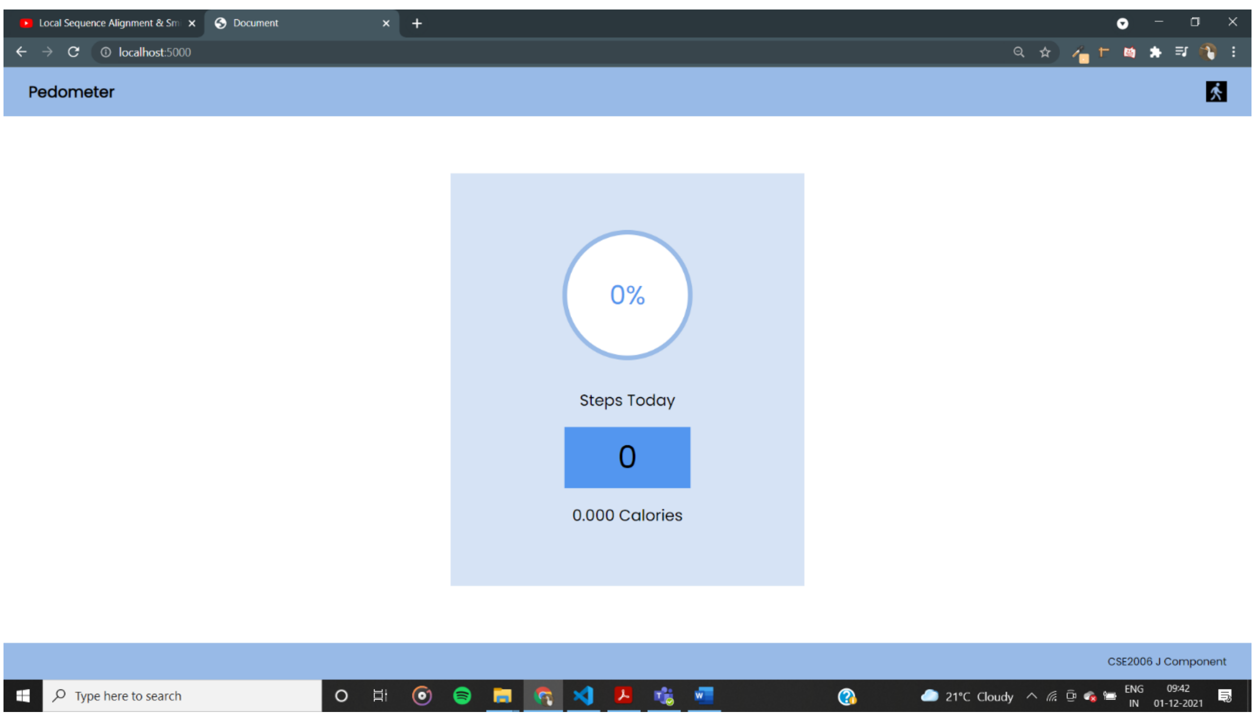




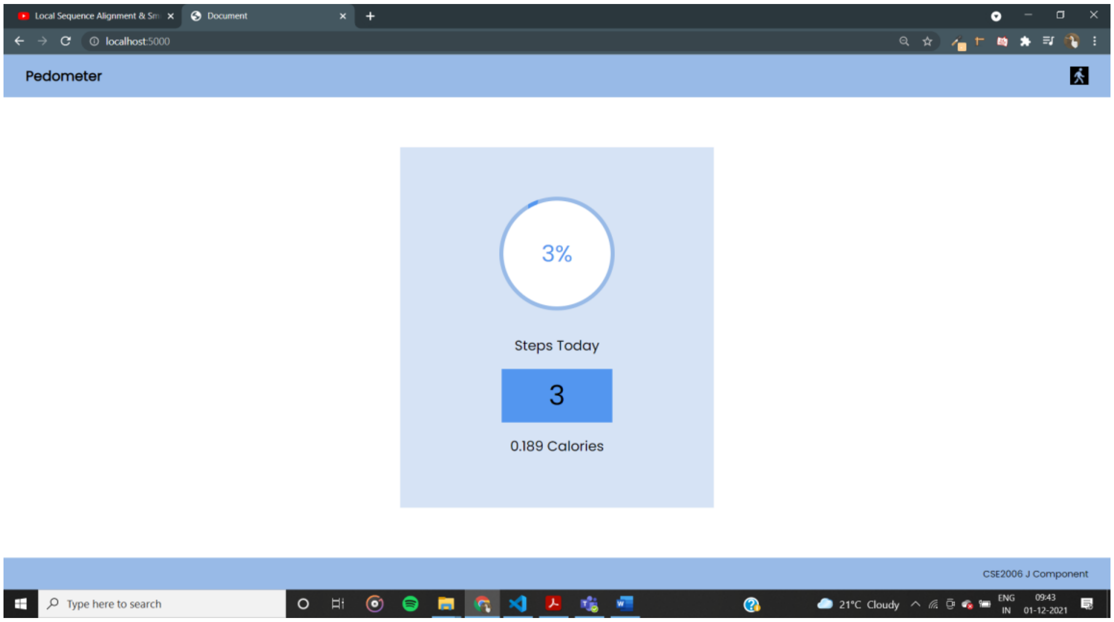
- Arduino serial monitor printing the number of steps taken



- Initial state of the web-app



- State of the web-app after some steps have been taken



XII. Limitations

- Accelerometers also have the inability to capture upper extremity movement, which may lead to underestimation of household activities such as washing dishes. Underestimation may also result when performing weight bearing activities.
- It is sensitive to temperature and operates over a limited temperature range.
- An Accelerometer can not measure rotation around its own axis of movement. ->Due to this, it is used in conjunction with a gyroscope to measure angular velocity.
- An Accelerometer measures chance in velocity only. It does not measure a constant velocity.
- This pedometer doesn't take into consideration your age, height, and weight. Instead, it uses a simple ratio of steps to calories.

XIII. Conclusion

The constant necessity to cater to the health and welfare of the general populace has only increased in recent years. Individual endeavours to retain health and strength are the only way to combat the lifestyle changes that have been imposed on the majority of the population, especially with the arrival of the pandemic and the unusual occurrences that have been making headlines in recent months. Health monitoring systems have been around for a long time, but it's past time to reform and rethink their rigid and inflexible designs to make them better, more accessible, and accurate for widespread use. We looked at a cost-effective, dependable, and incredibly responsive solution to the pedometer problem in our research. In order to monitor a person's everyday activity, step counting is becoming increasingly vital in the healthcare system. Many academics have attempted to implement step counting techniques by incorporating a range of sensors into mobile phones from various perspectives. Accelerometers are the greatest contenders for the 'step-count' issue, according to our research, and we've also recreated it with the help of reasonable, cost-effective solutions. This assures that a device as basic as a pedometer is inclusionary and unaffected by age, gait speed, angle with regard to the land, or any other variable that should not, in theory, interfere with pedometric estimates.

References

[1] Fontana, G. G., Tafur, A. J., Gardner, A. W., & Casanegra, A. I. (2014). Validity of smartphone pedometer apps as a tool to monitor physical activity remotely. Circulation, 130(suppl_2), A14796-A14796

[2] Wong, R. C. P., Yang, L., & Szeto, W. Y. (2021). Wearable fitness trackers and smartphone pedometer apps: Their effect on transport mode choice in a transit-oriented city. Travel behaviour and society, 22, 244-251.

[3] Fuzi, M. F. B. M. (2013). Pedometer.

[4] Nair, M. B., Kumar, S. R., Kishore, N. A., Mohan, N., & Anudev, J. (2018, August). Instantaneous feedback pedometer with emergency GPS tracker. In 2018 2nd International Conference on I-SMAC, 2018 2nd International Conference on (pp. 122-126). IEEE.

[5] Singh, R., Gehlot, A., Khilrani, J. K., & Mittal, M. (2020). Internet of Things–triggered and power-efficient smart pedometer algorithm for intelligent wearable devices. In Wearable and Implantable Medical Devices (pp. 1-23). Academic Press.

[6] Oliver, M., Badland, H. M., Shepherd, J., & Schofield, G. M. (2011). Counting steps in research: a comparison of accelerometry and pedometry. Open Journal of Preventive Medicine, 1(01), 1.

[7] Jayalath, S., & Abhayasinghe, N. (2013, April). A gyroscopic data based pedometer algorithm. In 2013 8th International Conference on Computer Science & Education (pp. 551-555). IEEE.

[8] Lim, B. Y., Shick, A., Harrison, C., & Hudson, S. E. (2010, January 22). *Pediluma: motivating physical activity through contextual information and social influence*. ACM.

[9] Karandikar, A., Deshpande, N., Lingayat, S., & Kulkarni, A. (2019, November). *IoT based Smart Fitness Tracker for Gymnasiums*. IRJET.