

Schallsensor

Benötigtes Material

- Arduino Uno
- USB-Kabel für den Arduino
- MAKERFACTORY MF-6402171 Mikrofon-Schallsensor
- 3x MW-Kabel
- Stimmgabel (optional)

Benötigte Software

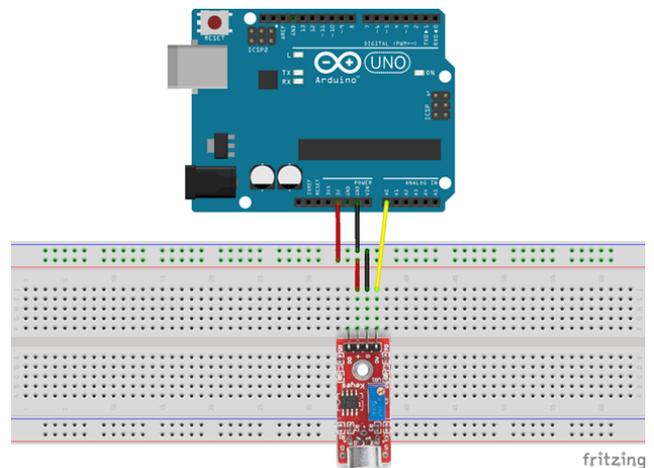
- arduinoFFT von Enrique Condes (Version 1.5.6)

Beschreibung

Im Folgenden messen wir mithilfe eines Schallsensor den Schalldruckpegel für eine gewisse Zeit und berechnen mithilfe der sogenannten diskreten Fourier-Transformation daraus die vorhandenen Frequenzen.

Verkabelung

MF-6402171	Kabel	Arduino
A0	Gelb	A0
G	Schwarz	GND
+	Rot	5V
D0		



Schaltplan

Code

```

*/
#include "arduinoFFT.h"

const int SAMPLES = 128; // SAMPLES-pt FFT. Muss eine Zweierpotenz sein.
                        // Maximal 128 für Arduino Uno.
const int SAMPLING_FREQUENCY = 2048; // Messwerte je Sekunde. Muss mindestens
                                     // doppelt so hoch wie die höchste
                                     // erwartete Frequenz sein (Stichwort
                                     // Nyquist).

arduinoFFT FFT = arduinoFFT();

unsigned int samplingPeriod;
unsigned long microSeconds;

double vReal[SAMPLES];
double vImag[SAMPLES];

```

```

void setup() {
  Serial.begin(9600); // Baudrate für den seriellen Monitor
  samplingPeriod = round(1000000 * (1.0 / SAMPLING_FREQUENCY)); // Periode in Mikrosekunden
  delay(1000);

  // Lese SAMPLES viele Messwerte ein
  for (int i = 0; i < SAMPLES; i++) {
    microseconds = micros();

    vReal[i] = analogRead(0); // Lies den Wert des Analog-Pins 0 (A0,
                             // quantisiere ihn und speicher ihn als Realteil.
    vImag[i] = 0;           // Der Imaginärteil ist immer 0.

    // Warte, bis eine Abtastperiode vergangen ist, um sicherzustellen, dass die
    // Werte in gleichmäßigen Abständen gemessen werden.
    while (micros() < (microseconds + samplingPeriod)) {
      // Tu nichts.
    }
  }

  // Ziehe den Durchschnittswert ab, um die Frequenz 0 zu entfernen.
  // Dies ist optional und kann auch auskommentiert werden.
  double vRealAvg = 0;
  for (int i = 0; i < SAMPLES; i++) {
    vRealAvg = vRealAvg + vReal[i];
  }
  vRealAvg /= SAMPLES;
  for (int i = 0; i < SAMPLES; i++) {
    vReal[i] = vReal[i] - vRealAvg;
  }

  /*
  Führe die schnelle Fourier-Transformation (FFT) auf den samples aus.
  Hierbei werden die alten Messwerte überschrieben.
  Möchte man diese behalten, sollte man aus Speicherplatzgründen nicht den
  Arduino Uno verwenden und die Werte vor der FFT in andere Vektoren
  kopieren.
  */
  FFT.Windowing(vReal, SAMPLES, FFT_WIN_TYP_HAMMING, FFT_FORWARD);
  FFT.Compute(vReal, vImag, SAMPLES, FFT_FORWARD);

  // Schreibe die Werte auf den seriellen Monitor, um diese in eine csv-Datei
  // kopieren zu können.
  Serial.print("Frequenz;Realteil;Imaginaerteil;Betrag");
  for (int i = 0; i < SAMPLES; i++) {
    Serial.println("");
    Serial.print(1.0 * SAMPLING_FREQUENCY * i / (SAMPLES - 1.0)); // Frequenz
    Serial.print(";");
    Serial.print(vReal[i]); // Realteil
    Serial.print(";");
    Serial.print(vImag[i]); // Imaginaerteil
    Serial.print(";");
    Serial.print(sqrt(sq(vReal[i]) + sq(vImag[i]))); // Absolutbetrag der
                                                    // komplexen Zahl Realteil
                                                    // + i*Imaginaerteil
  }
}

void loop() {

```

```
// Tue nichts im loop. Zum erneuten Ausführen muss der Arduino neugestartet  
// werden (roter Knopf).  
}
```

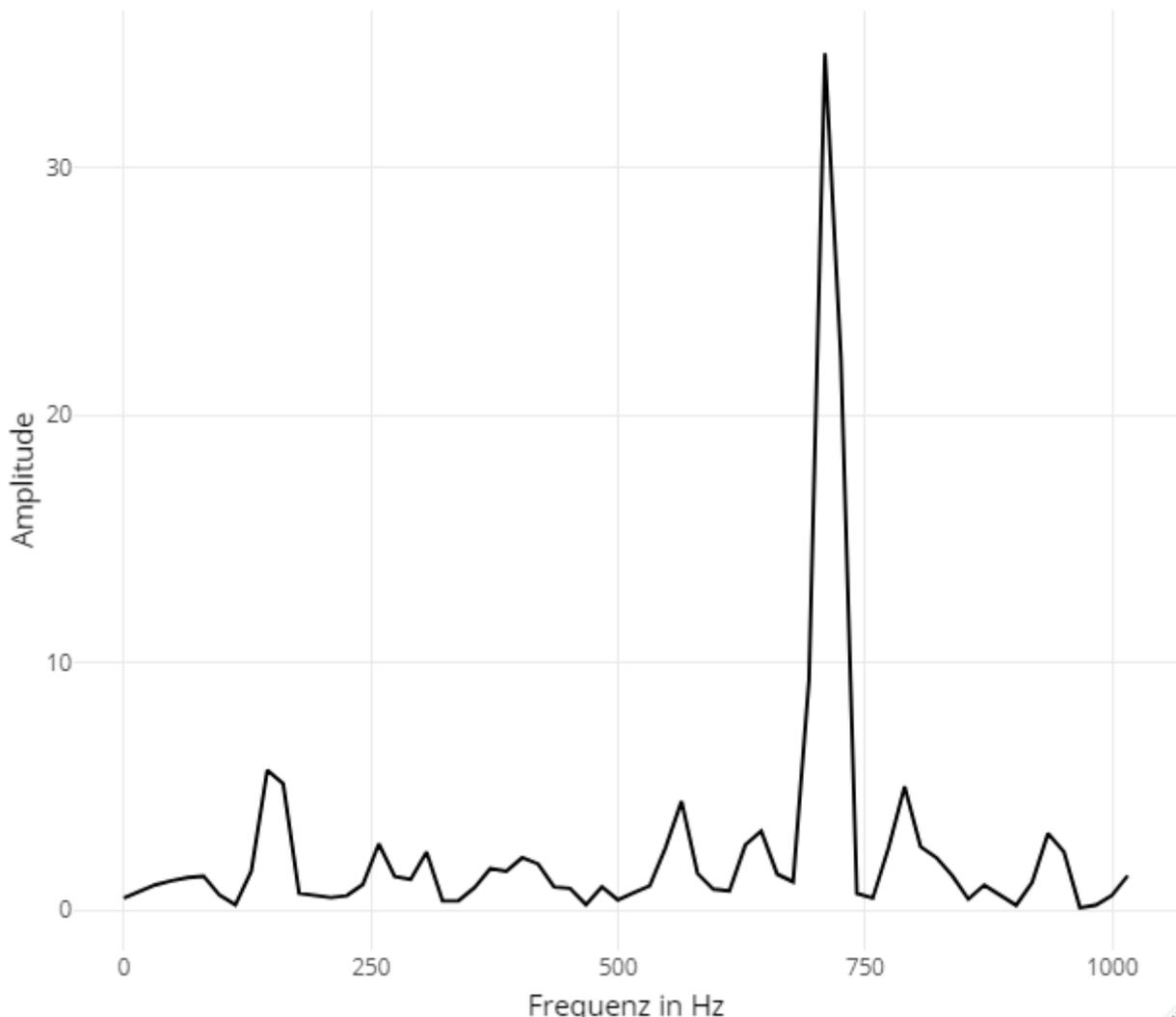
Ablauf

Nachdem der Arduino am Computer und der Schallsensor, wie oben beschrieben, am Arduino angeschlossen ist, öffne den seriellen Monitor und stelle die im Code angegebene Baudrate ein. Es sollten bereits Werte auf dem seriellen Monitor zu sehen sein. Lösche die bisherige Ausgabe. Starte dann den Arduino durch Betätigen des roten Knopfes neu und starte dein Signal, bspw. mithilfe einer Stimmgabel oder von Kopfhörern. Im Setup ist ein delay von einer Sekunde eingebaut, damit man Zeit hat, das Signal zu starten.

! Hinweis!

Beachte, dass nur für einen kurzen Zeitraum das Signal aufgenommen wird, da der Speicher des Arduinos sehr schnell voll ist. Bei einer Abtastfrequenz von 2048 Werten je Sekunde und einer Dauer von 128 Abtastungen dauert das Messen nur 62,5 Millisekunden.

Die neuen Werte, die während des Abspielens des Tons aufgenommen wurden, können nun in eine csv-Datei kopiert und anschließend mit Software wie Excel oder R visualisiert werden.



Frequenzspektrum eines 700Hz Signals

Grundsätzlich ist auch eine Visualisierung mithilfe des seriellen Plotters der Arduino IDE möglich. Allerdings ist es in diesem nicht ohne Weiteres möglich, die x-Achse anzupassen, sodass eine Interpretation der Werte

schwierig ist.

(Diskrete) Fourier-Transformation

Hinweis!

Die folgende Erklärung kann verwirrend sein, falls du im Unterricht noch keine komplexen Zahlen hattest. Für den Aufbau und den Code ist es aber nicht wichtig, die Mathematik hinter der Fourier-Transformation verstanden zu haben.

Die diskrete (schnelle) Fourier-Transformation („DFT“ für Discrete Fourier Transform bzw. „FFT“ für Fast Fourier Transform) ist ein mathematisches Werkzeug, welches ein diskretes, komplexwertiges Signal in seine Frequenzanteile aufspaltet. Intuitiv stellt die DFT das Signal durch eine Summe von Sinus- und Kosinusschwingungen variierender Frequenzen und Amplituden dar. Insbesondere hat ein reines Sinussignal genau eine Frequenz, welche nicht die Amplitude 0 hat. Der Wert dieser DFT ist wieder ein diskretes, komplexwertiges Signal. Mathematisch gesehen wird ein Vektor komplexer Zahlen auf einen weiteren Vektor komplexer Zahlen derselben Länge abgebildet. Da man jede reelle Zahl als komplexe Zahl auffassen kann, die einen Imaginärteil von 0 besitzt, ist die DFT auch für reellwertige Signale definiert. In der Praxis interessiert man sich auch meist nur für reellwertige Signale. Dass man jedoch trotzdem komplexe Zahlen verwendet, liegt daran, dass mit diesen die Darstellung relativ kompakt ist. Etwa steht in der Polarkoordinatendarstellung der Frequenzkoeffizienten der Radius für die Amplitude und der Winkel für die Phasenverschiebung des Signals. Solch eine Interpretation ist mit reellen Zahlen nicht möglich. Die DFT ist invertierbar. Das bedeutet, dass man das ursprüngliche Signal verlustfrei aus dem Frequenzspektrum rekonstruieren kann.

Aufgaben

- ① **Werden die Frequenzen vom Messgerät direkt gemessen?**

Antwort: _____

- ② **Warum ist die Frequenz nicht eine einzige Zahl?**

Antwort: _____

- ③ **Wie könnte man das Frequenzspektrum sinnvoll durch eine einzige Zahl darstellen?**

Antwort: _____

- ④ **Wie sieht ein Signal aus, welches aus nur einer einzigen, reellwertigen Frequenz besteht?**

Antwort: _____

- ⑤ **Wie sieht ein Signal aus, welches aus nur einer einzigen, komplexwertigen Frequenz besteht?**

Antwort: _____

- ⑥ **Was macht die Fourier-Transformation?**

Antwort: _____

- ⑦ **Warum zeigt das Frequenzdiagramm bei einem reinen 700Hz Signal auch andere Frequenzen?**

Antwort: _____

- ⑧ Visualisiere die von dir gemessenen Beträge der Frequenzen mit einem Programm deiner Wahl (z.B. Excel)!

Antwort: _____

- ⑨ Visualisiere die von dir gemessenen Frequenzen inklusive Real- und Imaginärteil in einem 3D-Plot! Was fällt dir auf?

Antwort: _____

- ⑩ Wie könnte man den Aufbau verbessern/erweitern?

Antwort: _____

- ⑪ Wie könnte man solch ein Frequenzmessgerät einsetzen?

Antwort: _____

- ① Werden die Frequenzen vom Messgerät direkt gemessen?

Antwort: _____

- ② Warum ist die Frequenz nicht eine einzige Zahl?

Antwort: _____

- ③ Wie könnte man das Frequenzspektrum sinnvoll durch eine einzige Zahl darstellen?

Antwort: _____

- ④ Wie sieht ein Signal aus, welches aus nur einer einzigen, reellwertigen Frequenz besteht?

Antwort: _____

- ⑤ Wie sieht ein Signal aus, welches aus nur einer einzigen, komplexwertigen Frequenz besteht?

Antwort: _____

- ⑥ Was macht die Fourier-Transformation?

Antwort: _____

- ⑦ Warum zeigt das Frequenzdiagramm bei einem reinen 700Hz Signal auch andere Frequenzen?

Antwort: _____

- ⑧ Visualisiere die von dir gemessenen Beträge der Frequenzen mit einem Programm deiner Wahl (z.B. Excel)!

Antwort: _____

- ⑨ Visualisiere die von dir gemessenen Frequenzen inklusive Real- und Imaginärteil in einem 3D-Plot! Was fällt dir auf?

Antwort: _____

10 Wie könnte man den Aufbau verbessern/erweitern?

Antwort: _____

11 Wie könnte man solch ein Frequenzmessgerät einsetzen?

Antwort: _____

Quelle

Dieses Projekt, insbesondere der Code und das Schaltbild, basiert auf einem Projekt von Clyde Lettsome, siehe <https://clydelettsome.com/blog/2019/12/18/my-weekend-project-audio-frequency-detector-using-an-arduino/>.

Das Material und dessen Inhalte sind - sofern nicht anders angegeben - lizenziert unter der Creative Commons Lizenz CC BY-NC-SA 4.0 (für den vollständigen Lizenztext siehe <https://creativecommons.org/licenses/by-sa/4.0/legalcode>)

