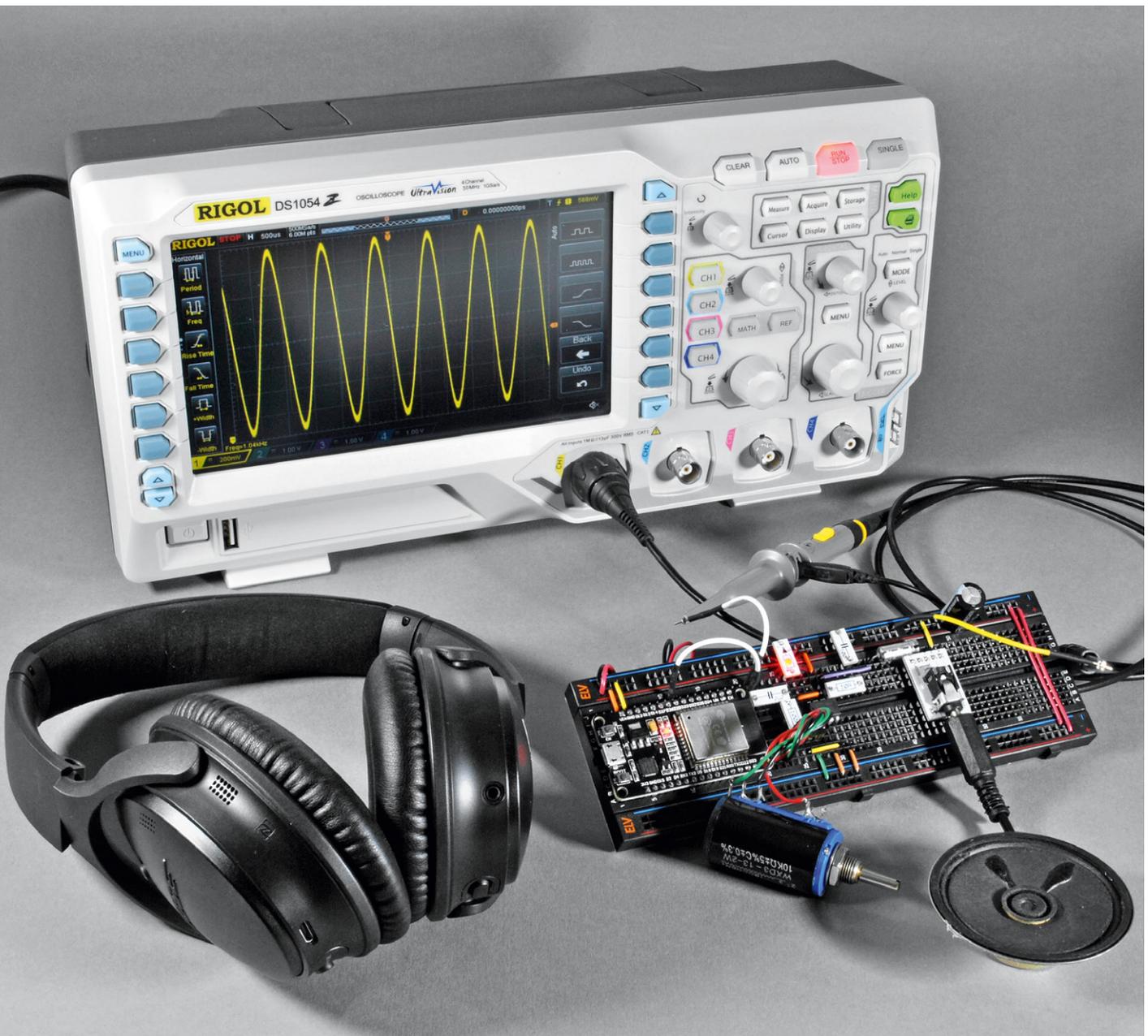




# Bioelektronik I

## Audiotechnik, Klang und Gehör

Biosensorik und Bioelektronik sind vergleichsweise junge Fachgebiete der Elektrotechnik. Aufgrund bahnbrechender Entwicklungen in der allgemeinen Messtechnik konnten auch im biomedizinischen Bereich enorme Fortschritte erzielt werden. In den folgenden Ausgaben des ELVjournals soll die Bioelektronik daher etwas genauer betrachtet werden. Im ersten Beitrag stehen die Themen Gehör und Akustik im Fokus. Neben der Ton- oder Klangerzeugung und Audioverstärkern wird auch ein Do-it-yourself-Hörtestgerät zur Bestimmung des Hörfrequenzgangs vorgestellt. Dieses erlaubt insbesondere den Nachweis für die Abnahme der Hörfähigkeit mit zunehmendem Alter.





## Vom Ton zum Klang

Wenn man sich näher mit der Audiotechnik befassen will, sollte man sich über einige Begriffe der Ton- und Klangerzeugung im Klaren sein. Ein Ton bezeichnet zunächst ein Schallereignis, das von Musikinstrumenten, der menschlichen Stimme oder auch elektronisch erzeugt werden kann. Prinzipiell kann jedem Ton eine mehr oder weniger exakte Tonhöhe zugeordnet werden. Während die physikalischen Eigenschaften als Kombination von Sinustönen und Geräuschkomponenten analysiert und beschrieben werden können, ist die subjektive Tonwahrnehmung von psychoakustischen Gegebenheiten, kulturellen Erfahrungen und ästhetischen Erwartungen abhängig.

Physikalisch entstehen Töne durch sich in der Luft ausbreitende Druckschwankungen. Diese werden durch mechanische Schwingungen erzeugt. Soll die Schwingung andauern, muss kontinuierlich Energie zugeführt werden. Das gilt auch für die elektronische Ton- und Klangerzeugung. Eine periodische Schwingung ist eine Zustandsänderung, die sich nach einem bestimmten Zeitintervall, der sogenannten Periode, identisch wiederholt. Eine Schwingung wird auch dann noch als periodisch betrachtet, wenn sich ihre Amplitude langsam mit der Zeit verändert (Bild 1). Die Amplitude bestimmt dabei die Lautstärke des Tons.

Die zweite wichtige Eigenschaft eines Tons ist seine Frequenz. Sie bestimmt die Tonhöhe. Hierbei gilt:

$$\text{Frequenz [Hz]} = 1 / \text{Periodendauer [s]}$$

Die einfachste periodische Schwingungsform ist die Sinusschwingung. Als Ton kommt der reine Sinus in der Natur und bei mechanischen Instrumenten nur sehr selten vor.

Ein Klang dagegen ist ein aus harmonischen Tönen zusammengesetztes akustisches Ereignis. Harmonische Töne werden so bezeichnet, wenn deren Frequenzanteile in einem ganzzahligen Verhältnis zueinander stehen. Ein Klang wird durch

- Klanghöhe,
- Klangdauer,
- Klangfarbe und
- zeitlich verlaufende Lautstärke

bestimmt. Die Klangfarbe ist das wesentliche Charakteristikum zur Unterscheidung zwischen einzelnen Instrumenten. Von besonderer Bedeutung sind hier die harmonischen Obertöne. Diese erzeugen aus einzelnen Sinusschwingungen den charakteristischen Klang eines Instruments. Dadurch können Konzertflügel, Geigen oder Posaunen völlig unterschiedlich klingen, auch wenn jeweils derselbe musikalische Ton, etwa der Kamerton a1 mit 440 Hz gespielt wird.

Bei Musikinstrumenten kann die Klangfarbe kaum beeinflusst werden. In der elektronischen Musik dagegen existieren die verschiedenartigsten Möglichkeiten, einen Klang beliebig zusammenzusetzen.

## Elektronische Erzeugung von Tönen und Klängen

Über viele Jahre hinweg wurden Töne und Klänge über analog-elektronische Verfahren erzeugt. Oszillatoren, Hüllkurvengeneratoren oder Filter erlaubten die Synthese einer außerordentlichen Vielfalt an neuen Klängen, die mit klassischen Instrumenten nicht erzeugt werden konnten. Die Technik gipfelte in den oft geradezu monströsen Analoysynthesizern der siebziger und frühen achtziger Jahre. Aber auch hier führte die Digitaltechnik zu einer Revolution. Moderne Synthesizer ba-

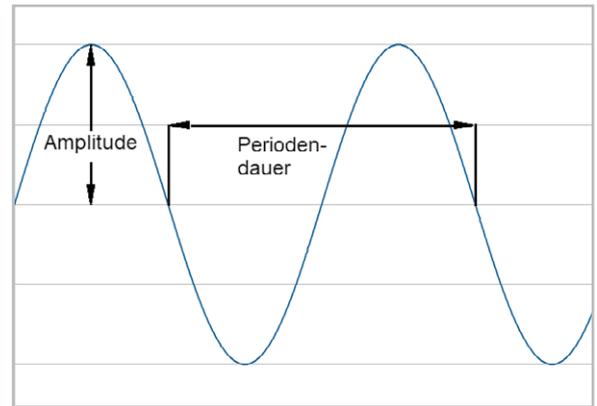


Bild 1: Amplitude und Periodendauer

sieren heute praktisch ausschließlich auf der digitalen Klangersynthese. Im Folgenden soll daher auf die digitale Klangerzeugung eingegangen werden. Es wird sich jedoch zeigen, dass man trotzdem nicht vollständig ohne analoge Schaltungstechniken auskommt.

Die Basis für praktische Anwendungsbeispiele bildet ein Mikrocontroller. Aufgrund seines günstigen Preises, der umfassenden Leistungsmerkmale und der weiten Verbreitung kommt der ESP32 der Firma Espressif zum Einsatz. Ein weiterer Vorteil dieses Controllers ist, dass er sehr einfach über die bekannte Arduino-IDE programmiert werden kann. Dazu ist ein Add-on verfügbar, mit dem das ESP32-Board über die Arduino-IDE programmierbar wird. Nach dem Starten der Arduino-IDE werden dazu die „Voreinstellungen“ geöffnet: Datei → Voreinstellungen

Der Menüpunkt „Zusätzliche Boardverwalter-URLs“ erlaubt hier das Eintragen der folgenden Adresse:

[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)

Nach einer Bestätigung mit „OK“ wird der Boardverwalter unter Werkzeuge → Board → Boardverwalter aktiviert.

Hier kann über den Eintrag „esp32“ im Suchfeld nach dem Espressif-Board gesucht werden. Schließlich kann dieses via „Installieren“ übernommen werden. Damit sollten die Informationen für das ESP-Board in der IDE zur Verfügung stehen (Bild 2 und Bild 3). Durch Auswählen des Eintrags „ESP32 Dev Module“ kann die von uns verwendete Entwicklungsplatine NodeMCU mit ESP32 (siehe Materialliste) programmiert werden.

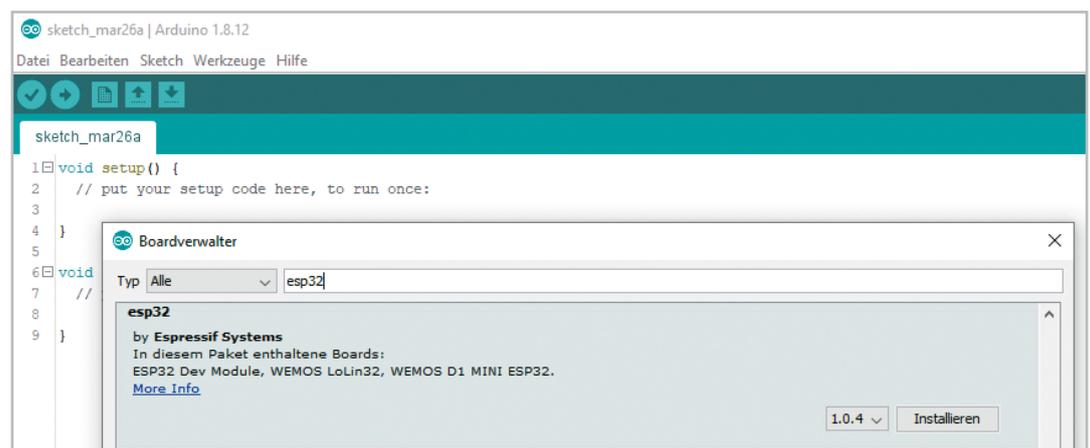


Bild 2: Im Boardverwalter werden erst die Informationen für das ESP32-Board installiert ...

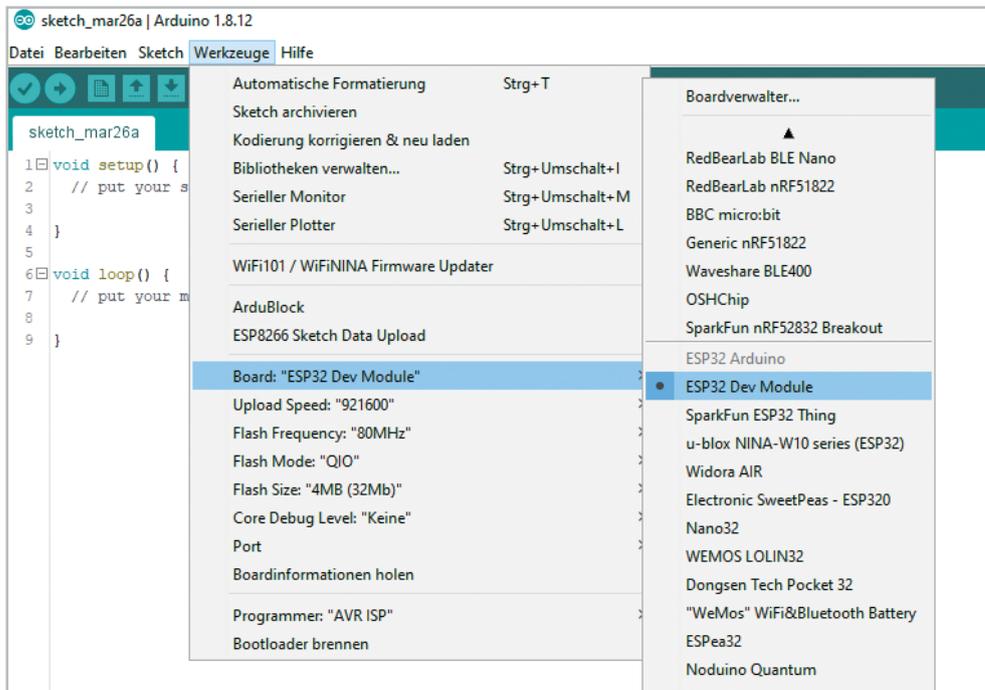


Bild 3: ... anschließend kann das Board ausgewählt werden.

Detaillierte Installationsanweisungen gibt es in der im Download-Bereich der Entwicklungsplatte verfügbaren Bedienungsanleitung.

## Tonerzeugung mit dem Mikrocontroller

Die einfachste Art, einen Ton mit digitaler Technik zu produzieren, ist die Erzeugung einer Rechteckfrequenz. Da Rechtecksignale sehr obertonhaltig sind, klingen sie jedoch hart und „technisch“. Der folgende Sketch erzeugt ein Rechtecksignal mit einer fest vorgegebenen Frequenz:

```
// square wave generator.ino

int freq = 1000;
int channel = 0;
int resolution = 8;           // 2^8 = 256 steps
int outputPin = 12;

void setup()
{ ledcSetup(channel, freq, resolution);
  ledcAttachPin(outputPin, channel);
  ledcWriteTone(channel, freq);
}

void loop()
{ }
```

Wir nutzen in diesem Fall eine Funktion für die Erzeugung einer Pulsweitenmodulation (PWM = Pulse-Width-Modulation), die eigentlich u. a. für LEDs gedacht ist.

In der Funktion `ledcSetup(channel, freq, resolution)` definieren wir zunächst einen Kanal. Der ESP32 hat insgesamt 16 Kanäle (0–15). Wir wählen den Kanal 0. Für die Frequenz wählen wir 1000 Hz (1 kHz) und die Auflösung des Duty-Cycles definieren wir mit 8 Bit, also in 256 Schritten. Die Ausgabe des PWM-Signals erfolgt an GPIO 12 (`outputPin`) und wird mit

`ledcAttachPin(outputPin, channel)` definiert. Schließlich wird der Ton mit `ledcWriteTone(channel, freq)` ausgegeben. <sup>1</sup>

Mithilfe eines Oszilloskops kann der erzeugte Spannungsverlauf an Pin 12 des ESP-Controllers dargestellt werden. Wie erwartet erhält man eine Rechteckkurve mit einer Frequenz von 1,000 kHz. Die Amplitude beträgt aufgrund der Betriebsspannung des ESP32-Chips 3,3 V (siehe Bild 4). Das Programm findet sich, wie auch alle folgenden, im Downloadpaket [1] zu diesem Beitrag.

Für einen weicheren Klangeindruck sind reine Sinuswellen erforderlich. Diese können nicht durch digitales Ein- und Ausschalten eines Ports erzeugt werden. Einen Ausweg bietet die Verwendung eines Digital-Analog-Konverters (DAC für engl. Digital-to-Analog-Converter). Damit können auch analoge Spannungswerte ausgegeben werden. Man muss dann lediglich dafür sorgen, dass ein digital erzeugter sinusförmiger Signalverlauf in ausreichend schneller Abfolge an den DAC ausgegeben wird. Das folgende Programm erzeugt eine Sinusschwingung unter Verwendung des im ESP32 integrierten DACs (s. a. Downloadpaket):

```
// sinus_generator.ino
// ESP32

#define steps 100

int SineValues[steps];           // an array to store our values for sine
float ConversionFactor = (2*PI) / steps;
float RadAngle;
int amplitude = 127;
int offset = 128;

void setup()
{ for(int alpha = 0; alpha < steps; alpha++)
  { RadAngle = alpha * ConversionFactor;
    SineValues[alpha]=(amplitude * sin(RadAngle)) + offset;
  }
}

void loop()
{ for(int i = 0; i < steps; i++)
  { dacWrite(25, SineValues[i]);
  }
}
```



Bild 4: Rechteck-Schwingung mit 1 kHz

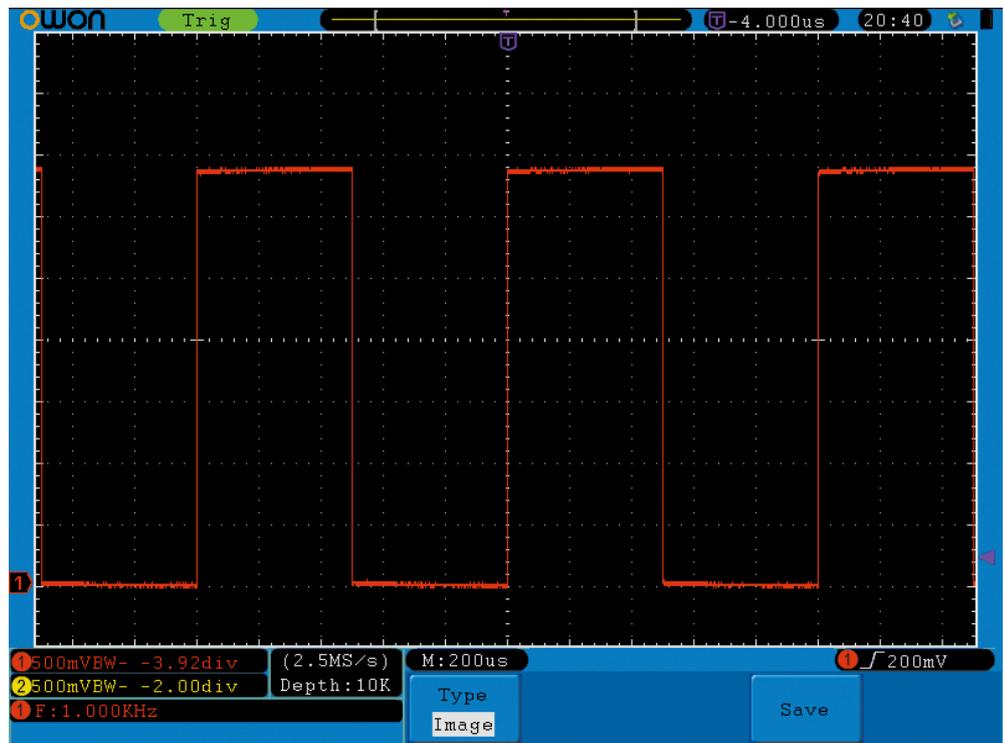
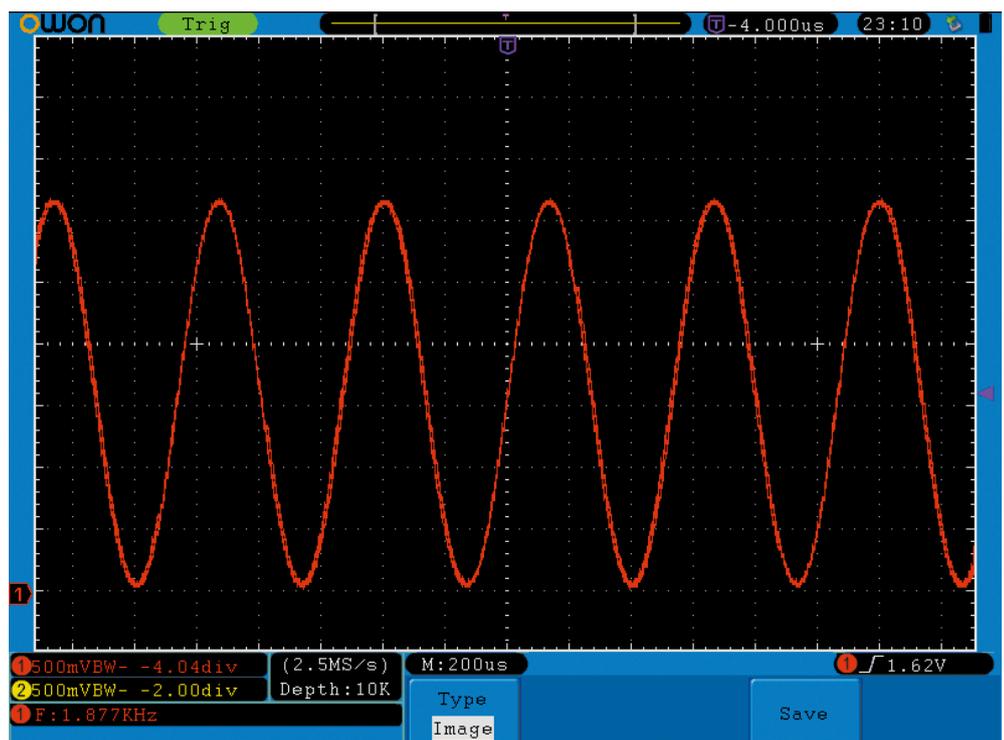


Bild 5: Sinusförmiger Spannungsverlauf



Da die Berechnung von Sinuswerten verhältnismäßig aufwendig ist, erfolgt eine Vorberechnung im Set-up-Teil des Programms. Die Variable „steps“ legt die Anzahl der Stützpunkte fest, die im Array „SineValues“ abgespeichert werden. In der Hauptschleife werden die Werte dann direkt an den DAC über Pin 25 des Controllers ausgegeben. Bild 5 zeigt das Ergebnis.

Trotz des Abspeicherns der Sinuswerte in einer Tabelle zeigt sich hier ein Problem: Das Verfahren ist vergleichsweise langsam. Mit 100 Stützpunkten

pro Periode ergibt sich eine Frequenz von lediglich 1,8 kHz. Dennoch lassen sich damit einige interessante Audio-Effekte realisieren. Anstelle von Rechtecktönen können jetzt auch einfache Klänge realisiert werden.

Das folgende Programmbeispiel liefert einen glockenähnlichen Klang, wie er als Warnsignal, etwa bei nicht angeschnallten Gurten in einem Fahrzeug, häufig zu hören ist. Der Sketch dazu sieht so aus (siehe auch Downloadpaket):



```
// digital_bell.ino
// bell like sounds using fast PWM
// ESP32

#define audioPin 25 // DAC audio output
#define steps 255

byte SineValues[steps];
byte amplitude = 127;
byte offset = 128;
float ConversionFactor = (2*PI) / steps;
float RadAngle;

void setup()
{ pinMode(audioPin, OUTPUT);

  // pre-calculation of sine values
  for(int alpha = 0; alpha < steps; alpha ++)
  { RadAngle = alpha * ConversionFactor;
    SineValues[alpha]=(amplitude * sin(RadAngle)) + offset;
  }
}

void loop()
{ for (int ampl = 25; ampl >=0 ; ampl--) // envelope
  { for (unsigned int d = 0; d < 20; d++) // cycles per step
    { for (unsigned int j = 0; j < 255; j++) // sine output
      { dacWrite(audioPin, SineValues[j]/25.0*ampl); delayMicroseconds(3);
      }
    }
  }
  delay(200);
}
```

Das Signal kann über einen hochohmigen Schallwandler ( $R > 200 \Omega$ ) direkt hörbar gemacht werden. Für den Anschluss eines normalen Lautsprechers ( $R = 8 \Omega$ ) ist dagegen ein Verstärker notwendig, wie er weiter unten im Beitrag beschrieben wird.

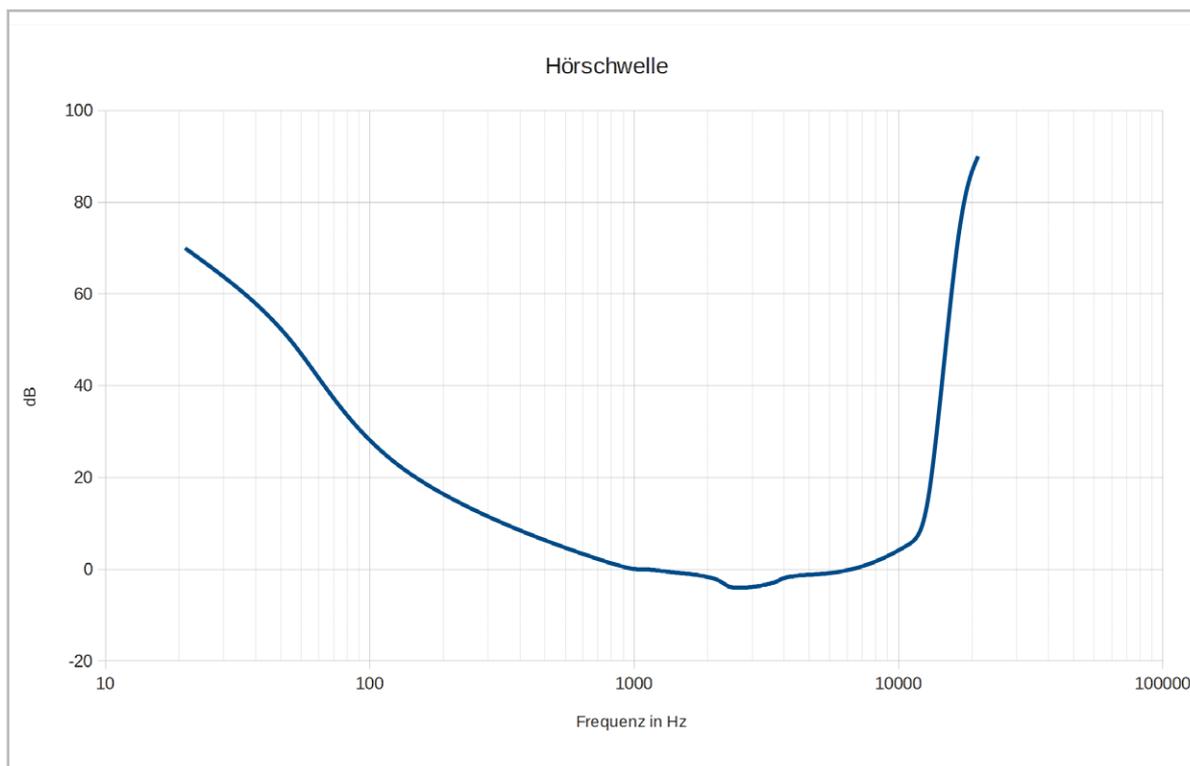


Bild 6: Hörschwelle



## Hörtestgerät im Eigenbau

Nachdem der Unterschied zwischen Tönen und Klängen nun klar sein sollte, stellt sich die Frage, zu welchen Leistungen das menschliche Gehör fähig ist. Von besonderer Bedeutung ist dabei die sogenannte Hörfläche. Damit wird der Frequenz- und Lautstärkebereich, der vom menschlichen Gehör wahrgenommen werden kann, umrissen. Die persönliche Hörfläche gibt Aufschluss über das Hörvermögen eines einzelnen Menschen.

Hörprobleme, von Einschränkungen bei einzelnen Frequenzen bis hin zur Altersschwerhörigkeit, werden in einer individuell erstellten Hörfläche deutlich. Die Hörfläche wird für niedrige Schallpegel von der sogenannten Hörschwelle, also dem gerade noch hörbaren Schalldruckpegel, begrenzt. Die obere Grenze bildet die Schmerzschwelle. Die tiefsten vom Menschen wahrnehmbaren Frequenzen liegen bei etwa 16 Hz bis 21 Hz. Die höchsten hörbaren Frequenzen von etwa 16 kHz bis maximal 20 kHz sind stark vom Lebensalter abhängig.

Das Hörvermögen erreicht sehr gute Werte zwischen 2 kHz und 5 kHz. Das ist vermutlich entwicklungsbiologisch bedingt, da hier die meisten Laute der gesprochenen Sprache auftreten. Darüber hinaus liegen in diesem Bereich auch andere ehemals lebenswichtige Geräusche wie etwa das Rascheln von bewegtem Laub oder Tierlaute.

Frequenzen unterhalb von 16 Hz werden als Infraschall bezeichnet. Dieser kann bei ausreichend hohen Pegeln als Vibration wahrgenommen werden, sodass es hier keine absolut genau definierte Grenze gibt. Die Wahrnehmbarkeit an der oberen Frequenzgrenze dagegen ist vergleichsweise scharf. Das Hörvermögen bricht oberhalb von ca. 20 kHz sehr schnell ab. In diesem sogenannten Ultraschallbereich können auch sehr hohe Schallpegel vom Menschen nicht mehr wahrgenommen werden. Viele Tiere wie etwa Katzen oder Hunde können dagegen auch deutlich höhere Frequenzen als der Mensch erfassen. Fledermäuse erreichen sogar Rekordwerte mit einem Wahrnehmungsvermögen von Schallfrequenzen bei über 100 kHz.

Die für die Beurteilung des Hörvermögens besonders wichtige Hörschwelle ist derjenige Schalldruckpegel, bei welchem Töne oder Geräusche gerade noch hörbar sind (Bild 6).

Bei einem gesunden Menschen liegt die Hörschwelle bei einer Frequenz von 2 kHz bei einem Schalldruck von 20  $\mu\text{Pa}$ . Dieser Wert wird daher als Schalldruckpegel von null Dezibel (dB) festgelegt. Sowohl bei tieferen als auch bei hohen Frequenzen ist die Hörschwelle zu höheren Pegeln hin verschoben. So werden Sinustöne von 30 Hz oder 15 kHz erst ab einem Schalldruckpegel von ungefähr 60 dB wahrgenommen. Dies entspricht gemäß der logarithmischen Dezibelskala einem Schalldruck der 1000 Mal höher ist als der Wert bei 2 kHz. Die höchste Empfindlichkeit des menschlichen Gehörs liegt zwischen ca. 2 und 4 kHz. Hier werden Werte von bis zu -5 dB, entsprechend einem Schalldruck von ca. 11  $\mu\text{Pa}$ , erreicht.

Neben dem Lebensalter können auch andere Einflüsse die Hörschwelle verschieben. Insbesondere lang andauernder Lärm oder auch lautstarke Musik

führen zu einer deutlichen Anhebung der Hörschwelle bis hin zu Schwerhörigkeit.

Die Ermittlung der Hörschwelle muss mit Sinustönen vorgenommen werden. Die in anderen Signalformen enthaltenen Obertöne würden das Ergebnis in nicht reproduzierbarer Weise verfälschen. Bei der klassischen Tonaudiometrie kommen daher Sinustöne mit festen Frequenzen (meist 250, 500, 1000, 2000, 3000, 4000, 6000 und 8000 Hz) zum Einsatz. Alternativ kann die Tonfrequenz aber auch langsam und kontinuierlich über den ganzen Frequenzbereich durchgestimmt werden.

Im hörmedizinischen Bereich werden exakt kalibrierte Kopfhörer verwendet. Im sogenannten Eingabelungsverfahren arbeitet man mit ansteigenden Schallpegeln. Der Prüftonepegel wird dabei bei verschiedenen Frequenzen aus dem unhörbaren Bereich bis zum Hörbarwerden gesteigert. Das Ergebnis ist ein sogenanntes Tonaudiogramm, in welchem die individuelle Hörschwelle in Abhängigkeit von der Frequenz aufgetragen wird. Ist das Hörvermögen beeinträchtigt, dann weicht die Hörschwelle von der Normkurve ab. Das Tonaudiogramm ist die Grundlage bei der Diagnose eines Hörschadens oder einer Hörbehinderung.

Mit der im nächsten Abschnitt vorgestellten Methode kann man sein Gehör für eine erste Einschätzung des Hörvermögens selber testen. Natürlich kann dieses Verfahren keinen professionellen Hörtest beim Audiologen ersetzen. Trotzdem ist es sinnvoll und zudem auch sehr interessant, einen Hörtest mit selbst entwickelten Geräten durchzuführen. Falls sich Schwachstellen offenbaren, sollten diese selbstverständlich umgehend von einem Facharzt abgeklärt werden.

## Mikrocontroller-basierter Tongenerator

Wie bereits gezeigt wurde, können über die klassische DAC-Ansteuerung nur Sinusfrequenzen von maximal ca. 2 kHz erzeugt werden. Für die Aufnahme eines Tonaudiogramms sollten aber Frequenzen von über 10 kHz zur Verfügung stehen. Will man sogar die obere Hörfrequenzgrenze testen, werden bis zu 20 kHz benötigt. Hier kommt ein besonderes Leistungsmerkmal des ESP32-Controllers zum Tragen. Neben den umfangreichen und bestens bekannten Kernfunktionalitäten des Chips wie WLAN und Bluetooth kann dieser auch mit einem integrierten Cosine-Waveform-Generator aufwarten. Dieser ist im technischen Referenzhandbuch zum ESP32 [2] und dem ESP-IDF Programming Guide [3] beschrieben.

Eine ausführlichere Beschreibung gibt es im GitHub-Repository von Krzysztof [4].

Allerdings wird der Cosine-Waveform-Generator von der ESP-IDF bislang nicht explizit unterstützt. Dennoch kann man den Generator sogar mit der Arduino-IDE erfolgreich und nutzbringend einsetzen.

Der Generator erlaubt die Erzeugung von Sinusfrequenzen mit bis zu 250 kHz in hoher Qualität und mit geringem Oberwellenanteil. Damit ist er bestens zum Aufbau eines Hörtestgerätes geeignet. Ein Sketch für die Erzeugung von Sinustönen kann so aussehen:



```
// AudioTester_sinus_generator_FAST.ino
// ESP32
// IDE 1.8.12

#include „soc/rtc_io_reg.h“
#include „soc/rtc_cntl_reg.h“
#include „soc/sens_reg.h“
#include „soc/rtc.h“
#include „driver/dac.h“

int clk_8m_div = 0;      // 0 ... 7: RTC 8M clock divider (division is by clk_8m_div+1, i.e. 0 -> 8MHz frequency)
int frequency_step = 1; // 1: 133 Hz ... 200: 26.500 Hz: Frequency for CW generator
int scale = 1;          // 0: 100 % - 1: 50% - 2: 25% - 3: 12.5% of full scale
int offset = 0;        // 0 = no offset
int invert = 2;         // invert MSB for correct sine waveform

int freq_array[] = {1, 2, 3, 5, 8, 10, 15, 18, 25, 30, 50, 80, 90, 92, 93, 95, 98, 100, 102, 105, 110, 115, 121, 123, 125};

void setup()
{ Serial.begin(250000);
}

void loop()
{ for (int n = 1; n <= 23; n++)
  { frequency_step = freq_array[n];

    Serial.println(frequency_step);

    // Enable generator
    SET_PERI_REG_MASK(SENS_SAR_DAC_CTRL1_REG, SENS_SW_TONE_EN);
    // connect generator to this channel
    SET_PERI_REG_MASK(SENS_SAR_DAC_CTRL2_REG, SENS_DAC_CW_EN1_M);
    // Invert MSB, otherwise part of waveform is inverted
    SET_PERI_REG_BITS(SENS_SAR_DAC_CTRL2_REG, SENS_DAC_INV1, 2, SENS_DAC_INV1_S);

    dac_output_enable(DAC_CHANNEL_1);
    // set SINE generator registers
    REG_SET_FIELD(RTC_CNTL_CLK_CONF_REG, RTC_CNTL_CK8M_DIV_SEL, clk_8m_div);
    SET_PERI_REG_BITS(SENS_SAR_DAC_CTRL1_REG, SENS_SW_FSTEP, frequency_step, SENS_SW_FSTEP_S);
    SET_PERI_REG_BITS(SENS_SAR_DAC_CTRL2_REG, SENS_DAC_SCALE1, scale, SENS_DAC_SCALE1_S);
    SET_PERI_REG_BITS(SENS_SAR_DAC_CTRL2_REG, SENS_DAC_DC1, offset, SENS_DAC_DC1_S);
    SET_PERI_REG_BITS(SENS_SAR_DAC_CTRL2_REG, SENS_DAC_INV1, invert, SENS_DAC_INV1_S);

    float frequency = RTC_FAST_CLK_FREQ_APPROX / (1 + clk_8m_div) * (float) frequency_step / 65536;
    vTaskDelay(2000/portTICK_PERIOD_MS);

    delay(3000);
  }
}
```

Auch dieses Programm ist wieder im Downloadpaket zu diesem Beitrag enthalten.

Das Spektrum in [Bild 7](#) zeigt, dass ein nahezu oberwellenfreies Sinussignal erzeugt wird.

Das Signal steht seitens des ESP32 an Pin 25 mit konstanter Amplitude zur Verfügung. Über eine 3,5-mm-Buchse kann es direkt in einem geeigneten Kopfhörer wiedergegeben werden. Zur Reduktion der Lautstärke und zum Schutz des Controllerausgangs sollte mindestens ein 220-Ohm-Widerstand in Serie zum Kopfhörer geschaltet werden ([siehe Bild 8](#)).



### Wichtiger Hinweis:

Achtung! Testen Sie vorsichtig die Lautstärke – je nach verwendetem Kopfhörer/Lautsprecher kann diese sehr hoch sein.

Um ein Audiogramm aufnehmen zu können, muss aber noch die Lautstärke, d. h. die Signalamplitude,



Bild 7: Der ESP32 erzeugt ein nahezu oberwellenfreies 16-kHz-Sinussignal.

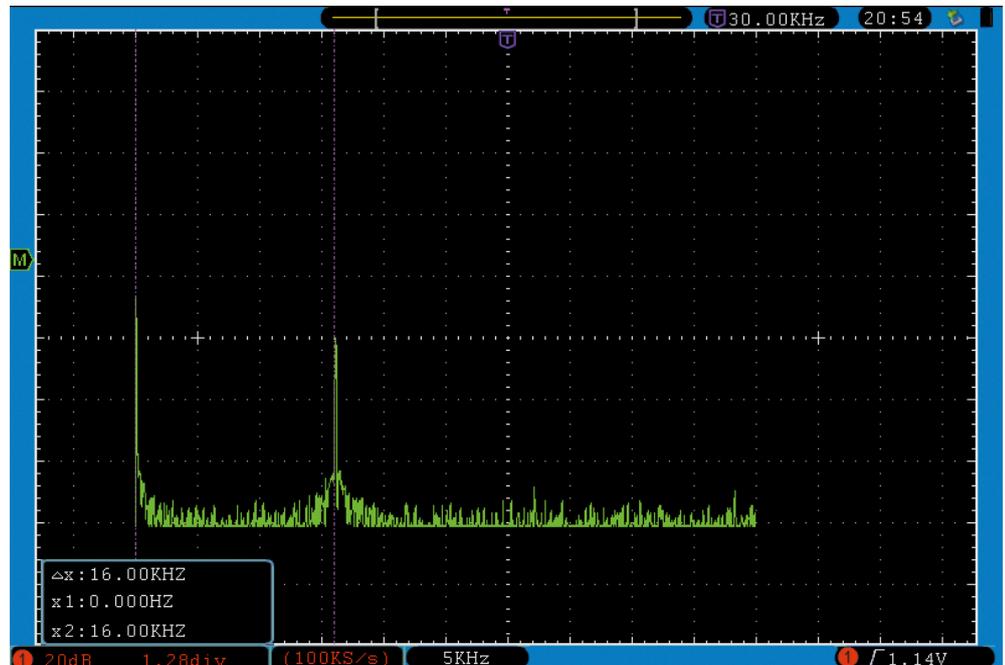
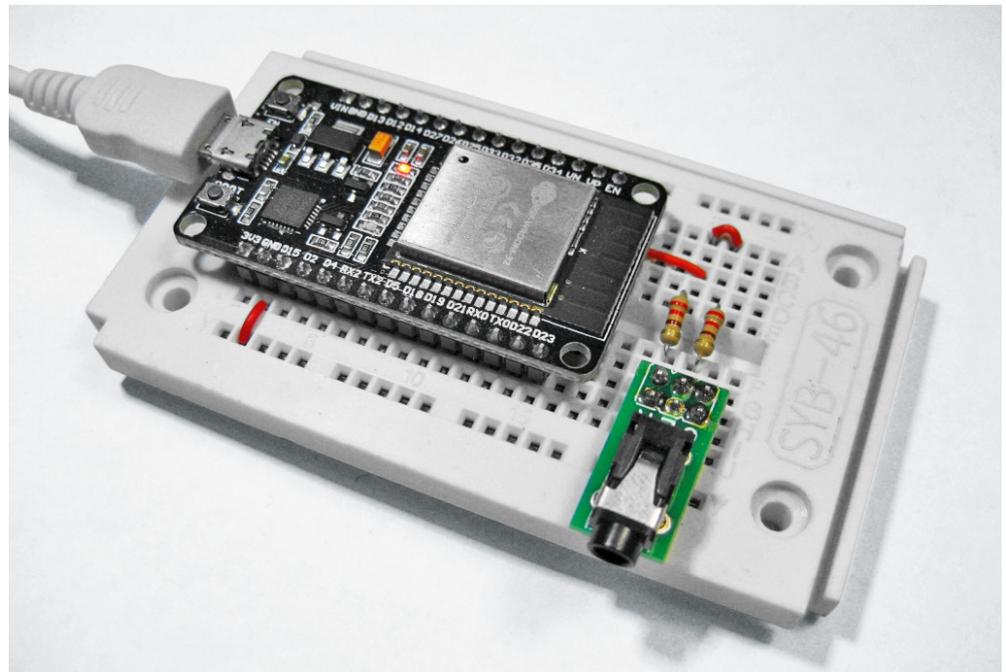


Bild 8: ESP32 mit Audio-Ausgang



variiert werden. Prinzipiell ist die Signalamplitude zwar auch digital beeinflussbar, allerdings zeigt es sich, dass eine analoge Variante einfacher und auch leichter handhabbar ist. Ein Analogverstärker mit einem präzisen Potentiometer ist für diese Aufgabe bestens geeignet. Grundsätzlich ist jedes Potentiometer verwendbar. Da es hier lediglich als Spannungsteiler dient, ist der Nominalwiderstand unkritisch und Werte von 5 bis 100 k $\Omega$  können eingesetzt werden.

Präzisionspotentiometer (sogenannte HeliPots, siehe Bild 9) sind jedoch besonders gut geeignet, da sie eine exakte Ablesung des eingestellten Wertes erlauben. Der in Bild 9 gezeigte Drehknopfzähler kann im Bedarfsfall durch einen gewöhnlichen Potentiometerknopf ersetzt werden. In diesem Fall kann man sich beispielsweise mit einer selbst gefertigten Skala behelfen und die vollen Umdrehungen mitzählen.



Bild 9: HeliPot

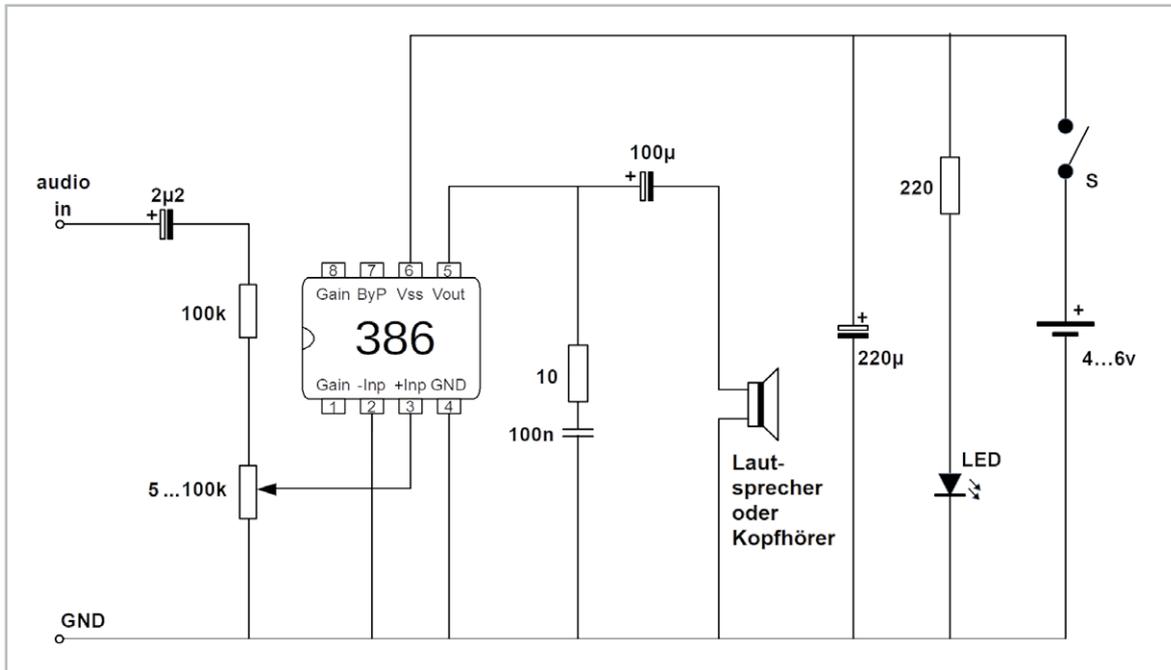


Bild 10: Schaltbild zum Audioverstärker

Wie im folgenden Abschnitt dargelegt, kommt es ohnehin weniger auf die absoluten Messwerte an als vielmehr auf den Vergleich von Potentiometerwerten.

Bild 10 zeigt das Schaltbild für einen geeigneten Audioverstärker, Aufbauvorschlage dazu sind in Bild 11 (mit Prototypenadaptern) und Bild 12 (mit Gehuse) zu sehen.

Fur den Verstarker kommt der Standardbaustein LM386 zum Einsatz. Er liefert fur diese Anwendung eine ausreichende Leistung und ist zudem fur die Verstarkung von Frequenzen von bis zu 100 kHz geeignet. Am Eingang wird uber einen 2,2-µF-Kondensator der Gleichspannungsanteil des Mikrocontrollersignals abgetrennt. Danach folgt ein Spannungsteiler, der die Regelung der Lautstarke ermoglicht. Der hier eingesetzte 100-kΩ-Widerstand kann bei Bedarf auch auf 1 MΩ erhohet werden. Damit wird die maximale Lautstarke deutlich reduziert, sodass das Helipot praziser eingestellt werden kann. Die R-C-Kette aus 10 Ω und 100 Nanofarad am Ausgang kann unter Umstanden entfallen. Sie dient lediglich zur Unterdruckung von Schwingneigungen und muss nur eingebaut werden, falls unerwartete Pfeiff- oder Brummgerausche horbar sind.

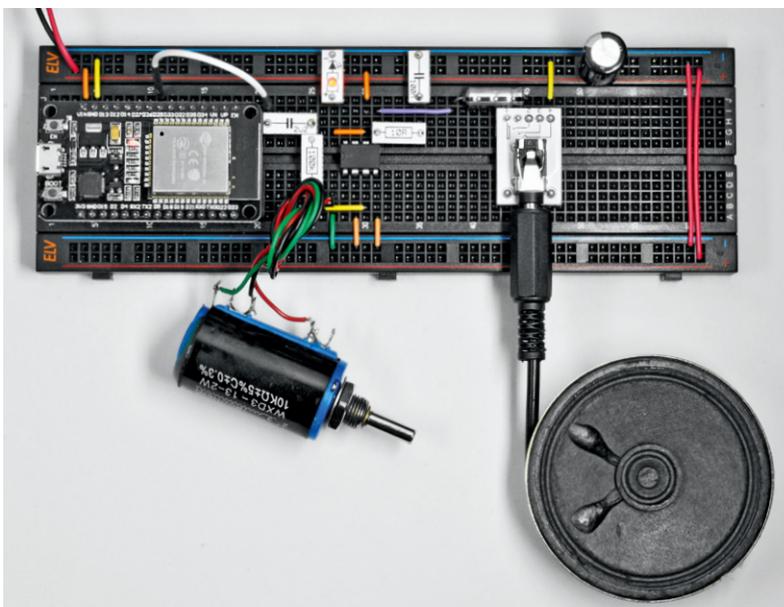


Bild 11: Aufbauvorschlag der Verstarkerschaltung mit ESP32



### Wichtiger Hinweis:

Bitte stets darauf achten, dass die Lautstarke im Kopfhorer nicht zu hoch wird. Es empfiehlt sich, fur erste Tests den Kopfhorer noch nicht aufzusetzen, sondern zunachst mit gewissem Abstand zu prufen, ob die Lautstarke nicht unangenehm ist.

**Zu hohe Lautstarken konnen Horschaden verursachen!**

### Bioelektronik als Familienereignis: Aufnahme von Audiogrammen

Die Messung lauft nun folgendermaen ab: Nach dem Start des Programms AudioTester.ino wird der tiefste Ton (133 Hz) ausgegeben. Der zugehorige Frequenzschritt wird im seriellen Monitor der Arduino-IDE angezeigt. Nun wird das Potentiometer so lange von null

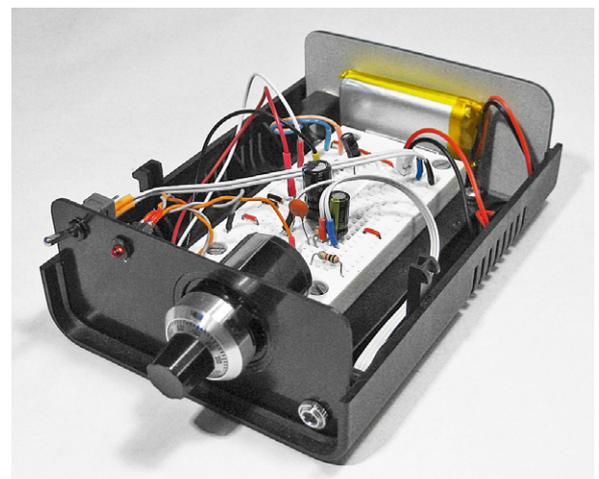


Bild 12: Aufbauvorschlag im Gehuse



beginnend aufgedreht, bis die Testperson den Ton hört. Dieser Potentiometerwert wird in eine Audiogrammtabelle eingetragen. Ein passendes Excel-Template dazu findet sich im Downloadpaket.

Nach drei Sekunden folgt der nächste Ton (266 Hz) und die Prozedur wird wiederholt. Die Messung ist abgeschlossen, wenn der letzte Wert (125 mit einer Frequenz von 16,625 kHz) protokolliert wurde. **Bild 13** zeigt ein Beispiel, in welchem die Hörkurven von vier verschiedenen Testpersonen im Alter von 15 bis 78 Jahren eingetragen sind. Hier zeigt sich, dass auch mit einfachen Mitteln bereits Unterschiede im Hörvermögen klar nachweisbar sind.

Die Frequenzen werden vom Mikrocontroller mit hoher Präzision erzeugt. Die Amplitude bzw. die Lautstärke können dagegen nur mit professionellen Messgeräten präzise festgelegt werden. Absolute Messwerte sind daher mit den hier vorgestellten Mitteln kaum möglich. Dies ist aber auch gar nicht notwendig, da vergleichende Messreihen ebenso aufschlussreich sein können.

Damit wird die Aufnahme der Audiogramme zum Familienereignis. Insbesondere sollten die in vielen Studien nachgewiesenen Ergebnisse auch hier sichtbar werden. Im Allgemeinen wird sich zeigen, dass sich bei vielen Menschen das Gehör ab einem Lebensalter von etwa 60 Jahren deutlich verschlechtert. Der Verlust der Hörfähigkeit ist am stärksten bei höheren Frequenzen ausgeprägt. Damit fällt es betroffenen Personen mit der Zeit immer schwerer, auch normale Sprache im Frequenzbereich zwischen 500 Hz und 3 kHz zu verstehen.

Im unteren Frequenzbereich zwischen 250 und 1000 Hz ist der Hörverlust bei Frauen und Männern nahezu identisch. Bei hohen Frequenzen von 2 bis 8 kHz ist der Hörverlust bei Männern dagegen stärker ausgeprägt. Der Unterschied gleicht sich mit zunehmendem Alter jedoch meist wieder aus. In jüngeren Altersgruppen sind Hörschäden dagegen wesentlich ungleicher verteilt. Hier kommt vor allem der Einfluss der individuellen Lebensführung zum Tragen. Partys, Rockkonzerte oder überlaut eingestellte Kopfhörer fordern ihren Tribut.

## Fazit und Ausblick

Töne und Klänge können mit elektronischen Mitteln leicht erzeugt werden. Das Spektrum reicht vom einfachen Warnsignal bis zu komplexen Synthesizerklängen. Die moderne Mikrocontrollertechnik erlaubt es dabei, ehemals hardwarebasierte Systeme vollständig in Software umzusetzen. Lediglich in der Filter- und Verstärkertechnik sind noch analoge Schaltungskomponenten erforderlich. Sogar biometrische Messungen können mit einfachen Mitteln durchgeführt werden. Auch hier trägt die Digitalisierung wesentlich zu einer Vereinfachung des Schaltungsaufbaus bei.

Mit den in diesem Artikel vorgestellten Methoden kann kein professioneller Hörtest ersetzt werden. Insbesondere im Vergleich mit anderen Personen können aber durchaus Hördefizite in Erscheinung treten. Diese sollten dann beim Audiologen überprüft werden. So sind unter Umständen sogar weitergehende Hörschäden vermeidbar.

Der nächste Artikel zu dieser Beitragsserie wird sich um die Erfassung von physiologischen Parametern drehen. Durch die Messung des Hautwiderstands kann der Einfluss von Gemütszuständen wie Nervosität oder Stress aufgezeigt

werden. Die hochpräzise Temperaturerfassung bis hin zur Fiebermessung zeigt die Einflüsse von Infekten oder auch körperlichen Aktivitäten auf die Hauttemperatur. Über die optische Pulsmessung kann die Abhängigkeit der Pulsfrequenz von der körperlichen Verfassung und der aktuellen physischen Belastung dargestellt werden. Auf diese Weise lassen sich eine ganze Reihe von subjektiven Gemütszuständen mit objektiven Messungen hinterlegen. **ELV**



## Weitere Infos:

- [1] Das Downloadpaket zu diesem Artikel findet sich im ELVshop unter dem Webcode: #10317
  - [2] Technisches Datenblatt ESP32 – Cosine-Waveform-Generator (Espressif): [https://www.espressif.com/sites/default/files/documentation/esp32\\_technical\\_reference\\_manual\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf)
  - [3] ESP-IDF Programming Guide (Espressif): <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/dac.html?highlight=cosine>
  - [4] GitHub-Repository zum Cosine-Waveform-Generator: <https://github.com/krzychb/dac-cosine>
- Alle Links finden Sie auch online unter [de.elv.com/elvjournal-links](http://de.elv.com/elvjournal-links)

Neben diversen Kleinteilen wie Widerständen, Schaltern und Kondensatoren werden die folgenden Komponenten benötigt:

Material	Best.-Nr.
Entwicklungsplatine NodeMCU mit ESP32	145164
Präzisions-Draht-Potentiometer 20 kΩ	114495
IC LM386N	019960
Breitbandlautsprecher	129823
Breadboard	125905
Universal-Gehäuse (für Kopfhörerbetrieb)	129126
Gehäuse (für Lautsprechereinbau)	030464

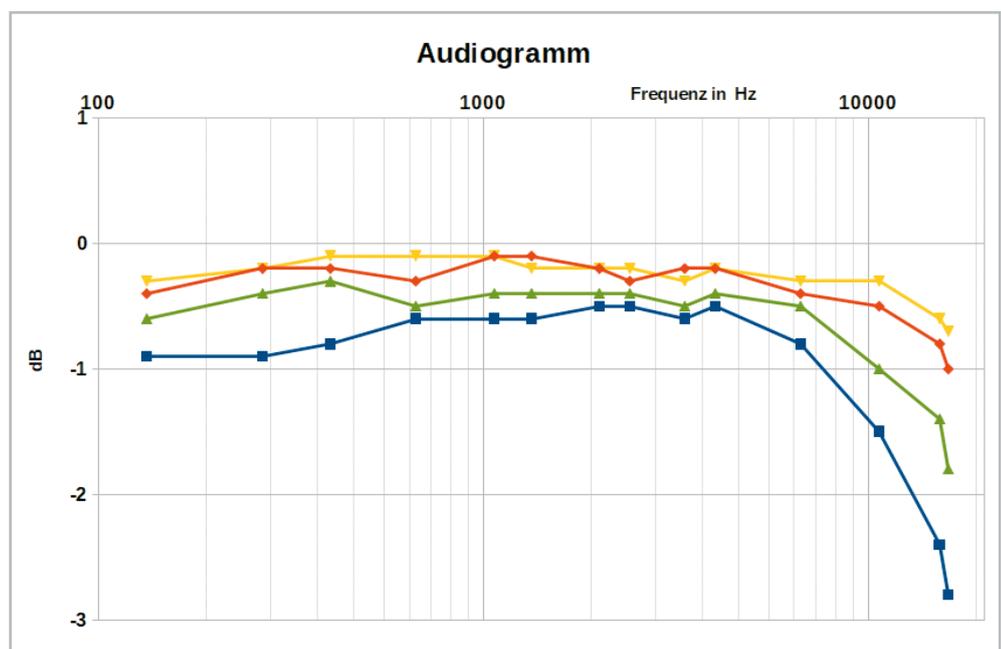


Bild 13: Audiogramm von vier Testpersonen: Gelb = weiblich, 15 Jahre; Orange = weiblich, 48 Jahre; Grün = männlich, 52 Jahre; Blau = männlich, 78 Jahre