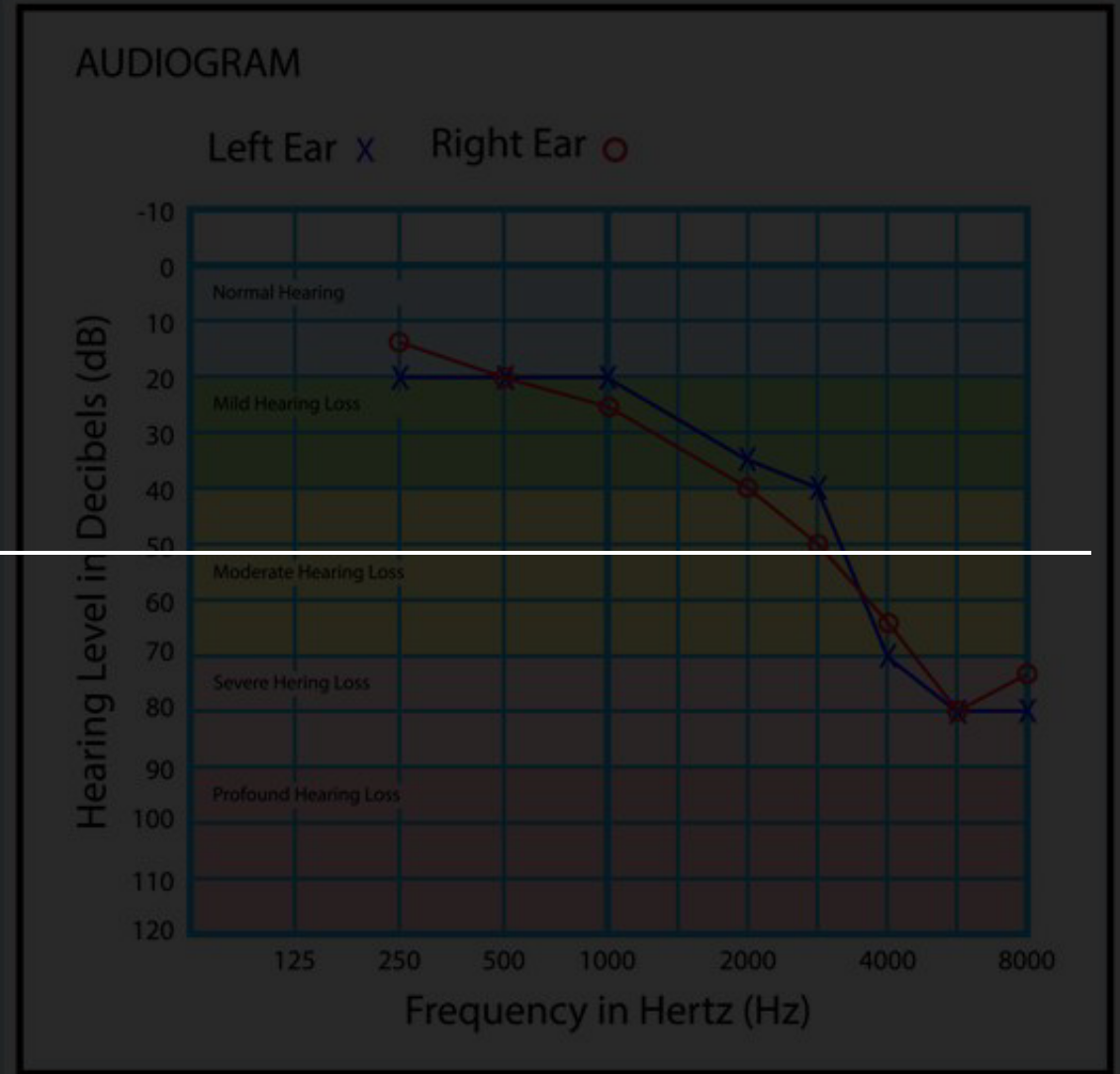




Das Audiometer

Gliederung

- Grundlegende Physik des Hörens
- Technische Grundlagen des Audiogramms
- Das erstellte System



Grundlegende Physik des Hörens

- **Schallwellen:**

- mechanische Welle, die durch Druckschwankungen in der Luft entsteht.
- **Frequenz** (Hertz, Hz) beschreibt die Tonhöhe
- **Amplitude** die Lautstärke (Schalldruckpegel).

- **Wahrnehmungsbereich des menschlichen Ohrs:**

- Frequenzen zwischen ca. **20 Hz bis 20 kHz**.
- Mit zunehmendem Alter nimmt die Hörfähigkeit ab

Grundlegende Physik des Hörens

- **Schalldruckpegel (dB-Skala):**
 - Eine logarithmische Skala wird verwendet, da das menschliche Ohr große Unterschiede in der Lautstärke wahrnimmt.
 - Beispiele: Flüstern bei 20 dB, normales Gespräch bei 60 dB, laute Musik über 85 dB (potenziell hörschädigend).

Dezibel-Tabelle

Schallpegel bekannter Umgebungsgeräusche



Technische Grundlagen des Audiogramms



- **Audiometer:**

- Ein Gerät zur genauen Messung von Hörschwellen über Luft- und Knochenleitung.
- **Bestandteile:** Kopfhörer (Luftleitung), Knochenleitungshörer (Knochenleitung) und Tonausgabe.

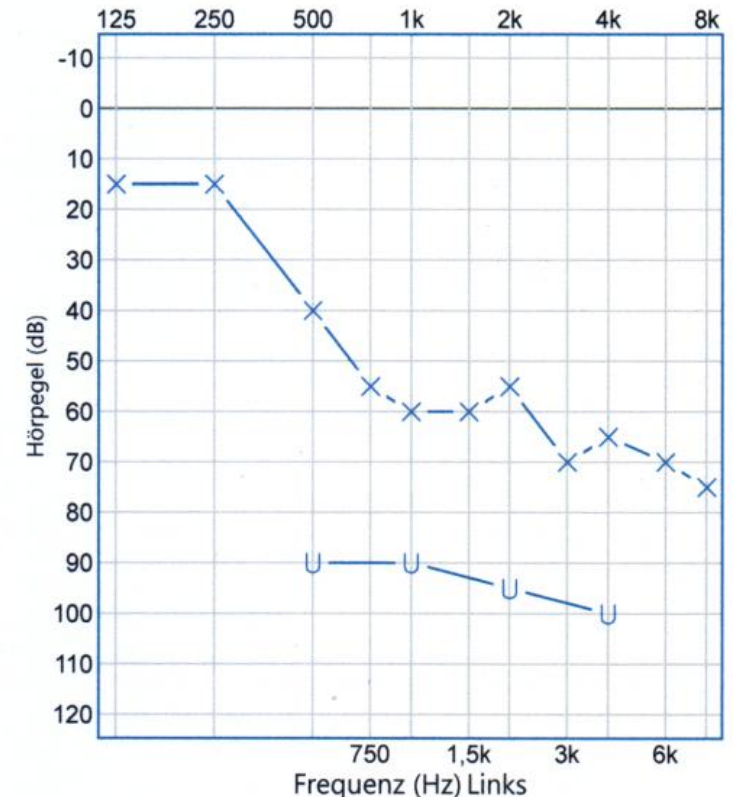
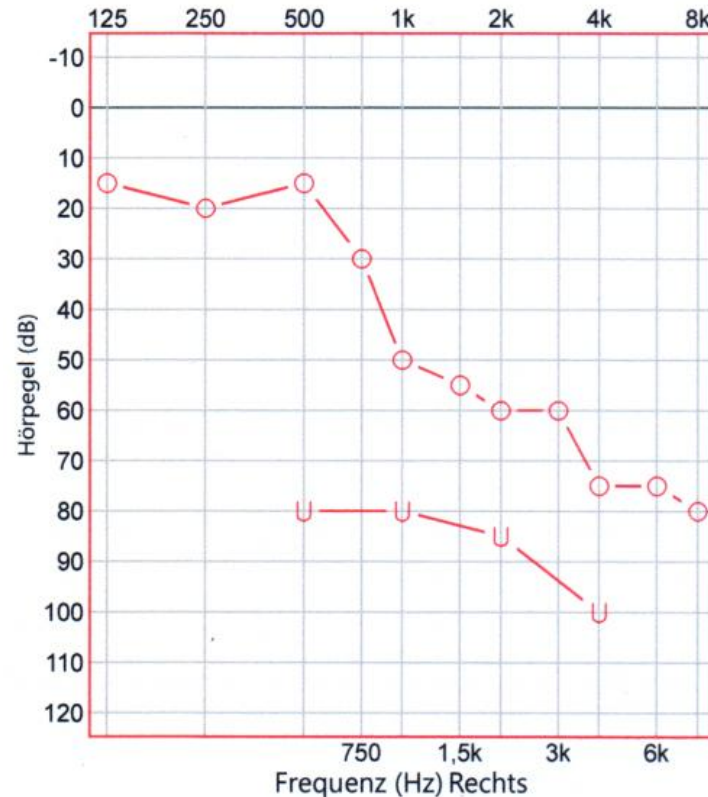
- **Audiogrammarten:**

- **Luftleitung:** Über Kopfhörer zur Messung des Gesamthörvermögens.
- **Knochenleitung:** Über Knochenleitungshörer zur Messung des Innenohrvermögens.

Technische Grundlagen des Audiogramms

- **Testablauf:**

Ein Audiologe spielt Töne in verschiedenen Frequenzen und Lautstärken ab und bestimmt die niedrigste Lautstärke, bei der der Patient die Töne hören kann.



https://doofe-ohren.de/wp-content/uploads/2020/08/2019-12-05.terzo_.audiogramm-1024x560.png

Unser Projekt

- Knöpfe zum Verstellen der Lautstärke
- Knopf zur Erhöhung der Frequenz
- LCD-Bildschirm zur Ausgabe der Werte
- **Update:**
 - Lautstärkenenerhöhung selbstständig
 - Ausgabe der Werte über Serial Print
 - Anleitung welches Ohr gerade getestet wird
 - Bibliotheken und deren Versionen wurden erfasst
 - Code in Github hochgeladen https://github.com/bonsaibauer/arduino_hearing_test

Unser Projekt



Unser Projekt

→ Lautstärkenenerhöhung selbstständig

```
58     void loop() {  
59 +     unsigned long currentTime = millis();  
60 +  
61 +     // Automatische Lautstärkeerhöhung alle 800 ms  
62 +     if (currentTime - lastVolumeIncreaseTime >= 800 && currentVolume < 30) {  
63 +         lastVolumeIncreaseTime = currentTime; // Zeitstempel aktualisieren  
64 +         currentVolume++; // Lautstärke erhöhen  
  
65 +         myDFPPlayer.volume(currentVolume); // Lautstärke einstellen  
66 +         updateDisplay(); // Anzeige aktualisieren
```

Unser Projekt

→ Ausgabe der Werte in Serial Printer

```
134 +  
135 + void sendToSerial(int frequencyIndex, int volume) {  
136 +   // Werte über serielle Verbindung im CSV-Format ausgeben  
137 +   Serial.print(frequencies[frequencyIndex - 1]);  
138 +   Serial.print(",");  
139 +   Serial.print(volume);  
140 +   Serial.print(",");  
141 +   Serial.println(isLeftEar ? "L" : "R"); // L oder R für das Ohr  
142 + }
```

Unser Projekt

→ Anleitung welches Ohr gerade getestet wird

```
41 + // "Rechtes Ohr" anzeigen, bevor der rechte Ohrtest startet
42 + showEarIndicator("Rechtes Ohr");
43 + delay(1000); // Anzeige für eine Sekunde
44 +
45 + // Start des Lautstärkentests
```

```
// Falls alle Frequenzen für das rechte Ohr durchlaufen sind, zum linken Ohr wechseln
if (currentFrequencyIndex >= sizeof(frequencies) / sizeof(frequencies[0])) {
    if (!isLeftEar) {
        showEarIndicator("Linkes Ohr"); // Linkes Ohr anzeigen
        delay(1000); // Anzeige für eine Sekunde
        isLeftEar = true; // Wechselt zum linken Ohr
        currentFrequencyIndex = 1; // Frequenzindex auf Anfang setzen
    }
}
```

Unser Projekt

→ Bibliotheken und deren Versionen wurden erfasst

```
1  + #include <Arduino.h> //AIPIC_Opta 1.2.0, ArduinoRS485 1.1.0, Arduino_DebugUtils 1.4.0,  
   Arduino_Opta_Blueprint 0.2.5, Firmata 2.5.9  
2  + #include <SoftwareSerial.h> //EspSoftwareSerial 8.1.0, Arduino_SerialUpdater 0.0.1  
3  + #include <DFRobotDFPlayerMini.h> //DFRobotDFPlayerMini 1.0.6  
4  + #include <LiquidCrystal_I2C.h> //LiquidCrystal I2C 1.1.2
```

Unser Projekt

→ Code in Github hochgeladen https://github.com/bonsaibauer/arduino_hearing_test

The screenshot shows the GitHub interface for the repository 'arduino_hearing_test' by user 'bonsaibauer'. The repository is public and has 1 branch (main) and 4 commits. The commit history shows two files: 'README.md' (initial commit, 50 minutes ago) and 'hearing_test.ino' (v2.0.1, 45 minutes ago). The README file is selected, showing the title 'arduino_hearing_test' and the description 'DHBW Mannheim Project with a Arduino and LCD Screen (hearing test)'. The right sidebar contains links to Readme, Activity, 0 stars, 1 watching, 0 forks, Releases (No releases published, Create a new release), and Packages (No packages published, Publish your first package).

arduino_hearing_test Public

Pin Unwatch 1 Fork 0 Star 0

main 1 Branch Tags Go to file Add file Code

bonsaibauer v2.0.1 ea56f0d · 45 minutes ago 4 Commits

File	Commit	Time
README.md	Initial commit	50 minutes ago
hearing_test.ino	v2.0.1	45 minutes ago

README

arduino_hearing_test

DHBW Mannheim Project with a Arduino and LCD Screen (hearing test).

Readme Activity 0 stars 1 watching 0 forks

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Quellen

- [Audiometer – Wikipedia](#)
- [Hörtest Teil 1: Audiogramm und Hörschwellenmessung | Doofe Ohren](#)
- <https://sonova-retail-media-prd.azureedge.net/media/ml/sonova.geers-de>
- [Tonaudiogramm – Wikipedia](#)
- Dieter Mrowinski; Thomas Steffens; Günter Scholz: *Audiometrie*. 5. Auflage. Thieme, Stuttgart 2017, [ISBN 978-3-13-240107-5](#).