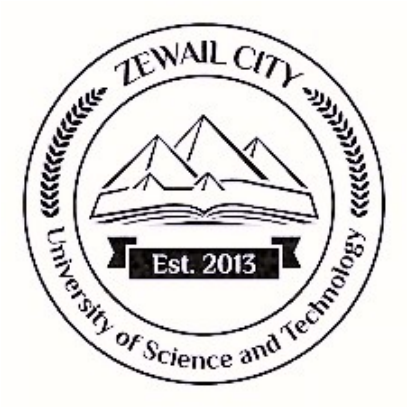
UNDERGRADUATE PROJECT

University of Science and Technology In Zewail City



Nanotechnology and Nanoelectronics Engineering Department

Echoes of Motion: Signal Processing and Data Transmission System Using a Piezoelectric Sensor

LaTeXed by:

Shadwa Amir Ahmed Elzoghbi

Instructor:

Prof. Ahmed S. Mohamed

Academic Year:

2024/2025

Team Members

Shadwa Amir[1]
Mahitab Swid[2]
Ahmed Yasser[3]
Amr Alshaarawy[4]

Contributions:

Amr Khaled Muhamed - Assisted with the format.

[1]Sophomore student at Nanotechnology and Nanoelectronics Engineering, UST ZC.
[2]Sophomore student at Nanotechnology and Nanoelectronics Engineering, UST ZC.
[3]Sophomore student at Nanotechnology and Nanoelectronics Engineering, UST ZC.
[4]Sophomore student at Nanotechnology and Nanoelectronics Engineering, UST ZC.

CONTENTS

Abstract

This project focuses on the design and development of a signal conditioning circuit for piezoelectric sensors, aimed at converting physical forces into measurable electrical signals. The system enhances the signal's readability by utilizing amplification and filtering techniques, allowing for accurate detection of weight and movement. The circuit includes key components such as a power supply, piezoelectric sensors, an amplifier, a microcontroller, and a display for real-time monitoring. This project specifically addresses attendance tracking in classrooms by using the piezoelectric sensor to detect vibrations caused by footsteps as people enter a room. The system processes and amplifies these signals, and then wirelessly transmits the data for real-time monitoring. By implementing this solution, the project offers a practical approach to improving attendance accuracy and automating the counting process.

1 Introduction

1.1 Project Overview

Signal Processing and Data Transmission System Using a Piezoelectric Sensor and ESP8266

In this project, we aimed to address attendance discrepancies in classrooms by developing an automated counter system that utilizes a piezoelectric sensor. The system detects and counts people entering a room by processing vibrations caused by footsteps. To achieve this, we implemented a signal processing and data transmission solution that includes filtering, amplifying, and transmitting data wirelessly for real-time monitoring.

1.2 System Overview

The system comprises a piezoelectric sensor, an ESP8266 microcontroller, and an accompanying web-based interface. The piezoelectric sensor captures analog signals corresponding to mechanical vibrations. These signals are processed through an external circuit that includes a filter and amplifier to enhance signal fidelity. The processed signal is then read by the ESP8266 microcontroller, which applies further signal interpretation logic to detect taps, representing footfalls.

2 Methodology

The methodology for this project revolves around the processing of signals from a piezoelectric sensor to enable accurate data acquisition, analysis, and conditioning. The overall process involves multiple stages, including signal acquisition, frequency analysis, filtering, and simulation, all of which contribute to enhancing the quality of the signal and enabling real-time monitoring of attendance. In this section, we will outline the steps taken to process the signal from a single piezoelectric sensor, providing the necessary foundation for further sensor integration and system development.

2.1 One Sensor Processing

Signal Acquisition of a piezoelectric sensor:

The raw signal from the sensor was acquired and processed using Analog Discovery 2, which provided reliable measurements. The sensor was connected to one channel of the device to provide output and initiate the process using the device's software. The connection was as follows:

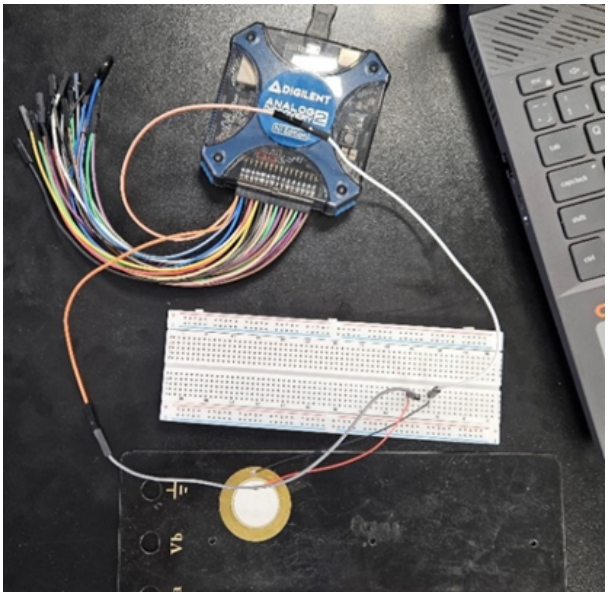


Figure 0.1: Sensor connection setup.

2.2 Signal Analysis

The objective is to analyze the various frequencies of the signal, identify the most dominant frequencies, and extract the necessary information that will help improve the quality of the signal later on in the project. This analysis is crucial for enhancing signal clarity, removing unwanted noise, and focusing on the relevant frequency components.

To achieve this, the spectrum analyzer was utilized to examine the waveform of the received signal. By using the "Peak Hold Continuous" mode on the spectrum analyzer, we were able to observe the frequency spectrum over time. This mode allows for the continuous capture and display of the peak values of the signal's frequency components, making it easier to identify key features. Through this analysis, the dominant low frequency, which represents the fundamental frequency of the signal, was identified. In addition to the fundamental frequency, the other harmonics (integer multiples of the fundamental) were also detected. These harmonics can provide valuable insight into the characteristics of the signal, such as its purity and potential sources of distortion. The identification of these frequencies is shown here:

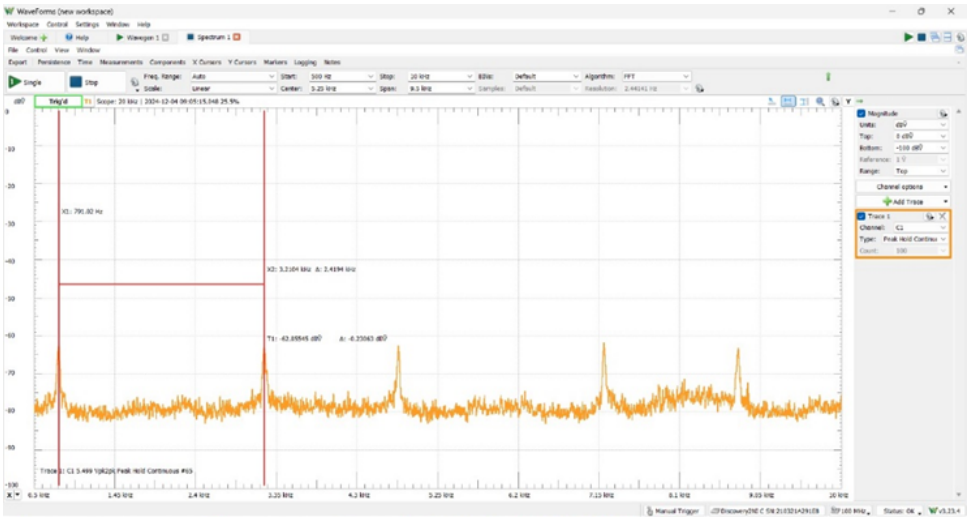


Figure 0.2: Frequency spectrum analysis using Waveforms.

Importing to LTSpice

Data was retrieved from the waveform, and a low-pass filter was designed to filter the frequencies, which was used to generate a simulation, that will be shown in the Filter circuit section.

3 Filter Circuit

The low-pass filter is built to filter at the cutoff frequency 0.465 Hz. The specific components are adjusted to suit the ones found in the market. It is built as identified in phase 2.

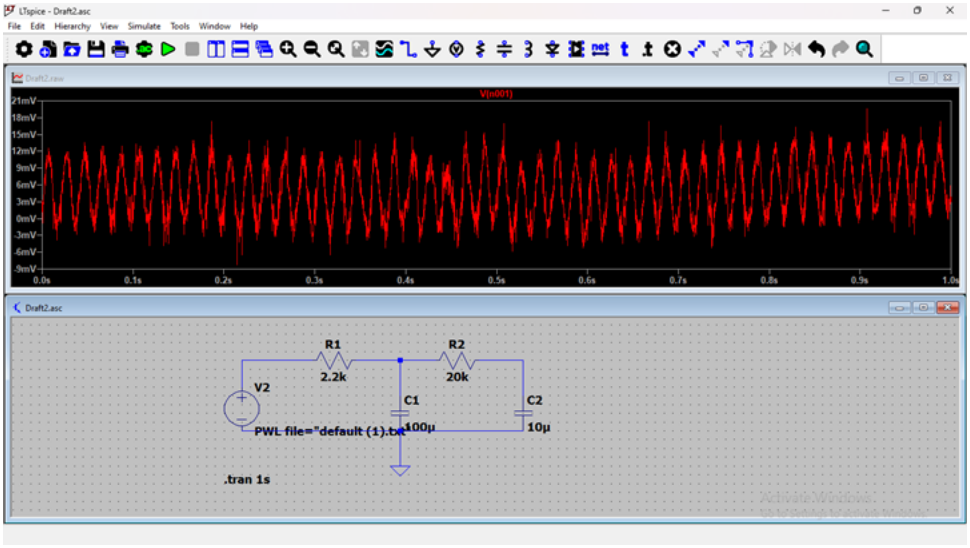


Figure 0.3: the raw signal before filtering

3.1 First order Filter

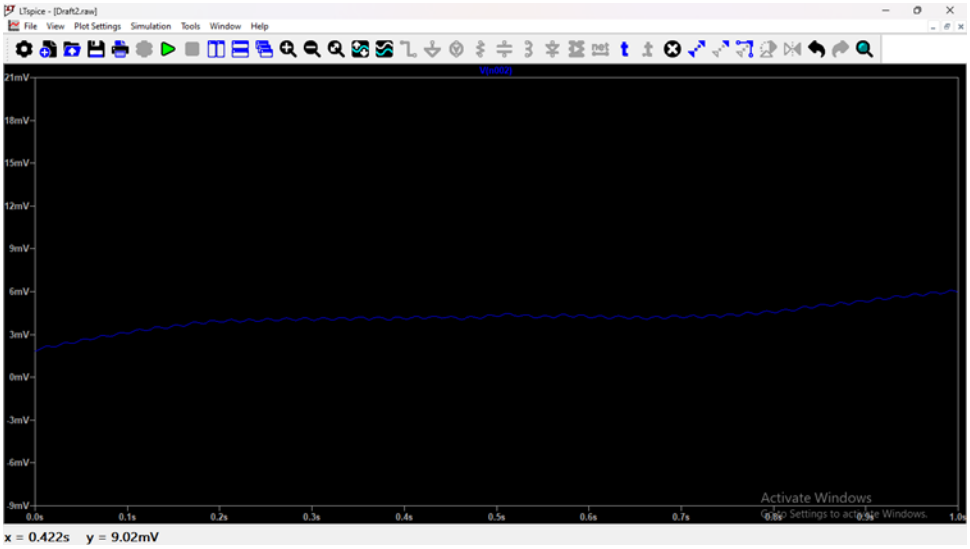


Figure 0.4: Waveform of the first order filtered signal

3.2 Second Order filtering

The circuit is designed to filter the input signal, allowing specific frequency components (0.465 Hz) to pass while attenuating others. This is essential for isolating the desired frequencies and reducing noise.

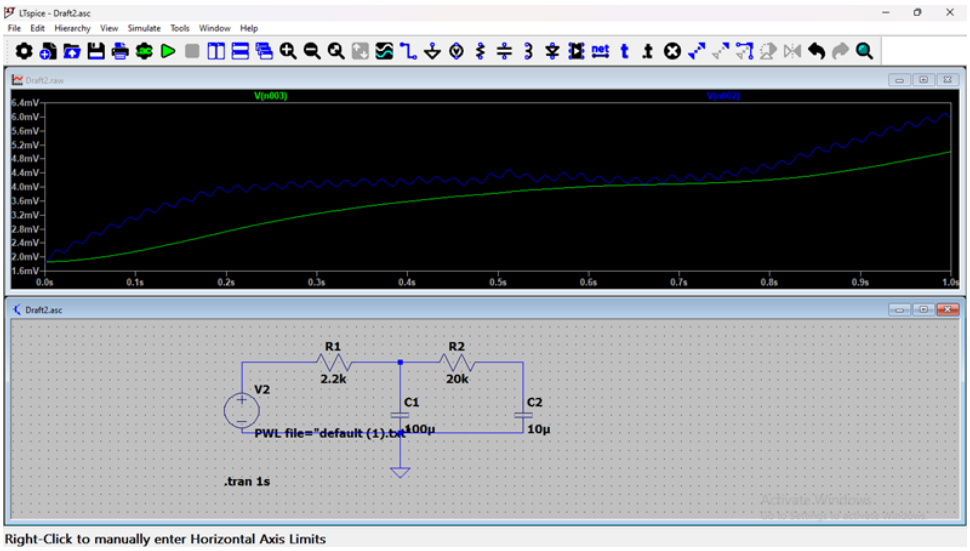


Figure 0.5: Waveform of the second order filtered signal

4 Amplifier Circuit

The amplifier circuit is built using the amplifier LM358. We connected its input to the output of the filter. The LM358 is a dual operational amplifier (op-amp) widely used for signal conditioning and amplification.

4.1 Signal Amplification

The LM358 amplifies the weak voltage signal generated by the piezoelectric sensor when it experiences mechanical stress. This ensures the signal strength is sufficient for subsequent filtering or digitization.

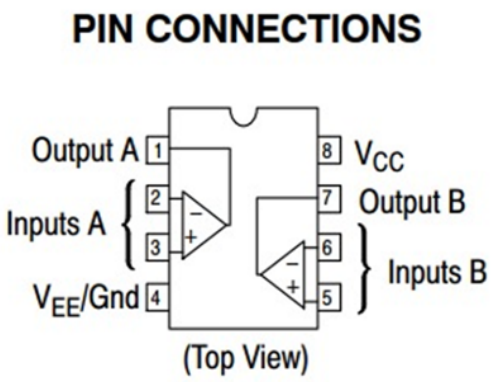


Figure 0.6: Pin Configuration of LM358 IC

4.2 Features of LM358

1. Dual Op-Amp Design:

It contains two independent op-amps in a single package, allowing flexibility in circuit design.

2. Low Power Consumption:

Operates with low quiescent current, making it efficient for battery-powered or low-energy applications.

3. Wide Supply Voltage Range:

Functions effectively in a range of 3V to 32V (single supply) or ±1.5V to ±16V (dual supply).

4. High Gain:

Provides a high open-loop voltage gain, essential for signal amplification.

5. Single Supply Operation:

Can operate with a single power supply, simplifying circuit design.

the gain is given by:

$$A_v = \frac{V_{\text{out}}}{V_{\text{in}}} = \left(1 + \frac{R_f}{R_{\text{in}}}\right)$$

5 Circuit on LTSpice

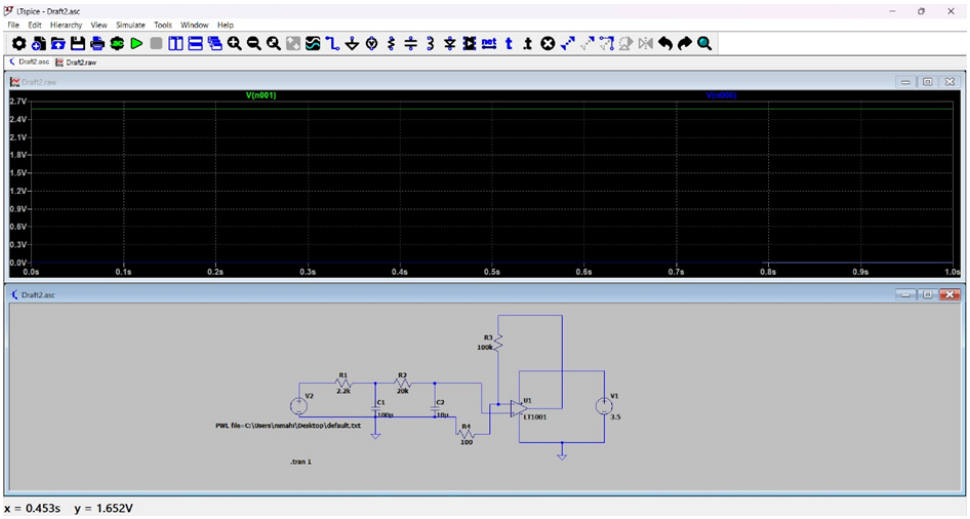


Figure 0.7: Circuit Design

6 Hardware and Simulation

This figure shows the final hardware implementation of the circuit:

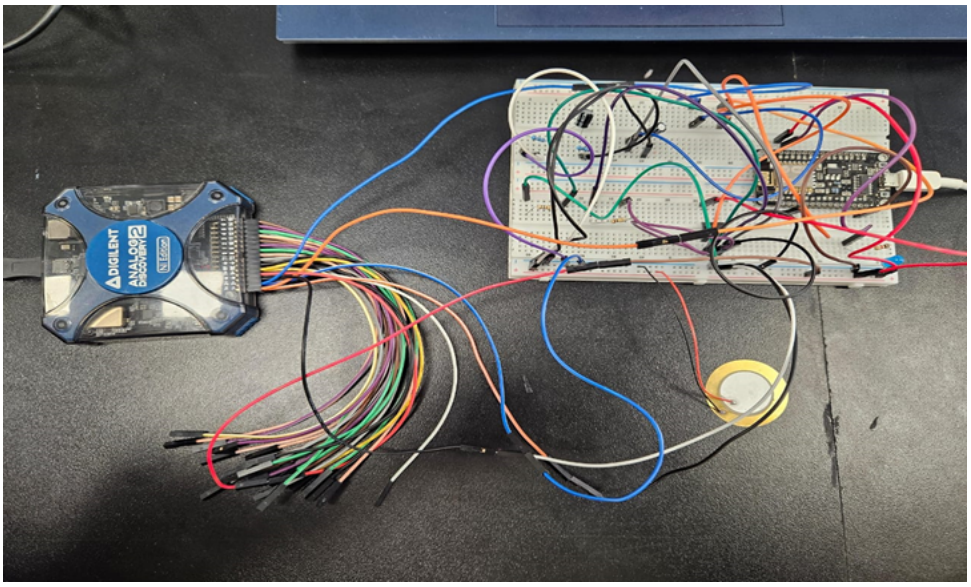


Figure 0.8: Circuit hardware implementation

Hardware Implementation

The system consists of a piezoelectric sensor, an amplifier circuit, a second-order low-pass filter, and the ESP8266 microcontroller. Each stage of the hardware is designed to process the signal from the piezoelectric sensor, detect taps, and communicate the data via a web server.

1. **Piezoelectric Sensor:**
   - Detects mechanical stress (taps) and generates a small voltage signal.
2. **Second-Order Low-Pass Filter:**
   - Filters out high-frequency noise using a Sallen-Key configuration with resistors and capacitors.
3. **Amplifier Circuit:**
   - LM358 Op-Amp amplifies the signal to a suitable level for the ESP8266.
   - Gain controlled by resistors R1 and R2.
4. **ESP8266 Microcontroller:**
   - Reads the filtered signal via the A0 pin.
   - Processes the signal to count taps and calculate intensity.
   - Displays the data via a web server.

7 Result On Waveforms

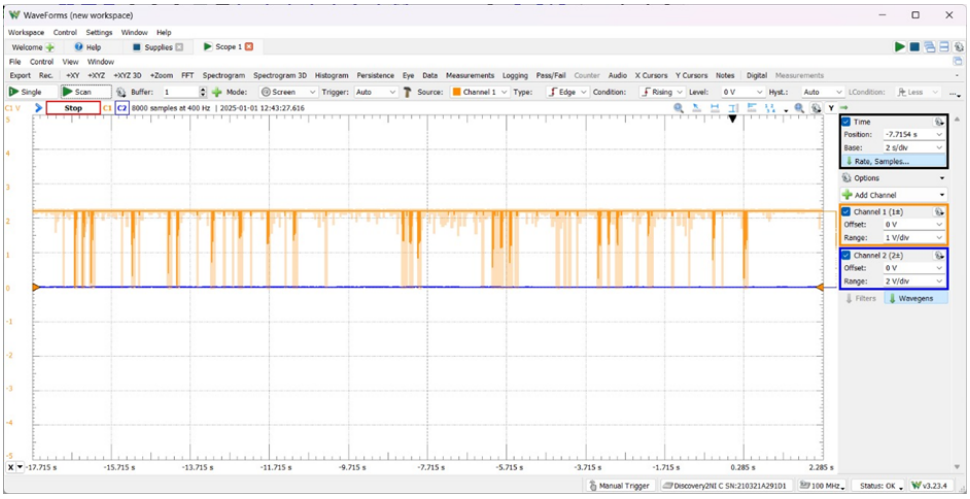


Figure 0.9: Simulation of the amplified signal

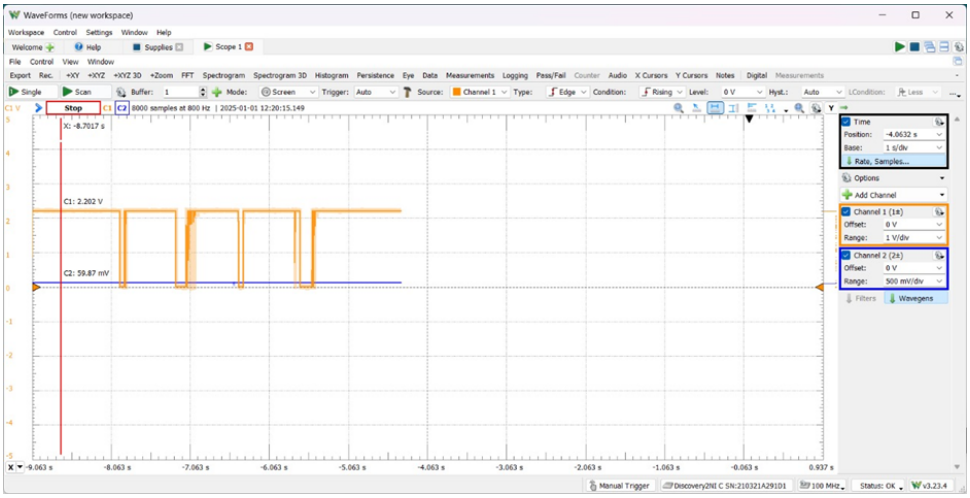


Figure 0.10: amplified signal pulses

8 Transferring the signal to readable measurements (Tap Counter)

8.1 Wireless Data Transmission and Visualization

To provide an intuitive and accessible platform for monitoring the data in real-time, the ESP8266 microcontroller was programmed to function as a Wi-Fi-enabled web server. This allows the system to transmit processed data wirelessly to any device connected to the same network, offering seamless access for users.

The processed data—comprising the tap count (the number of footfalls detected) and tap intensity (the magnitude of vibrations)—is sent from the ESP8266 to a web interface. To ensure efficient and easy communication between the microcontroller and the user interface, the data is formatted in JSON (JavaScript Object Notation). JSON is a lightweight data-interchange format that makes it easy for the web interface to parse and display the information dynamically in real time.

The web interface itself is developed using standard HTML, CSS, and JavaScript. This enables the creation of a simple, yet functional, dashboard that can be accessed from any modern web browser on connected devices, such as smartphones, tablets, or computers. The interface displays live updates of the tap count and intensity values, offering a visual representation of foot traffic through the room. Additionally, users can interact with the interface to observe trends or reset the counters as needed.

A significant advantage of the system is its fully manual development, ensuring that it does not rely on third-party platforms like Blynk. This approach allows for greater control over the data flow and visualization process, ensuring that the web interface can be customized to fit specific needs. The flexibility of the system also makes it easily scalable for various use cases, beyond just classroom attendance tracking. For instance, it could be adapted for monitoring foot traffic in other environments, such as offices, public spaces, or even for applications requiring real-time analysis of environmental factors or sensor data.

The combination of wireless transmission, real-time data processing, and a customizable web interface makes this system highly versatile and user-friendly, providing an effective solution for automatic, real-time monitoring in a variety of scenarios.
A key feature of the system is its integration with the Blynk platform, offering potential for mobile app monitoring in addition to the web-based visualization. The system provides scalability for multiple use cases, including classroom attendance tracking and other real-time monitoring applications.

9 Arduino Code Implementation

Below is the implementation of the system, where the ESP8266 reads signals from the piezoelectric sensor and handles web server requests for real-time data display:

```
// Include libraries
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>

// WiFi credentials
const char* ssid = "Galaxy␣S23+␣MMMS";
const char* password = "11042723";

// Web server
ESP8266WebServer server(80);

// Piezo input
const int piezoPin = A0;

// Variables
int tapCount = 0;
int lastAnalogValue = 0;
int peakValue = 0;
int threshold = 100; // Adjust this threshold value
unsigned long lastTapTime = 0;
const unsigned long debounceTime = 200;
const int maxAnalogReading = 1023;
float scalingFactor = 0.1;
```

Code functionality

System Initialization

This section establishes the foundation for the system's operation:

- **Libraries**: Enable Wi-Fi connectivity and HTTP web server functionality for real-time data transmission.
- **Wi-Fi Credentials**: Store the SSID and password needed for the ESP8266 to connect to a network.
- **Web Server**: Initializes an HTTP server on port 80 to handle data requests and serve a user interface.
- **Piezoelectric Sensor Input**: Defines the analog pin (`A0`) used to read vibrations from the piezoelectric sensor.
- **Variables**: Manage sensor data and system functionality:
  - Track tap count and signal intensity.
  - Prevent false detections with threshold and debounce logic.
  - Scale sensor values for meaningful representation.

This setup ensures seamless communication, accurate tap detection, and readiness for data processing.

```
void handleRoot() {
  String htmlPage =
       "<!DOCTYPE␣html>"
       "<html>"
       "<head>"
       "<title>Piezo␣Tap␣Counter</title>"
       "<script>"
       "␣␣setInterval(function()␣{"
       "␣␣␣␣var␣xhr␣=␣new␣XMLHttpRequest();"
       "␣␣␣␣xhr.open('GET',␣'/data',␣true);"
       "␣␣␣␣xhr.onload␣=␣function()␣{"
       "␣␣␣␣␣␣if␣(xhr.status␣===␣200)␣{"
       "␣␣␣␣␣␣␣␣var␣data␣=␣JSON.parse(xhr.responseText);"
       "␣␣␣␣␣␣␣␣document.getElementById('tapCount').textContent␣=␣data.
          tapCount;"
       "␣␣␣␣␣␣␣␣document.getElementById('intensity').textContent␣=␣data.
          intensity;"
       "␣␣␣␣␣␣}"
       "␣␣␣␣};"
       "␣␣␣␣xhr.send();"
       "␣␣},␣100);" // Fetch data every 100ms
       "</script>"
       "</head>"
       "<body>"
       "␣␣<h1>Piezo␣Tap␣Counter</h1>"
       "␣␣<p>Tap␣Count:␣<span␣id='tapCount'>0</span></p>"
       "␣␣<p>Intensity:␣<span␣id='intensity'>0</span></p>"
       "</body>"
       "</html>";
  server.send(200, "text/html", htmlPage);
}
```

Functionality of `handleRoot`

The `handleRoot` function is responsible for creating and serving an HTML webpage when a client accesses the root URL (`/`) of the ESP8266 web server. The functionality can be summarized as follows:

- **Webpage Creation:** The function dynamically constructs an HTML page that displays the current **tap count** and **intensity** values measured by the piezo sensor.
- **Dynamic Updates:**
  - The webpage includes a JavaScript script that uses the `setInterval` function to periodically (every 100ms) send GET requests to the server's `/data` endpoint.
  - The server responds with JSON data containing the latest tap count and intensity values.
  - The received data is parsed and used to update the HTML elements displaying the tap count and intensity, enabling real-time monitoring.
- **Content Delivery:** The webpage is served to the client using the `server.send` function, with an HTTP status code of `200` and a content type of `text/html`.

```
void handleData() {
  String jsonData = "{\"tapCount\":␣" + String(tapCount) +
                    ",␣\"intensity\":␣" + String(peakValue *
                       scalingFactor) + "}";
  server.send(200, "application/json", jsonData);
}

void setup() {
  pinMode(piezoPin, INPUT);
  Serial.begin(115200);

  // Connect to WiFi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting␣to␣WiFi...");
  }
  Serial.println("Connected␣to␣WiFi");

  // Start server
  server.on("/", handleRoot);
  server.on("/data", handleData);
  server.begin();
}

void loop() {
  int currentAnalogValue = analogRead(piezoPin);

  // Detect peak value
  if (currentAnalogValue > peakValue) {
    peakValue = currentAnalogValue;
  }

  // Check for tap detection
  if (currentAnalogValue > threshold && millis() - lastTapTime >
      debounceTime) {
    tapCount++;
    lastTapTime = millis();
  }

  // Reset peakValue periodically (optional)
  if (millis() % 1000 == 0) {
    peakValue = 0;
  }

  server.handleClient();
}
```

Code functionality

- `handleData()`: Sends real-time data (tap count and intensity) as a JSON object to connected clients.
- `setup()`: Configures the sensor, connects to Wi-Fi, and initializes the web server.

- `loop()`:
  - Reads sensor values to detect taps and track intensity.
  - Implements threshold and debounce logic to ensure valid tap detection.
  - Periodically resets intensity values and handles client requests.

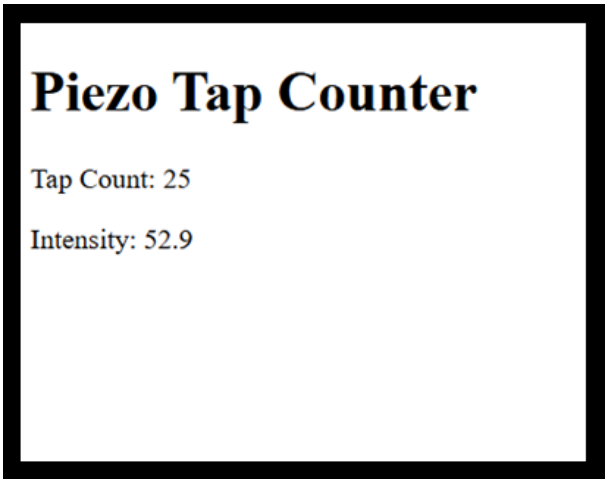


Figure 0.11: Web Interface of the Piezo Tap Counter

The figure above shows the web interface generated by the ESP8266 microcontroller, which acts as a Wi-Fi-enabled server to provide real-time monitoring of the piezoelectric sensor data. The interface is designed for easy access and usability, offering key information that reflects the current sensor activity:

- **Tap Count:** This field dynamically updates to display the total number of taps detected by the piezoelectric sensor. Each tap corresponds to a vibration caused by a footfall, and the count increases in real time as new footfalls are detected. The tap count is continuously refreshed without any user intervention, ensuring the number reflects the most current sensor readings. This field provides valuable data for monitoring foot traffic in the monitored area.
- **Intensity:** This field shows the peak intensity of the taps, representing the strength or magnitude of the detected vibrations. The intensity is presented on a scaled range for better readability and to ensure that the data is both informative and user-friendly. Higher intensity values indicate stronger, more forceful footfalls, while lower values correspond to softer steps. By scaling the intensity values, the system provides users with a clearer picture of not only the frequency of foot traffic but also the physical force behind each footfall, which can be useful for a variety of applications.

To ensure the web interface reflects the latest data from the piezoelectric sensor without requiring manual interaction, the webpage is updated every 100 milliseconds using AJAX (Asynchronous JavaScript and XML) requests. AJAX allows the interface to asynchronously retrieve updated data from the ESP8266 microcontroller without having to refresh the entire webpage. This method minimizes delays and ensures that the user sees a continuous, real-time feed of the tap count and intensity values. As a result, users can monitor the sensor activity seamlessly, without the need to reload the page, providing an efficient and responsive real-time experience.

By using AJAX, the system guarantees smooth and uninterrupted updates, making it suitable for applications that require instant access to dynamic data. This ensures that the sensor's real-time activity is always visible, supporting applications such as monitoring foot traffic, attendance tracking, or environmental sensing.