

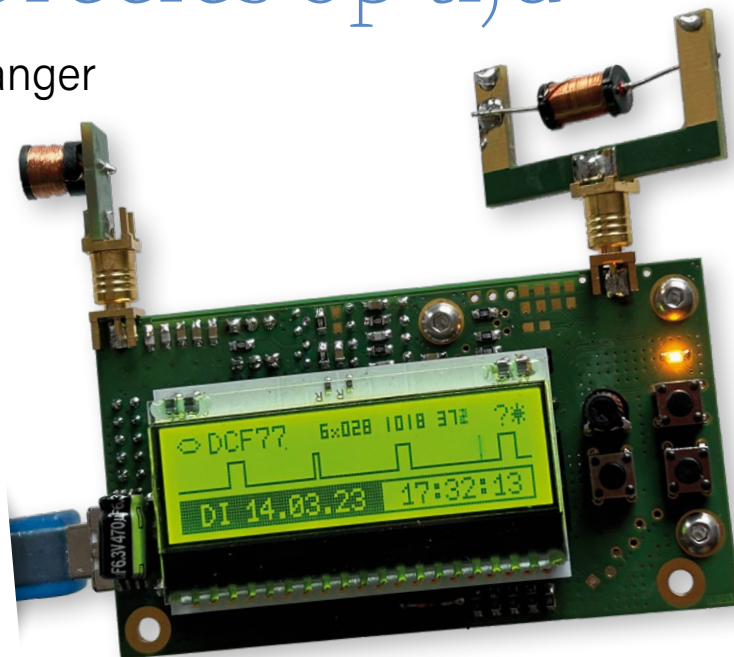
PIC voor tien

precies op tijd

ontwerp van een SDR-tijdseinontvanger

Marco Rohleder (Duitsland)

Als je een tijdseinontvanger ontwerpt, kun je eigenlijk niet meer om het concept van een Software Defined Radio (SDR) heen. Elektor heeft al veel artikelen gepubliceerd over de software voor het decoderen van dit soort signalen, maar hier concentreren we ons op de vereiste hardware. We willen graag toekomen met een minimum aan rekenvermogen en gebruiken dan ook een eenvoudige 8bit-microcontroller met periferie die onafhankelijk van de CPU werkt. Opmerkelijk genoeg bleek uiteindelijk dat de gebruikte PIC-MCU zelfs in staat was om twee verschillende tijdsignalen tegelijk te ontvangen en te decoderen.



tijd uitzenden. Bovendien hebben DCF39 en DCF49 een zendvermogen van 100 kW, aanzienlijk meer dan de 50 kW van DCF77. Er zijn ook nog steeds verschillende meteorologische diensten (zoals maritieme weerstations) actief op deze frequenties. Zou het niet mooi zijn om al die stations (en meer) op de langegolf te kunnen ontvangen?

Alle begin is moeilijk

Mijn eerste pogingen om het DCF77-sig-naal op te pikken met een op 77,5 kHz afgestemde ferrietantenne en dat te versterken, om de draaggolffrequentie met 1Hz-modulatie zichtbaar te maken op een oscilloscoop, leverden weinig op. Er is helaas erg veel storing op de langegolffband. Schakelende voedingen en LED-verlichtingssyste-men in alle denkbare soorten en maten, voornamelijk van Aziatische afkomst, zorgen voor een ruisniveau dat soms veel groter is dan het gewenste signaal. Het is bijna niet te geloven dat uit dit ratjetoe van signalen het DCF77-sig-naal kan worden ontvangen.

Hoe bouw je een tijdseinontvanger met gemakkelijk verkrijgbare onderdelen? Voor het decoderen van het tijdsig-naal is in elk geval een microcontroller (MCU) nodig, dus het ligt voor de hand om zoveel mogelijk gebruik te maken van de voorzieningen die de MCU al aan boord heeft. Dat betekent dat de ontvanger zoveel mogelijk software en zo min mogelijk extra hardware gebruikt.

We wilden ook het gebruik van exotische, moeilijk verkrijgbare componenten vermijden; we geven waar mogelijk de voorkeur aan standaardonderdelen. Dat geldt vooral voor SMD-componenten. Condensatoren en

weerstand-en in 0805-behuizing zijn gemakkelijk met de hand te solderen en besparen ruimte op de print. Maar een vierbeens-IC met een heel kleine pinafstand is een stuk lastiger. Naast het DCF77-sig-naal op 77,5 kHz uit Mainflingen, dichtbij Frankfurt, zijn er nog vele andere zenders in Europa en elders in de wereld die tijdseinen uitzenden, zoals MSF60 op 60 kHz in Groot-Brittannië, TDF162 op 162 kHz in Frankrijk en WWVB op 60 kHz in de Verenigde Staten. Verder zijn er ook ripple control-zenders, zoals DCF49 op 129,1 kHz (ook in Mainflingen) en DCF39 op 139 kHz in Burg dichtbij Magdenburg, die de actuele

De in de handel verkrijgbare modules zijn blijkbaar ontworpen als rechthoek-richtaan ontvangers met een zeer smal banddoorlaatfilter met een 77,5kHz-kristal. Dat is een eenvoudige en goedkope aanpak, maar daarmee kunnen alleen signalen van 77,5 kHz ±10 Hz worden ontvangen. Uiteindelijk werd het me duidelijk dat zonder stevig (en, zoals later bleek, digitaal) filteren, het doel van betrouwbare tijdseinontvangst onhaalbaar was.

Kijk om te beginnen eens naar het blokschema van de volledig geconfigureerde tijdseinontvanger (figuur 1). We zullen de diverse blokken één voor één bespreken en uitverken.

Preselector met MOSFET-‘varicap’

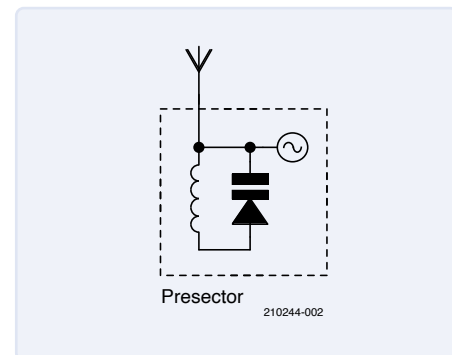
Zelfs de meest effectieve en steile software-filters en de beste en snelste A/D-converters zijn nutteloos als ze worden overspoeld

door storing waardoor het gewenste signaal verdrinkt in de ruis. We ontkomen dus niet aan het gebruik van een preselector die de meeste ruis blokkeert.

Vaak wordt een LC-kring met een variabele condensator (figuur 2) gebruikt als preselector. De resonantiefrequentie van de afgestemde kring volgt uit de formule:

$$f_0 = \frac{1}{2\pi\sqrt{LC}}$$

Als je een ferrietantenne met een typische zelfinductie van 1500 µH gebruikt, zoals gebruikelijk voor de langegolfband, dan hebt je een variabele condensator met een bereik van 400 pF tot 6,7 nF nodig voor een frequentiebereik van 50 tot 200 kHz. Helaas bestaan er geen varicap-diodes met die specificatie. Gelukkig heeft elke halfgeleidercomponent een inherente (en doorgaans ongewenste)



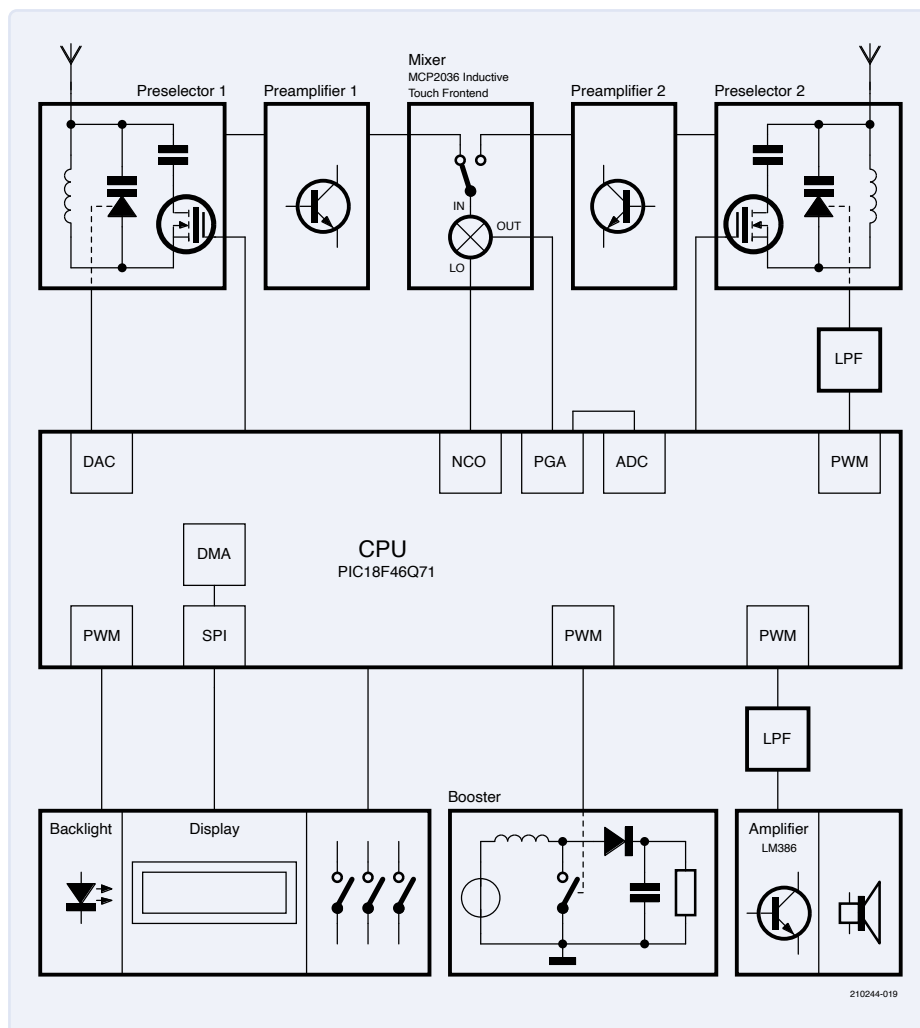
Figuur 2. Principeschema van de preselector.

junctiecapaciteit, die over het algemeen afhankelijk is van de dikte van de sperlaag en van de spanning. Mijn eerste metingen aan een vermogens-MOSFET zagen er veelbelovend uit. Uit verschillende types gaf een IRF640 de beste resultaten, met een capaciteit van 500 pF bij 20 V_{UDS} en 3 nF bij 1 V_{UDS}. Om wat gemakkelijker af te stemmen binnen het gewenste frequentiebereik, heb ik een aantal extra condensatoren (C30...C34) toegevoegd, die met behulp van transistoren (T30...T34) parallel met de vermogens-MOSFET's (T35 en T36) geschakeld worden als dat nodig is. Met deze configuratie kan de preselector in figuur 3 worden afgestemd over een bereik van ongeveer 30 kHz tot 300 kHz met een enkele zelfinductie van 1500 µH. Om het afstembereik van de MOSFET's volledig te benutten, is een hogere spanning nodig dan de DAC in de microcontroller kan leveren. Daarom wordt een LM358-opamp gebruikt voor het versterken van het signaal van 0...4,2 V naar het vereiste spanningsbereik van 0...30 V.

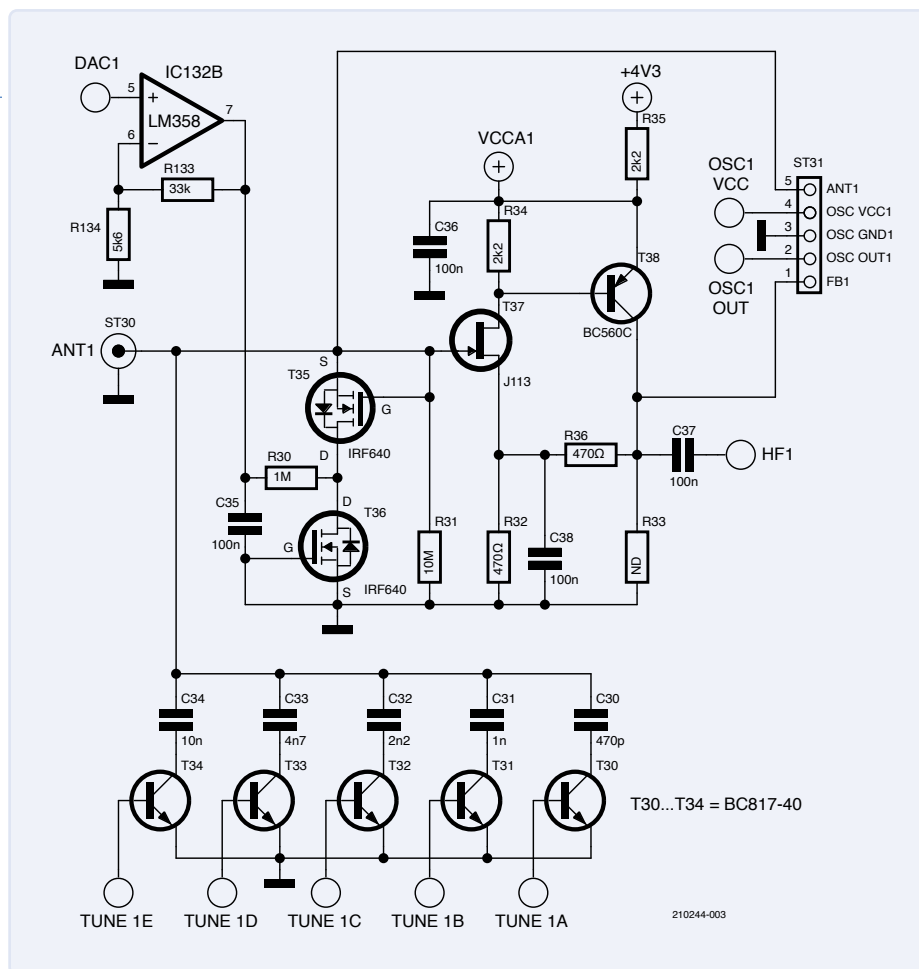
Alternatieve resonantiekeringen

Om verschillende ideeën te kunnen testen, heb ik een connector gebruikt (ST31, figuur 3), zodat dit deel van de schakeling kan worden uitgewisseld.

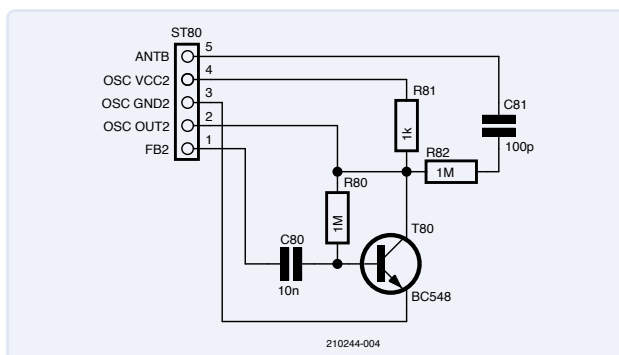
In figuur 4 zie je een eenvoudige manier om de terugkoppeling te implementeren. De benodigde faseverschuiving en extra versterking worden gerealiseerd met behulp van een extra transistor (T80). Deze oplossing heeft het voordeel dat de resonantieschakeling



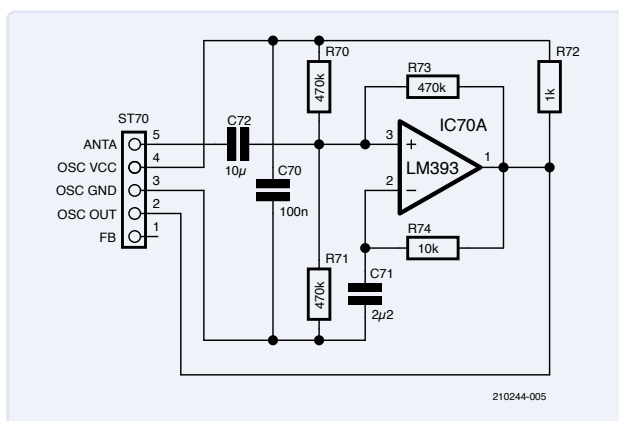
Figuur 1. Blokschema van de tijdseinontvanger.



Figuur 3. De preselector met de versterker voor het antennesignaal.



Figuur 4. Oscillator geïmplementeerd met transistor-terugkoppeling.



Figuur 5. Opamp-gebaseerde oscillator.

nauwelijks wordt gedempt, maar het bleek wel moeilijk een betrouwbare resonantie over een breed frequentiebereik te bereiken.

De terugkoppeling verandert de preselector in een resonantiekkring waarvan de trillingsfrequentie ongeveer dezelfde is als de resonantiefrequentie, maar dan in de langegolfband. Er is een klein probleem: bij de bedoelde ontvangstfrequentie verandert helaas de faseverschuiving van de ingang naar de terugkoppeling, dus de schakeling resonanceert niet altijd op de verwachte resonantiefrequentie. Voor een tweede poging heb ik een oude schakeling van een LC-meter met een comparator aangepast (**figuur 5**). Deze resonanceert over een groot bereik van inductie- en capaciteitswaarden en levert heel netjes de verwachte resonantiefrequentie. Maar, zoals zo vaak het geval is in de elektronica: een verbetering op één punt gaat ten koste van iets anders. De aansluitingen en terugkoppeling zijn geïmplementeerd met een zeer hoge impedantie, maar de ingangsimpedantie was aanzienlijk hoger dan bij de eerder beschreven oplossing. Drie 470kΩ-weerstanden (R70, R71, en R73) die (voor HF) parallel staan, geven samen 150 kΩ, wat de resonantieschakeling sterk dempt. In de praktijk had dit nauwelijks invloed op ontvangstkwaliteit, alleen moest de totale versterking iets worden opgevoerd. Voor deze variant had ik graag één van de in de MCU geïntegreerde comparatoren gebruikt. De pinmultiplexer in de MCU biedt veel mogelijkheden om signalen naar verschillende I/O-pinnen te leiden, maar bij intensief gebruik van de analoge periferie lopen we toch tegen grenzen aan, zeker als we willen vermijden dat het kwetsbare antennesignaal kriskras over de print wordt gevoerd. Daarom heb ik gekozen voor een externe comparator (IC70), zodat de signaalpaden zo kort mogelijk blijven. Uiteindelijk ben ik uitgekomen op het compromis van **figuur 6**. Hier wordt de resonantieschakeling betrouwbaar geëxciteerd met een korte puls (200 ns) op ST31/90 (pin 2) via R90/D90. Maar zonder terugkoppeling duurt de resonantie maar enkele perioden, wat te kort is om nauwkeurig genoeg te meten met de geïntegreerde timers. Daarom wordt het versterkte signaal via ST31 (pin 4) naar één van de comparatoren in de MCU gevoerd. En de uitgang van de comparator op ST31 (pin 2) levert extra energie om de schakeling aan het resoneren te houden (zie ook figuur 2).

De referentiespanning op de comparator, die met een DAC kan worden ingesteld, bepaalt hoe goed we de verwachte resonantiefrequentie kunnen benaderen. Misschien weet één van onze lezers hoe dit automatisch zou kunnen worden aangepast tijdens het gebruik zodat het betrouwbaar werkt bij verschillende frequenties.

Welke van deze drie mogelijkheden je ook kiest: uiteindelijk heb je een digitaal signaal met een redelijke amplitude dat je naar één van de tellers in de microcontroller kunt voeren. De frequentieresolutie hoeft niet beter te zijn dan 100 Hz, dus het is voldoende om de frequentie te meten met een tijdvenster van 10 ms (1/100 s).

Bij het opstarten meet de software eerst de frequentiebereiken die mogelijk zijn met het bijgeschakelen van de condensatoren C30...C34; voor elke combinatie wordt gemeten met de minimale en maximale afstemspanning. Dan wordt het best passende frequentiebereik geselecteerd voor de definitieve afstemming en de afstemspanning wordt stapsgewijs omhoog of omlaag aangepast, beginnend op de helft van het spanningsbereik. De totale tijd om af te stemmen op een bepaalde ontvangstfrequentie is minder dan 100 ms.

Je kunt de resonantieschakeling ook exciteren met een bekende frequentie en de capaciteit zó instellen dat je de minste demping en de hoogste amplitude krijgt. Je kunt daarvoor een hoogohmige weerstand verbinden met de aansluitpunten (ST31, pin 2, pin 5) en een zeer nauwkeurige referentiefrequentie op de I/O-pin zetten om het circuit aan te sturen (we komen daar nog op terug).

Ik heb ook een koperspoor rondom de hele print gelegd, dat ook volstaat om de antenne exciteren met de gewenste afstemfrequentie. Dat werkt heel goed, maar in vergelijking met de eerder beschreven oplossing is het

wel traag, omdat veel iteraties nodig zijn. Daarom lijkt het mij het handigste om beide oplossingen te combineren. Benader eerst de doelfrequentie ruw met de oscillator, en stem dan nauwkeuriger af door te exciteren en af te stemmen op maximale resonantie. **Figuur 7** toont een typische kalibratie en afstemming op 77,5 kHz.

Helaas is de capaciteit niet recht evenredig met de aangelegde spanning, zoals te zien is in **figuur 8** (let op de logaritmische schaal van deze grafiek). Met andere woorden: de capaciteitsverandering is groot bij lage spanningen en wordt steeds kleiner bij hogere spanningen. Om dat te verhelpen, kunnen we gebruik maken van een bijzondere eigenschap van de DAC in de microcontroller: hij kan gevoed worden met een variabele interne referentiespanning. Voor lage uitgangsspanningen kiezen we een referentiespanning van 1,024 V, wat leidt tot een klein spanningsbereik. Voor hogere afstemspanningen kan de referentiespanning worden opgevoerd tot 4,096 V.

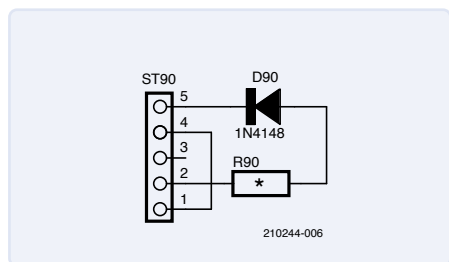
Het signaal van de preselector is nu beschikbaar voor verdere verwerking. Helaas heeft de in de microcontroller geïntegreerde A/D-converter een maximale conversiesnelheid van 300 kHz.

Mixer en meer

Omdat de bemonsteringsfrequentie voor een software-implementatie van de ontvanger op zijn minst tweemaal de hoogste te ontvangen frequentie moet zijn, en liefst zelfs vier maal hoger, moet het ontvangen signaal eerst naar een lagere frequentie worden gemengd voordat we het verder kunnen verwerken in de microcontroller.

Min of meer toevallig hoorde ik over een interessante component, de MCP2036 van Microchip, die eigenlijk bedoeld is als front-end voor aanraaksensoren, maar die ook een mengtrap bevat (**figuur 9**).

Het te mengen signaal gaat naar de twee ingangen LREF en LBTN. De lokale oscillatorfrequentie wordt via de CLK-input ingevoerd als een digitaal signaal, wat leidt tot een middenfrequentie van een paar kilohertz op de uitgang. Deze chip biedt ook andere nuttige functionaliteit, zoals een referentie-spanningsbron, een driver (hier niet gebruikt) en een opamp voor het versterken van het signaal en, indien nodig, het filteren van het uitgangssignaal om ongewenste mengproducten te verwijderen.

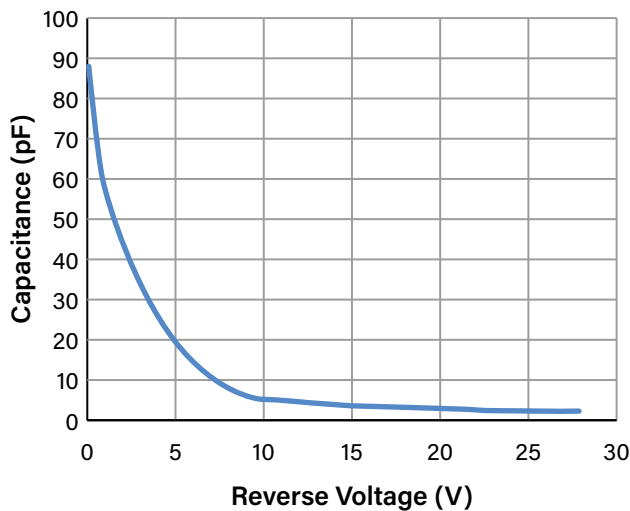


Figuur 6. De oscillator wordt geëxciteerd met een weerstand en een diode.

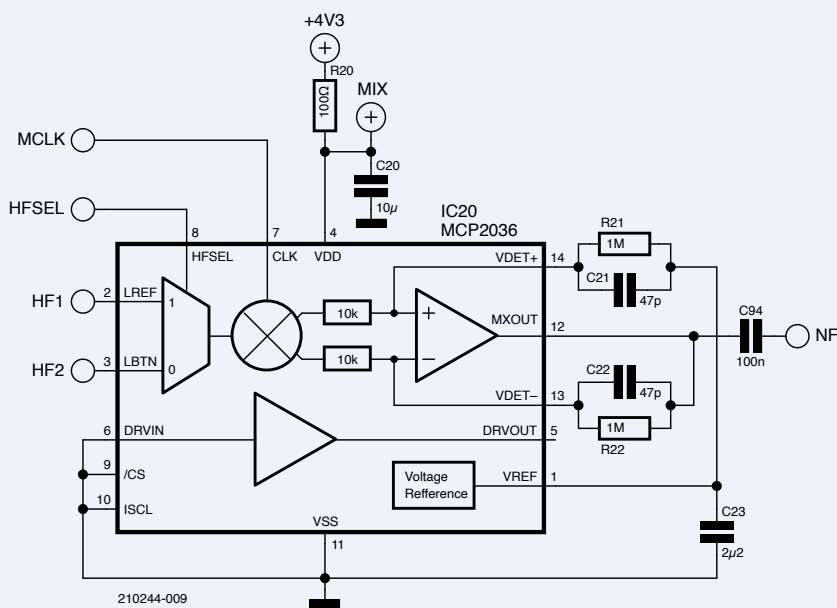
CHECKING TUNER ...		
Cx	F-LO	F-HI
0	127,000	227,200
1	106,300	147,500
2	91,400	113,900
3	82,700	98,200
4	74,400	84,900
5	69,400	77,700
6	64,800	71,400
7	61,300	66,900
8	56,000	60,000
9	53,800	57,300
10	51,400	54,600
11	49,700	52,600
12	47,700	50,200
13	46,300	48,700
14	44,800	46,900
15	43,700	45,600
16	39,800	41,200
17	39,000	40,400
18	38,100	39,400
19	37,400	38,600
20	36,500	37,600
21	35,900	36,900
22	35,200	36,100
23	34,600	35,600
24	33,500	34,400
25	33,000	33,800
26	32,500	33,200
27	32,100	32,800
28	31,500	32,200
29	31,100	31,700
30	30,600	31,300
31	30,300	30,800

TUNING ... (5)		
[512]	77,760	>
[256]	76,570	<
[384]	77,280	<
[448]	77,540	>
[416]	77,420	<
[432]	77,480	<
[440]	77,510	>
[436]	77,490	<
[438]	77,500	

Figuur 7. Controle van de tuner.



Figuur 8. Het niet-lineaire verloop van de junctiecapaciteit (bron: Digikey – <https://tinyurl.com/3js37yjn>).



Figuur 9. Interne structuur van de MCP2036 (bron: Microchip).

Dan wordt het volledig geconditioneerde en omlaag gemengde signaal naar de A/D-converter geleid. Daar wordt het signaal bemonsterd met viermaal de middenfrequentie om de I/Q-componenten te verkrijgen voor de digitale demodulatie van het ontvangen signaal.

Om amplitudegemoduleerde stations zoals DCF77 te kunnen ontvangen, moet wel de

versterking van de ingangstrap worden geregeld zodat de input voor de A/D-converter in het juiste bereik blijft: niet te hoog maar ook niet te laag. Daarvoor heb ik de opamp in de PIC18F46Q71-microcontroller gebruikt. Hij is geconfigureerd als inverterende versterker en in serie met de ingang aangesloten. Een prettige bijkomstigheid is dat er ook een weerstandsnetwerk is voor het instellen van

de versterking en waarmee een versterkingsfactor tot wel 15x bereikt kan worden. Net als bij de DAC is de referentiespanning van de ADC instelbaar tussen 1,024 V en 4,096 V, wat nog een extra versterkingsfactor vier oplevert. En als we maar acht van de twaalf bits van de A/D-converter gebruiken, hebben we nog een extra instelbare 16-voudige versterking. Alles bij elkaar betekent dit dat de versterking kan worden aangepast binnen een bereik van 1:14.400 ($14.400 = 15 \times 15 \times 16 \times 4$), zoals te zien in **figuur 10**.

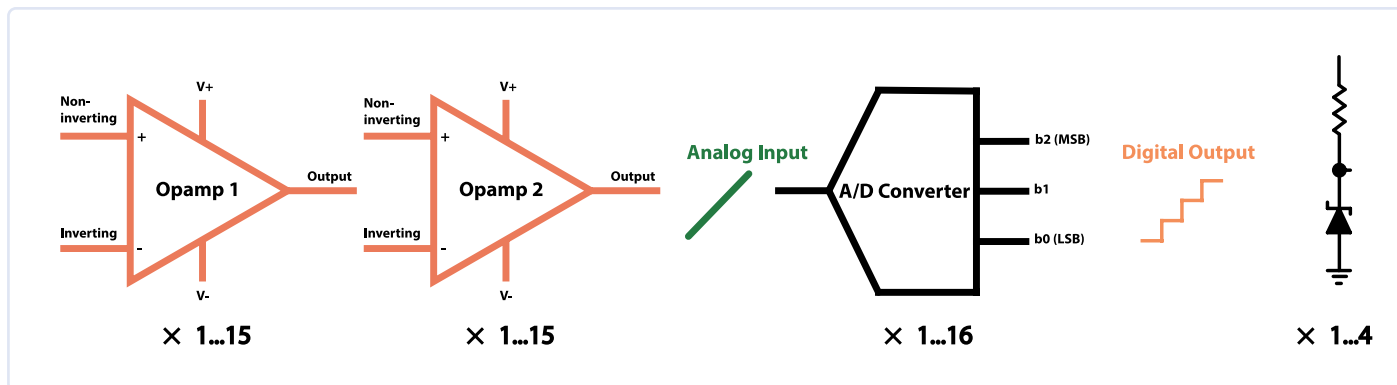
Omdat zes bits eigenlijk al genoeg zijn voor het betrouwbaar decoderen van het signaal (zoals we nog zullen aantonen), neemt het dynamische bereik toe tot 1:57.600 (nog eens vier maal meer), dat is meer dan 95 dB. Dat zou moeten volstaan – of de afstand tot de zender nu 2.000 km is of maar een paar meter. Nu hoeven we alleen nog maar een variabele lokale oscillatorfrequentie voor de mengtrap te genereren. En ook daarvoor heb ik de periferie van de microcontroller gebruikt. De meeste microcontrollers bezitten een timer/counter die het kloksignaal kan delen door een factor n . Dat is voldoende voor veel toepassingen, maar de resolutie is erbarmelijk. Stel dat de referentiefrequentie 16 MHz is. Een deelfactor van 80 levert een uitgangsfrequentie van 200 kHz, maar met de eerstvolgende deelfactor (81) is de uitgangsfrequentie 197,5 kHz: een verschil van 2,5 kHz!

Naast deze standaard-timers heeft de hier gebruikte PIC nog een perifere component: de zogenaamde *numeric controlled oscillator* (NCO). Zoals je ziet in **figuur 11** is dit een omhoogteller die telkens een vooringestelde waarde toevoegt aan de huidige waarde voor een nieuw resultaat. Dat gebeurt in een zeer hoog tempo, in dit geval met de microcontrollerklokkrequentie van 64 MHz.

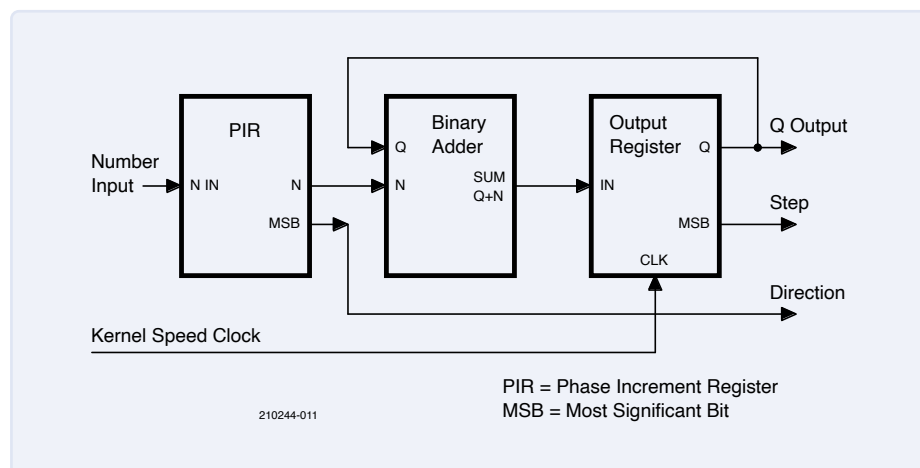
Telkens wanneer de opteller een overflow genereert, wordt ook een resultaat gegenereerd of een uitgang omgeschakeld. Bij lage frequenties krijgen we zo een resolutie van enkele tienden van een hertz. Natuurlijk zit er een zekere jitter in het uitgangssignaal, maar die kunnen we hier negeren.

De microcontroller

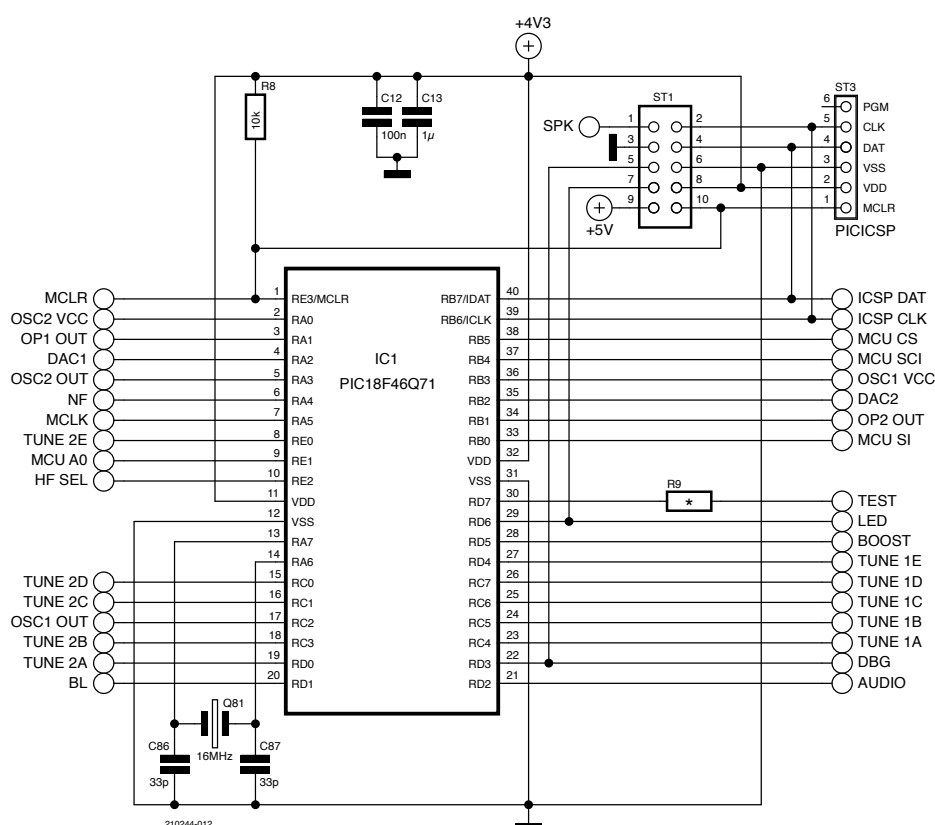
Ik heb al veel over de microcontroller gezegd. Met de vele functies en taken die deze MCU moet verzorgen, is het heel opmerkelijk dat ik een geavanceerde 8bit-microcontroller (Microchip PIC18F46Q71 [1]) heb gekozen,



Figuur 10. Instelbare totale versterking.



Figuur 11. Numeriek bestuurd oscillator (NCO).



Figuur 12. Het brein van de tijdseinontvanger: de PIC18F46.

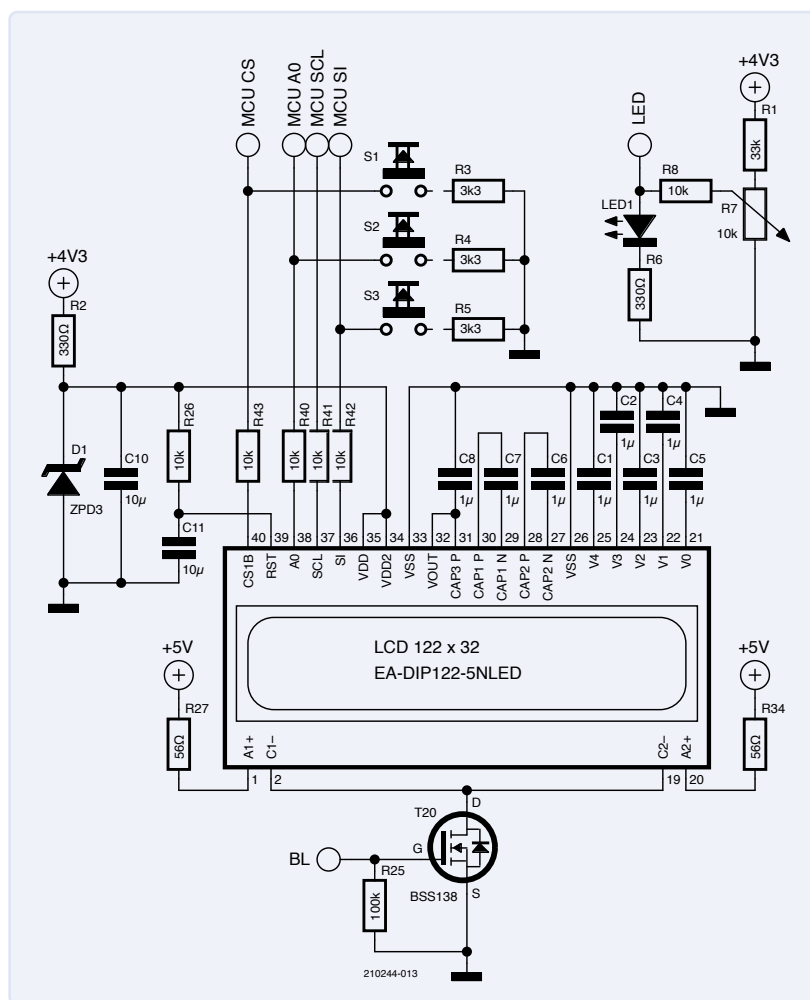
in plaats van een krachtig 32bit-exemplaar. De PIC18F46Q71 is in elk geval verkrijgbaar in een zelfbouwvriendelijke plastic DIP-behuizing. Hij is rijk uitgerust met periferie, waar ik uitgebreid gebruik van heb gemaakt. Tot de buitengewone functionaliteit in zijn klasse behoren onder meer een 12bit-ADC met een bemonsteringsfrequentie van 300 kHz, twee geïntegreerde opamps elk met een versterking/bandbreedte-product van 5,5 MHz, een meerkanaals DMA-controller, meerdere snelle PWM-kanalen met dode tijd-generator, twee UART's en een numeriek bestuurd oscillator (het hart van de ontvangerschakeling). Hij bevat nog meer interessante periferie, zoals programmeerbare logische cellen (die bijvoorbeeld gebruikt kunnen worden voor het genereren van een kwadratuursignaal met een faseverschuiving van 90 graden), twee snelle comparatoren en een hardware CRC-eenheid, die we in dit project niet gebruiken.

De microcontroller is ook uitgerust met een hardware-vermenigvuldiger, zodat hij voldoende rekenvermogen heeft voor ingewikkelde berekeningen, zoals het draaien van een Hilbert-filter om de beeldfrequentie te onderdrukken.

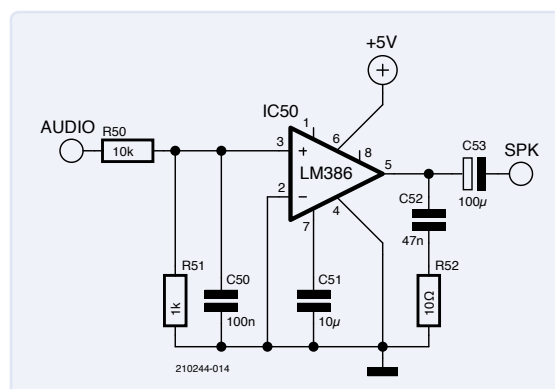
In **figuur 12** zie je de microcontroller met al zijn ingangs- en uitgangssignalen. ST3 is de ICSP-programmeerpoort, en meerdere I/O's zijn naar connector ST1 geleid om te debuggen. Hiermee kan seriële data van de UART worden verstuurd, bijvoorbeeld om het ingestelde bereik van de preselector weer te geven, of voor het configureren van extra PWM-signalen met hoge resolutie voor het debuggen van de individuele trappen van de SDR-demodulator.

Gebruikersinterface

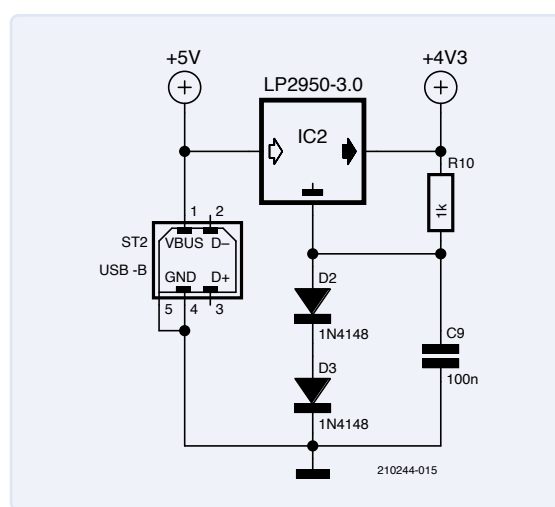
Natuurlijk moet de ontvanger ook een gebruikersinterface in de vorm van een klein grafisch display en drukknoppen voor de bediening hebben, plus een potentiometer (**figuur 13**). Ik heb gekozen voor één van de bekende *Display-On-Glass*-modules (DOG), met een



Figuur 13. De gebruikersinterface bestaat uit een display, een LED en enkele drukknoppen.



Figuur 14. Audio met de oude vertrouwde LM386.



Figuur 15. De 5V-voeding van de USB-adapter wordt teruggebracht tot 4,4 V voor het digitale gedeelte...

SPI-interface verbonden met de microcontroller. Her is wel irritant dat de SPI-interface van het display éénrichtingsverkeer heeft: we kunnen data naar het display schrijven, maar teruglezen is niet mogelijk. Daarom moet een grafische afbeelding eerst worden opgebouwd in het RAM-geheugen en dan periodiek naar het display worden gestuurd. Ook daarvoor gebruik ik weer een periferie-element van de MCU, in dit geval de DMA-controller. Die stuurt de inhoud van het displaygeheugen periodiek naar het display via de SPI-interface. Dit gebeurt helemaal op de achtergrond; het doet geen beroep op de rekenkracht van de CPU. Het totale stroomverbruik van het display is slechts een paar honderd μA , dus ik kon de extra voedingsspanning van 3,3 V implementeren met een zenerdiode. Voor de conversie van de signaalniveaus gebruik ik serieweer-

standen in combinatie met de klemdiodes in het display.

Audio

Omdat er ook enkele radiostations binnen het ontvangstbereik vallen (bijvoorbeeld BBC World op 198 kHz), heb ik nog een luidspreker en een versterker toegevoegd. Ik heb zoveel mogelijk gebruik gemaakt van de periferie in de microcontroller voor het implementeren van de versterker. Aanvankelijk had ik gekozen voor een klasse-D ontwerp, met uitgangstransistoren die door de microcontroller in- of uitgeschakeld werden. Interferentie met de ontvanger kon ik vermijden door een geschikte klokfrequentie te kiezen met zo weinig mogelijk harmonischen in het ontvangstbereik. Maar wat zo mooi werkte bij de implementatie van de spanningsboost (zie onder) bleek hier

niet mogelijk te zijn, misschien vanwege het aanzienlijk grotere vermogen of de behoorlijk lange kabels naar de luidsprekers, of allebei. Dus ik heb toch maar een conventionele analoge versterker gebouwd rondom de oude, vertrouwde LM386. Het audiosignaal wordt eerst digitaal verwerkt door de microcontroller, omdat het in digitale vorm aanwezig is als I/Q-componenten na de A/D-omzetting. Als output worden de berekende samples naar een PWM-eenheid met een hoge klokfrequentie geleid, waarna ze via een laagdoorlaatfilter naar de ingang van het versterker-IC gaan (figuur 14). De gebruikte PWM-eenheid kan rechtstreeks uit de 64MHz-systeemklok worden aangestuurd, wat een PWM-frequentie van 15,625 kHz ($64 \text{ MHz} / 2^{12}$) met een resolutie van 12 bit oplevert. Daarmee kan naast de audio-output ook het volume worden bepaald.

De gekozen luidspreker meet 40×40 mm.

Voeding

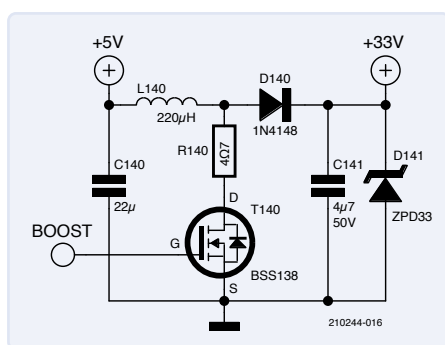
Over de voeding kunnen we kort zijn (**figuur 15**): het was de bedoeling om de hele ontvanger te voeden vanuit één USB-poort. Omdat de spanning op de USB-poort niet zo goed gestabiliseerd is, maakt de 3V-LDO-spanningsregelaar IC2 eerst een schone voedingsspanning van de USB-spanning voor het digitale deel van de signaalverwerking. Het massapotentiaal van de spanningsregelaar wordt opgetild door twee siliciumdiodes, zodat de uitgangsspanning (VDD) 4,3 V is. Dat is laag genoeg om spanningsvariaties op de USB-poort te compenseren en hoog genoeg voor de referentiespanningsbronnen in de microcontroller.

Maar nu terug naar het begin: voor de afstemspanning moet de USB-spanning van 5 V worden opgekrakt naar een veel hoger niveau. Dat gaat met een niet-gestabiliseerde boost-converter met een verrassend eenvoudig ontwerp (**figuur 16**). Het PWM-sig-naal om T70 aan te sturen wordt door de microcontroller gegenereerd. De zenerdiode begrenst de uitgangsspanning tot een maximum van 33 V. Deze schakeling is niet erg efficiënt, maar het stroomverbruik is laag en wordt in feite alleen bepaald door de ruststroom van de opamp, dus het rendement is hier niet zo belangrijk.

Antennes

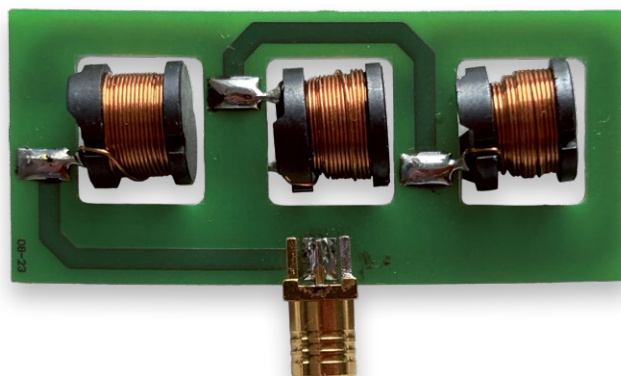
De antenne wordt met de ontvanger verbonden door een stel SMB-connectors. Zodoende kan de antenne worden gedraaid, zodat we hem zo goed mogelijk op de zender kunnen richten.

Voor eerste ontvangstexperimenten gebruikte



Figuur 16. ...en opgevoerd naar 33 V voor de afstemspanning.

Figuur 17. Zelfbouw-lusantenne.



Figuur 18. Goede ontvangst met vaste spoeltjes in een ongebruikelijke configuratie.

ik een ferrietantenne uit een oude radiowekker, die ik had vastgelijmd op een stuk printplaat. Natuurlijk moest ik eerst de afstemcondensator verwijderen, want het afstemmen gaat nu met de preselector.

Omdat kant-en-klare ferrietantennes en zelfs ferrietstaven steeds moeilijker te vinden zijn, en omdat niet iedereen dol is op het wikkelen van spoelen, ben ik op zoek gegaan naar alternatieve antennes. Op de uitverkooppagina van een elektronica-postorderaar vond ik een lusantenne en die heb ik op een stuk print gelijmd. De ontvangstresultaten waren vergelijkbaar met die van de ferrietantenne. Daarom heb ik een print ontworpen met inkepingen als hulp bij het wikkelen (**figuur 17**) en zelf de antenne gemaakt. Ik moest heel dun geëmailleerd koperdraad gebruiken om een antenne met het juiste aantal windingen te maken. Het was een hele klus, maar het is te doen.

Tussen mijn voorraad onderdelen vond ik een zakje spoelen dat ik een tijdje geleden op de kop had getikt. Zou ik daar ook een antenne mee kunnen maken? Door een paar van die spoelen in serie aan te sluiten, kreeg ik iets dat leek op een ferrietantenne, maar dan in een paar stukken (**figuur 18**). Het bleek dat ik ook

hiermee vergelijkbare ontvangstresultaten kon bereiken. De signaalamplitude was een stuk lager, maar de ontvangstkwaliteit was verrassend goed: ik kon DCF77 gemakkelijk ontvangen over een afstand van 800 km. Ik heb het geprobeerd met de spoelen in een radiale en axiale opstelling, en dat werkte allebei. Door de beide varianten te combineren (afwisselend radiale en axiale richting), zou het mogelijk moeten zijn om een omnidirectionele antenne te maken. Als bijvoorbeeld de veldlijnen door de axiaal gerichte spoelen op een maximum zijn, zien de radiaal gerichte spoelen vrijwel geen veldlijnen, en andersom.

Software

De software werkt in feite als een conventionele SDR, dat betekent dat de I/Q-componenten worden gebruikt voor de demodulatie van het signaal. Hoe dat werkt is al vaak beschreven in Elektor [2] [3], dus daar hoeven we nu niet dieper op in te gaan. Het codefragment in het kader **Twee ontvangers zijn beter dan één** is daarentegen echt nieuw is en verdient enige extra aandacht.

Deze schakeling maakt gebruik van veel van de geïntegreerde periferie, bijvoorbeeld voor het genereren van de lokale oscillatorfre-

Twee ontvangers zijn beter dan één

De schakeling was eigenlijk alleen ontworpen voor het ontvangen van data- en tijdseinuitzendingen op de lange golf, maar de mengtrap heeft een tweede schakelbare ingang. Daarom heb ik besloten om een extra preselector met een eigen antenne toe te voegen. We kunnen daar componenten plaatsen voor lange- of middengolfontvangst. Zoals bleek bij het ontwikkelen van de software, kunnen we dan zelfs signalen op twee verschillende frequenties tegelijk ontvangen.

Helaas heeft de microcontroller die ik hier gebruikt heb maar één NCO, dus ik moest mijn toevlucht nemen tot een truc om de tweede lokale oscillatorfrequentie te genereren. Ik heb één van de twee universele 16bit-timers gebruikt om de laagste lokale oscillatorfrequentie te genereren. Hoewel de timer wordt geklokt door de 64MHz-systeemklok, is het mogelijk om redelijk dicht bij de gewenste lokale oscillatorfrequentie te komen. Bij 77,5 kHz (DCF77) is de fout bijvoorbeeld slechts 200 Hz. Als de tweede universele timer wordt gebruikt voor het genereren van de samplefrequentie voor de middenfrequentie, kan die worden geregeld tussen 2000 Hz en 2200 Hz met een resolutie van ongeveer 1 Hz, om te compenseren voor de eerder beschreven afwijking. De NCO van de andere ontvangerschakeling moet worden aangepast aan de resulterende nieuwe middenfrequentie.

Het kristal heeft natuurlijk geen absolute nauwkeurigheid. Als we uitgaan van een onnauwkeurigheid van 50 ppm, dan is de fout bij 77,5 kHz 4 Hz. Dat is te veel voor het zeer smalbandige filter voor de decodering. Toch kan de fout worden bepaald door software. Als het signaal van de gewenste zender met een voldoende signaalniveau ontvangen kan worden, kan de frequentiecorrectie automatisch plaatsvinden. De software moet gedurende vier perioden meten of de I/Q-vector linksom of rechtsom draait. Als dat over een bekende periode wordt gedaan, geeft het een indicatie over hoeveel de werkelijke lokale oscillatorfrequentie verschilt van de gewenste frequentie. Een besturingsfunctie op een hoger niveau kan dan de middenfrequentie reduceren of verhogen om het verschil dicht bij nul te brengen. De listing hiernaast laat zien hoe in C kan worden geïmplementeerd.

```
#define FILTERDEPTH 25
typedef struct {
    int16_t Ihistory[FILTERDEPTH];
    int16_t Qhistory[FILTERDEPTH];
    int32_t Iaverage;
    int32_t Qaverage;
    uint8_t index;
    uint8_t lastQuadrant;
    int8_t phaseChange;
} IQ;
IQ *iq;
void IQcallback( int16_t I, int16_t Q ) {

    uint8_t quadrant =0;

    // moving average lowpass filter I
    iq->Iaverage -= iq->Ihistory[iq->index];
    iq->Ihistory[iq->index] = I;
    iq->Iaverage += I;

    // moving average lowpass filter Q
    iq->Qaverage -= iq->Qhistory[iq->index];
    iq->Qhistory[iq->index] = Q;
    iq->Qaverage += Q;

    iq->index++;
    if( iq->index >= FILTERDPHT ) {
        iq->index =0;
    }
    // integrate phase shift > phase error >
    // estimate frequency offset of crystal
    if( iq->Qaverage<0 ) quadrant =1;
    if( iq->Iaverage<0 ) quadrant ^= 3;
    if( iq->lastQuadrant == ( quadrant+1 ) & 3 ) )
        iq->phaseChange--;
    if( iq->lastQuadrant == ( quadrant-1 ) & 3 ) )
        iq->phaseChange++;
    if( iq->phaseChange < -90 )
        iq->phaseChange = 90;
    if( iq->phaseChange > 90 )
        iq->phaseChange = -90;
    iq->lastQuadrant = quadrant;
}
```

quentie voor de mengtrap en de DMA-controller voor het display, dus de belasting van de microcontroller zelf is zeer beperkt. Daardoor heeft zelfs deze 8bit-microcontroller voldoende power voor het demoduleren van het ontvangen signaal.

De laatste schakel in de keten is de GUI, die nieuw berekende waarden weergeeft op het display.

Vooruitblik

Het moge bekend zijn: na het bouwen van het eerst prototype komen er massa's ideeën over hoe het beter zou kunnen bij een tweede

aanloop. Toch heb ik besloten aan mijn eerste benadering vast te houden en de schakeling te implementeren zoals gepland. Als ik hem opnieuw zou moeten ontwerpen, zou ik dit anders doen:

- een draai-encoder toevoegen aan de gebruikersinterface voor het instellen van de frequentie;
- de PIC18F26Q71 (het kleine broertje van de PIC18F46Q71) proberen, maar dan twee stuks. Dat neemt nauwelijks meer plaats in op de print, maar dan hebben we twee stuks NCOx, vier

opamps, en vier comparatoren. Vanuit een HF perspectief zou dit een betere print-layout mogelijk maken. Er is dan wel communicatie nodig tussen de twee microcontrollers, maar die zou beperkt kunnen blijven. Eén CPU zou kunnen zorgen voor de digitalisering en het verwerken van de ontvangen data, terwijl de andere de GUI ondersteunt. Omdat beide microcontrollers dezelfde klok zouden gebruiken, zou een one-wire UART-interface op Mb/s-tempo goed mogelijk zijn.

- de SMB-connector, die bijna onbetaal-

baar zijn geworden, vervangen door ouderwetse jack-pluggen. Een 2,5mm-jack zou heel geschikt zijn, en de antenne kan dan nog steeds gedraaid worden. Er zijn ook drie- en vierpolige jacks en -pluggen verkrijgbaar, waarmee het zelfs mogelijk zou zijn om een deel van de schakeling, zoals de afstemcondensatoren, te verplaatsen naar de antenneconstructie.

- de preselectoren aanpassen om ook signalen op de middengolf (AM-omroep-frequenties) te ontvangen of gebruik voor amateurradio mogelijk wordt. Praktische experimenten hebben aangetoond dat de mengtrap tot in het MHz-gebied bruikbaar is.

Je ziet: SDR hoeft niet per se te worden geïmplementeerd met snelle 32bit-microcontrollers, die vaak alleen maar verkrijgbaar zijn in behuizingen die moeilijk met de hand te solderen zijn. Door slim gebruik te maken van de periferie van een 8bit-microcontroller, is het mogelijk één of meer softwarefuncties 'uit te besteden' en zo een gemakkelijk te implementeren ontwerp te krijgen.

Zoals altijd staat of valt een project met het aantal deelnemers dat een bijdrage levert. Vooral het implementeren van de software kost veel tijd. Maar misschien heeft iemand zin om dit project vooruit te helpen? Laat het ons weten! 

210244-03 (vertaling: Evelien Snel)

Vragen of opmerkingen?

Hebt u technische vragen of opmerkingen naar aanleiding van dit artikel? Stuur een e-mail naar de auteur via marco@happy-electrons.com, of naar de redactie van Elektor via redactie@elektor.com.

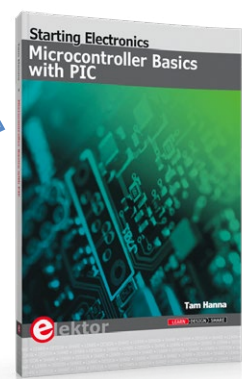
Over de auteur

Marco is geboren in 1970 en was al van jongs af aan geïnteresseerd in elektronica. Hij werkte aan elektronica in zijn tienerjaren, studeerde ooit elektrotechniek, en is 35 jaar actief geweest in onderzoek en ontwikkeling. Hij werkt nu voor een halfgeleiderfabrikant in de Verenigde Staten. Marco is ook een radioamateur; zijn callsign is DD9DD.



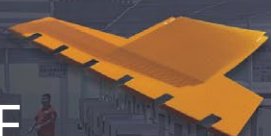
Gerelateerde producten

- **Bert van Dam, PIC Microcontrollers (Elektor 2008)** (eBook, PDF) www.elektor.nl/18093
- **Tam Hanna, Microcontroller Basics with PIC (Elektor 2020)** (Paperback) www.elektor.nl/19188 (e-Book, PDF) www.elektor.nl/19189



WEBLINKS

- [1] MCU PIC18F46: <https://microchip.com/en-us/product/PIC18F46Q71>
- [2] Martin Ossmann, "Software Defined Radio met AVR", zesdelige serie, Elektor maart 2012 tot en met oktober 2012: <http://www.elektormagazine.nl/magazine/elektor-201203/16514>
- [3] Martin Ossmann, "SDR-radioklokken", Elektor januari 2023: <http://www.elektormagazine.nl/magazine/elektor-290/61471>



JOUW ULTIEME PARTNER VOOR HET PRODUCEREN VAN PCB'S!

WAAROM KIEZEN VOOR PCBWAY?

www.pcbway.com
service@pcbway.com

- **TOPKWALITEIT PCB'S**
Onze geavanceerde technologie zorgt voor foutloze PCB's voor jouw projecten.
- **SUPERSNELLE DOORLOOPTIJD**
Zo kort als 24 uur voor de PCB-prototyping service.
- **GEMAKKELIJK BESTELLEN**
Upload eenvoudig jouw ontwerp, en wij regelen de rest!
- **TOEGEWIJDE KLANTENSUPPORT**
Ons deskundige supportteam zorgt voor een naadloze ervaring van begin tot eind.