

Thesen zur Verteidigung

- Durch Blindleistungskompensation kann der Stromverbrauch in einer Schaltung deutlich reduziert werden.
- Ein Parallelschwingkreis hat bei seiner Resonanzfrequenz den geringsten Stromfluss.
- Verglichen mit einer Sinusspannung führt das Betreiben eines Schwingkreises mit einer Rechteckspannung zu erhöhtem Stromfluss.
- Die Schallabgabe von Piezolausprechern wird verstärkt, indem man sie in ein geschlossenes Metallgehäuse klebt.
- Die Schallemission der Lautsprecher ist außerhalb eines Bienenstocks nicht mehr relevant.
- Mehrere kleine, parallel geschaltete Piezolausprecher erzeugen eine höhere Lautstärke als einzelne, größere Piezolausprecher.
- Eine Reihenschaltung von Lautsprechern führt zu einer Dämpfung der Lautstärke.
- Die schnellste Methode, um mit dem Mikrocontroller ATmega328 Spannungen zu messen, ist der "Free-Running-Mode" des ADC's.
- Ein Ultraschallgerät kann für unter 25 Euro pro Stück hergestellt werden.

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides Statt, dass ich die vorliegende Arbeit selbstständig und nur unter Zuhilfenahme der ausgewiesenen Hilfsmittel angefertigt habe.

Sämtliche Stellen der Arbeit, die im Wortlaut oder dem Sinn nach anderen gedruckten oder im Internet verfügbaren Werken entnommen sind, habe ich durch genaue Quellenangaben kenntlich gemacht.

Leipzig, 31.10.2018

Ort, Datum

Unterschrift Lena Voelker

Inhaltsverzeichnis

Thesen zur Verteidigung	i
Eidesstattliche Erklärung	ii
Abbildungsverzeichnis	v
Tabellenverzeichnis	vii
1 Einführung	1
1.1 Motivation	1
1.2 Grundlegendes	2
1.2.1 Die Varroamilbe	2
1.2.2 Bekannte Bekämpfungsmethoden	3
1.2.3 Behandlung mittels Ultraschall	3
1.3 Piezolausprecher und dynamische Lautsprecher	4
1.4 Schwingkreis und Blindleistungskompensation	5
1.5 Bestimmung der Resonanzfrequenz	7
1.6 Berechnung des realen LC-Schwingkreises	8
2 Lautstärkemessung	11
3 Der erste Prototyp	16
3.1 Funktionsweise	16
3.2 Fazit	18
4 Der zweite Prototyp	19
4.1 Funktionsweise	19
4.2 Platinenherstellung	20
4.3 Operationsverstärker LM386	22
4.4 Operationsverstärker LM324	24
4.5 Programmierung des ATmega328	27
4.6 Spannungsüberwachung	27
4.7 Externe Ansteuerung	30

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

Inhaltsverzeichnis

4.8	Programmcode für den Mikrocontroller	30
4.8.1	Spannungsmessung	31
4.8.2	Programmcode - Modell 1	35
4.8.3	Programmcode-Modell 2	37
4.8.4	Stufeneinstellung	39
5	Das Gehäuse	44
6	Vorbereitung einer Kleinserienfertigung	45
	Literaturverzeichnis	46
	Anhang	48
	Schaltplan Prototyp 1	48
	Programmcode Prototyp 1	49
	Programmcode 1 Prototyp 2	51
	Programmcode 2 Prototyp 2	54
	Gesamtschaltung Prototyp 2	61
	Programmablaufplan Programmcode 2	62
	Bills-of-Material-Liste	63

Abbildungsverzeichnis

1.1	Varroabefall	
	www.nzgeo.com/wp-content/uploads/1970/01/66_Varroa_02-600x535.jpg	2
1.2	Milbenfall bei einer Behandlung, laut bienenhirte.de	4
1.3	Dynamischer Lautsprecher und Piezolautsprecher	5
1.4	Zeigerdiagramm der Spannung vor und nach der Kompensation	6
1.5	Realer LC-Schwingkreis	8
1.6	Frequenzgang RLC-Parallelschwingkreis	9
2.1	Messaufbau der Lautstärkemessung	11
2.2	Sonogramm der Frequenzfindung	15
3.1	Platine des ersten Prototyps	16
3.2	Ausgänge der H-Brücke ohne Last	17
3.3	Ausgang der H-Brücke mit kapazitivem Lautsprecher	18
3.4	Ausgang der H-Brücke mit Parallelschwingkreis	18
4.1	Der zweite Prototyp	19
4.2	Entwickelte Platine	20
4.3	PCB-Design in Altium Designer	21
4.4	Bestückte Platine	22
4.5	Schaltung LM386 in LT-Spice	23
4.6	Verhältnis zwischen Ein- und Ausgangsspannung LM386	23
4.7	Schaltung des LM324	24
4.8	Spannungen am Tiefpassfilter	25
4.9	Amplituden- / und Phasengang RC-Tiefpass	25
4.10	Rauschglättung	26
4.11	Header für ISP und FTDI	27
4.12	Schaltung der Spannungsüberwachung	28
4.13	Schaltung der Laderegulung	29
4.14	Verbesserte Laderegulung	29
4.15	Schaltung zur externen Ansteuerung	30

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

Abbildungsverzeichnis

4.16	Arbeitsweise eines Analog-Digital-Umsetzers	
	Frank Ohnhäuser. [9]	32
4.17	Zeiteinteilung des Analog-Digital-Umsetzers	
	- datasheet.octopart.com/ATMEGA328-AU-Atmel-datasheet-99479556.pdf . .	33
4.18	Schaltung der Spannungsmessung	34
4.19	Schema der Spannungsmessung	35
4.20	Programmablaufplan Code 1	37
4.21	Schema der Messung Code 2	38
4.22	Schaltung der LED's zur Anzeige der Betriebsstufen	40
5.1	CAD-Modell des Gehäuses	44
6.1	Schaltung des LM324	61
6.2	Programmablaufplan des Setup-Codes	62
6.3	Programmablaufplan des Loop-Codes	62
6.4	BOM-Liste des Gerätes	63

Tabellenverzeichnis

2.1	Einzelmessung der Lautsprecher	12
2.2	Parallelschaltung zweier Lautsprecher	12
2.3	Parallelschaltung mehrerer Piezos mit \varnothing 20mm	13
2.4	Reihenschaltung zweier Lautsprecher	13
2.5	Lautsprecher im Gehäuse	14
2.6	Lautstärke außerhalb des Bienenstocks	14

1 Einführung

1.1 Motivation

Im Rahmen eines Forschungsprojektes des Forschungs- und Transferzentrums Leipzig e.V in Kooperation mit dem Institut für Elektronik und biomedizinische Informationstechnik der Hochschule für Technik, Wirtschaft und Kultur Leipzig wurde ein Prototyp eines Gerätes entwickelt, das zur Erforschung der Schädlingsbekämpfung in Bienenstöcken eingesetzt werden soll. Das Gerät soll eine variable Frequenz besitzen, möglichst günstig produziert werden, einen geringen Stromverbrauch haben und mehrere Stufen der Schallabgabe besitzen. Ziel ist es, ein Gerät zu entwickeln, das für Forschungszwecke geeignet ist.

Das Bienensterben schreitet weiter voran und bisher fehlt eine effektive, einfache und rückstandsfreie Behandlungsmethode bei einem Bienenmilbenbefall. Ein Aussterben der Biene hätte für die Umwelt und den Menschen fatale Folgen. Das Gerät soll für die Erforschung der Wirksamkeit einer Ultraschallbehandlung bei Varroamilbenbefall geeignet sein.

1.2 Grundlegendes

1.2.1 Die Varroamilbe

Die Varroamilbe ist unter dem wissenschaftlichen Namen *Varroa destructor*, die Zerstörermilbe, bekannt. In den 1970er Jahren kam sie durch asiatische Honigbienenimporte nach Europa und ist seither einer der bedeutendsten Ursachen des weltweiten Bienensterbens. Lange Winter in Europa, Pflanzenschutzmittel und Monokulturen auf europäischen Feldern begünstigen das Bienensterben. Mittlerweile stehen sechzig Prozent aller Bienenarten auf der roten Liste der bedrohten Tierarten. In den letzten 30 Jahren nahm die Zahl der Bienenvölker in Deutschland um vierzig Prozent ab.[2]

Für die Umwelt ist ein solches Bienensterben fatal. Ein Großteil der Pflanzen und Nutzpflanzen werden von Bienen bestäubt, somit sind sie für den Erhalt des Ökosystems notwendig. Ein Aussterben des Insekts hätte auch für den Menschen schlimme Folgen. Bienen nehmen einen großen Einfluss auf die Landwirtschaft. Die UNO (= United Nations Organization) sieht in der Milbe eine Bedrohung für die Nahrungsgrundlage des Menschen und führte 2018 den Weltbienentag ein, um auf das Problem aufmerksam zu machen.



Abbildung 1.1: Varroabefall

Bei der Varroamilbe handelt es sich um einen Parasiten der Familie Varroidae, der ausschließlich Honigbienen befällt und somit nur in Bienenstöcken zu finden ist. Ein Befall des Volkes - eine Varroose - ist eine Tierseuche und in Österreich anzeigepflichtig. Ein verseuchtes Bienenvolk verendet ohne eine Behandlung nach drei bis vier Jahren. In der Zwischenzeit ist die Leistungsfähigkeit der Volkes stark beeinträchtigt.

Der Parasit nimmt Nahrung durch das Saugen an ausgewachsenen Bienen auf und schwächt somit die Tiere. Zudem können Viren übertragen werden. Zur Fortpflanzung befällt der Parasit die Bienenbrut. Die heranwachsenden Bienen sind deutlich kleiner und schwächer als gesunde Tiere. [3]

1.2.2 Bekannte Bekämpfungsmethoden

Nach dem Bundesamt für Sicherheit im Gesundheitswesen (BASG) dürfen Bienen nur mit zugelassenen Stoffen (Verordnung (EU) Nr. 37/2010) behandelt werden (§ 4 (5) TAKG-Tierarzneimittelkontrollgesetz). Diese zugelassenen Präparate beinhalten Ameisensäure, Milchsäure, Oxalsäure oder Thymol. Diese Mittel führen zu einer schnellen Befallsreduktion, allerdings kann es zu Rückständen im Honig oder Schäden an den Bienenvölkern kommen. Zudem sind spezielle Vorkehrungen, wie beispielsweise Schutzhandschuhe und Schutzbrillen nötig. Die Mittel müssen zu bestimmten Zeiten eingesetzt werden und die Wirksamkeit ist witterungsabhängig.

Seit zwei Jahren forscht die Universität Hohenheim an einem geeigneten Bekämpfungsmittel. Anfang des Jahres 2018 wurde ein Erfolg veröffentlicht. Lithiumchlorid soll Milben rückstandslos in nur acht Stunden abtöten und dabei keinerlei Schäden an Bienen oder Honig verursachen. Es müssen jedoch noch einige Tests abgeschlossen werden, bevor das Mittel auf den Markt kommen kann.[7]

Neben den Behandlungen mit diesen Präparaten existieren weitere ökologische Behandlungsmethoden, die keine Rückstände im Honig hinterlassen können, wie zum Beispiel Fangwabenverfahren, eine Hitzebehandlung, die Behandlung mittels Pseudoskorpionen oder die gezielte Entnahme von Bienenbrut. Auch diese Methoden können nur zu bestimmten Zeiten eingesetzt werden und sind überwiegend zeitaufwendig.[1]

1.2.3 Behandlung mittels Ultraschall

Eine weitere mögliche Behandlungsmethode bei einem Varroamilbenbefall ist die Behandlung mittels Ultraschall. Bereits in den 90er Jahren existierte ein Gerät mit dem Namen "Der Schallomat", der einen Erfolg in der Schädlingsbekämpfung versprach. Diese Idee wurde von dem Imker Rainer Krüger neu aufgegriffen. Er entwickelte den Varroa-Killer-Sound. Ein Gerät, das laut Angaben des Erfinders mit 15000 Hz und einer Lautstärke von 90 dB arbeitet. Ein Gerät kostet 86 Euro und besitzt einen großen Bleiakku. Die Frequenz des Schalls oder die Beschallungsdauer können nicht variiert werden. Es sind bereits Ultraschallgeräte für andere Milbenarten, wie zum Beispiel die Staubmilbe, auf dem Markt.

Der Schall soll die Parasiten orientierungslos machen, in der Nahrungsaufnahme und der Fortpflanzung stören und schließlich zum Tod der Schädlinge führen.

Eine Behandlung mittels Ultraschall ist vollständig rückstandsfrei, umweltschonend und zeitsparend. Derzeit existiert noch keine wissenschaftliche Arbeit, die eine Wirksamkeit der Behandlung mittels Ultraschall bei Varroamilben oder Staubmilben belegt. Dementsprechend sind viele Imker skeptisch. Aus diesem Grund wurden vom Hersteller Testgeräte an verschiedene Imker ausgegeben.[4]

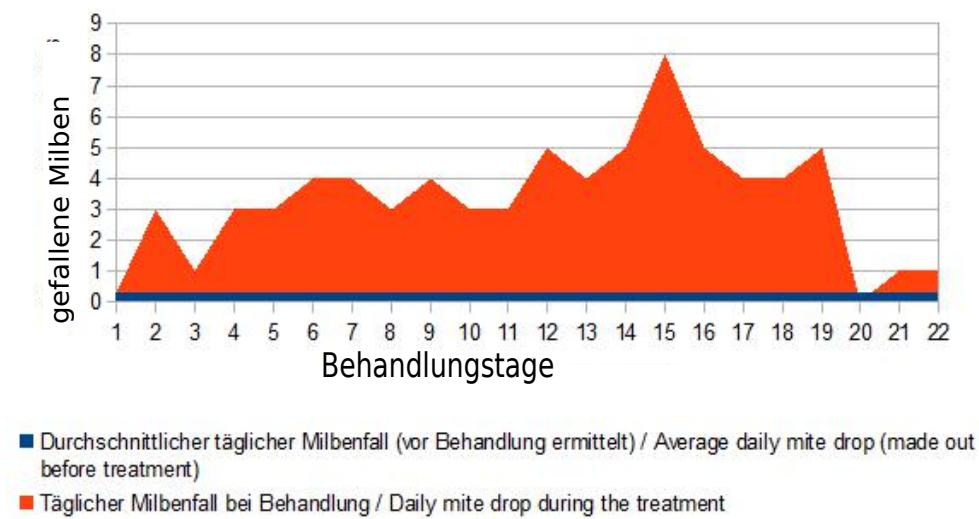


Abbildung 1.2: Milbenfall bei einer Behandlung, laut [1]

Laut eines Artikels¹ wurden bereits Erfolge mit dem Varroa-Killer-Sound gemessen. Der Erfolg einer Varroamilbenbehandlung lässt sich anhand des Milbenfalls bestimmen. Hierzu werden täglich am Boden liegende Milben und tote Bienen gezählt. Die Behandlung mit dem Varroa-Killer-Sound soll eine 4-20-fach höheren Milbenfall bewirken als ohne eine Behandlung. Zudem bestätigen veröffentlichte Kommentare auf der Internetseite die Wirksamkeit der Ultraschallbehandlung. [5]

Da es sich hierbei nur um Erfahrungsberichte und keine wissenschaftlichen Arbeiten handelt, bleibt die Wirksamkeit der Ultraschallbehandlung weiterhin unbestätigt.

Ziel soll es nun sein, ein kostengünstiges Gerät mit variabel einstellbaren Parametern zu Forschungszwecken zu entwickeln.

1.3 Piezolausprecher und dynamische Lautsprecher

Dynamische Lautsprecher setzen mithilfe eines Magneten und einer Schwingspule eine Membran in Bewegung. Durch die Induktivität besitzen sie bei hohen Frequenzen eine hohe Impedanz. Es wird viel Leistung benötigt.

Piezolausprecher arbeiten mit dem inversen Piezoeffekt. Der Piezoeffekt bzw. piezoelektrische Effekt oder direkter Piezoeffekt beschreibt das Entstehen einer elektrischen Spannung bei dem Verformen von Festkörpern.

Diese elektrische Spannung ist abhängig von der Kraft F der mechanischen Beanspruchung, der Kapazität C und der piezoelektrischen Ladungskonstante d . Sie lässt sich mit der Formel

¹<https://bienenhirte.de/blog/varroa-killer-sound/#lightbox/2/>

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

1 Einführung

$$U = \frac{d \cdot F}{C}$$

bestimmen. [6]

Vorteile von Piezolausprechern gegenüber dynamischen Lautsprechern sind

- kleine, flache Bauform
- Stückkosten
- Robustheit
- Leistung bei hohen Frequenzen
- geringe Impedanz bei hohen Frequenzen

Aus diesen Gründen wurde für das Ultraschallgerät Piezolausprecher verwendet.

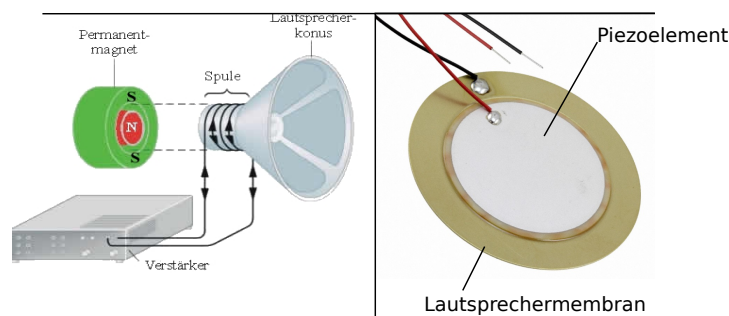


Abbildung 1.3: Dynamischer Lautsprecher und Piezolausprecher

1.4 Schwingkreis und Blindleistungskompensation

In einem Parallelschwingkreis tritt bei Resonanzfrequenz ein Impedanzmaximum auf. Nach dem ohmschen Gesetz ist der Strom dann am geringsten.

Betrieibt man eine Kapazität oder eine Induktivität mit einer Wechselspannung treten, aufgrund des ständigen Auf- / und Abbaus elektrostatischer und elektromagnetischer Felder, Blindströme auf. In Schaltungen sind diese meist unerwünscht, da die Energie nicht genutzt werden kann, den Verstärker und das Netz aber dennoch belastet. Dieser Energieaustausch soll bei Resonanz nur zwischen Induktivität und Kapazität erfolgen. Dafür muss folgende Resonanzbedingung gelten:

$$\omega L = \frac{1}{\omega C}$$

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

1 Einführung

mit $\omega = 2\pi f$

Der Blindwiderstand der Kapazität und Induktivität müssen gleichgroß sein. Bei diesem Arbeitspunkt ist die Phasenverschiebung zwischen Spannung und Strom $\Phi = 0^\circ$, da sich die Blindwiderstände aufheben und nur noch der Wirkwiderstand Einfluss auf die Schaltung hat.



Abbildung 1.4: Zeigerdiagramm der Spannung vor und nach der Kompensation

Diese Formel kann nun beliebig nach C, L oder f umgestellt werden, um benötigte Größen zu bestimmen. Diese Beziehungen gelten nur für (oberschwingungsfreie) Sinusspannungen. Eine Spule zur Blindstromkompensation lässt sich also mit der Formel

$$L = \frac{1}{\omega^2 C} = \frac{1}{(2\pi f)^2 \cdot C}$$

bestimmen. Wenn die Induktivität und die Kapazität bei einer anderen Frequenz betrieben werden, wird ein höherer Strom fließen als bei der vorher bestimmten Resonanzfrequenz. Soll die Frequenz des Modells also variabel sein, muss entweder mit erhöhtem Stromverbrauch gerechnet werden, oder die Kompensationsspule entsprechend der neuen Frequenz ausgetauscht werden.

In unserem Fall soll die Resonanzfrequenz zwischen 15 und 20 kHz liegen. Die gewählten Piezolausprecher haben laut Datenblatt eine Kapazität von $C = 10nF \pm 30\%$ bei einer Frequenz von 1000 Hz. In einer Parallelschaltung von Kapazitäten liegt an allen Kapazitäten die gleiche Spannung an. Die Gesamtkapazität ist gleich der Summe der einzelnen Kapazitäten.

$$C_{ges} = C_1 + C_2 + C_3 \dots$$

Nach dieser Rechnung müssten alle acht verwendeten Lautsprecher eine Kapazität von $C_{ges} = 80nF \pm 30\% = 80nF \pm 24nF$ besitzen. Die parallelen Piezolausprecher haben zusammen eine Kapazität von $C = 57,93nF \approx 60nF$. Dieser Wert wurde mit dem LCR-Meter ESCORT ELC-1331 bestimmt.

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

1 Einführung

Die Spule muss bei einer gewünschten Frequenz von 15 kHz also den Wert

$$L = \frac{1}{(2\pi \cdot 15000 \text{ Hz})^2 \cdot 0.00000006 \text{ F}} = 0.0018763 \text{ H} \approx 1.9 \text{ mH}$$

besitzen.

1.5 Bestimmung der Resonanzfrequenz

Im Resonanzfall geht der Strom I im Schwingkreis gegen 0. Der Parallelschwingkreis hat bei seiner Resonanzfrequenz ein Impedanzmaximum. Die kapazitiven Piezolausprecher haben eine Gesamtkapazität von ungefähr $C = 60 \text{ nF}$ und somit einen Blindwiderstand von

$$X_C = \frac{1}{\omega C}.$$

Mit $\omega = 2\pi f$ folgt

$$X_C = \frac{1}{2\pi f \cdot 60 \text{ nF}}$$

Für den Blindwiderstand der Spule gilt

$$X_L = \omega L = 2\pi f L$$

Die Blindwiderstände sind bei Resonanzfall gleich groß. Im Resonanzfall gilt

$$X_L = X_C = 2\pi f L = \frac{1}{2\pi f C}$$

Da die Kapazität des Piezolausprechers und die gewünschte Frequenz bekannt sind, konnte dank dieser Formel die Induktivität zur Blindleistungskompensation berechnet werden.

$$L = \frac{1}{(2\pi f)^2 C} = \frac{1}{(2\pi \cdot 15000 \text{ Hz})^2 \cdot 0.00000006 \text{ F}} = 1876.3 \mu\text{H} \approx 0.0019 \text{ H}$$

Die für den Prototyp gewählten Spulen besitzen eine Induktivität von $L = 1800 \mu\text{H}$.

Für die Resonanzfrequenz gilt dann

$$f_0 = \frac{1}{2\pi\sqrt{LC}} = \frac{1}{2\pi\sqrt{0.0018 \text{ H} \cdot 0.00000006 \text{ F}}} = 15314.69 \text{ Hz}$$

Bei Resonanzfrequenz gilt für $X_C = \frac{1}{2\pi \cdot 15314.69 \text{ Hz} \cdot 60 \text{ nF}} = 173.2 \Omega$

und für die Spule

$$X_L = 2\pi f L = 2\pi \cdot 15487 \cdot 120 \mu\text{H} = 173.2 \Omega$$

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

1 Einführung

Bei diesen Werten der Blindwiderstände existiert ein Impedanzmaximum im Parallelschwingkreis. Nur noch reine Wirkwiderstände wirken auf die Schaltung.

1.6 Berechnung des realen LC-Schwingkreises

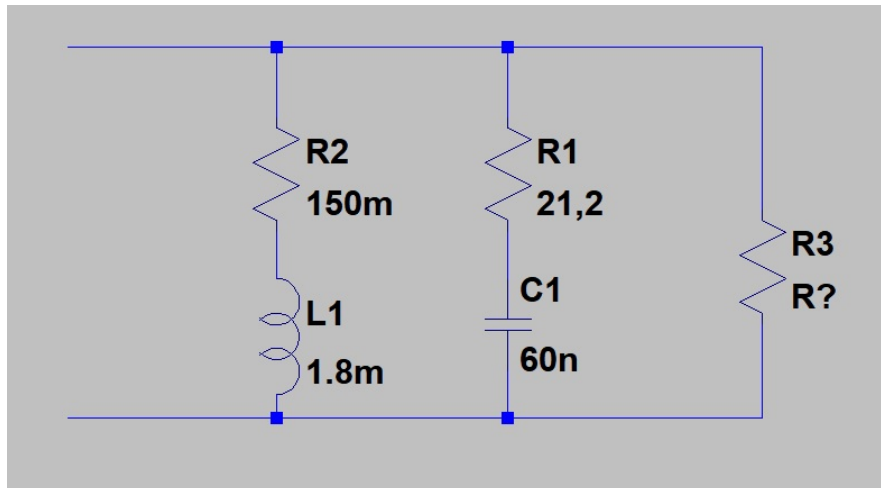


Abbildung 1.5: Realer LC-Schwingkreis

Die Spule besitzt einen ohmschen Reihenwiderstand von $R_L = 150m\Omega$ und somit eine Impedanz von $\underline{Z}_L = R_L + jX_L = 150m\Omega + j\omega 1800\mu H$. Die Impedanz bei Resonanzfrequenz ist dann $\underline{Z}_{LR} = 150m\Omega + j173.2\Omega$.

Betrachtet man eine Zylinderspule ohne magnetischen Kern (Luftspule) gilt für die Induktivität

$$L \approx \mu_0 \cdot \frac{N^2 \cdot A}{l}$$

Mit μ_0 - magnetische Feldkonstante, N - Windungszahl, A Fläche einer Wicklung und l - Länge der Spule.

Auf eine Multiplikation mit der magnetischen Feldkonstante μ_r kann in diesem Fall verzichtet werden, da Luftspulen keinen ferromagnetischen Kern besitzen und die Permeabilitätszahl von Luft $\mu_r = 1$ ist. Setzt man in diese Formel verschiedenste Werte ein stellt man fest, dass die für das Gerät benötigte Spule sehr groß oder sehr viele Windungen benötigt. Eine Luftspule dieser Dimension ist zudem verhältnismäßig teuer. Aus diesem Grund wird eine Spule mit einem ferromagnetischen Kern verwendet. Der Kern erhöht die Permeabilität und die magnetische Flussdichte. Die Spule benötigt weniger Windungen, um eine gewünschte Induktivität zu erreichen.

Die Piezolautsprecher haben eine Impedanz von $\underline{Z}_C = R_C + jX_C$. Der ohmsche Wirkwiderstand der Piezolautsprecher war vorerst nicht bekannt.

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

1 Einführung

Mithilfe des LCR-Meters HM8118 von der Firma Rohde & Schwarz konnte die Impedanz bestimmt werden. Auch dieses Messgerät zeigte eine Kapazität der Piezolausprecher von $C = 53.98 \text{ nF} \approx 60 \text{ nF}$ an. Bei einer Frequenz von $f = 15 \text{ kHz}$ wurde ein Reihenwiderstand von $R_S = 21.13 \Omega$ und eine Gesamtimpedanz von $|Z| = 197.65 \Omega$ gemessen. Die Gesamtimpedanz lässt sich wie folgt bestimmen.

$$|Z| = \sqrt{R^2 + X_C^2}.$$

Mit $X_C = \frac{1}{\omega C} = \frac{1}{2\pi \cdot 15000 \text{ Hz} \cdot 53.98 \text{ nF}} = 198.56 \Omega$ folgt

$$|Z| = \sqrt{(21.13 \Omega)^2 + (198.56 \Omega)^2} = 197.69 \Omega.$$

Zusätzlich ist anzunehmen, dass durch die mechanische Schwingung der Piezolausprecher ein weiterer Parallelwiderstand existiert. Dieser wurde mithilfe der Güte des Parallelschwingkreises ermittelt. Die Güte Q eines Parallelschwingkreises ist definiert als

$$Q = R\sqrt{\frac{C}{L}}.$$

Zudem ist bekannt, dass die Bandbreite durch $B = \frac{f_0}{Q}$ ausgedrückt werden kann.[13]

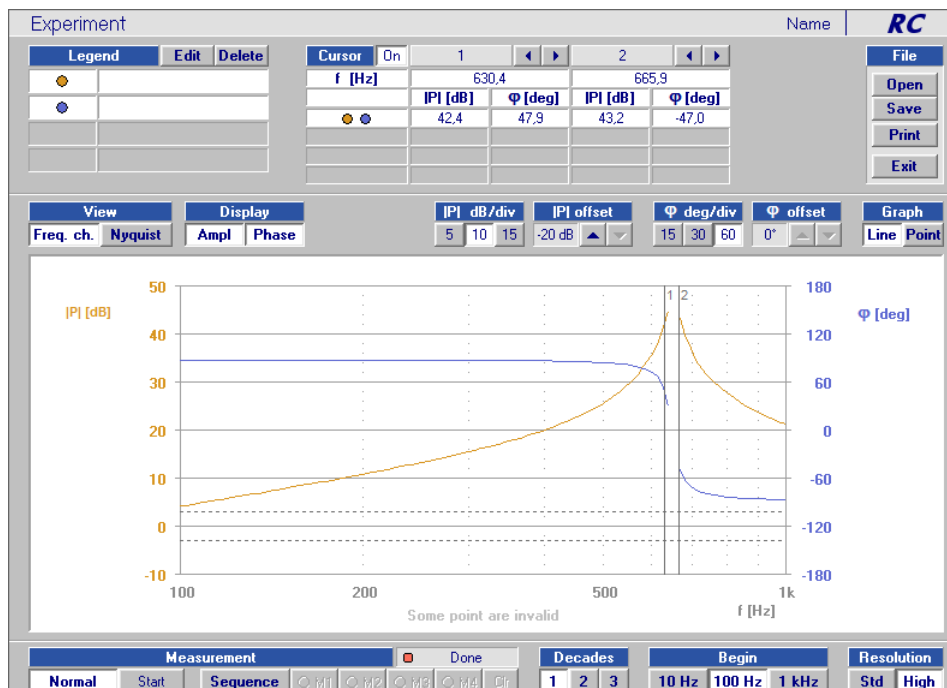


Abbildung 1.6: Frequenzgang des RLC-Parallelschwingkreises

Diese Formeln nach dem gesuchten Parallelwiderstand umgeformt ergibt

$$R = \frac{f_0}{B \cdot \sqrt{\frac{C}{L}}}.$$

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

1 Einführung

In dem Versuchsaufbau wurde eine Spule mit $L = 1H$ und die Piezolautsprecher mit $C = 60nF$ verwendet. Die Bandbreite betrug $B = 35.5Hz$ und $f_0 = 653.8Hz$. Die Formel ergab einen Wert von

$$R_M = \frac{653.8Hz}{35.5Hz \cdot \sqrt{\frac{60nF}{1H}}} = 75186.7\Omega.$$

Die Admittanz $\underline{Y} = \underline{Z}^{-1}$ des Schwingkreises wird

$$\underline{Y} = \frac{1}{R_M} + \frac{1}{R_C} + j\omega C + \frac{1}{R_L + j\omega L}$$

$$\underline{Y} = \frac{1}{R_M} + \frac{1}{R_C} + j\omega C + \frac{1}{R_L + j\omega L} \cdot \frac{R_L - j\omega L}{R_L - j\omega L}$$

$$\underline{Y} = \frac{1}{R_M} + \frac{1}{R_C} + j\omega C + \frac{R_L - j\omega L}{R_L^2 + (\omega L)^2}$$

$$\underline{Y} = \frac{1}{R_M} + \frac{1}{R_C} + j\omega C + \frac{R_L}{R_L^2 + (\omega L)^2} - j \frac{\omega L}{R_L^2 + (\omega L)^2}$$

$$\underline{Y} = \frac{1}{R_M} + \frac{1}{R_C} + \frac{R_L}{R_L^2 + (\omega L)^2} + j \left(-\frac{\omega L}{R_L^2 + (\omega L)^2} + \omega C \right)$$

Da der Strom bei Resonanz im Schwingkreis gegen 0 geht, soll der Mikrocontroller die Spannung über einen Shunt messen. Der Mikrocontroller soll mit dem DDS-Generator Frequenzen zwischen 15-20 kHz erzeugen und die gemessenen Spannungen vergleichen. Nach dem ohmschen Gesetz gilt $I = \frac{U}{R} = \frac{U}{|Z|} = U \cdot |Y|$.

Mit der Resonanzkreisfrequenz $\omega = \omega_r = \frac{1}{\sqrt{LC}}$ folgt für die Resonanzfrequenz $f_r = \frac{\omega_r}{2\pi}$.

Aus der Resonanzbedingung für den Parallelschwingkreis folgt

$$\omega C - \frac{\omega L}{R_L^2 + (\omega L)^2} = 0$$

Wird diese Formel nun nach ω umgeformt, ergibt sich für die Resonanzfrequenz

$$\omega = \sqrt{\frac{1}{LC} - \frac{R_L^2}{L^2}}$$

$$f_r = \frac{1}{2\pi} \cdot \omega = \frac{1}{2\pi} \cdot \sqrt{\frac{1}{LC} - \frac{R_L^2}{L^2}}$$

$$f_r = \frac{1}{2\pi} \sqrt{\frac{1}{1800\mu H \cdot 60nF} - \frac{(0.15\Omega)^2}{(1800\mu H)^2}} = 15314.68Hz$$

2 Lautstärkemessung

Zur Messung der Lautstärke des Ultraschallgenerators konnte das Gerät "Apollo" von der Firma "Sinus Messtechnik GmbH" aus Leipzig geliehen werden. Das Gerät ist ein professioneller Akustikanalysator mit einem 24-bit AD-Wandler. Zudem besitzt es einen USB-Anschluss und lässt sich mit einem Computer und der zugehörigen Software "SAMURAI 2.0" bedienen.

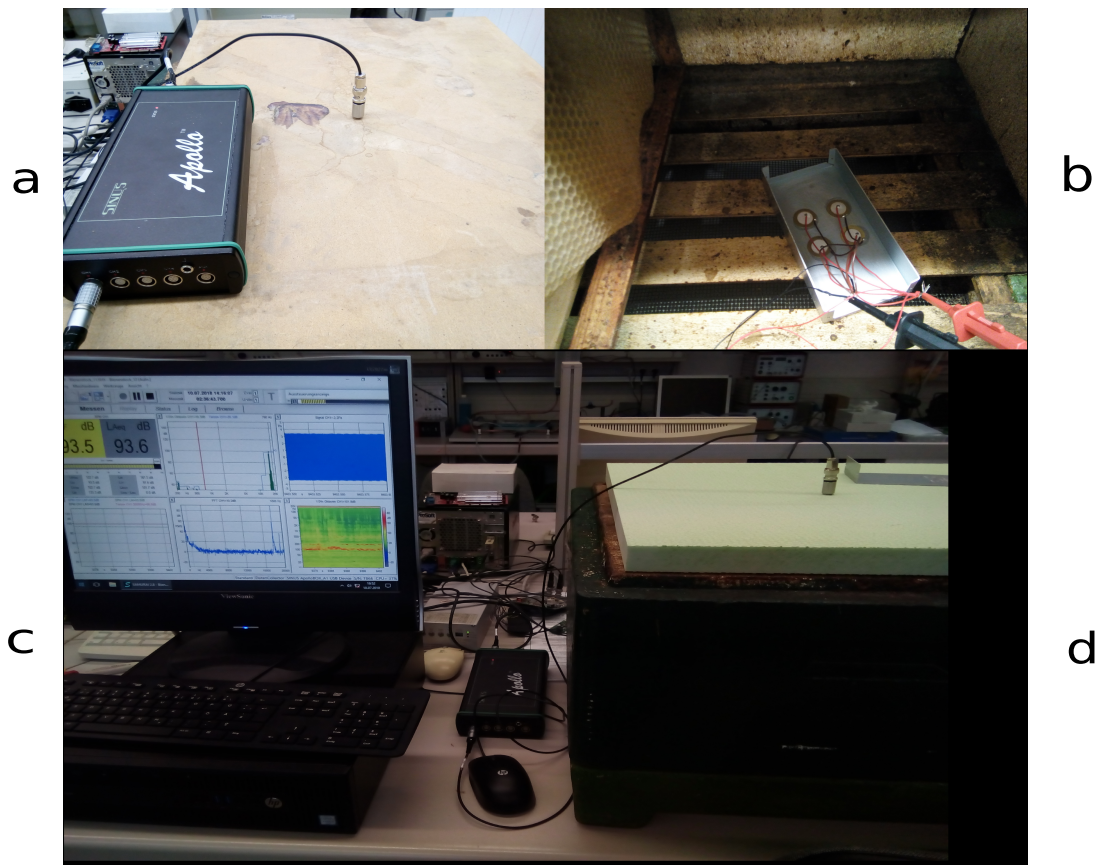


Abbildung 2.1: Messaufbau der Lautstärkemessung

Es sollte überprüft werden, welche Spannungen am Piezolautsprecher und an dem Lautsprecher SPS 41 von Sonitron die beste Leistung erzielt. Zudem sollte überprüft werden, ob eine Verstärkung der Schallabgabe erfolgt, wenn man den Piezolautsprecher direkt am Metallgehäuse

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

2 Lautstärkemessung

befestigt und wie das Gerät die Umwelt beeinflusst. Für die Messungen stand ein handelsüblicher Bienenstock zur Verfügung. Die Lautsprecher wurden mittig unter die Bienenwabenkästen gelegt. Das Mikrophon wurde ebenfalls mittig am Deckel des Bienenkastens angebracht. Zunächst wurden die verschiedenen Lautsprecher einzeln getestet. Die Lautsprecher wurden mit einer sinusförmigen Spannung mit $2.5 V_{pp}$ und Frequenzen zwischen 15 - 20 kHz betrieben.

Tabelle 2.1: Einzelmessung der Lautsprecher

f/kHz	dB	f/kHz	dB	f/kHz	dB
15	67,8	15	64,9	15	62,3
16	58,2	16	68,3	16	67,7
17	57,3	17	73,6	17	68,2
18	67,4	18	60,3	18	60,9
19	62,9	19	60,5	19	62,5
20	55,9	20	61,4	20	61,7
Sonitron SPS 41		Piezo \varnothing 35mm		Piezo \varnothing 20mm	

Bei der Einzelmessung ist zu erkennen, dass alle Lautsprecher ähnliche Werte aufweisen. Der Piezolausprecher mit einem Durchmesser von 35mm besitzt bei einer Frequenz von 17 kHz die maximale Lautstärke. Durchschnittlich erreicht der Sonitron SPS 61,58 dB. Der Piezolausprecher mit einem Durchmesser von 35mm erreicht durchschnittlich 64,83 dB und der Lautsprecher mit einem Durchmesser von 20mm 63,88 dB. Eine Rechteckspannung an den Lautsprechern führt zu gleichen Ergebnissen. Eine Rechteckspannung wirkt sich also weder positiv, noch negativ auf die erzeugte Lautstärke aus.

Im nächsten Schritt wurde eine Parallelschaltung zweier Lautsprecher mit einer sinusförmigen Spannung untersucht.

Tabelle 2.2: Parallelschaltung zweier Lautsprecher

f/kHz	dB	f/kHz	dB	f/kHz	dB
15	77,4	15	70,3	15	72,8
16	69,6	16	61,5	16	67,3
17	54,8	17	63,9	17	73,0
18	57,9	18	68,1	18	66,9
19	60,1	19	60,2	19	54,2
20	62,4	20	62,3	20	59,0
Sonitron SPS 41		Piezo \varnothing 35mm		Piezo \varnothing 20mm	

Es lässt sich eine Erhöhung der Lautstärke feststellen. Der SPS 41 erzeugt die höchsten Werte, gefolgt von den Piezolausprechern mit einem Durchmesser von 20mm. Da der Lautsprecher SPS 41 mit den Maßen 43.5 x 41 x 4.5mm recht sperrig ist und nicht mehr als zwei Stück parallel in einem Gerät verbaut werden können, wurden weitere Messungen mit dem Piezolausprecher mit einem Durchmesser von 20mm gemacht. Es wurden mehrere Piezolausprecher

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

2 Lautstärkemessung

parallel geschaltet.

Tabelle 2.3: Parallelschaltung mehrerer Piezos mit \varnothing 20mm

f/kHz	dB	f/kHz	dB
15	88,7	15	92,1
16	80,7	16	81,0
17	74,7	17	79,3
18	76,0	18	81,4
19	73,3	19	73,2
20	67,2	20	80,2
Vier Stück parallel		Acht Stück parallel	

Bei den Messungen wurde herausgefunden, dass mehrere parallele Piezoscheiben mit einem Durchmesser von nur 20 mm sehr gute Werte aufweisen. Sie erzeugen eine höhere Lautstärke als beispielsweise zwei parallele Sonitron SPS 41. Desto mehr parallele Piezolautsprecher, desto höher ist die erzeugte Lautstärke. Zuvor wurde angenommen, dass ein einzelner, großer Lautsprecher die besten Ergebnisse liefert.

Kleine Piezoscheiben sind für den Ultraschallgenerator ideal. Sie sind klein, besonders flach, besitzen eine geringe Impedanz bei hohen Frequenzen, robust und zudem um einiges günstiger als beispielsweise der SPS 41. Zusätzlich wurde eine Reihenschaltung aus zwei Lautsprechern getestet.

Tabelle 2.4: Reihenschaltungschaltung zweier Lautsprecher

f/kHz	dB	f/kHz	dB
15	65,1	15	68,2
16	66,7	16	62,4
17	65,2	17	69,7
18	71,4	18	62,1
19	57,7	19	51,7
20	65,0	20	56,5
Zwei Sonitron SPS 41		Zwei Piezo \varnothing 20mm	

Eine Reihenschaltung verschiedener Lautsprecher führt wie erwartet zu einer Abschwächung der Lautstärke gegenüber einer Parallelschaltung. Die Piezolautsprecher schwingen nicht synchron und behindern sich gegenseitig bei der Schallabgabe.

Im nächsten Versuch wurden vier Piezoscheiben mit dem Durchmesser von 20mm mit handelsüblichem Flüssigklebstoff direkt an dem Metallgehäusedeckel befestigt.

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

2 Lautstärkemessung

Tabelle 2.5: Lautsprecher im Gehäuse

f/kHz	dB	f/kHz	dB
15	94,3	15	91,7
16	84,5	16	84,9
17	83,7	17	79,9
18	102,7	18	97,4
19	80,4	19	84,7
20	91,4	20	90,1

Offenes Gehäuse Geschlossenes Gehäuse

Die Lautstärke erhöht sich so um bis zu 26 dB. Bei 18 kHz wurde ein Maximalwert von 102,7 dB gemessen. Die Schwingung der Piezolautsprecher wird an das Metallgehäuse übertragen und so die Abstrahlfläche erhöht. Zudem dient das Gehäuse als eine Art Resonanzkörper. Die Kabel der parallel geschalteten Piezoscheiben sollten alle die selbe Länge haben. So biegen sich die Lautsprecher synchron und es wird vermieden, dass sich die Schallwellen gegenseitig beeinflussen und die Lautstärke gedämpft wird.

Zusätzlich sollte die Lautstärke außerhalb des Bienenstocks gemessen werden. Es sollte herausgefunden werden, in welcher Entfernung die Piezolautsprecher noch hörbar sind und in wie weit sie damit die Umwelt beeinflussen. Bestenfalls sind sie nach nur kurzer Entfernung nicht mehr wahrnehmbar.

Tabelle 2.6: Lautstärke außerhalb des Bienenstocks

Abstand/cm	dB
0	55
20	47
40	42
60	37
80	36
100	35

Verhältnis Abstand/Lautstärke

Es stellte sich heraus, dass die Lautstärke durch den Bienenstock stark abgeschwächt wird. Es wurde eine möglichst ruhige Messumgebung aufgebaut. Dort wurden 35.3 dB Grundlautstärke gemessen. Ein Piezolautsprecher wurde in den Bienenstock gelegt und mit einer sinusförmigen Spannung mit $2.5 V_{pp}$ und 15 kHz angeregt. Die fast 90 dB im Bienenstock waren selbst unmittelbar neben dem Bienenstock stark geschwächt. Dort wurden 55 dB gemessen. Mit größerem Abstand verringerte sich die Lautstärke immer mehr. Ab ca. 50 cm Abstand zum Bienenstock wurden rund 40 dB gemessen. Dieser Wert liegt nah an der Grundlautstärke des Raumes. Es lässt sich somit bestätigen, dass das Ultraschallgerät kaum Einflüsse auf die Umwelt außerhalb des Bienenstocks hat und somit keine Menschen oder Tiere in der Nähe des Bienenstocks

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

2 Lautstärkemessung

stören wird.

Folgende Abbildung zeigt die Frequenzfindung des zweiten Prototyps in einem Sonogramm.

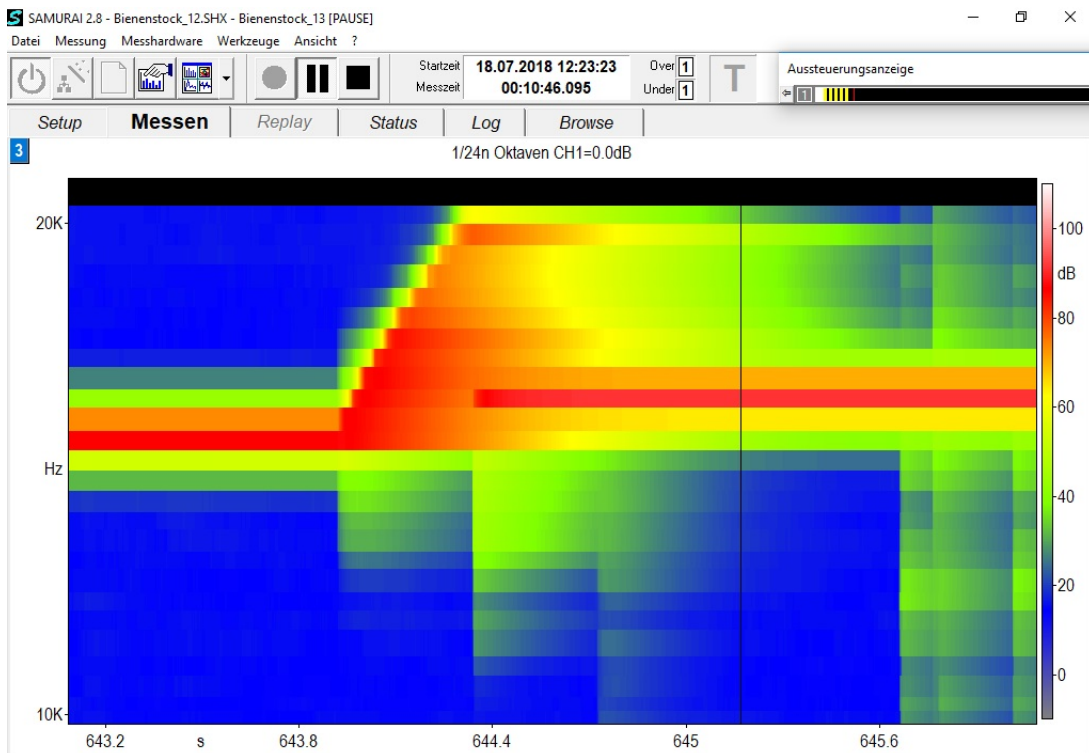


Abbildung 2.2: Sonogramm der Frequenzfindung

Es lässt sich erkennen, dass das Gerät mithilfe des Mikrocontrollers und des DDS-Chips mit einer Frequenz von 15 kHz beginnt. Stufenweise erhöht sich die Frequenz auf 20 kHz und anschließend wird die Resonanzfrequenz des Schwingkreises mithilfe der Spannungsmessung des Mikrocontrollers eingestellt. Es wurde ein Schwingkreis mit einer Resonanzfrequenz von etwa 15.9 kHz gewählt. Die Frequenzfindung dauert in etwa eine halbe Sekunde. Mit der Initialisierung des DDS-Chips dauert die vollständige Frequenzfindung etwas über eine Sekunde.

3 Der erste Prototyp

Aus einer Vorarbeit stand ein erster Prototyp zur Verfügung, der fertiggestellt und untersucht wurde. Erste Überlegungen waren es den Parallelschwingkreis mit einer Rechteckspannung zu versorgen, da diese Art von Spannung einfach mittels Mikrocontroller erzeugt werden kann.

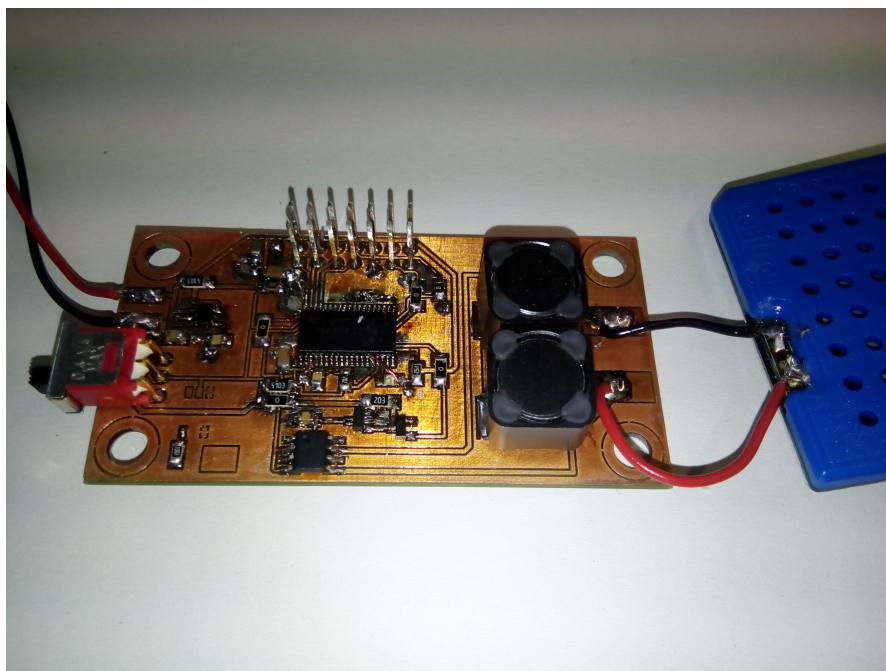


Abbildung 3.1: Platine des ersten Prototyps

Es musste geprüft werden, wie sich der Schwingkreis mit einer Rechteckspannung verhält und ob eine sinusförmige Spannung besser geeignet ist. Den vollständigen Schaltplan des Prototyps siehe Anhang "Schaltplan Prototyp 1".

3.1 Funktionsweise

Das Modell steuert mittels Timer im Mikrocontroller MSP430 und H-Brücke den Piezolautsprecher. Eine H - Brücke ist eine Schaltung aus vier Transistoren mit einer Umschaltung der Polarität der Spannung.

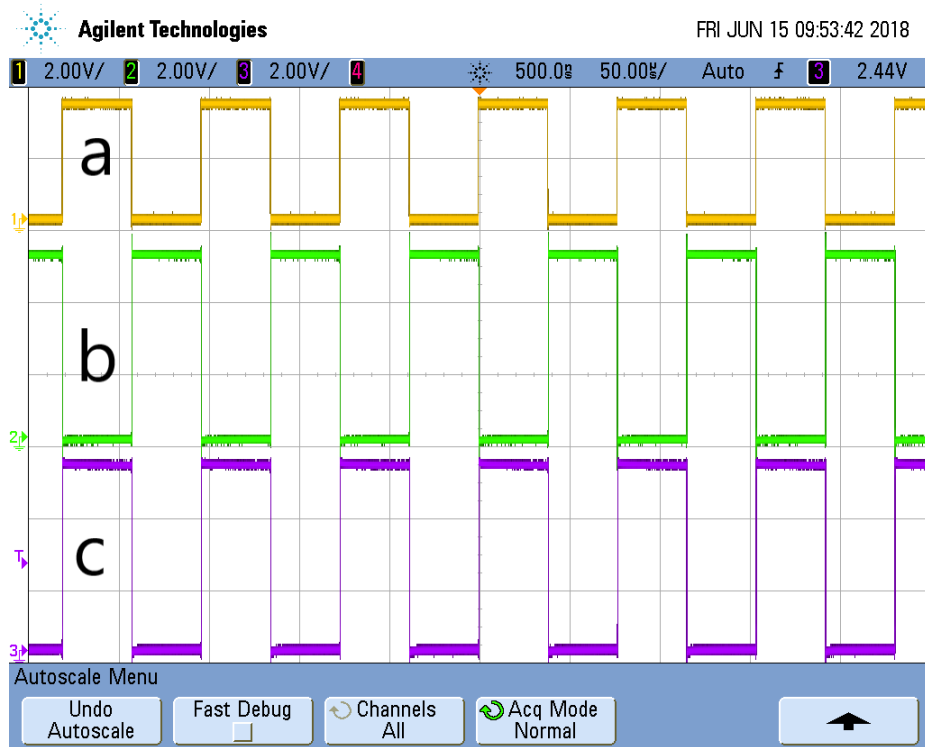


Abbildung 3.2: Ausgänge der H-Brücke ohne Last

In Abb.3.2 ist das PWM-Signal (a, gelb), welches vom Mikrocontroller erzeugt wird, sowie die beiden Ausgänge der H-Brücke (b, grün und c, lila) zu sehen. Abwechselnd schalten sich die Ausgänge 'high' und 'low' und erzeugen so ein gegenphasiges Signal über der Spule und dem Piezo-Lautsprecher.

Zudem besitzt es eine Spannungsüberwachung, einen Schalter und Spannungsregler, der die Spannung auf die vom MSP430 benötigten 3.3 V regelt.

Bei dem Prototyp wurde ein Stromverbrauch von $I = 76mA$ gemessen. Es folgte eine Überprüfung und Messung aller Bauteile. Es stellte sich heraus, dass der Parallelschwingkreis durch das Betreiben mit einer Rechteckspannung einen hohen Stromverbrauch hat. Zu Testzwecken wurde ein ohmscher Widerstand als Last verwendet. Der Stromverbrauch des Prototypen fiel auf $I = 20mA$. Abb. 3.3 zeigt die Spannung am Ausgang der H-Brücke mit einem Piezo-lautsprecher als Last. Die Spannung hatte eine sägezahnähnliche Form mit Spannungsspitzen und Spannungsabfällen. Man konnte Auf- und Entladekurven der Kapazität erkennen. In Abb. 3.4 ist die Spannung am Ausgang der H-Brücke mit einem LC-Parallelschwingkreis zu sehen. Auch hier sind Spannungsspitzen und Spannungsabfälle zu erkennen. Es wurde ein Strom von $I = 49mA$ gemessen.

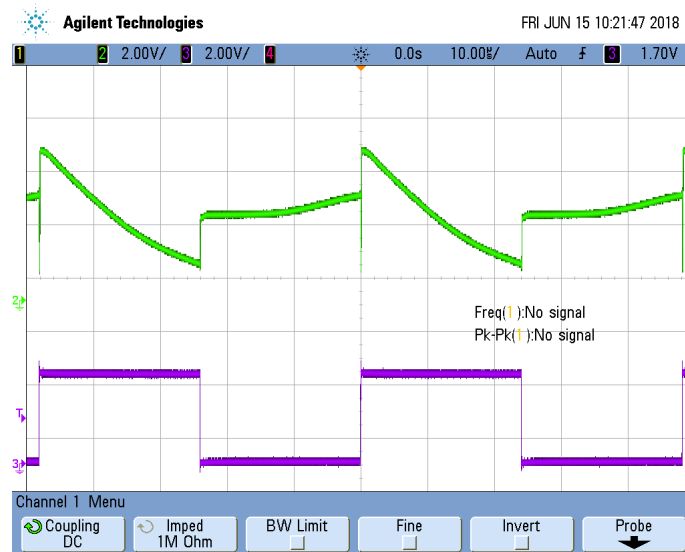


Abbildung 3.3: Ausgang der H-Brücke mit kapazitivem Lautsprecher

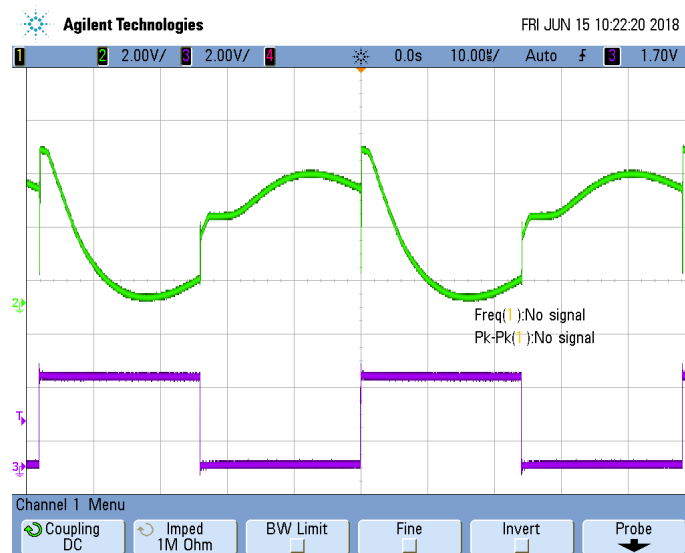


Abbildung 3.4: Ausgang der H-Brücke mit Parallelschwingkreis

3.2 Fazit

Die Versorgung eines LC-Schwingkreises mit einer Rechteckspannung führt zu einem hohen Stromfluss. Eine Periode einer Rechteckspannung kann als Schaltvorgang betrachtet werden, der bei Kapazitäten und Induktivitäten zu Spannungsspitzen und Stromspitzen führt.

Akustisch konnte kein Unterschied zwischen dem Betreiben eines Piezolausprechers mit einer Rechteckspannung oder mit einer Sinusspannung festgestellt werden. Dieser Prototyp beinhaltet nur die Grundfunktion eines Ultraschallgerätes. Es konnte weder den besten Arbeitspunkt eigenständig suchen, noch die Dauer der Schallabgabe variieren. Da ein möglichst stromsparendes Gerät entwickelt werden sollte wurde ein weiterer Prototyp ausgearbeitet, der den Schwingkreis mit einer sinusförmigen Spannung betreibt.

4 Der zweite Prototyp

4.1 Funktionsweise

Der zweite Prototyp wurde mit dem Mikrocontroller Arduino ProMini und dem Modul "Mini-Gen" von der Firma Sparkfun realisiert.

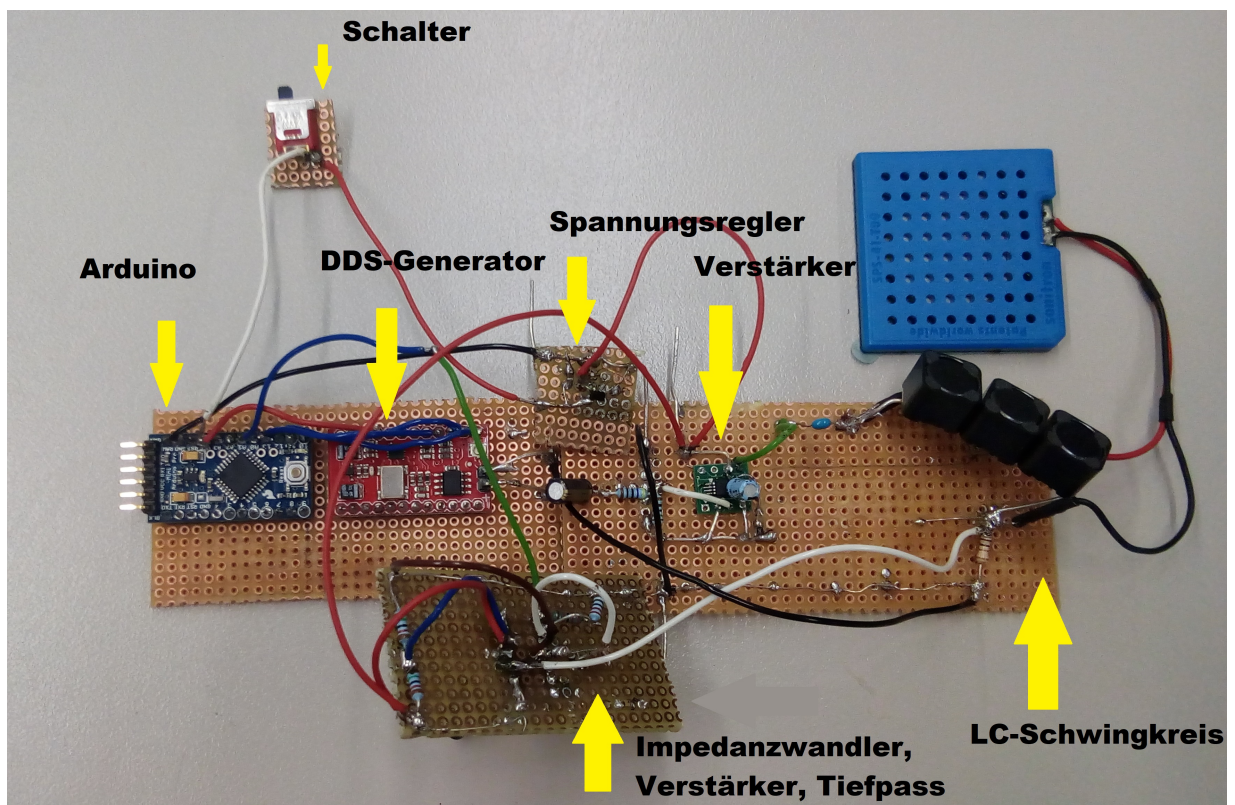


Abbildung 4.1: Der zweite Prototyp

Das Modul arbeitet mit dem DDS-Generator-Chip AD9837, der Signale in Sinus- Rechteck- oder Dreieckform mit bis zu 3 MHz erzeugen kann. Zudem besitzt das Modul einen sehr schnellen Operationsverstärker. Signale werden mit einer Amplitude von $U_{pp} = 1V$ ausgegeben. Dieser Wert kann nicht variiert werden.

Der Arduino kommuniziert mit dem Bussystem SPI (Serial Peripheral Interface) mit dem MiniGen-Modul. Dieses serielle, synchrone Dreidrahtbussystem besitzt zwei Leitungen für den Datenaustausch und eine weitere für den Takt. Der Mikrocontroller wurde mit der Arduino

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

4 Der zweite Prototyp

Entwicklungsumgebung Arduino IDE 1.8.3 programmiert.

Ein Grund für die Wahl auf das Modul von Sparkfun für den Prototypen war die vorhandene Bibliothek zum programmieren des MiniGen-Moduls. Diese Bibliothek beinhaltet viele Funktionen, durch die das Programmieren stark vereinfacht und beschleunigt wird. Diese Eigenschaften machen das Modul nützlich für die Entwicklung eines Prototypen. Es stellte sich heraus, dass der Operationsverstärker AD8045 und der vorhandene Quarz auf dem MiniGen-Modul viel Strom verbrauchen. Deshalb wurden auf dem entwickelten Gerät andere Bauteile verwendet.

Das Ultraschallgerät soll damit beginnen eine sinusförmige Spannung mit 15 kHz zu erzeugen und anschließend den stromsparendsten Arbeitspunkt finden. Dieser liegt bei der Resonanzfrequenz von dem LC-Schwingkreis. Dazu wird die Frequenz kontinuierlich erhöht und bei jeder Frequenz die Spannung über einem Messwiderstand gemessen. Nachdem der Arbeitspunkt eingestellt wurde sollen verschiedene Stufen einstellbar sein, welche die Dauer der Schallabgabe verändern. Um zusätzlich Strom zu sparen soll der Mikrocontroller und der DDS-Generator in Schlafmodi versetzt werden, sofern keine neue Aktion durchgeführt werden muss.

4.2 Platinenherstellung

Die Platine des Gerätes wurde mit der Software Altium Designer von Altium Limited entwickelt. Dieses EDA (Electronic Design Automation)- Programm wird von namhaften Firmen wie Bosch oder Intel zur Leiterplattenentwicklung genutzt und erhielt zahlreiche Auszeichnungen.[12]

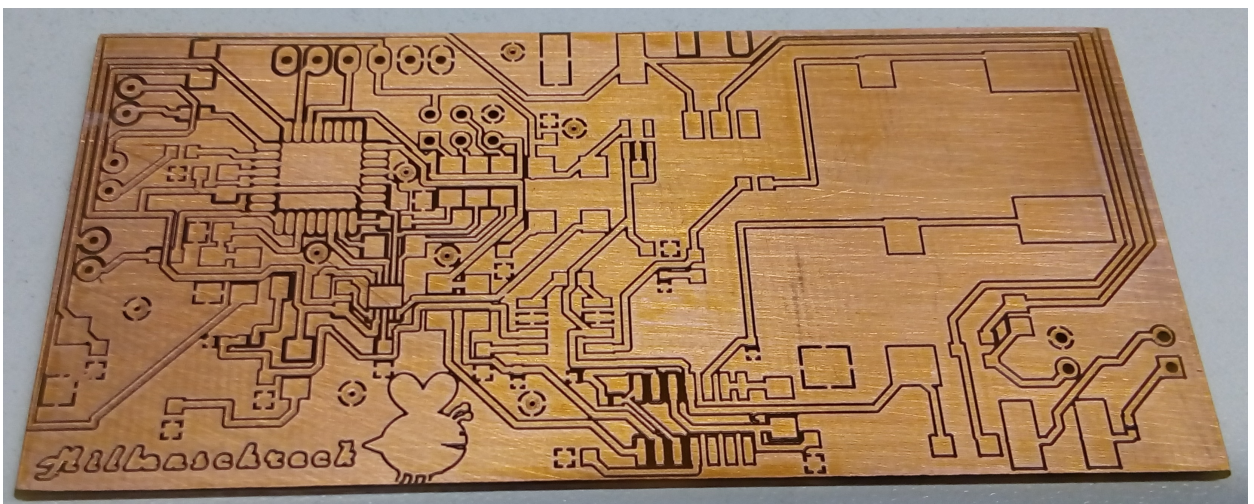


Abbildung 4.2: Entwickelte Platine

Alle Bauteile befinden sich auf dem "Top Layer", also der Vorderseite der Platine. Die gesamte Kupferschicht der Platine hat eine Masseverbindung. Dies gilt natürlich nicht für Leiterbahnen und Bauteile. Der "Bottom-Layer" hat ebenfalls eine Masseverbindung. Dank sogenannter "Via's", also Durchkontaktierungen auf der Platine, kann eine Verbindung mit Masse auch an

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

4 Der zweite Prototyp

schwer erreichbaren oder durch Leiterbahn eingeschlossene Stellen erreicht werden. Es wurde eine Leiterbahndicke von 0.405 mm gewählt. Zudem wurden einige 0 Ohm Widerstände zur Überbrückung von Leiterbahnen benötigt. Zwischen den Pads dieser Widerstände musste die Leiterbahndicke verringert werden.

Neben den vorhandenen Bauteilen in der Altium Bibliothek mussten weitere Bauteile hinzugefügt werden. Diese Daten in der Bibliothek beinhalten eine "Schematic-Datei", sowie eine Datei für den Footprint des Gehäuses von dem Bauteil. Viele Online-Händler für Elektronik bieten kostenlose PCB-Designs ihrer Waren direkt zum downloaden an. Größtenteils sind diese Dateien allerdings für die Software "Eagle". Dank des sogenannten Import Wizard von Altium Designer können diese Eagle-Dateien direkt in PCB-Dateien für Altium konvertiert werden. Für die auf dem Gerät verwendete Spule konnte keine PCB-Datei gefunden werden. Für diesen Fall hat Altium Designer die Funktion "IPC Compliant Footprint Wizard". Mit dieser Funktion und mithilfe von Vermessungsdaten eines Bauteils können Footprints sehr leicht automatisch erstellt werden. Es müssen lediglich die Größenangaben eines Bauteils eingetragen werden und das Programm erstellt automatisch passende Pads für das Bauteil. Diese Vermessungsdaten findet man in jedem Datenblatt eines Bauteils.

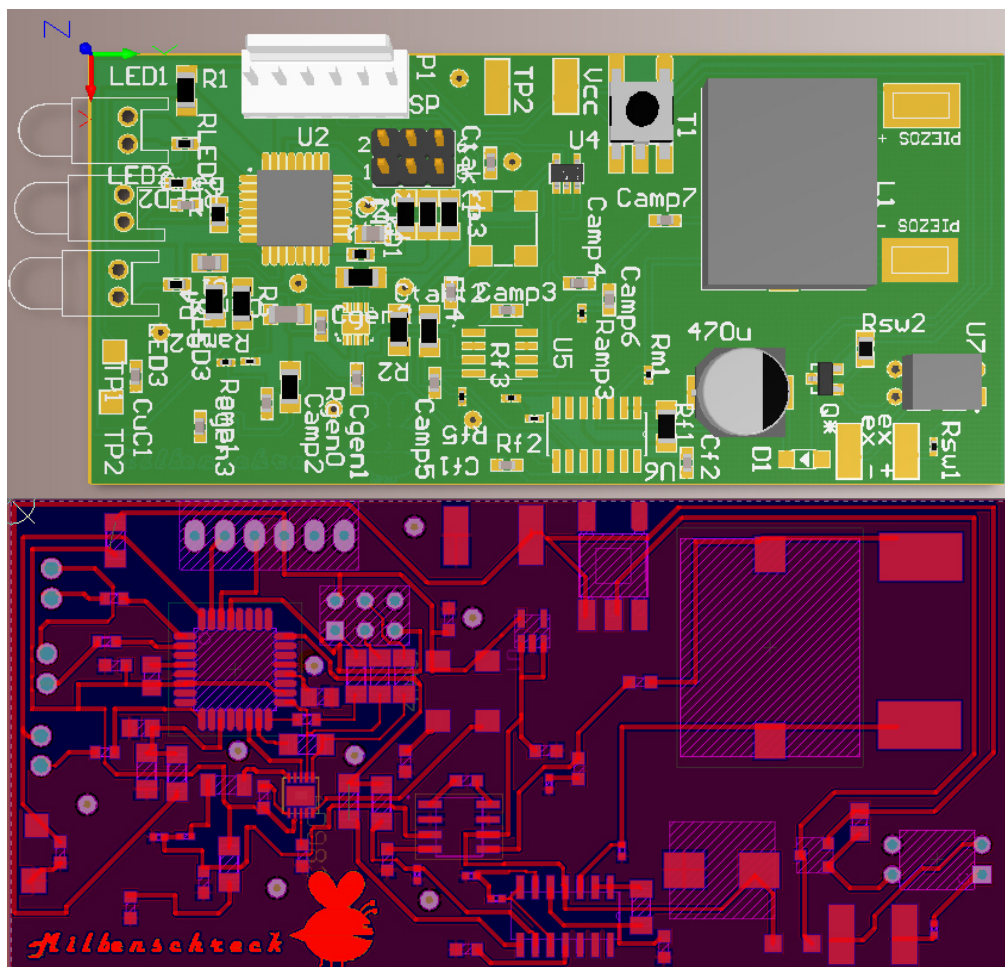


Abbildung 4.3: PCB-Design in Altium Designer

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

4 Der zweite Prototyp

Zusätzlich zu allen benötigten Footprints und Leiterbahnen wurde ein Logo auf die Platine implementiert. Das Logo besitzt den Schriftzug "Milbenscheck" und den Umriss einer Biene. Dieses Logo konnte mit dem Script "AltiumLogoCreator" eingefügt werden.

Zum Schluss mussten "Gerber"- und "Drill"-Dateien erstellt werden. Diese Dateien werden für das Fräsen einer Platine benötigt.

Nach dem Fräsen wurden die Platine per Hand mit Bauteilen bestückt.

Nach der Bestückung wurden Messungen durchgeführt und einige Fehler behoben. Diese Fehler bestanden meist aus Lötfehlern. Bei zwei Bauteilen bestand keine Masseverbindung. Dieser Fehler entstand durch fehlerhaftes Routing der Leiterbahnen im PCB-Design.

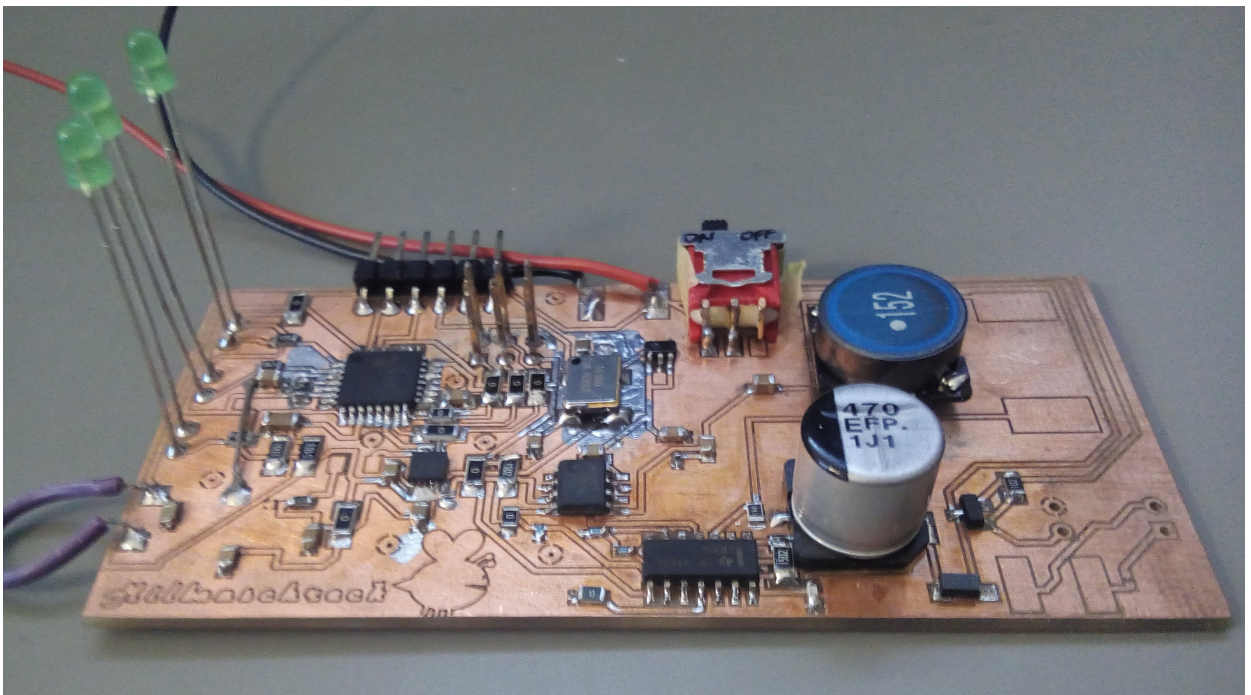


Abbildung 4.4: Bestückte Platine

4.3 Operationsverstärker LM386

Es wurde herausgefunden, dass der Operationsverstärker AD8045 auf dem MiniGen-Modul 18 mA verbraucht. Da dieser Wert zu hoch ist wurde sich entschieden einen anderen Verstärker zu nutzen. Die Wahl fiel auf den LM386 von Texas Instruments. Dieser Audio-Operationsverstärker kann je nach Schaltung eine Verstärkung von 20 - 200 erreichen und arbeitet dabei mit 4 - 8 mA bei einer Versorgungsspannung von 6V. Abb. 4.2 zeigt die verwendete Schaltung, die direkt an den Ausgang des DSS-Generator-Chips AD9837 angeschlossen wird. Der Widerstand R7 und der Kondensator C7 zwischen den Pins "g1" und "g8" sollen eine Verstärkung von 50 festlegen. Die Kondensatoren C5 und C2 sind Koppelkondensatoren zur Filterung des Gleichanteils

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

4 Der zweite Prototyp

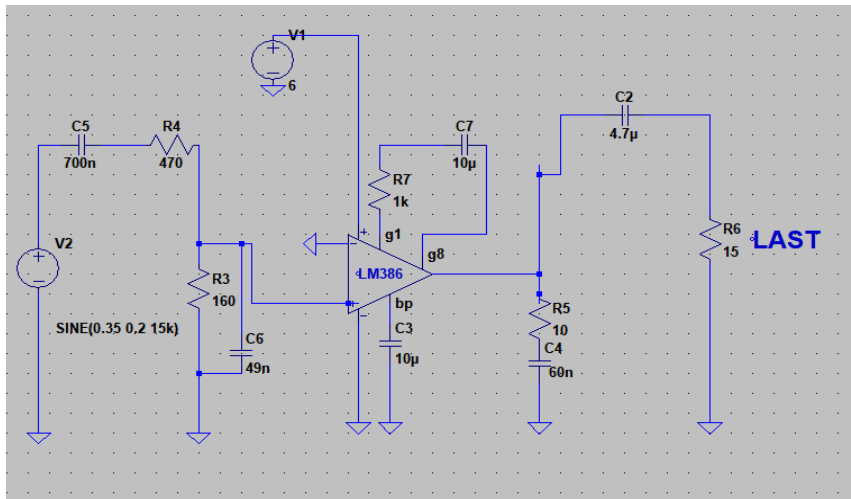


Abbildung 4.5: Schaltung LM386 in LT-Spice

der Spannung. Ein Spannungsteiler aus R3 und R4 vor dem Eingang des Operationsverstärkers führt zu einer geringen Eingangsspannung. Der Verstärker darf eine maximale Eingangsspannung von $-0.4V < U_I < 0.4V$ haben. Desto kleiner die Spannung ist, desto sauberer wird ein Sinussignal erzeugt. Übersteigt der Wert 45 mV bei einer Versorgungsspannung von 6 V, wird die Spitze des Sinusausgangssignal abgeschnitten.

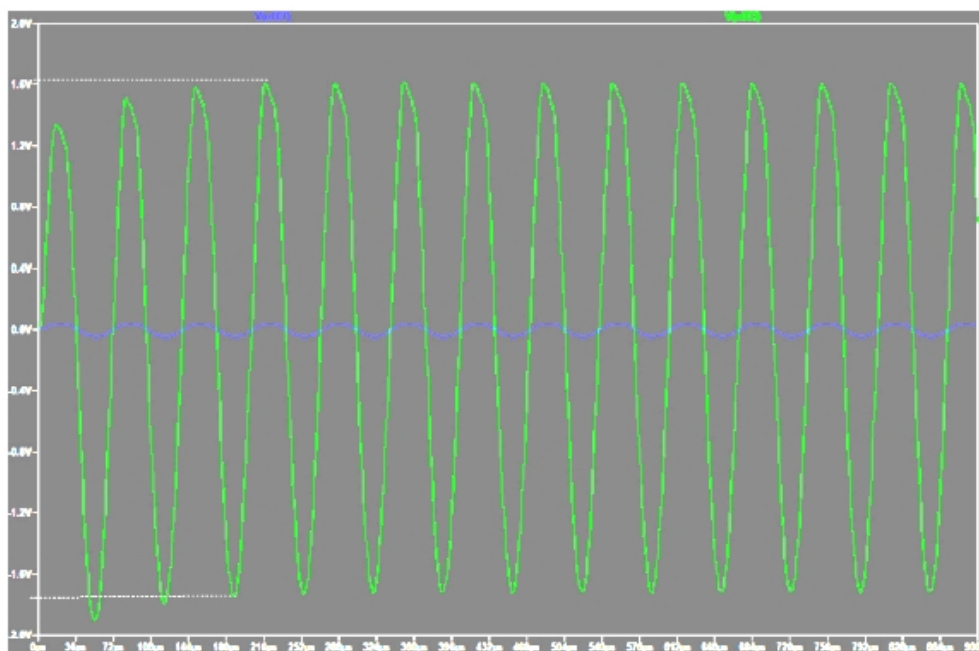


Abbildung 4.6: Verhältnis zwischen Ein- und Ausgangsspannung LM386

Eine Simulation mit dem Programm LT-Spice der Schaltung aus Abb. 4.2 ergab eine Verstärkung von

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

4 Der zweite Prototyp

$$v = \frac{U_A}{U_E} = \frac{1.6V}{0.04V} = 40$$

Es wird ein Spitze-zu-Spitze Ausgangssignal $U_{E_{pp}} = 3.2V$ erzeugt.

4.4 Operationsverstärker LM324

Die Spannung über dem Shunt zur Spannungsmessung mit dem Mikrocontroller war sehr verrauscht. Dies führte zu ungenauen Messungen und folglich zu einer falschen Resonanzfrequenzeinstellung des ATmega328. Aus diesem Grund wurde eine Schaltung zur Glättung und Verstärkung entwickelt. Der LM324 ist ein vierfacher Operationsverstärker in nur einem Gehäuse. Drei dieser OPV's wurden verwendet. Die Schaltung besteht aus einem Impedanzwandler, einem Tiefpassfilter erster Ordnung und einem Verstärker. Der Impedanzwandler hat

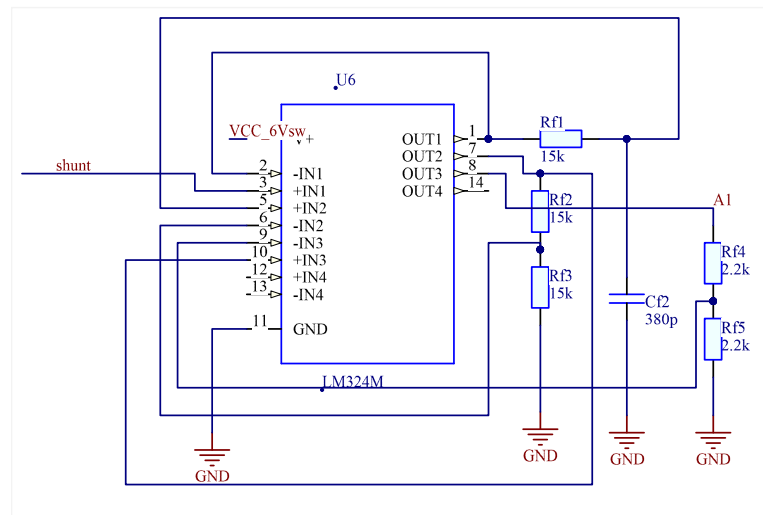


Abbildung 4.7: Schaltung des LM324

eine sehr hohe Eingangsimpedanz und eine geringe Ausgangsimpedanz. An dem Ausgang des Impedanzwandlers wurde ein Tiefpassfilter 1. Ordnung integriert. Der Tiefpass hat die Werte

$$R = 15 k\Omega$$

$$C = 380 pF$$

Es ergibt sich eine Grenzfrequenz von

$$f_G = \frac{1}{2 \cdot \pi \cdot 15 k\Omega \cdot 380 pF} = 27922 Hz$$

In der Praxis weicht dieser Wert allerdings etwas ab. Dies ist auf nicht beachtete Leitungswiderstände und Toleranzen der Bauteile zurückzuführen. Im nachfolgendem Bild sind Eingangs-/

Entwicklung eines Gerätes zur Schädlingbekämpfung in Bienenstöcken mittels Ultraschall

4 Der zweite Prototyp

und Ausgangsspannung des Tiefpassfilters zu sehen. Bei der Grenzfrequenz haben Ein-/ und Ausgangssignal eine Phasenverschiebung von 45° . Bei $1 V_{pp}$ besitzt das Ausgangssignal eine Amplitude von $0.707 V_{pp}$. Die Grenzfrequenz liegt in unserem Fall bei etwa 31 kHz.

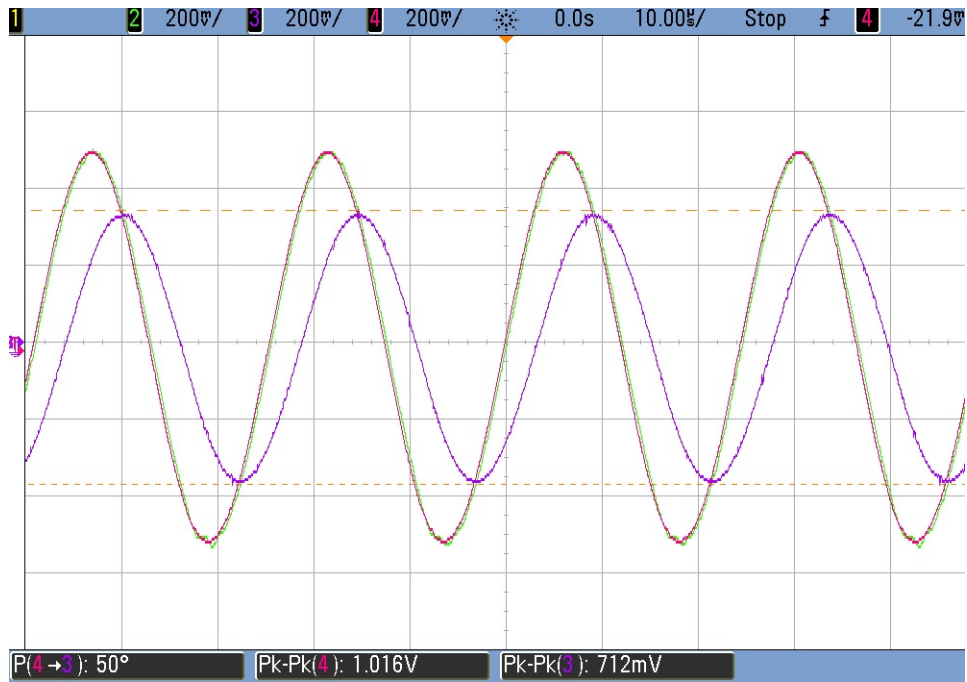


Abbildung 4.8: Spannungen am Tiefpassfilter

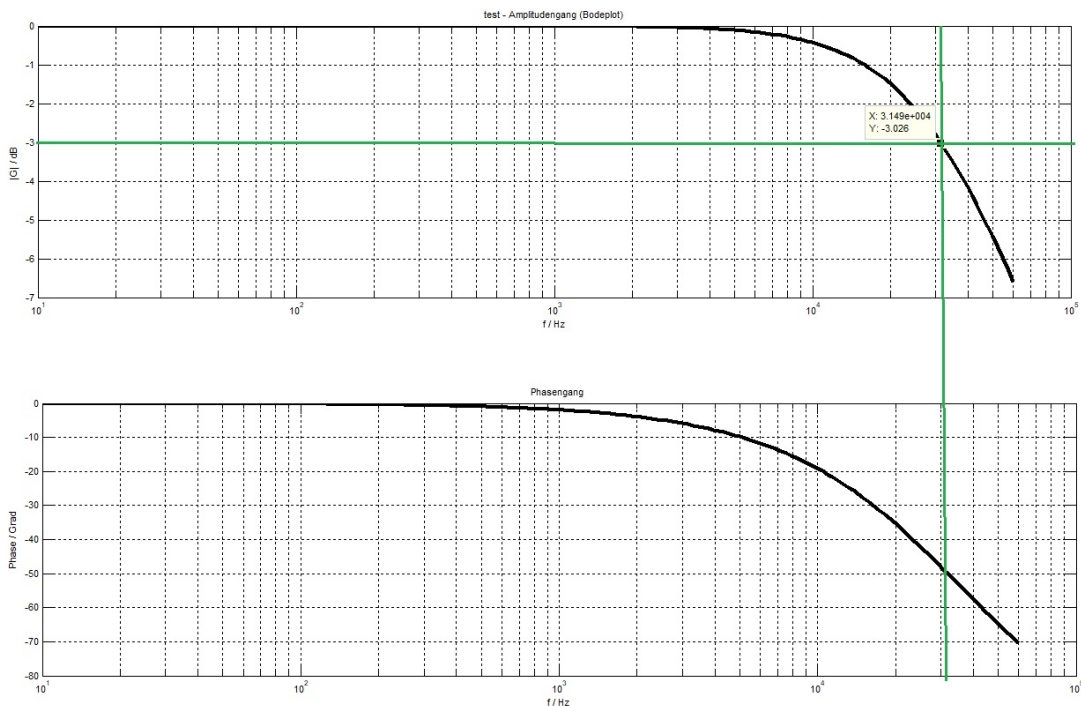


Abbildung 4.9: Amplituden-/ und Phasengang RC-Tiefpass

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

4 Der zweite Prototyp

Der Tiefpassfilter wurde auf seine Funktionalität getestet. Hierzu wurde ein Rauschen mit der 'Noise'-Funktion eines Funktionsgenerators erzeugt. In folgender Grafik (Abb. 4.10) lässt sich gut erkennen, dass der Tiefpassfilter das verrauschte Signal (gelb) glättet.

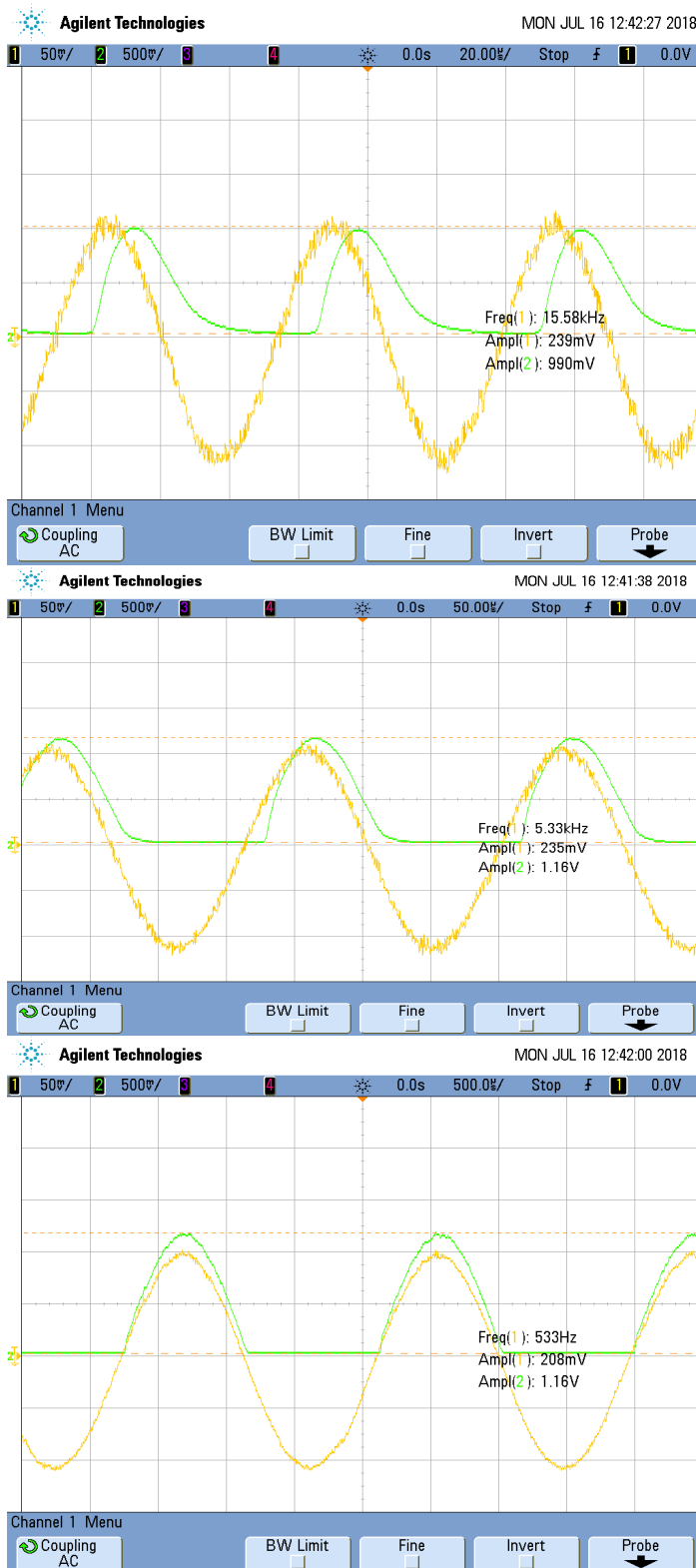


Abbildung 4.10: Rauschglättung

Der Operationsverstärker verstärkt die positive Flanke des Signals mit einer Verstärkung von ca. 11. Bei einer Frequenz von 15 kHz ist eine deutliche Phasenverschiebung und eine Dämpfung des Ausgangssignals zu erkennen. Die Spannung über dem Shunt wird so weit verstärkt, dass die 1.1 V Referenzspannung des Mikrocontrollers vollständig erreicht wird. Die Abbildung 4.10 zeigt das Eingangssignal und das Ausgangssignal des Tiefpassfilters bei verschiedenen Frequenzen. Bei einer Frequenz von $f = 533 \text{ Hz}$ sind beide Signale in Phase. Eine Verbesserung des Rauschens ist nicht zu erkennen. Bei einer Frequenz von $f = 5.33 \text{ kHz}$ macht sich eine geringe Phasenverschiebung bemerkbar. Die Rauschglättung ist zu erkennen. Bei Frequenzen ab $f = 15 \text{ kHz}$ lässt sich eine deutliche Phasenverschiebung feststellen. Das Ausgangssignal ist gedämpft und ein Rauschen nicht mehr erkennbar.

Im nächsten Schritt wurde die Schaltung mit dem Hauptmodell verbunden und getestet, ob der Mikrocontroller nun genauer über dem Shunt messen kann.

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

4 Der zweite Prototyp

Es stellte sich heraus, dass die Frequenz nun genauer eingestellt wird. Vergleicht man das Signal über den Shunt mit dem, welches vom Tiefpass und dem OPV geglättet und verstärkt wird, sieht man eine deutliche Verbesserung. Dennoch ist ein geringes Rauschen und Spannungsspitzen zu erkennen. Die Frequenz, die vom Mikrocontroller eingestellt wird weicht um bis zu 200 Hertz von der eigentlichen, gemessenen Resonanzfrequenz ab.

4.5 Programmierung des ATmega328

Für den zweiten Prototypen werden zwei Header zur Programmierung benötigt. Der erste Header "P_ISP" ist für die einmalige In-System-Programmierung (ISP) zuständig. Der ATmega328 benötigt zur Programmierung über UART einen sogenannten "Bootloader". Es handelt sich um ein kleines Programm, das es möglich macht, den ATmega über serielle Schnittstellen zu programmieren. Die Arduino-Boards haben bereits beim Kauf einen "Bootloader" installiert, jedoch nicht die separat erhältlichen ATmega328. Erst nachdem der Mikrocontroller mit dem Bootloader beschrieben wurde, kann der zweite Header für die serielle Programmierung über UART mit einem FTDI-Programmer verwendet werden. Die ISP-Header haben eine vorgege-

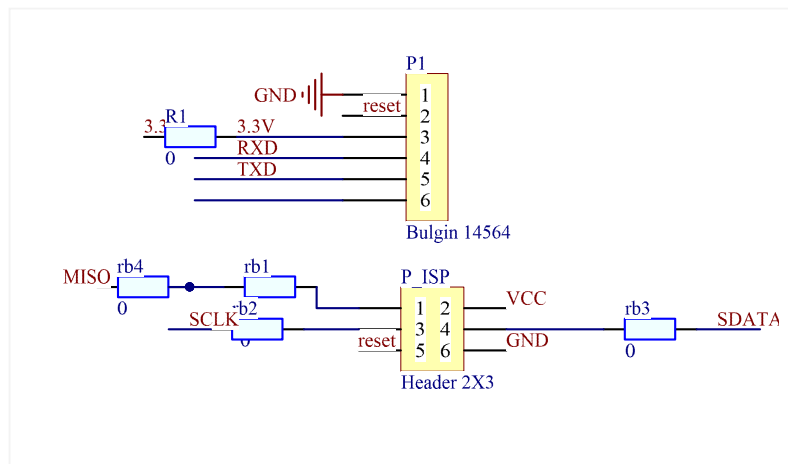


Abbildung 4.11: Header für ISP und FTDI

bene Form und Funktion der Pins. Bei dem Gerät sind zusätzlich 0 Ohm Widerstände verbaut, die als Brücken über Leiterbahnen dienen.

4.6 Spannungsüberwachung

Das Ultraschallgerät soll die Batteriespannung überwachen und die Ladung der Batterie abbrechen, wenn sie voll geladen ist. Dies geschieht mit einem Analogeingang des ATmega328 und über einen Spannungsteiler. Die Gesamtspannung über dem Spannungsteiler entspricht

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

4 Der zweite Prototyp

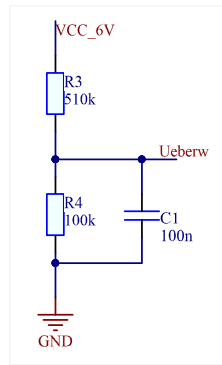


Abbildung 4.12: Schaltung der Spannungsüberwachung

der Batteriespannung $U_{ges} = 6V$. Die Spannung, die der Mikrocontroller misst entspricht der Spannung über dem Widerstand $R4$.

Es ergibt sich ein Spannungsverhältnis von

$$\frac{U_{R4}}{U_{ges}} = \frac{R4}{R3+R4}$$

$$U_{R4} = \frac{R4}{R3+R4} \cdot U_{ges} = \frac{100k\Omega}{510k\Omega+100k\Omega} \cdot 6V = 0.98V$$

Bei einer eingestellten internen Referenzspannung von $U_{ref} = 1.1V$ und einem 10-Bit ADC mit $2^{10} = 1024$ Werten entspricht dies dem Wert

$$\frac{1.1V}{1024} = \frac{0.98V}{x}$$

$$x = \frac{1024 \cdot 0.98V}{1.1V} \approx 912.$$

Der ADC des Mikrocontrollers wird also bei einer vollen Batteriespannung den Wert $x = 912$ zurück geben. Wird die Batteriespannung kleiner, verringert sich auch dieser Wert und es können Vorkehrungen zur Ladung der Batterie eingeleitet werden.

Die Laderegulung sollte möglichst einfach aufgebaut sein und mit möglichst wenig Bauteilen arbeiten. Die entwickelte Schaltung benötigt nur zwei Bauteile. Ein n-Kanal Feldeffekttransistor wird als Schalter verwendet. Das Gate des Transistors ist mit einem Ausgang des Mikrocontrollers verbunden. Für diese Art der Schaltung wird ein sogenannter "Logic Level FET" benötigt, da diese Art der Transistoren wenige Volt Spannung am Gate benötigen, um zu schalten. Ist der Pin den Mikrocontrollers auf "Low", liegen 0V am Gate des Transistors an. Die Gate-Spannung ist gleich der Source-Spannung und der Transistor ist gesperrt. In dieser Position wird die Batterie von dem Solarpanel geladen. Wird der Pin des Mikrocontrollers auf "High" gesetzt, öffnet sich der Transistor. Das Solarpanel wird kurzgeschlossen. Anders als die Batterie ist ein Solarpanel wie eine Stromquelle. Es kann problemlos kurzgeschlossen werden. Um einen Kurzschluss an der Batterie zu verhindern dient eine Diode.

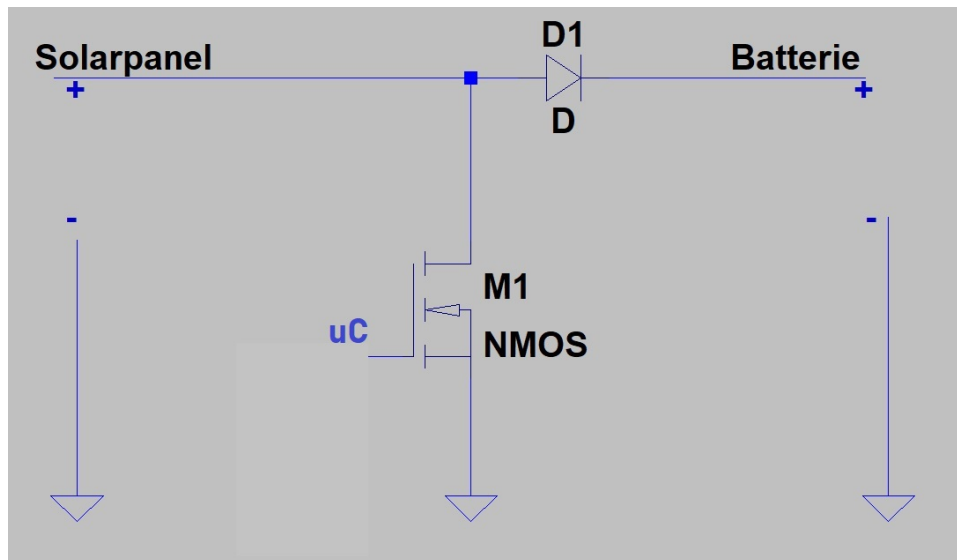


Abbildung 4.13: Schaltung der Laderegelung

Diese Schaltung ist zwar möglich, jedoch fällt bei kurzgeschlossenem Solarpanel kontinuierlich Spannung über dem N-FET ab. Eine Erwärmung der Platine wäre denkbar. Aus diesem Grund wurde eine weitere Schaltung zur Laderegelung entwickelt.

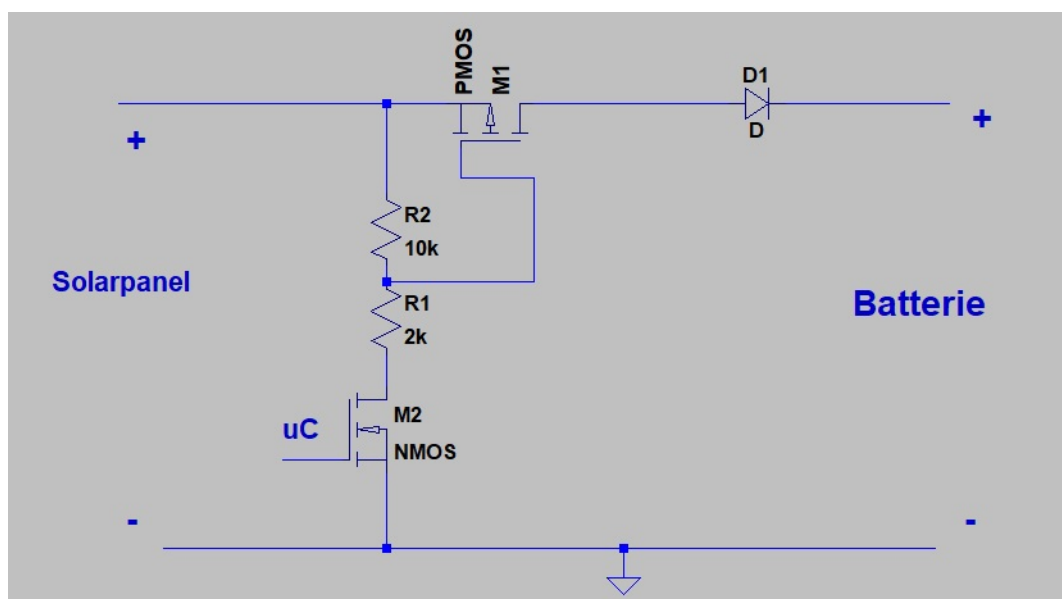


Abbildung 4.14: Verbesserte Laderegelung

In dieser Schaltung wird ebenfalls ein N-FET mit dem Mikrocontroller gesteuert. Wird der Pin des Mikrocontrollers auf "High" gesetzt, liegt eine höhere Spannung als am Source-Pin an. Der N-FET öffnet sich und stellt eine Masseverbindung her. An der Basis des P-FET liegt nun eine Spannung mit einem niedrigeren Potential an, als die Spannung am Source-Eingang des P-FET. Der P-FET als Schalter öffnet sich und die Batterie kann von Solarpanel geladen werden.

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

4 Der zweite Prototyp

Ist der Pin des Mikrocontrollers auf "Low " und somit ,dank des N-FET's, keine Masseverbindung vorhanden, fließt kein Strom durch die Schaltung.

4.7 Externe Ansteuerung

Neben der Betätigung eines Schalters soll das Gerät zusätzlich extern angesteuert werden. Ein Beispiel für ein solches externes Gerät wäre ein Arduino oder Raspberry Pi.

Hierzu wurde ein Optokoppler verwendet, der die Schaltung zusätzlich galvanisch trennt.

Liegt keine Spannung an dem Optokoppler an, liegen an der Basis des PNP-Transistors die 6V Betriebsspannung an. Der Transistor ist gesperrt und das Ultraschallgerät lässt sich nur durch die Betätigung des Schalters "T1 " mit der Betriebsspannung versorgen.

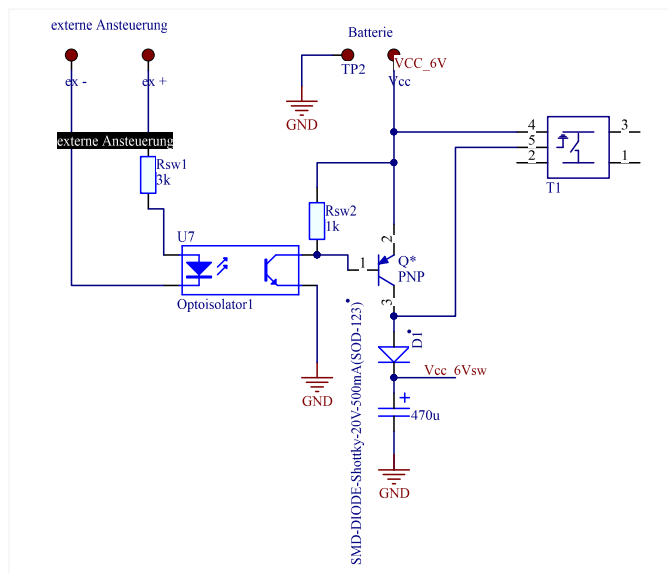


Abbildung 4.15: Schaltung zur externen Ansteuerung

Liegt nun eine Spannung an dem Optokoppler an , schaltet der NPN-Transistor des Bauteils durch. Das Potential an der Basis des PNP-Transistors ist geringer als das der Versorgungsspannung und der PNP-Transistor schaltet durch. So kann das Ultraschallgerät auch ohne die Betätigung eines Schalters mit der Betriebsspannung versorgt werden.

4.8 Programmcode für den Mikrocontroller

Mithilfe der "MiniGen"- Bibliothek kann der DDS-Generator mit nur wenigen Funktionen initialisiert werden.

```
Milbenschreck.reset();
```

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

4 Der zweite Prototyp

```
Milbenschreck.setMode(MiniGen::SINE);  
Milbenschreck.setFreqAdjustMode(MiniGen::FULL);  
  
freqReg = Milbenschreck.freqCalc(frequency);  
Milbenschreck.adjustFreq(MiniGen::FREQ0, freqReg);  
Milbenschreck.selectFreqReg(MiniGen::FREQ0);
```

Das MiniGen-Modul wird zurückgesetzt. Die Funktion

```
Name.setMode(newMode)
```

legt die Signalform fest. Hier ist Rechteckspannung, Dreieckspannung oder Sinusspannung möglich. Durch die Funktion

```
Name.selectFreqReg(Reg)
```

kann eines der beiden Frequenzregister ausgewählt werden. Durch zwei Frequenzregister kann der Benutzer einfach zwischen zwei Frequenzen wechseln. Um möglichst schnell eine Frequenz in das Frequenzregister zu schreiben existiert die Funktion

```
freqReg= Name.freqCalc(Frequenz)
```

Diese Funktion gibt einen Wert vom Datentyp "uint32_t" zurück, der direkt mit der Funktion

```
Name.adjustFreq(reg, freqReg)
```

in das Frequenzregister geschrieben werden kann. Zum Schluss kann mit der Funktion

```
Name.selectFreqReg(MiniGen::Reg)
```

das gewünschte Frequenzregister ausgewählt werden[11].

4.8.1 Spannungsmessung

Um mit einem Mikrocontroller Spannungen zu messen muss ein Pin mit einem Analog-Digital-Konverter (ADC) verwendet werden und als Eingang deklariert werden. Der 10-bit ADC des Arduinos kann $2^{10} = 1024$ Werte zurück geben. Die Spannung am Pin wird mit einer einstellbaren Referenzspannung verglichen. Der Wert 0 entspricht einer Spannung von 0V und der

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

4 Der zweite Prototyp

Wert 1023 entspricht einer Spannung, die äquivalent zu der Referenzspannung ist. Es können nur positive Spannungen gemessen werden. Die Spannung wird über einem Messwiderstand abgegriffen.

3-Bit Parallel-ADC

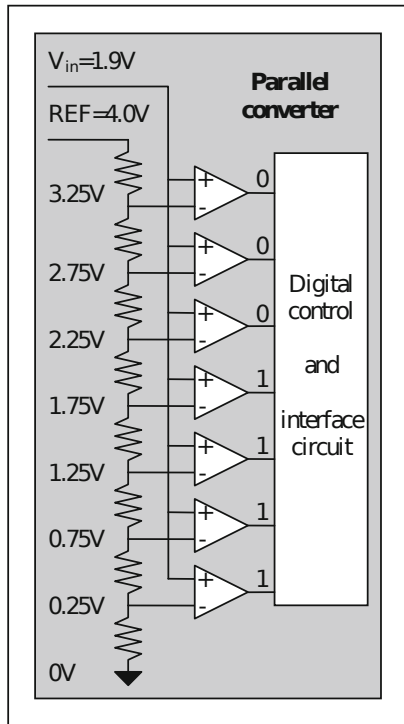


Abbildung 4.16: Analog-Digital-Umsetzer [9]

Ein Parallel-Analog-Digital-Umsetzer arbeitet mit Spannungsteilern, die eine Referenzspannung stufenweise teilen. Die Referenzspannung ist mit den nicht-invertierenden Eingängen der Komparatoren verbunden. Sie vergleichen die Spannung an den Spannungsteilern mit der Eingangsspannung und geben eine logische 1 aus, falls die Eingangsspannung gleich oder größer als die Spannung der Spannungsteiler ist. Ein n-Bit ADC gibt 2^n Werte zurück und benötigt 2^{n-1} Komparatoren. [9]

Bei dieser Arbeitsweise eines ADC's sollte sich die Eingangsspannung nicht ändern. Daher besitzen ADC's eine Abtast-Halte-Schaltung (Sample-and-Hold-Schaltung), die den Wert der Eingangsspannung zwischenspeichert. Dies geschieht analog mittels Kondensator und Schalter.

Besonders bei Wechselspannungen mit hohen Frequenzen ist eine Messung mit einem Mikrocontroller daher schwierig.

Die Bibliothek des Arduino besitzt eine Funktion zur einfachen Messung einer Spannung.

```
analogRead(Pin);
```

Die Funktion liest einen Analog-Pin und gibt einen Wert von 0-1023 zurück.

Diese Funktion dauert laut Datenblatt des ATmega328 ca. 0.0001 s. Ein Signal kann mit

$$f = \frac{1}{T} = \frac{1}{0.0001s} = 10000Hz$$

abgetastet und gemessen werden. Somit ist diese Funktion für die Messung in dem Ultraschallgerät nicht geeignet.

Der ADC des Arduinos kann in drei Modi betrieben werden. Der "Single Conversion Mode" misst einmalig einen Spannungswert und muss für mehrere Messungen immer wieder im Programmcode ausgelöst werden (analogRead()). In dem "Auto-Trigging-Mode" wird eine

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

4 Der zweite Prototyp

Messung des ADC's kontinuierlich von einem Timer, Interrupt oder einem Pin ausgelöst. Eine andere Methode der Spannungsmessung mit dem ADC eines Mikrocontrollers ist der "Free-Running-Mode ". Dieser Modus des ADC's ist der schnellste. Es beginnt eine neue Messung unmittelbar nachdem die Vorherige abgeschlossen ist.

ADC Timing Diagram, Free Running Conversion

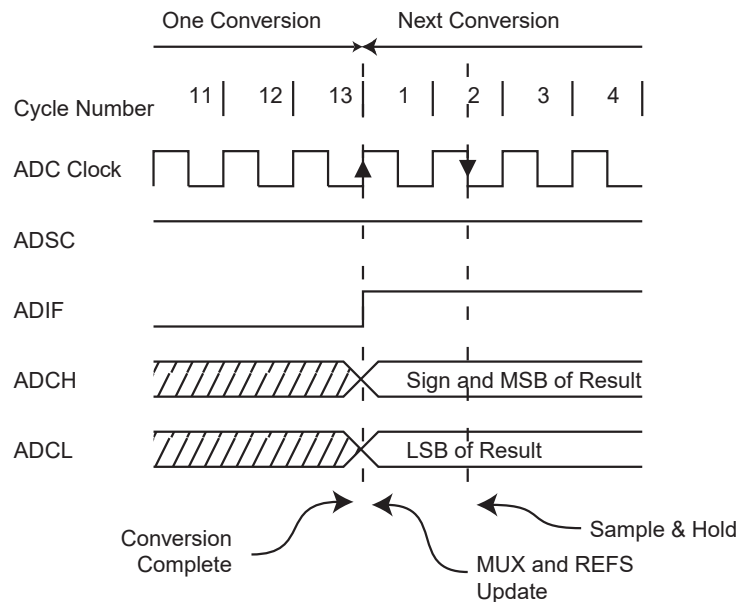


Abbildung 4.17: Zeiteinteilung des Analog-Digital-Umsetzers

Eine Messung des Analog-Digital-Umsetzers endet, wenn das "ADCH "und "ADCL"Register beschrieben sind und das ADC Interrupt-Flag "ADIF"gesetzt ist. Eine Messung dauert laut Datenblatt 13 Taktzyklen. Der ADC wurde wie folgt programmiert.

```
ISR(ADC_vect)
{
  ADC_val = ADCL;
  ADC_val = (ADCH << 8) + ADC_val;

  newReading = true;
  counter2++;
}

void setup() {
  ADMUX |= 0x00; //clear

  ADMUX |= (1 << REFS1) | (1 << REFS0); //1.1V Referenzspannung

  cbi(ADMUX, ADLAR);
  ADCSRA |= 0x02; //prescale 4

  // enable the Interrupt
```


Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

4 Der zweite Prototyp

```
sbi(ADCSRA, ADIE);

// free running mode
ADCSRB &= 0xF8;

// enable the Auto Trigger
sbi(ADCSRA, ADATE);

// enable the ADC
sbi(ADCSRA, ADEN);

sbi(ADCSRA, ADSC); //start Conversion
}
```

“ADC_vect “ist der Interrupt, der ausgelöst wird, wenn eine Messung beendet ist. Die Referenzspannung des ADC's wurde auf $U_{ref} = 1.1V$ gesetzt. Alternativ können noch $U_{ref} = 3.3V$ (Versorgungsspannung) oder $U_{ref} = A_{ref}$ gewählt werden. A_{ref} ist die Spannung, die an dem Referenz-Pin des Mikrocontrollers angelegt werden kann.

Für den Analog-Digital-Wandler muss ein Vorteiler gewählt werden. Dieser Wert bestimmt die Frequenz des ADC's.

$$f_{adc} = \frac{f_{takt}}{prescale} = \frac{8MHz}{4} = 2MHz$$

Da eine Messung 13 Taktzyklen dauert, erhält man eine Abtastfrequenz von

$$f_{sample} = \frac{f_{adc}}{13} = 153846Hz$$

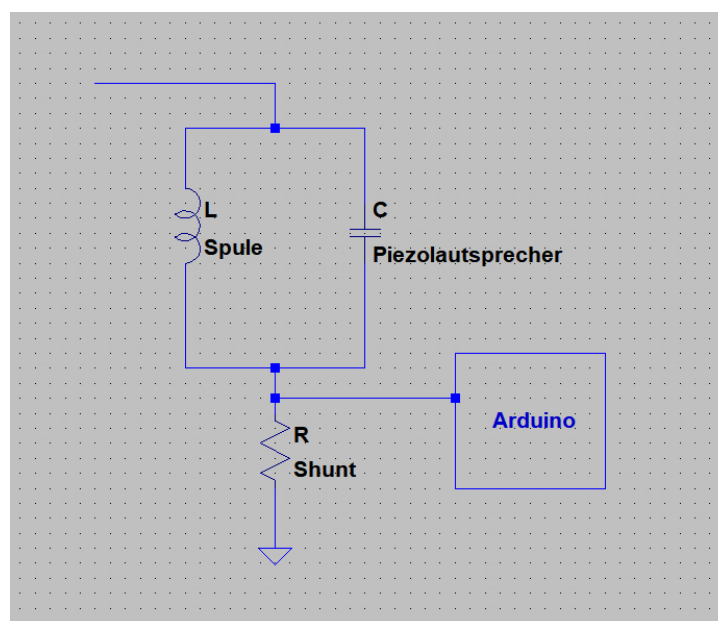


Abbildung 4.18: Schaltung der Spannungsmessung

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

4 Der zweite Prototyp

Die Abtastfrequenz eines Signales sollte nach dem WKS-Abtasttheorem (für Whittaker, Kotelnikow und Shannon) mindestens doppelt so groß sein, wie die maximale Frequenz des abzutastenden Signals.[10] Da das Gerät Spannungen mit Frequenzen zwischen 15 - 20 kHz erzeugt, ist diese Abtastfrequenz vom Analog-Digital-Umwandler ausreichend.

4.8.2 Programmcode - Modell 1

Es wurden zwei Programmcodes zur Findung der Resonanzfrequenz geschrieben. Bei der ersten Methode wurde die analogRead()-Funktion verwendet. Der ADC misst 250 Mal

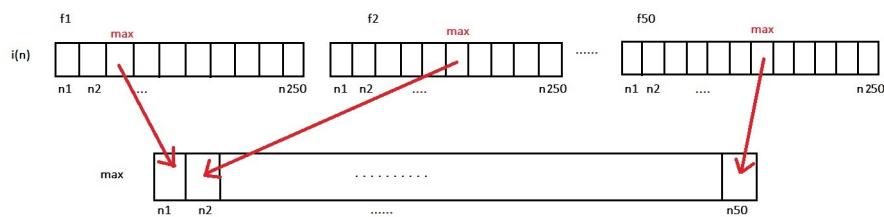


Abbildung 4.19: Schema der Spannungsmessung

pro Frequenz die Spannung. Diese Werte des ADC's werden in Spannung und Strom berechnet und in einem Array gespeichert. Wenn der "counter 2 "bei 250 angekommen ist, wird er wieder auf 0 gesetzt, der Maximalwert der Messung in einem Array gespeichert, die Frequenz um 100 Hz erhöht und nochmals gemessen. Dies geschieht so lange bis "counter 1 "bei dem Wert 50 angekommen ist und die Frequenz somit bei 20 kHz ist.

```
while (counter1 < 50) {
    stromTemp = 0;
    counter2 = 0;

    while (counter2 < 250) {

        shuntValue = analogRead(A2);
        float shuntVoltage = shuntValue * FAKTOR;
        strom = (shuntVoltage / MESSWIDERSTAND) * 1000;
        //strom = shuntVoltage*83.333333333;
        if (0.0 < strom > stromTemp) stromTemp = strom;

        counter2++;

    } //end while 2

    maxArray[counter1] = stromTemp;

    frequency += 100;
```

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

4 Der zweite Prototyp

```
freqReg = Milbenschreck.freqCalc(frequency);
Milbenschreck.adjustFreq(MiniGen::FREQ0, freqReg);
Milbenschreck.selectFreqReg(MiniGen::FREQ0);
counter1++;

} //end while 1
```

Im Anschluss wird das Array mit den Maximalwerten nach dem geringsten Wert durchsucht

```
float Min = maxArray[0];
for (int i = 1; i < 50; i++) {
if (maxArray[i] < Min) {
Min = maxArray[i];
}
}
```

und die passende Frequenz eingestellt.

```
for (i = 0 ; i < 50; i++) {

    if (maxArray[i] == Min) {

        frequency = (i * 100) + 15000;

    }

}
```

Trotz der langen Dauer der analogRead()-Funktion liegt die eingestellte Frequenz nah an der gemessenen Resonanzfrequenz. Die Abweichung beträgt maximal $\Delta f = \pm 600 \text{ Hz}$. Dieses Ergebnis ist annehmbar aber nicht ideal. Deswegen wurde ein weiterer Programmcode geschrieben, der öfter und schneller messen kann. Es ergibt sich folgender Programmablaufplan.

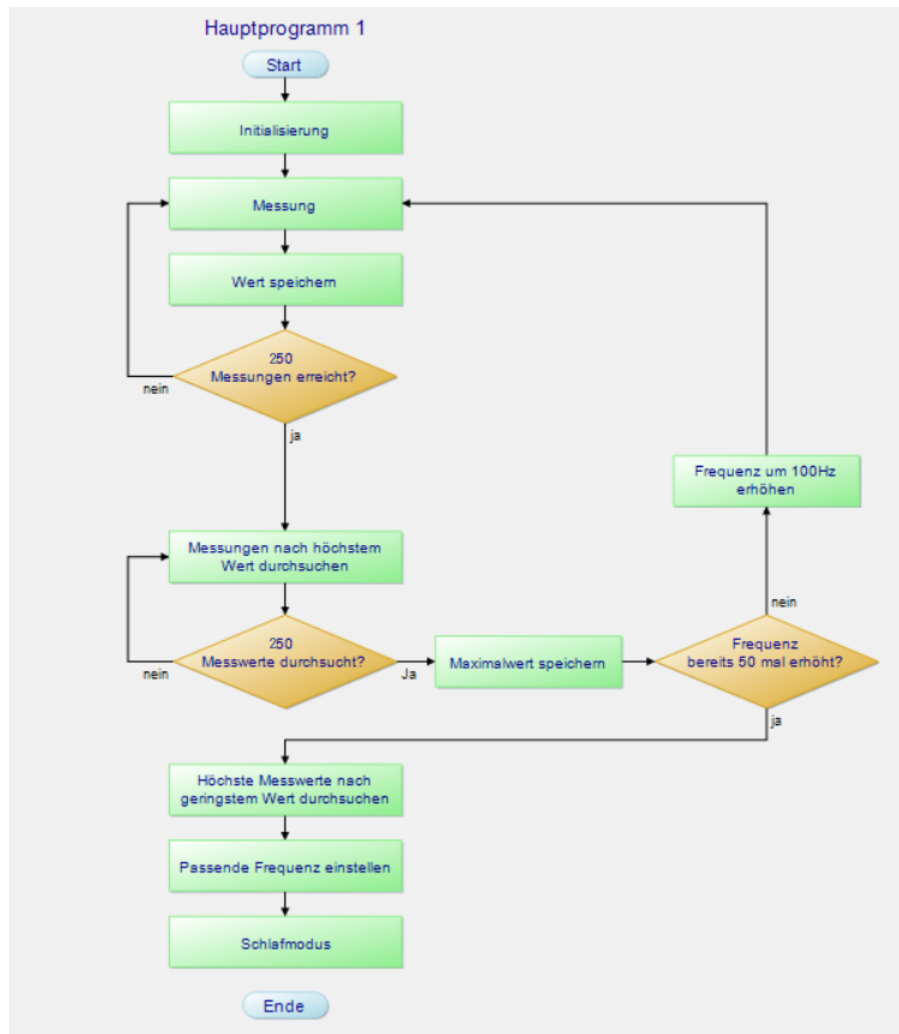


Abbildung 4.20: Programmablaufplan Code 1

Der vollständige Code ist im Anhang unter der Überschrift "Programmcode 1 Prototyp 2" zu finden.

4.8.3 Programmcode-Modell 2

Der zweite Programmcode beginnt ebenfalls mit der Initialisierung des MiniGen-Moduls und beginnt mit einer Frequenz von 15 kHz. Der Analog-Digital-Wandler wird in den "Free-Running-Mode" gesetzt. Ist eine Messung abgeschlossen wird folgender Interrupt ausgelöst:

```
ISR(ADC_vect)
{
  ADC_val = ADCL;
  ADC_val = (ADCH << 8) + ADC_val;

  newReading = true;
  counter2++;
}
```

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

4 Der zweite Prototyp

Die Messung hat folgendes Schema:

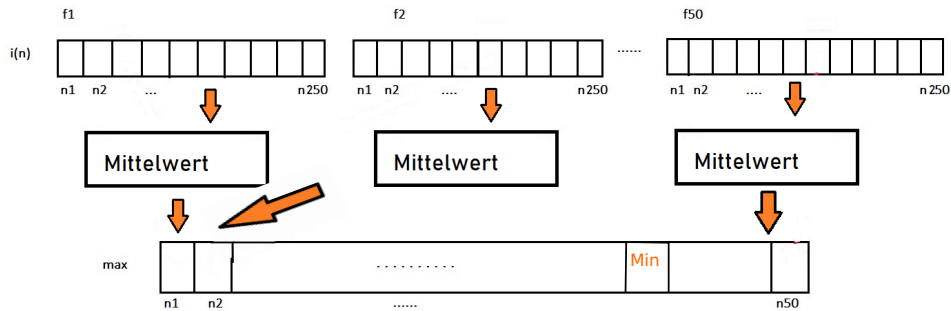


Abbildung 4.21: Schema der Messung Code 2

Der Wert des ADC's wird in die Variable "ADC_val "geschrieben und eine Zählvariable erhöht. Zudem signalisiert eine weitere Variable, dass eine Messung fertig ist. Aus 250 Messungen wird der Mittelwert gebildet, dieser Wert in einem Array gespeichert, die Frequenz um 100 Hz erhöht und weitere 250 mal gemessen. Dies geschieht bis 20 kHz erreicht sind. Im Anschluss wird genau wie beim Programmcode 1 der geringste Wert aus den Mittelwerten gesucht und die passende Frequenz eingestellt.

Neben dem ADC- Interrupt besitzt der Code einen Watchdog-Timer-Interrupt.

```
ISR(WDT_vect) {  
    }  
}
```

Dieser Interrupt wurde so eingestellt, dass er alle 2 Sekunden auslöst.

```
void watchdogSetup()  
{  
    cli(); // disable all interrupts  
    wdt_reset(); // reset the WDT timer  
    MCUSR &= ~(1 << WDRF);  
  
    // Enter Watchdog Configuration mode:  
    WDTCR |= (1 << WDCE) | (1 << WDE);  
    // Set Watchdog settings: interrupte enable, 0111 for timer -> 2Sek  
    WDTCR = (1 << WDIE) | (0 << WDP3) | (1 << WDP2) | (1 << WDP1) | (1 << WDP0);  
    sei();  
    Serial.println("finished watchdog setup");  
}
```

Der Watchdog-Interrupt soll den Mikrocontroller aus seinem Schlafmodus wecken und somit die Ausschaltzeiten der verschiedenen Betriebsstufen kontrollieren. Bevor der Atmega328 in

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

4 Der zweite Prototyp

den Schlafmodus geht, wird der DDS-Generator-Chip ebenfalls in einen Schlafmodus geschickt. Dazu müssen über das SPI Daten an den Chip gesendet werden.

```
void SPIWrite(uint16_t data) {
    digitalWrite(_FSYNCPin, LOW);
    SPI.transfer((byte)(data >> 8));
    SPI.transfer((byte)(data));
    digitalWrite(_FSYNCPin, HIGH);
}
```

Zum Schreiben von Daten muss der FSYNC Pin auf "Low" gesetzt werden und die 16-Bit Daten auf jeweils 8-Bit aufgeteilt werden. Nach dem Transfer der Daten muss der FSYNC-Pin wieder auf den Status "High" gesetzt werden. Um den DDS-Chip in den Schlafmodus zu schicken ist dann folgender Code nötig.

```
void Minigen_Sleep() {

    SPIWrite(0x00C0); //Sleepbit1 & Sleepbit12 im control register
}
```

Es können so 4 mA gespart werden.

Ein weiterer externer Interrupt stellt die Betriebsstufen des Prototyps ein. Der Programmablaufplan ist im Anhang unter der Überschrift "Programmablaufplan Code 2" zu finden.

4.8.4 Stufeneinstellung

Das Ultraschallgerät soll über mehrere einstellbare Betriebsstufen verfügen. Diese Stufen sollen die Abstrahldauer des Tones verändern. Während kein Ton erzeugt wird, können der DDS-Generator und der Mikrocontroller in einen Schlafmodus versetzt werden. Es kann somit Strom gespart werden. Zur Anzeige der aktuellen Stufe dienen drei Leuchtdioden.

4.8.4.1 Hardware

Die Leuchtdioden sind jeweils über einem Vorwiderstand mit einem Pin des Mikrocontrollers verbunden. Diese Pins wurden als digitale Ausgänge deklariert. Wird ein Pin auf "High" geschaltet, stehen am Pin $U = 3.3V$ zur Verfügung.

Die gewählten LED's "Kingbright THT" in der Farbe grün haben eine Durchlassspannung von $V_F = 2.2V$. Die restliche Spannung soll an einem Vorwiderstand abfallen. Für die LED's ist ein Strom von ca. $I = 3mA$ vorgesehen. Nach dem ohmschen Gesetz folgt

$$R = \frac{U}{I} = \frac{1.1V}{3mA} = 366.66\Omega$$

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

4 Der zweite Prototyp

Es wurde ein Vorwiderstand von $R_V = 460\Omega$ gewählt. Es fließt ein Strom von

$$I = \frac{U}{R} = \frac{1.1V}{460\Omega} = 2.39mA$$

Ein weiterer Pin des Mikrocontrollers ist als Analogeingang definiert. Zudem wurde der Pull-Up-Widerstand an diesem Pin verwendet. Für einen externen Schalter wurden zwei Pads eingefügt. Wird der externe Schalter betätigt, wird der Pin auf Masse gezogen und erhält den Zustand "Low ". Ein Kondensator dient zum Entprellen des Schalters.

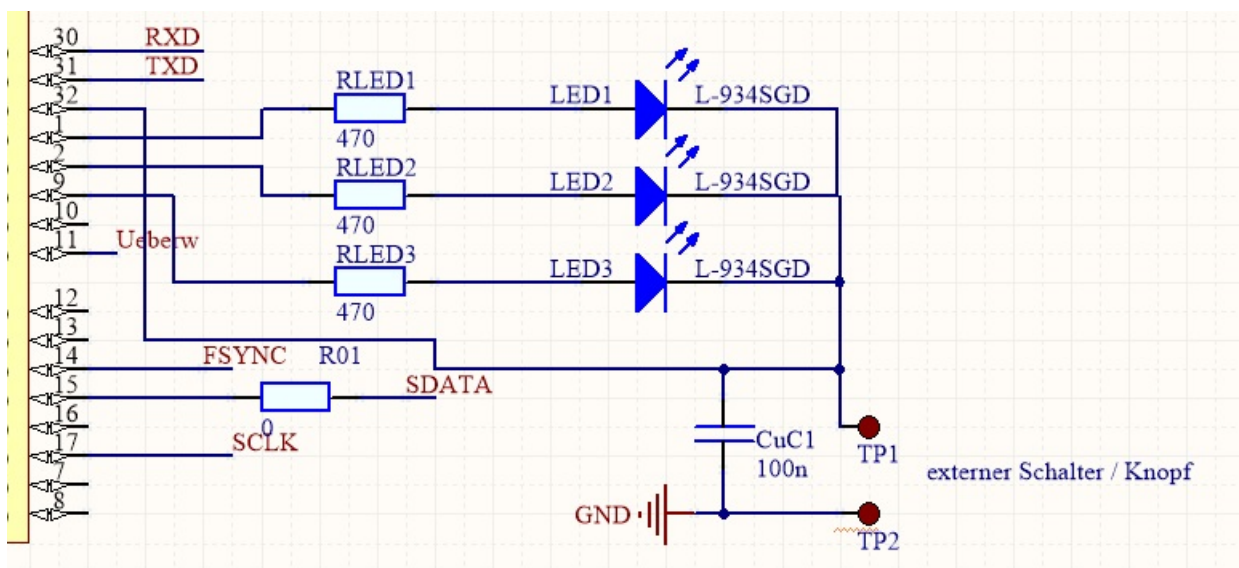


Abbildung 4.22: Schaltung der LED's zur Anzeige der Betriebsstufen

Pull-Up und Pull-Down-Widerstände werden verwendet, um einem Pin mit einem Schalter einen definierten Zustand zu geben. Ein Eingang, der weder mit der Betriebsspannung, noch mit Masse verbunden ist, reagiert sensibel auf äußere Einflüsse. Schon die Berührung mit der Hand kann ein fehlerhaftes Signal erzeugen. Bei der Nutzung eines Pull-Up-Widerstands hat der Eingangspin den Mikrocontrollers den definierten Zustand "High ". Bei der Betätigung eines Tasters, der mit Masse verbunden ist, erhält der Eingang den Wert "Low ".

4.8.4.2 Software

Der Pin des Mikrocontroller, welcher mit dem Schalter verbunden ist benötigt einen externen Interrupt.

```
attachInterrupt(digitalPinToInterrupt(2), Interruptfunktion, FALLING);
```

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

4 Der zweite Prototyp

Der Interrupt wird bei einer fallenden Flanke des Signals am Pin abgelöst. Dies geschieht, wenn der Schalter betätigt wird und der Pin somit auf Masse gezogen wird.

Direkt nach der Initialisierung des DDS-Chips und der Frequenzfindung wird der Status des Gerätes auf den Wert "0" gesetzt. Insgesamt besitzt es 4 einstellbare Stufen.

```
if (Status == 0) {
Serial.println ("Status:0");

digitalWrite(LED1, LOW);
digitalWrite(LED2, LOW);
digitalWrite(LED3, LOW);

delay(50);
}
```

Dieser Code für die verschiedenen Stufen ist in einer sich ständig wiederholenden Funktion.

```
void loop () {
...}
```

Bei jeder Wiederholung springt der Mikrocontroller mithilfe von if-Anweisungen in den richtigen Code für die jeweiligen Stufen.

Ist der Status der Stufe gleich 0 soll die Frequenz vom DDS-Chip dauerhaft ausgegeben werden. Das bedeutet, dass er nicht in einen stromsparenden Schlafmodus versetzt werden kann. Wurde der Taster betätigt, erhöht sich die Stufe um 1. Hat die Stufe den Status 1, wird eine der drei LED's an geschaltet. Da der Schalter mit Masse verbunden ist, leuchte die LED nur so lange, wie der Schalter betätigt wird. Diese Verschaltung spart zusätzlich Strom.

```
if (Status == 1) {

//Serial.println ("Status 1");

digitalWrite(LED1, HIGH);
digitalWrite(LED2, LOW);
digitalWrite(LED3, LOW);

freqReg = Milbenschreck.freqCalc(frequency);
Milbenschreck.adjustFreq(MiniGen::FREQ0, freqReg);
Milbenschreck.selectFreqReg(MiniGen::FREQ0);

delay(2000);

Minigen_Sleep();
}
```


Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

4 Der zweite Prototyp

```
wdt_reset();

sleep();
}
```

Im Status 1 wird begonnen die vorher gefundene Resonanzfrequenz auszustrahlen. Nach einer Verzögerung von 2000 ms wird der DDS-Generator in den Schlafmodus geschickt und stoppt das Erzeugen der Frequenz. Der Zähler des Watchdog-Timers wird erneut gestartet. Ab diesem Zeitpunkt wird der Interrupt des Watchdog-Timers in 2 Sekunden ausgelöst. Direkt im Anschluss wird der Atmega328 in den Schlafmodus geschickt und wird 2 Sekunden schlafen. In der zweiten Stufe werden die LED's 1 und 2 an geschaltet. Die Zählzeit des Watchdog-Timers wird früher als bei der Stufe 1 neu gestartet und die Frequenz ausgegeben. Nach einer Verzögerung von 900 ms werden DDS-Chip und anschließend der Mikrocontroller in den Schlafmodus versetzt.

```
if (Status == 2 ) {

delay(10);
digitalWrite(LED1, HIGH);
digitalWrite(LED2, HIGH);
digitalWrite(LED3, LOW);

freqReg = Milbenschreck.freqCalc(frequency);
Milbenschreck.adjustFreq(MiniGen::FREQ0, freqReg);
Milbenschreck.selectFreqReg(MiniGen::FREQ0);
wdt_reset();

delay(900);

Minigen_Sleep();

sleep();

}
```

In der dritten Stufe wird der Watchdog-Timer noch vor dem an schalten der drei LED's zurück gesetzt. Die gefundene Frequenz wird erzeugt und nach deiner Verzögerung von 380 ms werden DDS-Chip und ATmega328 in den Schlafmodus geschickt.

```
if (Status == 3 ) {
```

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

4 Der zweite Prototyp

```
wdt_reset();

digitalWrite(LED1, HIGH);
digitalWrite(LED2, HIGH);
digitalWrite(LED3, HIGH);

freqReg = Milbenschreck.freqCalc(frequency);
Milbenschreck.adjustFreq(MiniGen::FREQ0, freqReg);
Milbenschreck.selectFreqReg(MiniGen::FREQ0);

delay(380);

Minigen_Sleep();

sleep();

}
```

Aufgrund der unterschiedlich langen Verzögerungen und der verschiedenen Positionierung des Zurücksetzens Watchdog-Timers werden drei Betriebsstufen erzeugt, die mit einer unterschiedlichen zeitlichen Länge die Resonanzfrequenz ausstrahlen.

Ist der Status der Stufe größer als 3, wird die Stufe wieder auf den Status 0 gesetzt.

```
if (Status > 3) Status = 0;
```

5 Das Gehäuse

Das Gehäuse für das Ultraschallgerät sollte möglichst flach sein, um es im Bienenstock unter die Bienenwaben zu legen. Der Schall soll von unten mittig auf die Bienenwaben abgestrahlt werden. Zusätzlich muss das Gehäuse robust sein und die Platine vor äußeren Einflüssen schützen. Es darf den Bienen nicht möglich sein in das Gehäuse zu gelangen.

Es werden mehrere Bohrungen benötigt. Drei LED's zum Anzeigen der aktuellen Betriebsstufe sollen von außen sichtbar sein. Zudem soll ein Schalter zum Ein- und Ausschalten, ein Knopf zum Einstellen der Betriebsstufen und zwei Kabel für ein externes Steuergerät von außen erreichbar sein. Da die Piezolausprecher direkt an den Deckel des Gehäuses befestigt werden sollen, wird ein Metallgehäuse verwendet. Die Schwingung der Piezolausprecher soll bestmöglich an das Gehäuse übertragen werden.

Da der "Bottom-Layer" - also die Unterseite der Platine - eine Masseverbindung hat, muss die Platine von dem Gehäuse isoliert sein.

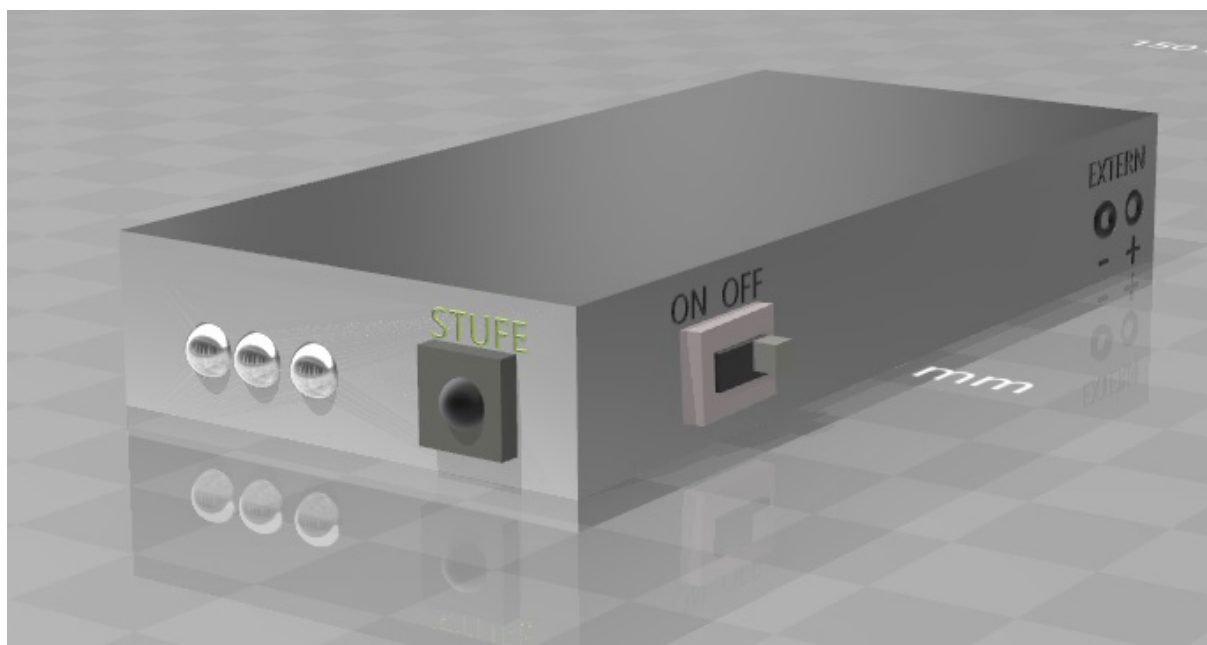


Abbildung 5.1: CAD-Modell des Gehäuses

6 Vorbereitung einer Kleinserienfertigung

Für die Vorbereitung einer Kleinserienfertigung müssen alle für das Gerät relevanten Dateien vorhanden und vollständig sein. Dazu gehört das mit Altium Designer entwickelte Printed-Circuit-Board (PCB)-Design und die ebenfalls mit Altium Designer erzeugten Dateien zum Fräsen der Platine. Zudem muss eine Bills-of-Material (BOM)-Liste vorhanden sein. Auf dieser Materialliste stehe alle verwendeten Bauteile, mit Informationen über Preis, Menge, Werte und verwendeten Footprints. Auch geeignete Lieferanten können aufgeführt sein. Mithilfe der Liste kann eine einfache Kostenkalkulation der Materialkosten erfolgen und problemlos erneut die richtigen Bauteile bestellt werden.

Da es sich bei dem Ultraschallgerät um eine Einzel-/ bzw. Kleinserienfertigung handelt und das Gerät außerhalb einer Firma entwickelt wurde, wurden die Bauteile zu üblichen Preisen im Einzelhandel erstanden. Ebenfalls wurden, außer bei SMD Widerständen und Kondensatoren, keine Mengenrabatte genutzt.

Die Gesamtmaterialekosten für das Ultraschallgerät belaufen sich auf etwa 17,50 Euro/Stück. Zuzüglich dazu müssen die Kosten für eine Platine addiert werden.

Um das Gerät zukünftig günstiger produzieren zu lassen, könnten Bauteile vom asiatischen Markt mit Mengenrabatten erworben werden. Zudem wäre es denkbar die Platinen dort fertigen zu lassen.

Eine vollständige BOM-Liste des entwickelten Ultraschallgerätes ist im Anhang zu finden.

Literaturverzeichnis

- [1] Rudolf Moosbeckhofer , Hemma Köglberger, u.A . *Varroa-Bekämpfung ; einfach - sicher - erfolgreich*. Biene Österreich. 2.Auflage, 2015.
- [2] Jan Grossarth. Das Bienendrama. *Frankfurter Allgemeine Zeitung*. <http://www.faz.net/aktuell/wirtschaft/warum-die-welt-ohne-bienen-ein-befremdlicher-ort-waere-15529920.html>, Letzter Aufruf: 03.09.2018.
- [3] Johannes Janus. Der bedeutsamste Bienenschädling – die Varroamilbe. <http://varroamilbe.de/>, Letzter Aufruf 03.09.2018.
- [4] Stephen Zechendorf. Mit 15.000 Hertz gegen Völkersterben. <https://www.volksstimme.de/lokal/burg/bienen-mit-15000-hertz-gegen-voelkersterben>, 2017. Letzter Aufruf 04.09.2018.
- [5] Wolfgang Zell. Varroa-Killer-Sound. <https://bienenhirte.de/blog/varroa-killer-sound/#lightbox/2/>, 2017. Letzter Aufruf 04.09.2018.
- [6] piezosystem Jena GmbH. Piezofibel - Wissenswertes zur Piezomechanik. <https://www.piezosystem.de/piezopedia/piezothorie/#c10697>, Letzter Aufruf 13.09.2018.
- [7] Barsch/Klebs. Bienen-Parasit : Forscher entdecken Wirkstoff gegen Vorroa-Milbe
https://www.uni-hohenheim.de/pressemitteilung?tx_ttnews%5Btt_news%5D=38676&cHash=6eec546b4790d34639e64eb7e1ce0463 , 2018. Letzter Aufruf 14.09.2018.
- [8] Parallelresonanz. <https://de.wikipedia.org/wiki/Parallelresonanz> , Letzter Aufruf 14.09.2018
- [9] Frank Ohnhäuser. Analog-Digital Converters for Industrial Applications Including an Introduction to Digital-Analog Converters. Springer Verlag 2015. S. 14 ff
- [10] Nyquist-Shannon-Abtasttheorem <https://de.wikipedia.org/wiki/Nyquist-Shannon-Abtasttheorem>. Letzter Aufruf 21.09.2018
- [11] MiniGen Hookup Guide. SparkFun Electronics ®.
<https://learn.sparkfun.com/tutorials/minigen-hookup-guide> . Letzter Aufruf 24.09.2018

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

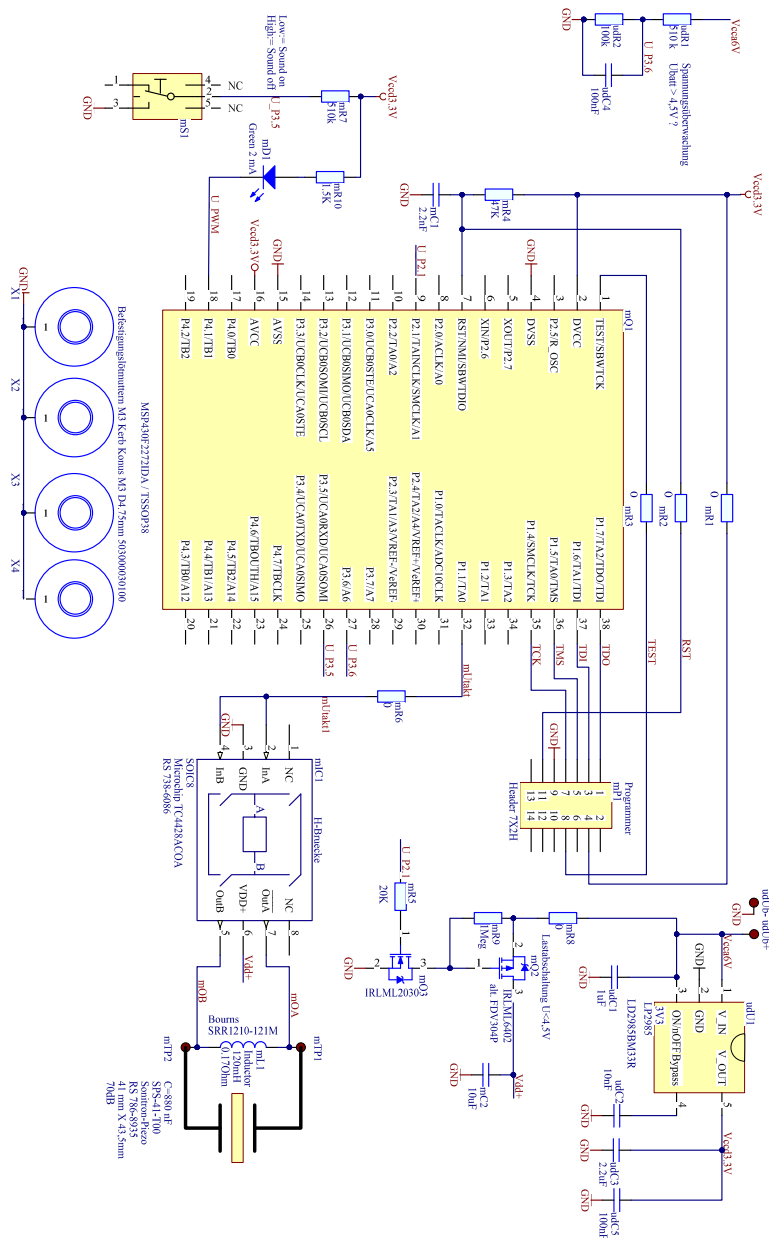
Literaturverzeichnis

- [12] Altium Designer https://de.wikipedia.org/wiki/Altium_Designer . Letzter Aufruf
04.10.2018
- [13] Gütefaktor <https://de.wikipedia.org/wiki/Gütefaktor> Letzter Aufruf 30.10.2018

Anhang

Schaltplan Prototyp 1

Schaltplan des ersten Prototyps



Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

Anhang

Programmcode Prototyp 1

```
#include <msp430.h>
// #define cycles 62500u          /* 500ms - Number for frequency adjustment*/
#define cycles 32u                /* TimerA 66us (15 kHz) - Number for frequency ↔
    adjustment*/
// "u" or "U" is short format for "unsigned",
#define period 1000u             /* TimerB PWM period 1 kHz*/
#define duty 200u
#pragma INTERRUPT (ISR_TimerA)
#pragma vector = TIMERA0_VECTOR
void ISR_TimerA(void)
{
    P1OUT ^= BIT1;              /* toggle P1.1 H-Bridge */
}

#pragma INTERRUPT (ISR_TimerB)
#pragma vector = TIMERB0_VECTOR
void ISR_TimerB(void)
{
    P4OUT ^= BIT0;              // toggle P4.0 LED_PWM
}

int main(void) {
    WDTCTL = WDTPW | WDTHOLD;    // Stop watchdog timer
    DCOCTL = CALDCO_8MHZ;        // MCLK 8MHz
    BCSCTL1 = CALBC1_8MHZ;       // Basistakt 8Mhz = 0,125 ns pro cycle
    BCSCTL2 |= DIVS_0;

    // define Port directions

    P1DIR |= BIT1;               // Set P1.1 to output direction H-Bridge clock
    P2DIR |= BIT1;
    P4DIR |= BIT0;
    P4SEL |= BIT0;
    P2OUT |= BIT1;
    // P3DIR |= BIT5;
    P3DIR &= ~BIT5;

    // init Timer A in capture compare with variable "cycles"
    TACTL |= TASSEL_2 + ID_3;    /* SMCLK (TASSEL_2) , 8x divider (ID_3) */
    TACCR0 = cycles;            /* Timer stops after N+1 cycles in periode-Register*/
    TACCTLO |= CCIE;            /* enable TimerA interrupt-flag when "cycles" counted */
    TACTL |= MC_1;

    // TBCCTL0 = OUTMOD_7 + CLLD_2 + TBIE; // OUTMOD_3 (ReSet (TBCCR1) / set (TBCCR0)),
```


Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

Anhang

```
// CLLD_2 (up to TBCL0,TBCLx loads when TBR counts to 0 (up or continuous mode)
//TBIE (Timer B interrupt enable)
TBCCR0= period ;

TBCCTL1 = OUTMOD_3; //OUTMOD_3 (Set (TBCCR1) /reset (TBCCR0)), ct11
TBCCR1 = duty; // duty 100u close to 10% low active duty cycle

TBCTL = TBSSEL_2 + TBCLR + ID_3 + MC_2; //ID_3;//TBSSEL_2 (SMCLK), ID_3 (Teiler 8)
/*
//{/
TBCTL = MC_2; // Stop the timer
TBCTL = TASSEL_2 + TBCLR; // ACLK, Clear Tar
TBCCR0 = period; // PWM Period
TBCCTL2 = OUTMOD_7; // CCR2 reset/set
TBCCR2 = duty; // CCR2 PWM duty cycle
P4DIR |= 0x01; // P4 output
P4SEL |= 0x01; // P4 option
TBCTL |= MC_2; // Start Timer_B in up mode
*/

_EINT(); // enable interrupt */

//_BIS_SR(LPM0_bits + GIE); // Low Power Mode on, enables interrupts

while(1){

__delay_cycles(20000);
// for(int i = 0; i <= 500000; i++);
//char state = P3IN.5;
if((P3IN & BIT5) == BIT5)
{
P2OUT |= BIT1;

}
else
{
P2OUT &= ~BIT1;

}

//P4OUT |= BIT0;
//P1OUT |= BIT1; // Last LED ein - Betrieb
//P1OUT &= ~BIT2; // Rote LED aus - Warnung
__delay_cycles(20000);
//P4OUT &= ~BIT0; // Rote LED aus - Warnung
//P1OUT &= ~BIT1; // Rote LED aus - Warnung
//P2OUT |= BIT2;

}

return 0;
}
```

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

Anhang

Programmcode 1 Prototyp 2

```
#include "SparkFun_MiniGen.h";
#include "avr/sleep.h"
#include "SPI.h";

#define FAKTOR 0.003222 //3.3/1024
#define MESSWIDERSTAND 12

MiniGen Milbenschreck;
int frequency = 15000;
float stromTemp = 0;
int shuntValue;
float strom = 0;
unsigned long freqReg;
float maxArray [51];
int i;
float min;
//void MiniGenInit(void);

void setup() {

Serial.begin(115200);

analogReference(DEFAULT);
pinMode(10, OUTPUT); //CS Pin -> an FSYNC
pinMode(11, OUTPUT); //MOSI Pin -> an SDATA
pinMode(13, OUTPUT); //SCK, CLK Pin -> an SCLK
pinMode(A2, INPUT);

MiniGenInit();
delay(2000);

byte counter1 = 0;
int counter2 = 0;

while (counter1 < 50) {
stromTemp = 0;
counter2 = 0;

while (counter2 < 250) {

shuntValue = analogRead(A2);
float shuntVoltage = shuntValue * FAKTOR;
strom = (shuntVoltage / MESSWIDERSTAND) * 1000;
//strom = shuntVoltage*83.333333333;
if (0.0 < strom > stromTemp) stromTemp = strom;

counter2++;

//delayMicroseconds(2);
} //end while 2

maxArray[counter1] = stromTemp;
```

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

Anhang

```
frequency += 100;
freqReg = Milbenschreck.freqCalc(frequency);
Milbenschreck.adjustFreq(MiniGen::FREQ0, freqReg);
Milbenschreck.selectFreqReg(MiniGen::FREQ0);
counter1++;

} // end while 1

float min = maxArray[0];
for (int i = 1; i < 50; i++) {
if (maxArray[i] < min) {
min = maxArray[i];
}
}
//Serial.println(min);

for (i = 0 ; i < 50; i++) {
Serial.println(maxArray[i]);
if (maxArray[i] == min) {

frequency = (i * 100) + 15000;
}
}
Serial.print(frequency);

freqReg = Milbenschreck.freqCalc(frequency);
Milbenschreck.adjustFreq(MiniGen::FREQ0, freqReg);
Milbenschreck.selectFreqReg(MiniGen::FREQ0);
delay(200);
sleep();

} //end setup
void loop() {

} // end loop

#####
#####

void MiniGenInit() {

Milbenschreck.reset();
```

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

Anhang

```
Milbenschreck.setMode(MiniGen::SINE);
Milbenschreck.setFreqAdjustMode(MiniGen::FULL); ←
//FULL (each count *0.0596Hz) oder ←
COARSE(each count +976.5Hz) oder FINE(each count +0.0596Hz)

freqReg = Milbenschreck.freqCalc(frequency);
Milbenschreck.adjustFreq(MiniGen::FREQ0, freqReg);
Milbenschreck.selectFreqReg(MiniGen::FREQ0);
}

#####
#####

void GibGummi (int frequency) {
freqReg = Milbenschreck.freqCalc(frequency);
Milbenschreck.adjustFreq(MiniGen::FREQ0, freqReg);
Milbenschreck.selectFreqReg(MiniGen::FREQ0);
}

#####
#####

void sleep(void)
{
set_sleep_mode(SLEEP_MODE_IDLE);
sleep_enable();
sleep_mode();
}

#####
#####
```

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

Anhang

Programmcode 2 Prototyp 2

```
#include "SparkFun_MiniGen.h";
#include "avr/sleep.h"
#include "SPI.h";

#include "avr/wdt.h";

#define LED1 3
#define LED2 4
#define LED3 5
#define Knopf 2
#define _FSYNCPin 10

#ifndef cbi
#define cbi(sfr, bit) (_SFR_BYTE(sfr) &= ~_BV(bit))
#endif
#ifndef sbi
#define sbi(sfr, bit) (_SFR_BYTE(sfr) |= _BV(bit))
#endif

MiniGen Milbenscheck;

int volatile Status = 0;
int volatile druck;
int frequency = 15000;
float tempVal = 0;
int shuntValue[250];
unsigned long freqReg;
float maxArray [51];
int i;
float Min;

bool volatile newReading = false;
int volatile ADC_val;
byte volatile counter2 = 0;

ISR(ADC_vect)
{
    ADC_val = ADCL;
    ADC_val = (ADCH << 8) + ADC_val;

    newReading = true;
    counter2++;
}

ISR(WDT_vect) {

}

void setup() {
    Serial.begin(115200);
```

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

Anhang

```
pinMode(LED1, OUTPUT);
pinMode(LED2, OUTPUT);
pinMode(LED3, OUTPUT);
pinMode(Knopf, INPUT_PULLUP);
pinMode(10, OUTPUT);
pinMode(11, OUTPUT);
pinMode(13, OUTPUT);
pinMode(A0, INPUT);

attachInterrupt(digitalPinToInterrupt(2), Interruptfunktion, FALLING);

ADMUX |= 0x00;

ADMUX |= (1 << REFS1) | (1 << REFS0); //1.1V Referenzspannung

cbi(ADMUX, ADLAR);
ADCSRA |= 0x02;

// enable the Interrupt
sbi(ADCSRA, ADIE);

// free running mode
ADCSRB &= 0xF8;

// enable the Auto Trigger
sbi(ADCSRA, ADATE);

// enable the ADC
sbi(ADCSRA, ADEN);

sbi(ADCSRA, ADSC);

Milbenschreck.reset();
Milbenschreck.setMode(MiniGen::SINE);
Milbenschreck.setFreqAdjustMode(MiniGen::FULL);

freqReg = Milbenschreck.freqCalc(frequency);
Milbenschreck.adjustFreq(MiniGen::FREQ0, freqReg);
Milbenschreck.selectFreqReg(MiniGen::FREQ0); /*Freq-Register waehlen*/

delay(2000);

byte counter1 = 0;
counter2 = 0;

while (counter1 < 50) {
    tempVal = 0;
    counter2 = 0;
    //shuntValue-Array beschreiben
    while (counter2 < 250) {
        if (newReading = true) {

            shuntValue[counter2] = ADC_val;
        }
    }
}
```

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

Anhang

```
        counter2++;
        newReading = false;

    } //end while 2
    //max. Wert aus Array finden
    for (int i = 0; i < 250; i++) {

        //      if ( shuntValue[i] > tempVal) {
            tempVal += shuntValue[i];
        }
        tempVal = tempVal * 0.004; // /250
        maxArray[counter1] = tempVal;
        //      }
    } //}

    /*Frequenz erhoehen*/
    frequency += 100;
    freqReg = Milbenschreck.freqCalc(frequency);
    Milbenschreck.adjustFreq(MiniGen::FREQ0, freqReg);
    Milbenschreck.selectFreqReg(MiniGen::FREQ0);

    counter1++;

} // end while 1

float Min = maxArray[0];
for (int i = 1; i < 50; i++) {
    if (maxArray[i] < Min) {
        Min = maxArray[i];
    }
}

for (i = 0 ; i < 50; i++) {

    if (maxArray[i] == Min) {

        frequency = (i * 100) + 15000;

    }
}

freqReg = Milbenschreck.freqCalc(frequency);
Milbenschreck.adjustFreq(MiniGen::FREQ0, freqReg);
Milbenschreck.selectFreqReg(MiniGen::FREQ0);

delay(300);
Serial.println("MiniGen");

cbi(ADCSRA, ADEN); //ADC off

watchdogSetup();
```

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

Anhang

```
}

void loop() {

if (Status == 1) {

    //Serial.println ("Status 1");

    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, LOW);

    freqReg = Milbenschreck.freqCalc(frequency);
    Milbenschreck.adjustFreq(MiniGen::FREQ0, freqReg);
    Milbenschreck.selectFreqReg(MiniGen::FREQ0);

    delay(2000);

    Minigen_Sleep();

    wdt_reset();

    sleep();

}
if (Status == 2 ) {

    // Serial.println ("Status:2");
    //wdt_reset();
    delay(10);
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, LOW);

    freqReg = Milbenschreck.freqCalc(frequency);
    Milbenschreck.adjustFreq(MiniGen::FREQ0, freqReg);
```


Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

Anhang

```
Milbenschreck.selectFreqReg(MiniGen::FREQ0);
wdt_reset();

delay(900);

Minigen_Sleep();

sleep();
// }
}
if (Status == 3 ) {

//Serial.println ("Status:3");
wdt_reset();

digitalWrite(LED1, HIGH);
digitalWrite(LED2, HIGH);
digitalWrite(LED3, HIGH);

//delay(20);

// while (Status == 3) {

    freqReg = Milbenschreck.freqCalc(frequency);
    Milbenschreck.adjustFreq(MiniGen::FREQ0, freqReg);
    Milbenschreck.selectFreqReg(MiniGen::FREQ0);

    delay(380);

    Minigen_Sleep();

    sleep();
    //delay(200);
    // }

}
if (Status == 0) {
Serial.println ("Status:0");

digitalWrite(LED1, LOW);
digitalWrite(LED2, LOW);
digitalWrite(LED3, LOW);

delay(50);

// sleep();

}

} //end loop

#####
#####
```

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

Anhang

```
void Interruptfunktion() {

Status++;

if (Status > 3) Status = 0;

}

void MiniGenInit() {

Milbenschreck.reset();
Milbenschreck.setMode(MiniGen::SINE);
Milbenschreck.setFreqAdjustMode(MiniGen::FULL);

freqReg = Milbenschreck.freqCalc(frequency);
Milbenschreck.adjustFreq(MiniGen::FREQ0, freqReg);
Milbenschreck.selectFreqReg(MiniGen::FREQ0);
}

#####
#####

void GibGummi (int frequency) {
freqReg = Milbenschreck.freqCalc(frequency);
Milbenschreck.adjustFreq(MiniGen::FREQ0, freqReg);
Milbenschreck.selectFreqReg(MiniGen::FREQ0);
}

#####
#####

void sleep(void)
{

set_sleep_mode(SLEEP_MODE_PWR_DOWN);
cli();
sleep_enable();
sei();
sleep_cpu();
sleep_disable();

Milbenschreck.reset();
Milbenschreck.setMode(MiniGen::SINE);
Milbenschreck.setFreqAdjustMode(MiniGen::FULL);

freqReg = Milbenschreck.freqCalc(frequency);
Milbenschreck.adjustFreq(MiniGen::FREQ0, freqReg);
Milbenschreck.selectFreqReg(MiniGen::FREQ0);

}
```

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

Anhang

```
#####  
#####  
  
void watchdogSetup()  
{  
cli(); // disable all interrupts  
wdt_reset(); // reset the WDT timer  
MCUSR &= ~(1 << WDRF); // because the data sheet said to  
/*  
WDTCR configuration:  
WDIE = 1 :Interrupt Enable  
WDE = 1 :Reset Enable - I won't be using this on the 2560  
WDP3 = 0 :For 1000ms Time-out  
WDP2 = 1 :bit pattern is  
WDP1 = 1 :0110 change this for a different  
WDPO = 0 :timeout period.  
*/  
// Enter Watchdog Configuration mode:  
WDTCR |= (1 << WDCE) | (1 << WDE);  
// Set Watchdog settings: interrupte enable, 0110 for timer  
WDTCR = (1 << WDIE) | (0 << WDP3) | (1 << WDP2) | (1 << WDP1) | (1 << WDPO);  
sei();  
Serial.println("finished watchdog setup"); // just here for testing  
  
}  
void Minigen_Sleep() {  
  
SPIWrite(0x00C0); //Sleepbit1 & Sleepbit12 im control register  
}  
#####  
void SPIWrite(uint16_t data) {  
digitalWrite(_FSYNCPin, LOW);  
SPI.transfer((byte)(data >> 8));  
SPI.transfer((byte)(data));  
digitalWrite(_FSYNCPin, HIGH);  
}  
}
```

Gesamtschaltung Prototyp 2

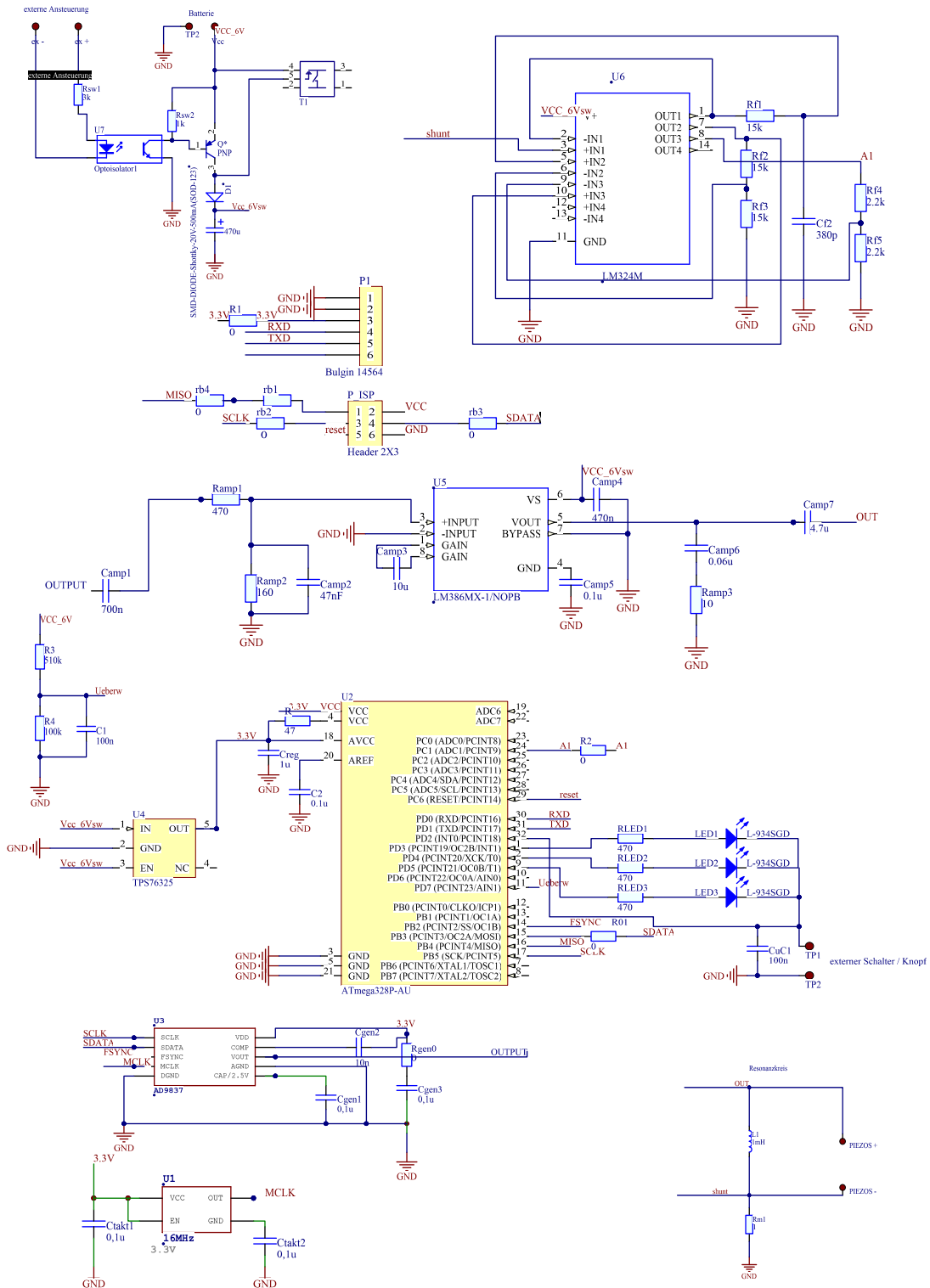


Abbildung 6.1: Schaltung des LM324

Programmablaufplan Programmcode 2

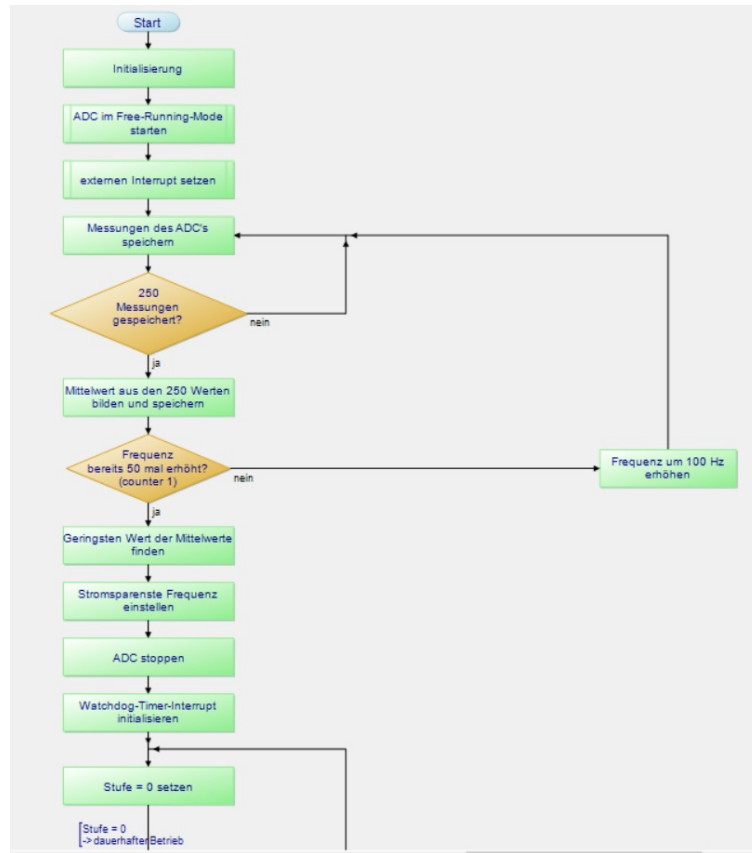


Abbildung 6.2: Programmablaufplan des Setup-Codes

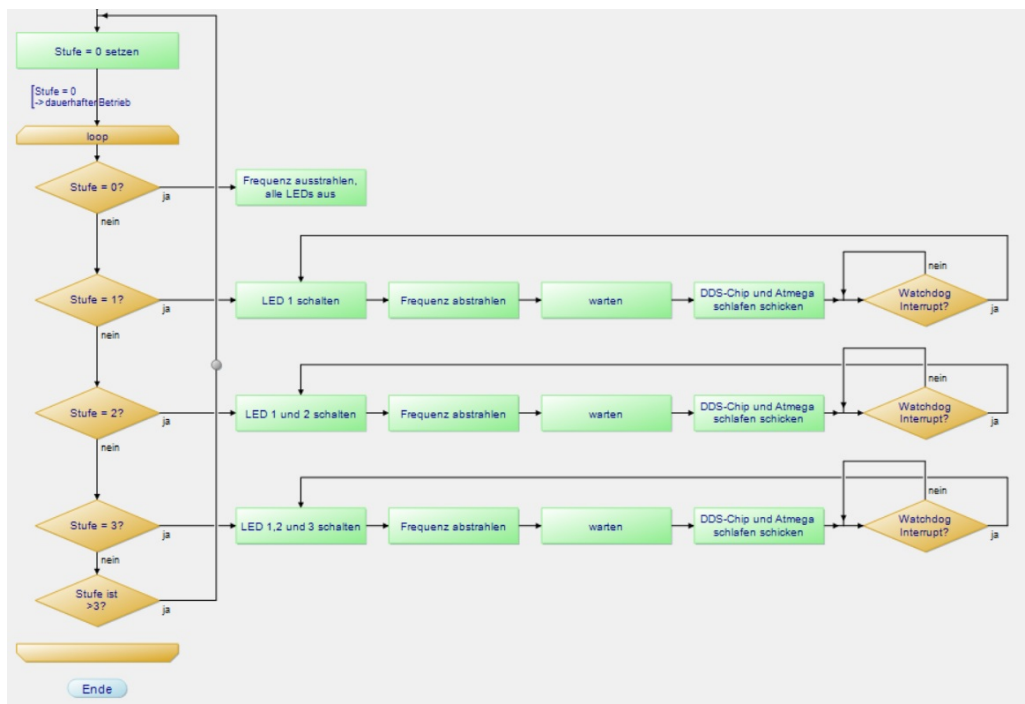


Abbildung 6.3: Programmablaufplan des Loop-Codes

Entwicklung eines Gerätes zur Schädlingsbekämpfung in Bienenstöcken mittels Ultraschall

Anhang

Bills-of-Material-Liste

Comment	Description	Designator	Footprint	LibRef	Quantity
Cap_p01	Kondensator	470u	CAP_TYP_E_RAD	Cap_p01	1
100n	Kondensator	C1	0805_Capacitor	Cap	1
0.1u	Kondensator	C2	0805_Capacitor	Cap	1
700n	Kondensator	Camp1	1206_capacitor	Cap	1
49n	Kondensator	Camp2	0603_Capacitor	Cap	1
10u	Kondensator	Camp3	0603_Capacitor	Cap	1
470n	Kondensator	Camp4	0603_Capacitor	Cap	1
0.1u	Kondensator	Camp5	0603_Capacitor	Cap	1
0.06u	Kondensator	Camp6	0603_Capacitor	Cap	1
47u	Kondensator	Camp7	0603_Capacitor	Cap	1
33u	Kondensator	C11	0603_Capacitor	Cap	1
380p	Kondensator	C12	0603_Capacitor	Cap	1
0.1u	Kondensator	Cgen1, Cgen3, Ctaekt1, Ctaekt2	0603_Capacitor	Cap	4
10n	Kondensator	Cgen2	0603_Capacitor	Cap	1
1u	Kondensator	Creg	0603_Capacitor	Cap	1
100n	Kondensator	CuJ1	0603_Capacitor	Cap	1
SMD-DIODE-肖特基-20V-500mA(SOD-123)	MMSZ4702T1 Series 500 mW 10 mA 15 V SMT Zener Voltage Regulator - SOD-123	D1	SOD-123_SMD-DIODE-ZENER-15V-500MW(SOD-123)(Primary)	SOD-123	1
Ind	Spule	L1	IND-ASP-1300T	Ind	1
L-9345GD	Typical INFRARED GaAs1 LED	LED1, LED2, LED3	LED-0	LED0	3
Buigin 14564	Buigin 14563 / 14564	P1	Buigin 14563	1x16 Buigin Shift	1
Header 2X3	Header 2X3	P_JSP	2x3G RM 2.54	Header 2X3	1
PNP	SOT23	Q*	SOT23	PNP	1
47	Widerstand	R	0805_Resistor	Res	1
0	Widerstand	r01, R1, R2, r02, r03, Rgen0	1206_resistor	Res	7
510k	Widerstand	R3	1206_resistor	Res	1
100k	Widerstand	R4	1206_resistor	Res	1
470	Widerstand	Ramp1	0402_resistor	Res	1
160	Widerstand	Ramp2	0402_resistor	Res	1
10	Widerstand	Ramp3	0402_resistor	Res	1
0	Widerstand	r04	0603_Resistor	Res	1
1k	Widerstand	R11	1206_resistor	Res	1
1k	Widerstand	R12, R13	1206_resistor	Res	2
2.2k	Widerstand	R14	1206_resistor	Res	1
470	Widerstand	R15	0402_resistor	Res	1
1k	Widerstand	RLED1, RLED2	0603_Resistor	Res	2
1	Widerstand	Rm1	0402_resistor	Res	1
3k	Widerstand	Rsw1	0402_resistor	Res	1
1k	Widerstand	Rsw2	0805_Resistor	Res	1
LSG.1301.93xx	Schrier LSG Series: SMD, höhenvariabel: 5 Pins	T1	LSG_1301.9314	LSG_1301.93xx	1
16MHz	Generic 5x3 and 7x5 oscillators	U1	CRYSTAL-SMD-7x5	OSCILLATORXAL-11225	1
ATmega328P-AU	8-bit AVR Microcontroller, 32KB Flash, 1KB EEPROM, 2KB SRAM, 32-pin TQFP, Industrial Grade (-40°C to 85°C)	U2	32A_M	ATmega328P-AU	1
AD9837	DDS generator	U3	AD9837	AD9837	1
TP5762Z5		U4	SOT23-5	TP5762Z5	1
LM386MX-1/NOPB	Wide Vm Low Power Audio Amplifier with Internal Gain 8-SOIC 0 to 70	U5	SOIC127P600X175-8N	LM386MX-1/NOPB	1
LM324M	Low Power Quad Operational Amplifier 14-SOIC 0 to 70	U6	SOIC127P620X175-14N	LM324M	1
Opotisolator1	4-Pin Phototransistor Optocoupler	U7	Dip-4	Opotisolator1	1

Abbildung 6.4: BOM-Liste des Gerätes