

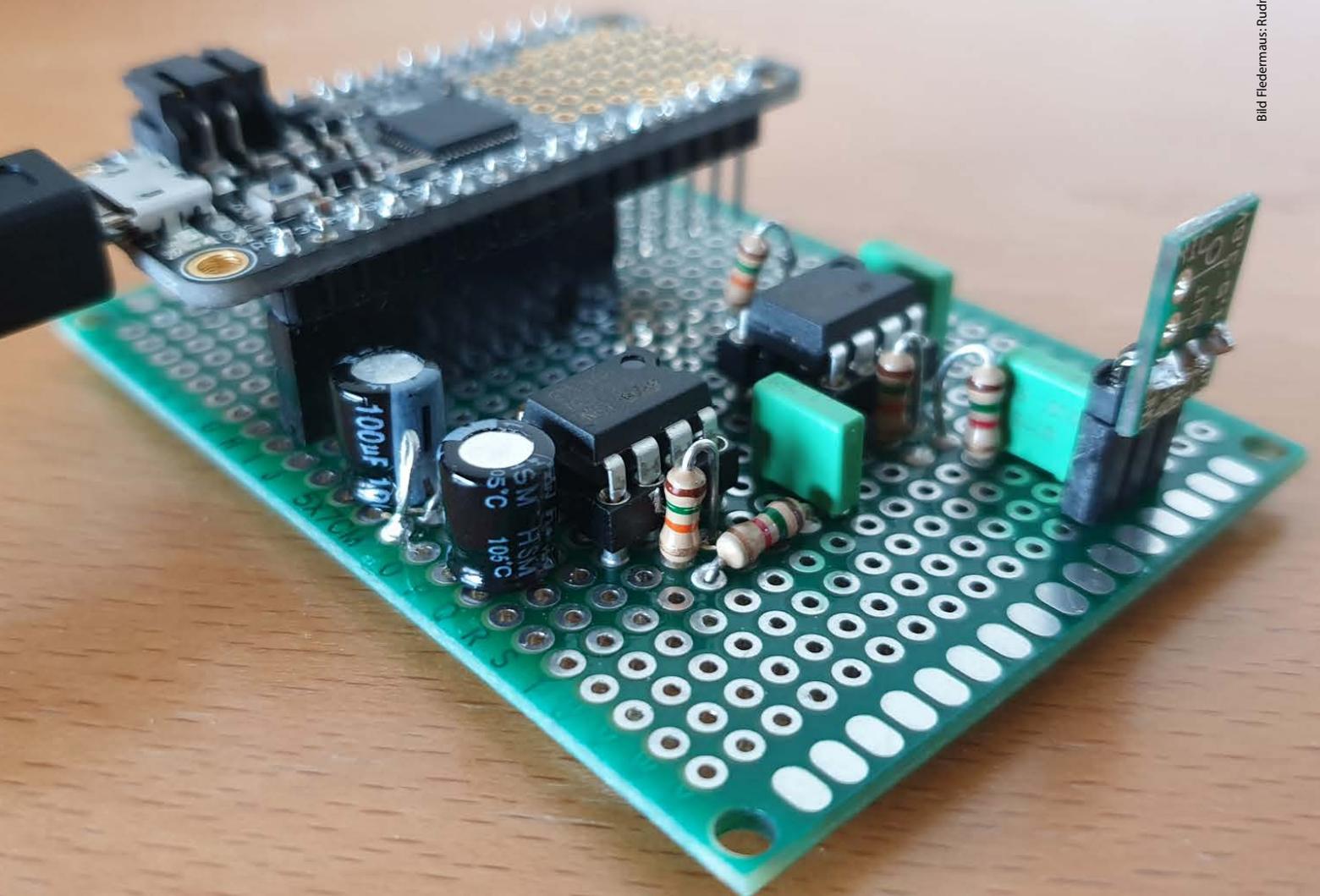
Fledermaus-Scanner

Dunkel wie die Nacht sind Fledermäuse nicht nur vor unseren Augen gut getarnt – sie kommunizieren und orientieren sich auch außerhalb unseres Hörbereichs. Mit einem MEMS-Mikrofon kommt man den Nachtschwärmern aber schnell auf die Spur.

von Ralf Stoffels



Bild Fledermaus: Rudmer Zwerver / Shutterstock.com



Als ich an langen und warmen Sommerabenden auf der Terrasse saß, fiel mir auf, dass in der Dämmerung reger Fledermaus-Flugbetrieb in unserem Garten herrschte. Vermutlich ist das die beste Jagdzeit, um viele Insekten zu erwischen, von denen sich hiesige Fledermäuse ernähren. Da ich kein Biologe bin, musste ich mein lange verblasstes Schulwissen erst mal im Web auffrischen, um dort zu lernen, dass es mehr als etwa dreißig Fledermausarten in Mitteleuropa gibt, von denen viele leider vom Aussterben bedroht sind.

Die Arten lassen sich meist anhand ihrer Ultraschall-Rufe identifizieren. Hierbei unterscheiden sich die Frequenzen, aber auch die Hüllkurven der Rufe – also die Geschwindigkeit, in der der Ruf an- oder abklingt. Fledermäuse erzeugen die Ultraschallsignale zur Erkennung von Beute und Hindernissen und zur Kommunikation untereinander. Die Frequenzen für Kommunikation und Navigation unterscheiden sich dabei deutlich. Die Frequenz der Signale liegt artspezifisch zwischen 20kHz und mehr als 120kHz – dort hören wir Menschen nichts mehr.

Daher gibt es jede Menge käuflicher Geräte, darunter auch Bausätze, die die Laute in den menschlichen Hörbereich von 100Hz bis ca. 16kHz verschieben. Hierbei wird eine ähnliche Methode wie bei Radioempfängern verwendet: ein sogenannter Mixer. Multipliziert man ein Signal der Frequenz f_1 mit einem sinusförmigen Signal einer Frequenz f_2 , so entstehen zwei Abbilder des Signals an den Frequenzpunkten f_1-f_2 und f_2+f_1 . So kann man z.B. ein 48kHz-Signal durch Multiplikation mit einem 40kHz-Sinus nach 8kHz verschieben und dadurch hörbar machen. Natürlich verschiebt sich nicht nur der eine Frequenzpunkt sondern das ganze Spektrum des Signals. Man bekommt also einen echten Höreindruck. Das Prinzip haben wir in Make 2/14 ab Seite 132 beschrieben.

Mir ging es aber darum, über den ganzen Abend aufzuzeichnen, welche Fledermausarten vorbeifledern. Daher habe ich mich für einen anderen Ansatz entschieden.

Abtasten statt mixen

Da ich keine Töne hören will, sondern die empfangenen Signale in einem Spektrum grafisch anzeigen möchte, genügt es, schnell genug abzutasten, um so ein breites Frequenzband digitalisieren zu können. Das Abtasttheorem zeigt, dass ein Signal dann verlustfrei digitalisiert werden kann, wenn die Abtastrate mehr als doppelt so hoch ist wie der höchste in dem Signal vorkommende Frequenzanteil.

Um zum Beispiel Musik, die alle Frequenzen von 0 bis 20kHz enthalten kann, verlustfrei zu digitalisieren, muss man mit mindestens 40.000 Abtastwerten pro Sekunde digitalisieren. Der klassische CD Player baut noch etwas

Kurzinfo

- » Fledermausfrequenzen abtasten und digitalisieren
- » Verstärkerschaltung mit MEMS-Mikrofon
- » Grafische Darstellung mit Processing

Checkliste



Zeitaufwand:
3 Stunden



Kosten:
30 Euro

Mehr zum Thema

- » Friederike Maier, Daniel Bachfeld, Radio in Software-Hand, Make 2/2014
- » Make Node-RED Special 2020
- » Thomas Euler, Krabbelroboter sendet Telemetrie, Make 7/2019, S. 112

Material

- » 1x Adafruit Feather M0 Express oder M0 Basic Proto Mikrocontrollerboard
- » 1x ELV Komplettbausatz Mikrofon MEMS1
- » 2x Dual Rail-to-Rail Operationsverstärker TS912
- » 3x Folienkondensator 100nF
- » 2x Elektrolytkondensator 10uF
- » Widerstände: 3x 150kOhm, 3x 15kOhm, 2x 4,7kOhm
- » 1x Lochrasterplatine, kupferbeschichtet

Alles zum Artikel im Web unter make-magazin.de/x8ra

Fledermäuse

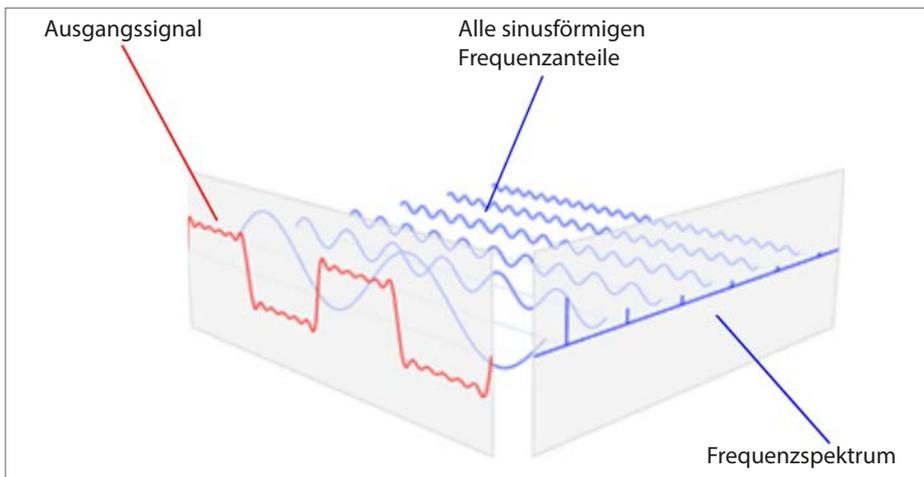
Weltweit gibt es etwa 1000 Fledermausarten. Neben den Flughunden sind sie die einzigen fliegenden Säugetiere. Durch ihre Fähigkeit, bei Nacht zu jagen, haben sie sich über viele Millionen Jahre eine erfolgreiche Nische geschaffen. Da Fledermäuse anders als Vögel ihre ungeborenen Jungen beim Fliegen mittransportieren, gebären sie üblicherweise nur ein Junges. Das macht Fledermauspopulationen sehr anfällig. Sie können sich nicht so schnell vermehren wie andere Kleinsäugetiere. Daher stehen alle Fledermausarten in Mitteleuropa unter Naturschutz.

Faszinierend sind ihre Fähigkeiten der Ultraschallortung: Fledermäuse erzeugen Ultraschallsignale im Kehlkopf, die sie entweder durch den Mund oder die Nase aussenden. Manche erreichen dabei einen Schalldruck von 140 dBA, was der Lautstärke eines Presslufthammers entspricht – zum Glück für Menschen unhörbar. Das reflektierte Echo ermöglicht es der Fledermaus, sich ein räumliches Bild der Umgebung zu machen. Je höher die Frequenz, desto feiner die räumliche Auflösung, aber desto geringer auch die Reichweite. Fledermäuse, die im Wald leben und auf kurze Distanz

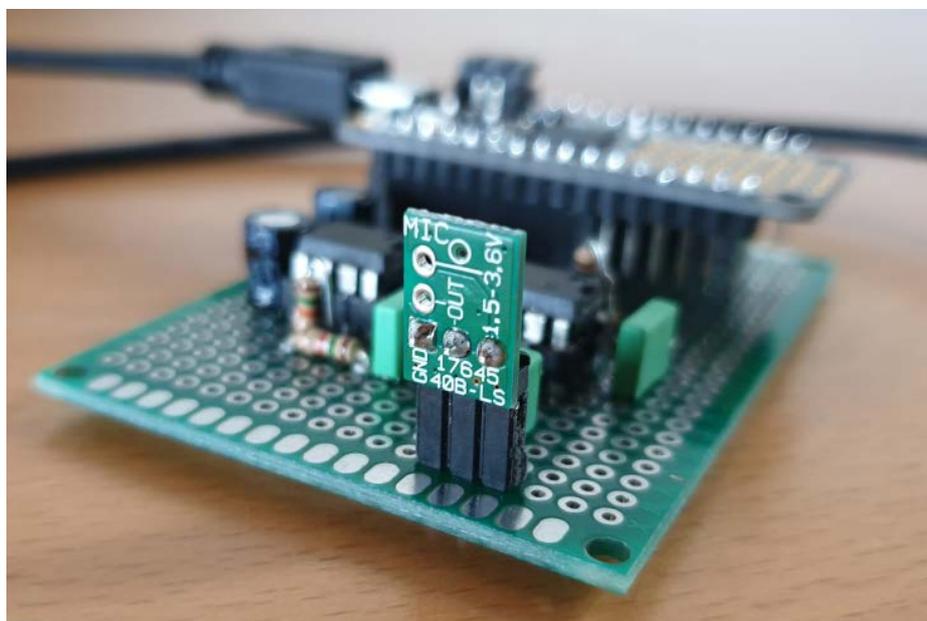
sehr genau navigieren müssen, verwenden daher eher höhere Frequenzen.

Fledermäuse sind außerdem in der Lage, das Echo des eigenen Signals aus den Echos anderer Fledermäuse herauszuhören. Dazu wird das Signal moduliert und erhält eine charakteristische Hüllkurve. Durch Laufzeitunterschiede zwischen den beiden Ohren kann die Richtung des Echos bestimmt werden. Den Dopplereffekt nutzen Fledermäuse, um die relative Geschwindigkeit von Objekten zu ermitteln. Ein fliegendes Insekt zerhackt das Echo mit seinen Flügelschlägen, was der Fledermaus weitere Information über die Art der Beute liefert.

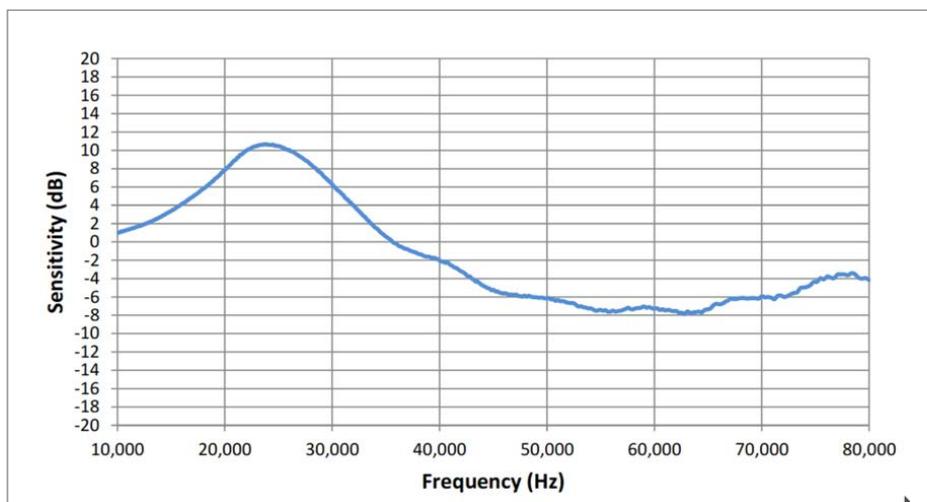
In unseren Breitengraden halten Fledermäuse im Winter in Höhlen und unbenutzten Eisenbahntunneln Winterschlaf. Einige Arten fliegen in den Süden. Hierbei legen sie einige hundert bis zweitausend Kilometer zurück. Daher kann man Fledermäuse in Mittel- und Nordeuropa nur in den Sommermonaten beobachten. Aber auch in dieser Zeit brauchen sie abgestorbene Bäume oder alte Gebäude und Dachböden für ihren Schlaf am Tage.



Das Ausgangssignal wird mittels Fouriertransformation in einzelne Sinusschwingungen zerlegt. Man erhält das Frequenzspektrum.



Das Mikro-Modul MEMS1



Frequenzgang des MEMS1

Reserve ein und verwendet eine Abtastrate von 44.1kps (kilo samples per second). Um also alle gängigen Fledermausarten abzudecken, brauchen wir mindestens 250 bis 300kps als Abtastrate. Die Digitalisierung der Werte erfolgt mit einem Analog/Digital-Wandler (ADC), so dass alle Abtastwerte digital im Speicher des Computers zur Weiterverarbeitung abgelegt werden können.

Frequenzspektrum ausrechnen

Wenn man für einen kurzen Moment mit dieser hohen Abtastrate das ganze Frequenzband von Null bis zur halben Abtastrate digitalisiert, erhält man eine Momentaufnahme aller zugleich vorkommenden Frequenzen in diesem Zeitraum. Das könnten zum Beispiel die Ortungssignale unterschiedlicher Fledermausarten sein. So wie man ein Lichtspektrum aus den unterschiedlichen Lichtwellenlängen mit einem Prisma oder Gitter zerlegen kann, nutzt man die mathematische Methode der Fouriertransformation, um ein Gesamtsignal in einzelne Frequenzanteile mit der jeweils dazugehörigen Amplitude (und Phase) zu zerlegen. Das funktioniert bei akustischen, elektrischen und mechanischen Schwingungen.

Wie bereits beschrieben gilt dabei für die maximale Bandbreite:

$$f_{\text{Max}} = f_{\text{Abtast}} / 2$$

Daraus lässt sich die Frequenzauflösung berechnen, wobei N die Anzahl der Abtastwerte ist:

$$df = f_{\text{Max}} / N \times 2$$

Taste ich zum Beispiel 500.000 mal pro Sekunde ab und sammle dabei 2048 digitalisierte Werte ein, dann decke ich ein Spektrum bis 250kHz mit einer Frequenzauflösung von circa 250Hz pro Frequenzpunkt im Spektrum ab. Je länger man Daten sammelt, desto besser wird also die Frequenzauflösung der Messung.

Da eine Fouriertransformation sehr rechenintensiv ist, hat sich die Fast-Fourier-Transformation (FFT) durchgesetzt, die durch das geschickte Ausnutzen von Symmetrien und Wiederverwenden von Berechnungsschritten die schnellste Möglichkeit der Berechnung darstellt. Um den Algorithmus bestmöglich zu nutzen, sollte die Anzahl der Abtastwerte eine Zweierpotenz sein, also zum Beispiel 1024, 2048 oder 4096.

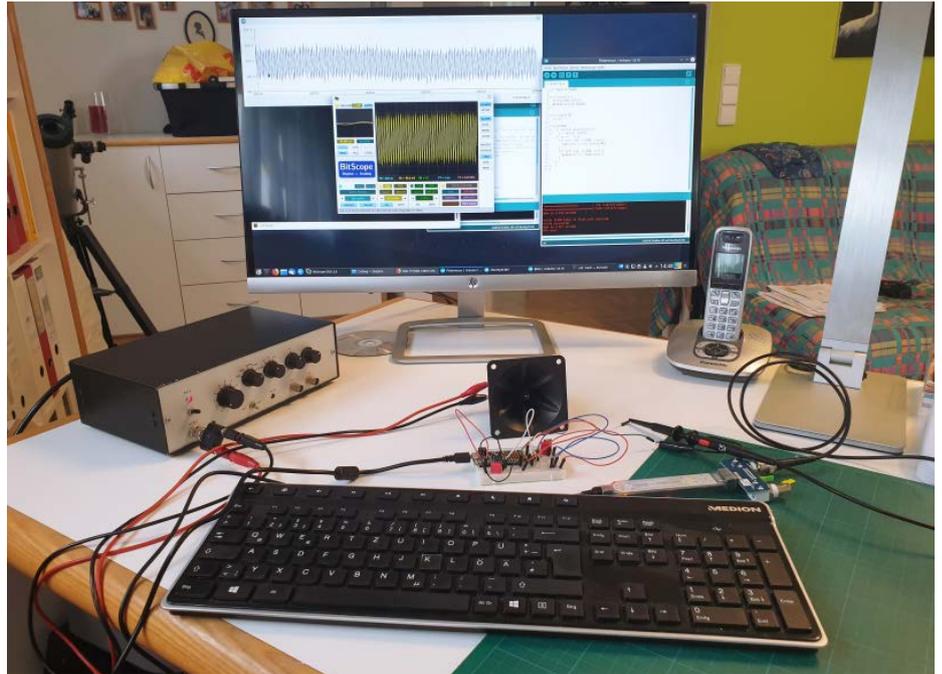
Von Arduino zu ARM M0+

Grundsätzlich ist es kein Problem, A/D-Wandler zu finden, die mindestens 300kps Abtastrate unterstützen. Will man aber einen A/D-Wandler verwenden, der in einem Mikrocontroller integriert ist, wird die Auswahl kleiner. Der klassische Arduino mit ATmega-Prozessor oder Derivaten bietet maximale Abtastraten bis ca. 20kHz, was noch nicht ein-

mal für das Spektrum im Hörbereich ausreicht. Für viele Messanwendungen mit empfindlicher Sensorik wird auch eher auf höhere Auflösung als auf hohe Frequenz optimiert.

Abhilfe schaffen ARM-M0-basierte Prozessoren, die einen Systemtakt von 48MHz und einen schnelleren A/D-Wandler mitbringen. Für mein Projekt habe ich daher das Feather M0+ Board von Adafruit verwendet. Es verfügt über einen SAMD21-Prozessor, der auf einem ARM M0+ Kern aufbaut. Sein 10-bit-ADC kann mit einer maximalen Abtastrate von 500kps betrieben werden. Für die Aufzeichnung unserer Audiosignale sind 10 Bit Auflösung mehr als genug.

Vermutlich ließen sich auch die neueren Arduino MKR Boards nutzen, die ebenfalls einen SAMD21 verwenden. Das habe ich aber noch nicht ausprobiert. Adafruit bietet schließlich noch eine bequeme FFT-Bibliothek an, die die Transformation der Abtastwerte auf ein Frequenzspektrum auf eine Zeile Code reduziert.



Testsignale auf dem BitScope-Oszilloskop

Das Mikrofon

Da wir uns hier für die Frequenzen interessieren, die Menschen nicht mehr hören, kommt man mit den üblichen Mikrofonen für Musik und Sprache nicht weiter. Inzwischen hat der Trend zur Miniaturisierung in Smartphones aber MEMS-Mikrofone hervorgebracht, deren Frequenzbereich bis nahe an 100kHz reicht. MEMS steht für „Micro Electrical-Mechanical Systems“, die mit ähnlichen Lithografieverfahren gefertigt werden wie Halbleiter-ICs.

Diese Bauteile sind allerdings so klein, dass sie für Hobbyelektroniker nicht zu verarbeiten sind. Zum Glück gibt es Breakoutboards, die Anschlüsse im 2,54-mm-Raster haben. Ich habe das MEMS1 Modul der Firma ELV verwendet, mit dem Mikrofon SPU0410LR5H-QB von Knowles.

Beim diesem Mikrofon ist der Frequenzgang zwar nicht konstant über den ganzen

Bereich, aber es sollte ausreichen, um die Signale zu detektieren. Wer möchte, kann die frequenzabhängige Mikrofonempfindlichkeit nach der FFT herausrechnen.

Experimente

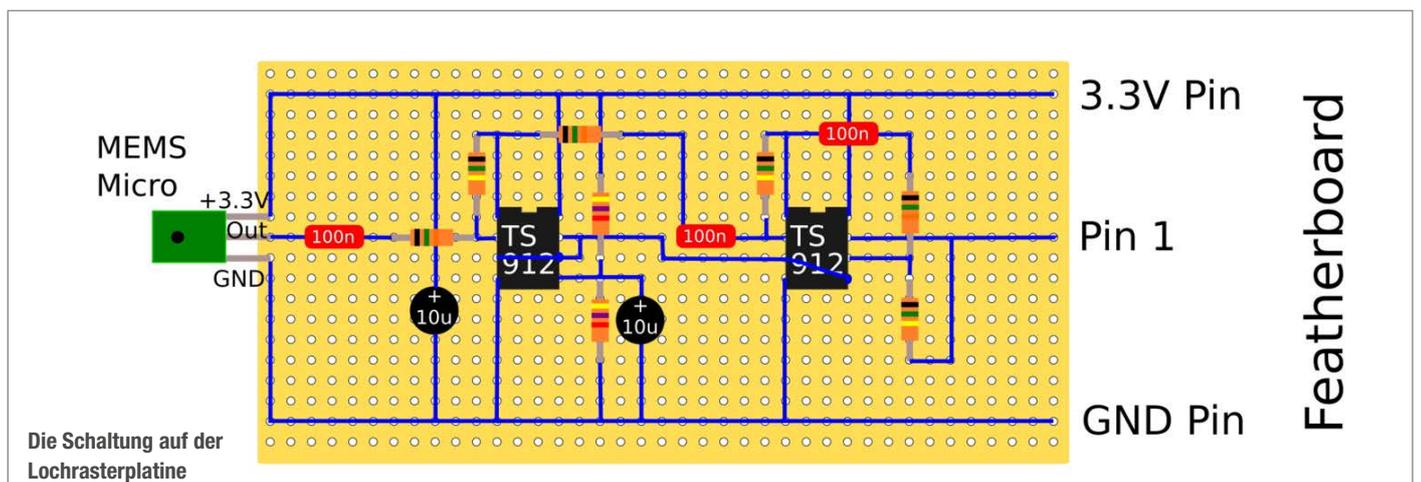
Um mir einen Eindruck von der Mikrofonempfindlichkeit zu verschaffen, habe ich das MEMS1 direkt an ein BitScope Micro Oszilloskop angeschlossen, dessen Software auf dem PC sowohl die Signalform im Zeitbereich als auch das Spektrum im Frequenzbereich anzeigen kann.

Mit einem Piezo-Hochtöner lassen sich Testsignale bis in den hohen Ultraschallbereich erzeugen. Dabei muss man davon ausgehen, dass auch der Hochtöner keinen konstanten

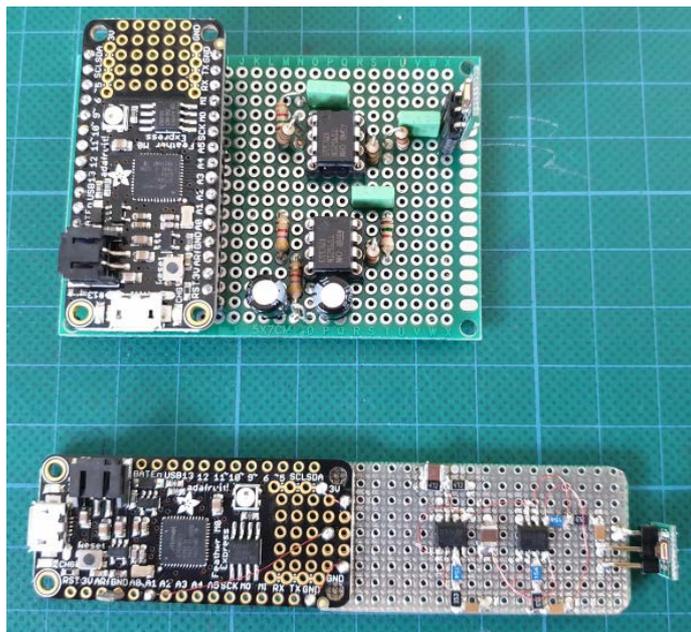
Frequenzverlauf hat. Man erhält also nur einen qualitativen Eindruck. Mir hat es trotzdem geholfen abzuschätzen, wie stark ich das Signal hinter dem Mikrofon verstärken muss, damit der Eingangsbereich des A/D-Wandlers von 3 Volt gut ausgenutzt wird. Ein Verstärkungsfaktor von 1000 schien sinnvoll.

Verstärkung

Mit Operationsverstärkern lässt sich die Verstärkerschaltung auf wenig Platinenplatz unterbringen. Ich habe mich für den Operationsverstärker TS912 entschieden – einen Rail-to-Rail-Typ, der bereits mit weniger als 3 Volt Betriebsspannung zurecht kommt und mit Signalen von 0 Volt bis zur Betriebsspannung umgehen kann. Da das MEMS-Mikrofon



Die Schaltung auf der Lochrasterplatine



Die große Scanner-Platine (oben) und die mobile Version



Oben im Gehäuse ist ein kleines Loch.

eine maximale Betriebsspannung von 3,3 Volt verträgt, kann ich die gesamte Schaltung mit 3,3 Volt betreiben.

Operationsverstärker haben eine Verstärkung von 1.000.000 oder mehr – allerdings nur bei sehr niedrigen Frequenzen. Sollen höherfrequente Signale verstärkt werden, ist das Verstärkungs-Bandbreite-Produkt (in Datenblättern: GBP – Gain/Bandwidth Product) der limitierende Faktor. Beim TS912 beträgt es 800kHz. Das bedeutet, dass er bei einem Signal mit einer Frequenz von 80kHz noch maximal um einen Faktor 10 verstärken kann.

Ich brauche also drei Stufen mit 10-facher Verstärkung, um den gewünschten Verstärkungsfaktor von 1000 bis 80kHz zu erreichen. Im DIP-8-Gehäuse des TS912 sind zwei dieser Verstärker integriert, so dass man mit zwei ICs auskommt. Der vierte Verstärker sorgt dafür, dass der Arbeitspunkt der Schaltung etwa bei 1,5 Volt liegt, also in der Mitte zwischen 0 Volt und der Betriebsspannung. So kann das Signal eine maximale Amplitude von 1,5 Volt annehmen, ohne dass es oben oder unten „anstößt“.

Die Schaltung baut man am besten auf einer Lochrasterplatine auf. Ich verwende dünnen Fädeldraht für die Verbindungen. Dieser Draht hat eine Lackisolation, die beim Lötens schmilzt. So dürfen sich die Drähte überkreuzen, da sie isoliert sind, und man muss nicht jeden Draht aufwendig abisolieren – das macht der LötKolben.

Die Platine hat nur drei Verbindungen mit dem Feather-Board: 3.3V, GND und die Verbindung vom Ausgang des dritten Operationsverstärkers zum Analogpin A1 des Mikrocontrollerboards. Für die Außenmontage habe ich noch eine kleinere Version gebaut, die in ein 32-mm-Isolationsrohr passt. Hierzu habe ich die SMD-Versionen der Operationsverstärker und passiven Bauteile verwendet. Diese Version erfordert etwas mehr Löterfahrung.

Für das Isolationsrohr habe ich zwei Endkappen auf dem 3D-Drucker gefertigt, so dass das Mikrofon durch ein kleines Loch mit der Außenwelt verbunden ist und ein halbwegs spritzwasserfester Auslass für das USB Kabel entsteht. Die Druckdateien stehen als Download zur Verfügung (siehe Link in der Kurzinfo).

Arduino-Software

Das Feather M0+ Board kann wie viele andere Mikrocontroller mit der kostenlosen Programmierumgebung von Arduino programmiert werden. Die IDE kennt das Board leider nicht von Haus aus. Es lässt sich jedoch mit wenigen Schritten nachinstallieren. Zunächst fügt man im Menü *Datei / Voreinstellungen* eine alternative *Boardverwalter-URL* ein: https://adafruit.github.io/arduino-board-index/package_adafruit_index.json

Im Menü *Werkzeuge – Board* sollten nun verschiedene Adafruit-Boards zur Auswahl bereitstehen. Hier fällt die Wahl auf das Feather M0+. Wenn man nun das Board per USB-Kabel mit dem PC verbindet, kann es für Windows-Benutzer erforderlich sein, einen spezifischen Treiber zu laden. Der sollte allerdings automatisch gefunden werden. Zum Start ist es eine gute Idee, den Sketch *Blink* zu laden (zu finden unter *Datei / Beispiele / 01.Basics*) und auf das Board zu übertragen, um zu prüfen, ob die neu installierte Infrastruktur funktioniert.

Für den Fledermausscanner nutze ich die Adafruit FFT-Bibliothek. Diese kann man über das Menü *Werkzeuge / Bibliotheken verwalten...* installieren, wenn man dort in der Suche „FFT“ eingibt. Weitere Infos zur Bibliothek und Installation finden Sie auch über den Link in der Kurzinfo.

Wer schon einmal den A/D-Wandler des Arduino benutzt hat, kennt den Befehl `analogRead()`. Leider lässt sich dieser hier nicht einsetzen, denn die maximale Geschwindigkeit von 500ksp/s wird nur erreicht, wenn der A/D-Wandler asynchron zum Programm läuft



Gehäuse-Variante für den dauerhaften Außeneinsatz

und die Ergebnisse direkt in den Speicher des Mikrocontrollers schreibt (DMA – Direct Memory Access). Wie das geht, ist in einer Application Note von Atmel beschrieben. Auf GitHub gibt es dazu einen Arduino-Sketch, den ich direkt in mein Programm eingebaut habe (siehe Link in der Kurzinfor).

Nach dem Herunterladen und Ausführen des Programms sendet das Board circa alle 100 Millisekunden eine komplette Liste aller berechneten Frequenzen (1024-Punkte-FFT) mit 115.200 bit/s über das USB-Interface. Der serielle Monitor der Arduino-IDE zeigt die Werte an.

Schickt man mit dem seriellen Monitor eine „0“ zum Board, schaltet es in den „raw-Modus“, in dem die abgetasteten Werte gesendet werden. Das hat mir beim Debugging geholfen. Das Senden einer „2“ bewirkt, dass das Board ab dann nur noch die Frequenzen über die USB-Schnittstelle sendet, bei denen ein bestimmter Schwellwert überschritten wird. „1“ schaltet zurück auf den Modus, bei dem das komplette Spektrum übertragen wird. In diesem Modus befindet sich der Controller auch nach jedem Reset oder Neustart.

Der serielle Monitor muss so konfiguriert sein, dass beim Senden der Befehle kein Return-Zeichen mitgesendet wird: Im Pull-Down-Menü des Monitors kann man dafür *no line ending* auswählen, wobei nur genau das Zeichen gesendet wird, das auf der Tastatur gedrückt und mit Return bestätigt wurde. Das Return selber wird nicht gesendet.

Grafische Darstellung

Um eine grafische Darstellung des Spektrums auf dem PC anzuzeigen, habe ich einen einfachen Sketch in Processing geschrieben. Mit dieser Programmierumgebung, die der Arduino-IDE sehr ähnelt, erstellt man mit wenig Aufwand einfache grafische Programme für den PC. Die IDE gibt es kostenlos – siehe Link in der Kurzinfor. Nach der Installation kann man den ebenfalls dort verlinkten Display-Tool-Sketch laden und ausführen. Lediglich die Bezeichnung der seriellen Schnittstelle muss angepasst werden. Während sie bei Linux `/dev/ttyACM0` oder `/dev/ttyACM1` heißt, lautet die Bezeichnung bei Windows `COMn`, wobei `n` eine Zahl zwischen 1 und 9 sein kann.

Das Programm schreibt einfach alle empfangenen Spektren übereinander, bis es neu gestartet wird. Wenn ein Schwellwert überschritten wird, wird zusätzlich noch die Frequenz mit ausgegeben. In einer späteren Version werde ich noch eine genauere Achsenkalibrierung hinzufügen.

Ausprobieren

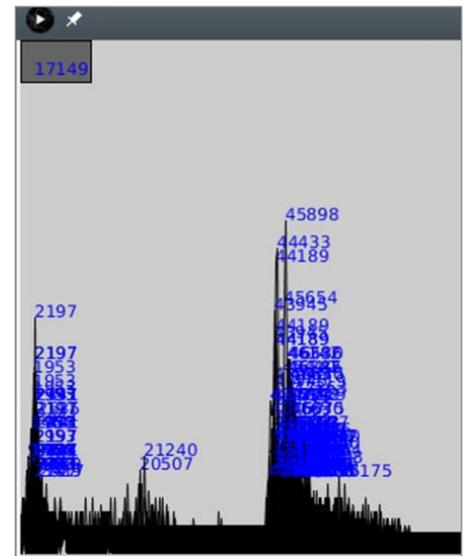
Nun hat natürlich nicht jeder einen Frequenzgenerator mit Piezo-Hochtöner zur Hand, wenn die Schaltung fertig ist, und es ist auch

unwahrscheinlich, dass sich sofort eine Fledermaus als Testkandidatin einfindet. Wer die Schaltung im Winter baut, muss sich bis zum Frühjahr gedulden, wenn die Fledermäuse aus dem Winterschlaf erwachen. Es sollten sich aber auch ohne Hilfsmittel sofort Gespräche oder Pfiffe im unteren Frequenzbereich bis 10kHz im Spektrum zeigen. Wischt man mit der Hand über Papier in der Nähe des Mikrofons kann man Frequenzen von 22kHz und mehr erzeugen. Wenn sie sich als Spitzen im Spektrum zeigen, funktioniert die Schaltung wie vorgesehen.

Fernhören

Wer nicht permanent mit einem Laptop bewaffnet im Dunkeln auf Fledermäuse lauschen will, kann die Schaltung auch stationär montieren und die Spektren im Web anzeigen. Dazu habe ich den Sensor wasserdicht verpackt, in dem ich das Rohr in ein 50mm HT-Rohr eingeschoben habe. Das HT-Rohr habe ich senkrecht montiert, mit einem wasserdichten Rohrstopfen am oberen Ende. Das untere Ende bleibt offen, so dass der Schall ungehindert am Sensor ankommt. Dort habe ich auch das USB-Kabel herausgeführt, statt ein Loch in das Rohr zu bohren, durch das Regen eindringen könnte.

Den fertig montierten Fledermaus-Scanner habe ich per USB an einen Raspberry Pi angeschlossen, auf dem der verbreitete MQTT-Broker Mosquitto, Node-Red.js mit installiertem Dashboard und ein Apache-Webserver laufen. Ein kurzes Python-Skript (siehe Link in Kurzinfor) liest die seriellen Daten vom Feather-Board, und schickt sie zum MQTT-Broker. Der Node-RED-Flow abonniert diese Nachrichten und zeigt sie grafisch auf einem Web-Dashboard an. Damit das funktioniert, muss der Raspberry Pi natürlich in Reichweite eines WLANs sein.



Die Signale bei ca. 46kHz entsprechen den Ortungssignalen von Zwergfledermäusen. Der kleinere Frequenzanteil bei ca. 20kHz deutet auf Soziallaute hin, also Kommunikation zwischen den Fledermäusen.

Eine genaue Beschreibung der MQTT/Node-Red-Implementierung auf dem Raspberry Pi würde den sinnvollen Umfang dieses Artikels überschreiten. Die Installation der erforderlichen Tools ist zum Beispiel im Make Node-RED Special und dem MQTT-Krabbelroboter (Make 7/19, S. 112) beschrieben, die ich in der Kurzinfor aufgeführt habe. Dort finden Sie auch den Link zur Webseite, auf der unser Dashboard live anzusehen ist. Es zeigt die Daten des Scanners im C-Turm am Stammheimer Feld in Calw, in dem im Sommer immer Fledermäuse in speziellen Höhlen nisten. Wer sich den Aufwand des Nachbaus sparen will, kann unseren Fledermäusen in der warmen Jahreszeit wieder mit dem Einsetzen der Dämmung im Web auflauern. —hch



Im Node-RED-Dashboard sind die empfangenen Frequenzen zu sehen.