

NF-Wobbler-Arduino

Heute werden preiswerte Spektrumanalysatoren(SA) angeboten, die sich auch ein Hobby-Bastler leisten kann. Alle haben aber den Nachteil, dass sie erst ab 9 kHz arbeiten. Es fehlt der Bereich der Niederfrequenz(NF). Es gibt zwar auch Signalgeneratoren, mit denen man wobbeln kann. Die haben aber leider kein Sichtteil. Man muss sie umständlich mit einem Sichtgerät(Oszillografen) verbinden. Ein Messgeräte-Aufbau, den man mal eben nicht so nebenbei zusammenstellt.

In meinem langen „Bastlerleben“ kam es aber doch hin und wieder vor, dass ich das Frequenzverhalten eines Übertragungsweges gern mal sichtbar gemacht hätte. Und sei es nur darum gegangen, die vielen Trägerfrequenzfilter, die ich irgendwann erstanden hatte, durchzumessen. Und auch die Sprachfilter im Funkbereich blieben mir in ihren Eigenschaften verborgen. Ich musste Annahmen treffen, um sie sinnvoll zu verwenden. Zuletzt erst hätte ich gern die Durchlasskurve eines NF-Schwingkreises dargestellt, als ich den Kriegsempfänger „Berta“ nachbauen wollte. Als Folge dieses Mangels in meiner Ausrüstung baute ich mir einen Frequenzumsetzer für meinen SA. Damit konnte ganz schnell mal auf einen Blick wobbeln. Die NF wurde in den Bereich um 10 MHz verschoben. Alle Filter im SA stehen mir daher zur Verfügung. Ebenso die Bestimmung des absoluten Pegels. Allerdings ist die Messdynamik sehr eingeschränkt. Durch die zweifache Frequenzumsetzung war nicht mehr als 30 dB zu erreichen. Also wünschte ich mir einen Wobbler, der mindestens 50 dB darstellen kann.

Ich besitze eigentlich so ein Gerät. Einen SA von W&G, der bei 0 Hz beginnt. Das ist aber kein Tischgerät, denn es wiegt mehr als 30 kg. Zu unhandlich, um es mal eben an einer anderen Messstelle einzusetzen. Und raumsparend ist es auch nicht. Ich trenne mich

nur ungern von den Präzisionsgeräten von R&S bzw. W&G. Sie haben mir viele Jahre lang Freude gemacht. Doch allmählich füllte sich meine kleine Werkstatt und ich leide heute unter Platzmangel. Das sollte sich jetzt ändern. Und so kam mir die Idee, mit einem Arduino die Steuerung einer Elektronik zu machen, die man ohne Probleme nachbauen kann.

Das Konzept

Es sollte ein einfacher Wobbler werden, kein Messgerät! Und die Bauteile müssen preiswert zu beschaffen sein. Der Funktionsumfang soll den grundlegenden Ansprüchen genügen. So hat die x-Achse eine lineare Darstellung. Nur die y-Achse hat eine logarithmische Teilung. Der Frequenzbereich aber soll den gesamten NF-Bereich abdecken, von 0 Hz bis 29 kHz.

Auf einer Sichtfläche von 45mm x 60mm kann man keine hohe Auflösung erwarten. Größere Displays für den Arduino sind ganz schnell im hohen Preisbereich. Meines ist ein TFT-LCD-Display und kostet in DL ca. 22 EUR. Dennoch wird die y-Achse einen Messbereich von 60 dB haben. Der AD-Wandler im Arduino misst mit 10 Bits. Er hat also einen Dynamikbereich von etwa $1 : 1000 = 60 \text{ dB}$, wenn man von einer Mindestamplitude von 1mV ausgeht. Aber für die Messpunkte verwende ich bereits einige Pixel, sodass man eine gut sichtbare Linie im Kurvenzug erhält. Das schränkt natürlich die präzise Darstellung ein. Und so kann man davon ausgehen, mit einer Genauigkeit von $\pm 5\%$ zu messen. Mir genügen diese Eigenschaften. Eine logarithmische Auflösung der Frequenzachse brauche ich nicht. Dafür gibt es auf dem Markt einige Softwareangebote, die das mit dem PC und der Soundkarte machen. Ich will keine akustische Vermessung eines Studios vornehmen. Mikrofone und Lautsprecher sind nicht auf meinem Basteltisch.

Wunschliste

Hier meine Ansprüche für das Funktionsmuster:

Frequenzbereich : 0 Hz – 29 kHz

Dynamik: > 50 db

Ausgangspegel / Eingangspegel regelbar; -20 dBm, auf eine Impedanz von 600 Ohm bezogen; das sind ca. 80 mV RMS

Ausgangswiderstand(Impedanz): 50 Ohm, 600 Ohm, 1 M

Eingangswiderstand „ : 50 Ohm, 600 Ohm, 1 M

Ein Menue nach dem Einschalten; Tastensteuerung, schrittweise Verstellung mit einem Encoder

Keine Speicherung der Daten, keine Übertragung des Bildes auf einen Stick oder PC. Keine Speicherung in einem internen Memory.

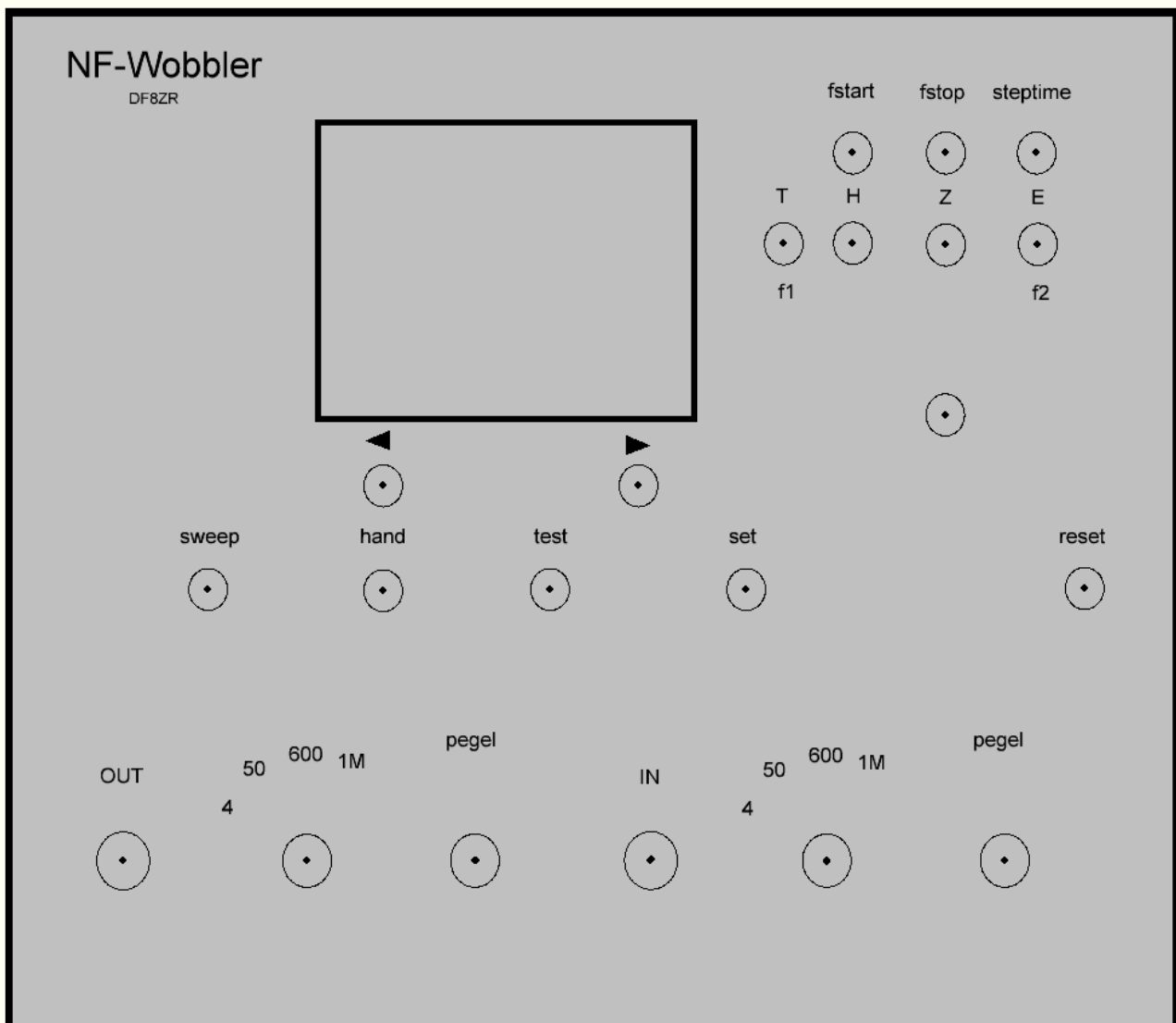
Handelsübliche Bauteile. Gesamtkosten nicht höher als 80,-- ... 100,-- EUR.

Man sieht also, dass es ein Bastelprojekt sein soll, das man als Anreiz für den Nachbau durchaus veröffentlichen kann.

Die Frontplatte

Nachfolgend ein Bild. Oberhalb des Displays sind im rechten Teil die Taster für die Eingabedaten. Das sind „fstart“ für die Startfrequenz, „fstop“ die für die Stopfrequenz. Ganz rechts der Taster für die Aktivierung der Eingabe der Darstellungszeit. Es wird die Zeitspanne eines Schrittes in x-Richtung festgelegt. Die Laufzeit ergibt sich aus der max. Länge der x-Achse. Die Kurve wird hier nur in 120 Schritten dargestellt. Das Display hat zwar 240 x 320 Bildpunkte. Man braucht aber links und rechts noch Freiraum. Für die Messkurve verbleiben 240 Pixel. Der Punkt hat

aber > 2 Pixel im Durchmesser.



Für die Anzeige der gerade aktiven Ziffernstelle(Dekade) zeichnet der Wobbler Unterstriche. Die Dekaden werden mit den Tastern „T,H,Z und E“ aufgerufen. Der Unterstrich läuft allerdings mit, wenn man die Zahlen erhöht oder erniedrigt. Die korrekte Darstellung dieser Automatik war ein wesentlicher Anteil meiner Programmierarbeit und brachte viele Ungereimtheiten zutage, die ich mit Kompromissen unterdrücken musste.

Unterhalb des Displays sind die Taster für die Aktivierung der Cursor „<<“ und „>>“. Hiermit wird der Sweep in x-Richtung bewegt. So kann man durch Abschätzen die -3dB-Bandbreite

bestimmen, da die Frequenzen oberhalb der Kurve angezeigt werden.

In der unteren Reihe links sind die Funktionstasten. „sweep“ löst den periodischen Messvorgang aus. Mit „hand“ kann man den Fortschritt in x-Richtung auch von Hand ausführen. „test“ zeigt eine Messkurve im Überblick. Hier wird von 0 Hz bis 29 kHz einem einem Durchlauf die Kurve gezeichnet.

„set“ übernimmt die Daten nach der Eingabe und wechselt in den Darstellungsmodus. Mit den Tastern „fstart“, „fstop“ oder „timeset“ gelangt man zurück ins Menue. Die Eingabedaten gehen nicht verloren. Nur wenn man auf den Knopf des Encoders drückt, erfolgt eine absolute Rücksetzung(RESET) der Software mit dem Verlust der Eingabe und der Frequenzkurve. Anfangs hatte ich diesen Switch für die Dekaden verwendet. Jedoch ging mir seine Schwergängigkeit auf die Nerven. Jedesmal verschob ich das Gerät beim Fortschalten der Dekaden. Daher wechselte ich nachträglich auf die leicht zu klickenden Tasten.

„reset“ setzt nur die Zahl hinter der gerade aktivierten Zeile auf Null.

So, das war nun hoffentlich nicht zu viel mit dieser einführenden Beschreibung. Es folgen jetzt die Hinweise für den Nachbau.

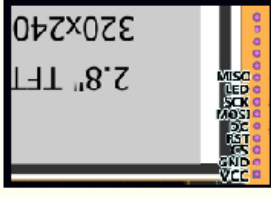
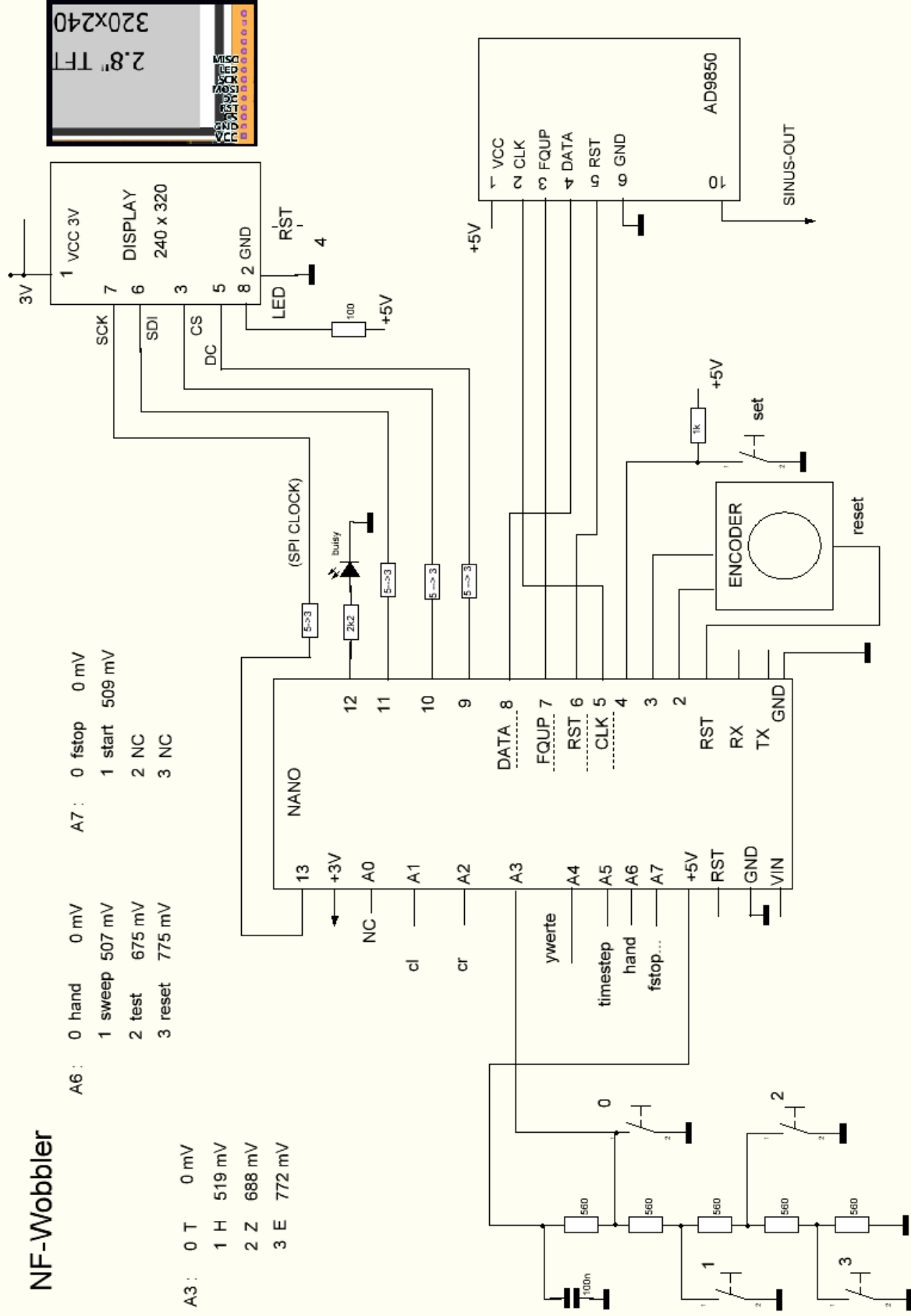
Schaltung

Das Bild zeigt die Schaltung im Überblick. Nachdem ich die Pegelanpassung für das Display zunächst mit Widerstandsteilern machte, habe ich zuletzt doch die käuflichen Module(1,60/Stck) genommen. Mit den Widerständen war die Funktion vom Zufall abhängig. Und die Ursache für so ein Fehlverhalten ist nur schwer zu finden. Man sucht vergebens zunächst in der Software. Es ist besser, beim Nachbau diese Probleme zu umgehen.

NF-Wobbler

A3: 0 T 0 mV
 1 H 519 mV
 2 Z 688 mV
 3 E 772 mV

A6: 0 hand 0 mV A7: 0 fstop 0 mV
 1 sweep 507 mV 1 start 509 mV
 2 test 675 mV 2 NC
 3 reset 775 mV 3 NC



Der AD9850 braucht mindestens 2ms für die Einstellung auf eine neue Frequenz. Ich habe deshalb 4ms vorgesehen. Das ist die Mindestdauer für einen Frequenzschritt. Aus der verfügbaren Länge der x-Achse folgt daher die kürzeste Wobbelzeit $250 \times 4 \text{ ms} = 1 \text{ s}$. Das ist auch für Abgleicharbeiten noch erträglich. Oft muss man bei Filtern die Einschwingzeit berücksichtigen. Dann kann man die Laufzeit bis 250 s (1000ms/Schritt) verlängern. Auch das sollte ausreichend sein. Bleibt ja immer noch die Möglichkeit, ganz langsam von Hand zu wobbeln.

Ports

Es waren nicht ausreichend viele Ports am Arduino verfügbar. Hier kam dann die Methode mit den Spannungsteilern an den Analogports zur Anwendung. Die Sache funktioniert ganz gut. Man gewinnt also Ports hinzu. Fehlauflösungen durch Störimpulse kann ich nicht beobachten.

Sonstige Hardware

Habe noch nicht die NF-Verstärker, Pegelwandler für die negative Betriebsspannung und Stromversorgung gebaut. Werde dazu später etwas schreiben. Im Augenblick programmiere ich noch und berichte jetzt darüber.

Software

Man muss konzentriert vorgehen. Das Programm wurde länger als üblich. Zunächst wollte der Encoder nicht korrekt arbeiten. Viele Vorschläge im Netz gehen vom Interrupt-Betrieb aus. Das hat sich zunächst nicht bewährt. Der Encoder machte dann aber Probleme mit einer akzeptablen Reaktion, weil der Code (das Programm) zu lang wurde. Ich schrieb es wieder auf Interrupt um.

Danach stellte sich heraus, dass die Markierung der aktiven Dekade viel Ärger machte und langes Nachdenken erforderte. Der Schriftzug der Ziffern wird von links nach rechts bewegt. Ist eine Dekade durch einen Unterstrich markiert, muss man bei

Änderungen des Wertebereichs darauf Rücksicht nehmen und in Abhängigkeit von dem Gültigkeitsbereich einer Dekade und dem angezeigten Wert den Unterstrich an der richtigen Stelle setzen. Auch hier waren Effekte wirksam, die ich bis heute nicht erklären kann. So passierte es z.B. beim Unterschreiten der Null, dass die Position auf dem Display verloren ging. Ein Unterstrich wurde unerwartet am linken Rand gezeichnet und der Encoder zählte positiv weiter. Die Ursache war ein println statt print. Habe auch mit Tricks gearbeitet, indem ich z.B. einfach das gesamte Bild dunkel schalte, wenn die Zahlen am Encoder im negativ Bereich sind.

Darstellungsproblem

Der Unterstrich sollte mitgeführt werden!

Wertebereiche:

Für die aktivierte Einerstelle(Dekade E).

0-9, 10-99, 100-999, 1000-9999, 10000-29000.

für Z. 10-99,100-999,1000-9999,10000-29000

für H: 100-999, 1000-9999, 10000-29000

für T. 1000-29000

Es wird nur in den letzten vier Stellen ein Unterstrich gezeichnet. Bei Zahlen über 9999 bleibt also die erste Stelle links ohne Unterstrich!

Mit möglichst wenig Code die Aufgabe zu lösen, war nicht einfach. Und dabei sollte auch die Übersicht gewahrt bleiben. Man muss viele ungewöhnlichen Bedienungsmöglichkeiten an den Tasten und unerlaubten Encoderwerten behandeln. Für jede

Dekade ließ sich das nicht ohne Aufwand codieren. Das Programm wurde hier leider sehr lang. Später habe ich es dann anders gelöst. Dazu mehr im PDF „Software“.

Analoge Ports

Durch den Einsatz von Spannungsteilern konnte ich die Bedienung mit mehreren Tasten realisieren. Man schreibt sich zunächst eine kurze Routine für das Einlesen der Spannungen. Dann kann man die if-Bedingungen sehr trennscharf formulieren. Man braucht also keine Spannungsmessungen mit dem Multimeter zu machen, denn die sind letztlich abweichend von den tatsächlich vom Arduino detektierten Spannungen. Dabei nicht zu lange Leitungen verlegen, um Fehlverhalten durch Störspannungen zu vermeiden. Für einen Test steht die Funktion `void valatest();` zur Verfügung.

AD9850

Dem Baustein habe ich einen Leistungsverstärker nachgeschaltet. So ist er auch bei Fehlabschlüssen vor Überlastung und Kurzschluss geschützt. Der Mitlaufgenerator des Wobblers muss in einen 600 Ohm-Widerstand ohne Spannungsabfall ca. 100mV RMS abgeben können. Ansonsten arbeitet der Baustein hervorragend im NF-Bereich. Der Sinus ist „sauber“, der Klirrfaktor hinreichend gering. Hält man den Eingangsverstärker noch störspannungsfrei, dann ist ein korrektes y-Signal zu erwarten. Das ist wichtig, denn hier werden keine Eingangsfiler verwendet, die das Rauschen vermindern. Dennoch sollte man im Gebrauch des Wobblers darauf achten, dass sich kein Brumm überlagert. Die Kurven würden verfälscht! Vielleicht werde ich später ein Kerbfilter für 50/100Hz nachrüsten. Für einen „Handscharter“ ist ja noch Raum auf der Frontplatte.

Sweepen

Der AD9850 braucht mindestens 2ms für die Einstellung auf eine neue Frequenz. Bei einer festen Schrittzahl von 250(Punkte auf der x-Achse) dauert das also mindestens 0,5 s. Ich habe die minimale Reaktionszeit auf 4ms eingestellt, da ja auch noch andere Verzögerungen durch den Algorithmus anfallen. Daher ist die kürzeste Durchlaufzeit eines Frequenzbereiches ca. 1s.

Das Programm überprüft diese Forderungen. Wird durch falsche Einstellungen die Zeit unterschritten, setzt es die Zeit zwischen zwei Steps automatisch auf 4ms. Der Anwender sollte jedoch wissen, was er einstellt.

Der AD8307

Ich verwende einen logarithmischen Dtektor. Er ist eigentlich für HF vorgesehen und sein Frequenzbereich geht bis 500 MHz. Im niederfrequenten Messbereich muss man die Kondensatoren anpassen. 10 uF sind hier aber genug, um auch von 0 Hz an ohne Artefakte zu arbeiten. Der „Strahl“ wird durch einen weiteren Glättungskondensator von 10u frei von Impulsen gehalten. Größere Werte sollte man nicht nehmen, denn sonst entstehen unerwünschte Einschwingvorgänge. Der Ad8307 ist preiswert zu erhalten.

Kosten und Probleme

Insgesamt belaufen sich die Kosten für alle Bauteile unter 100 EUR. So war es auch geplant, denn für dieses Geld kann man schon gebrauchte Altgeräte kaufen. Aber auch der Reiz des Selbstbauens ist ein Wert an sich, den man nicht unbeachtet lassen sollte. Es macht einfach Freude, wenn man eine Idee erfolgreich umsetzt. Dennoch war dieses Bastelprojekt für mich eine große Herausforderung. Viele kleine Probleme waren zu lösen. Und manchen Fehler habe ich erst nach zwei Wochen gefunden. So zeigte der AD9850 immer wieder einen Frequenzgang, der nicht tolerierbar war. Wenn man von Hand den Strahl von links nach

rechts bewegte, stimmten die Frequenzen und der Pegel blieb konstant. Machte man aber den periodischen Sweepbetrieb, dann fiel der Pegel regelmäßig ab. Auch dann, wenn man längere Pausen für die Frequenzsteps vorgab. Es hatte also nichts mit der Geschwindigkeit des Fortschreibens der Frequenz zu tun.



Erst als ich nach vielen Tagen vergeblicher Suche im eigenen Algorithmus mich entschloss, die aus dem Netz kopierte Bibliothek zu wechseln, verschwand dieser Mangel. Der Pegel blieb jetzt konstant und ich konnte alle bis dahin gebastelten Regelschaltungen verwerfen. Es war schlicht ein Fehler in der Bibliothek, den der Entwickler wahrscheinlich nicht einmal selbst bemerkte, weil er nur einen Funktionsgenerator damit betrieb und nicht wobbelte.

Ich habe nicht den Ehrgeiz, absolut perfekten Code zu erstellen. Das sollen die Berufsprogrammierer machen. Insofern gibt es noch einige kleine Mängel, die ich aber erst später mal abstellen werde. Nach einigen Wochen intensiver Sucharbeit in den fast 3000 Zeilen umfassenden Code habe ich im Augenblick nicht mehr die ursprüngliche Konzentration, die Bugs zu beseitigen. Die

Grundfunktionen laufen störungsfrei, lediglich bei machen Anzeigen sind Artefakte zu sehen.

Film

Wie immer dokumentiere ich die Arbeit mit einem Film bei YouTube:

<https://www.youtube.com/watch?v=xx30j56E4io>

Die Schaltungen sind in einem anderen Dokument zugänglich. Ebenso die Software, die man bei Arduino „Sketch“ nennt.

DF8ZR, 14.04.2022