

Raspberry Pi Pico als spectrumanalyzer

FFT's met goedkope hardware

Prof. dr. Martin Ossmann (Duitsland)

Dit artikel laat zien hoe je een software-spectrumanalyzer kunt bouwen met een eenvoudige Raspberry Pi Pico. Een basisversie met een resolutie van 12 bit en een samplesnelheid van 500 kS/s is zeer geschikt voor audiometingen. Met een externe ADC kan zelfs 50 MS/s worden bereikt!

Bij gebruik van de geïntegreerde ADC van de Raspberry Pi RP2040-microcontroller worden analoge signalen met maximaal 500 kS/s en met 12bit-resolutie bemonsterd. In het eenvoudigste geval wordt een PC gebruikt voor de eigenlijke frequentie-analyse en de weergave van de resultaten. Als eerste uitbreiding kan een LCD worden aangesloten, waardoor je al een zelfstandige analyzer hebt. Met een externe ADC kan in de volgende stap maximaal 50 MS/s worden gehaald, waardoor de analyzer geschikt is voor signalen tot 25 MHz.

500 kS/s aangesloten op PC

Voor de eerste experimenten wordt de Raspberry Pi Pico bedraad zoals getoond in **figuur 1**. R2 en R3 verleggen het nulpunt van de ADC naar de helft van het ingangsbereik. C1 verwijdert DC-componenten uit het ingangssignaal. De gebruikte ingang, ADC2, benut pin 34 samen met GPIO28. T1 wordt samen met R1 gebruikt om een TTL-niveau in te stellen voor data-uitgang naar de PC via de seriële interface, met 115.200 baud. Aangezien de schakeling vrij eenvoudig is, kan deze gemakkelijk op een breadboard worden opgebouwd, zoals te zien is in **figuur 2**.

De ADC werkt om te beginnen met de standaard bemonsteringsfrequentie van 500 kHz. De 12bit-monsters worden naar een FIFO-unit gestuurd, vanwaar ze in een buffer terechtkomen die N waarden kan opslaan. N heeft meestal een waarde van 1.024. Als de buffer vol is, worden de waarden serieel naar de PC gestuurd.

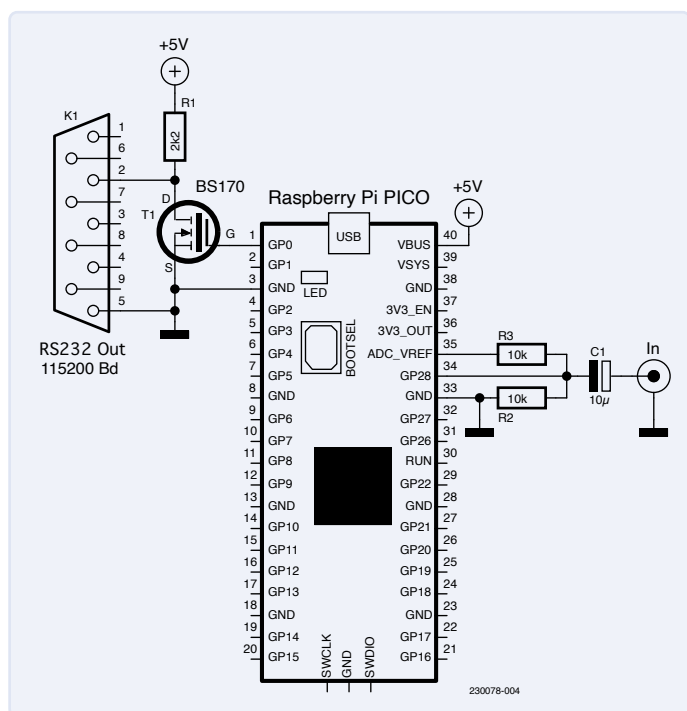
DFT en FFT

Het spectrum wordt berekend met behulp van de discrete fouriertransformatie (DFT). Dit algoritme neemt N samples s_n (met $n = 0$ tot $N-1$) van het ingangssignaal om N spectrumwaarden Z_k (met $k = 0$ tot $N-1$) te berekenen. De hier toegepaste transformatie luidt:

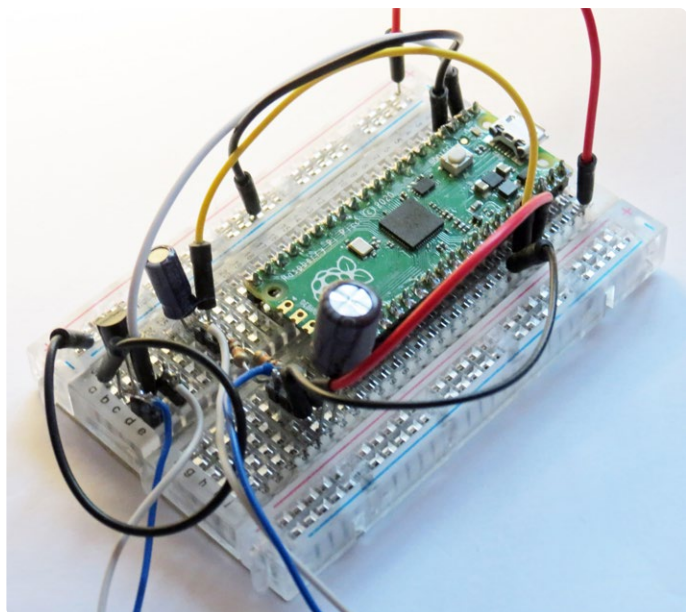
$$Z_k = \sum_{n=0}^{N-1} s_n e^{-j \frac{2\pi i k n}{N}} = \sum_{n=0}^{N-1} s_n \left[\cos\left(\frac{2\pi k n}{N}\right) - i \sin\left(\frac{2\pi k n}{N}\right) \right]$$

De DFT bestaat uit N complexe getallen. Als het ingangssignaal reëel is, zoals in dit geval, is de resulterende DFT symmetrisch en geldt het volgende:

$$Z_{N-k} = \bar{Z}_k$$



Figuur 1. Minimale bedrading van het Pico-board voor gegevensoverdracht naar de PC.



Figuur 2. Mijn breadboard-opstelling van de schakeling uit figuur 1.

Dus alleen de eerste $N/2$ waarden zijn relevant. Dit is in overeenstemming met het bemonsteringstheorema, dat zegt dat unieke waarden alleen worden verkregen tot de helft van de samplesnelheid. Daarom worden in het volgende alleen de eerste $N/2$ waarden gebruikt. Het berekende spectrum is dat van hetingangssignaal S voortgezet met de periode N . Om te begrijpen waarom de DFT eigenlijk een spectrumanalyse uitvoert, bekijken we eerst hoe hetingangssignaal $s_n = \cos(2\pi m n / N)$ wordt getransformeerd met bijvoorbeeld $m = 5$ en $N = 16$. Ditingangssignaal is een cosinus-trilling met de frequentie $f = f_s m / N$.

De resulterende getransformeerde waarden zijn $Z_m = Z_5 = N/2 = 8$, en de andere waarden, Z_k , zijn allemaal nul. Een trilling met frequentie m wordt dus precies merkbaar in de m -de getransformeerde waarde. Als in plaats van het cosinus-signaal een sinus-signaal getransformeerd zou worden, zou het imaginaire deel van Z_m niet nul zijn. Maar wat gebeurt er als we een signaal transformeren dat bestaat uit trillingen van verschillende frequenties? Ook dat is eenvoudig: de DFT is een lineaire transformatie, wat betekent dat de DFT van een som gelijk is aan de som van de DFT's van de afzonderlijke signalen. De resulterende formule is

$$\text{DFT}(u U_n + v V_n) = u \text{DFT}(U_n) + v \text{DFT}(V_n)$$

met de coëfficiënten u en v .

Als hetingangssignaal bestaat uit verschillende trillingen $s_n = \cos(2\pi m n / N)$ met verschillende frequenties m , dan resulteert dit in precies de individuele trillingen in het spectrum. Een signaal $s_n = A \cos(2\pi m n / N)$ met amplitude A resulteert in een waarde $Z_m = A N / 2$ in het spectrum. Deze waarde hangt af van het aantal samples. Om dit te elimineren wordt de DFT geschaald met een factor $2 / N$. Het nieuwe spectrum heeft dan de waarden $W_k = 2 Z_k / N$. Een cosinus-signaal met amplitude A resulteert dan in een spectrumwaarde van precies A . De amplitudewaarde A is de piekwaarde van het cosinus signaal. Voor de effectieve waarde moet deze gedeeld worden door $\sqrt{2}$.

Tabel 1. Rekentijd bij verschillende waarden van N .

N	DFT in s	FFT in s
256	0,590	0,008
1024	9,437	0,041
4096	150,995	0,197
16384	2415,919	0,918

Rekentijd

De rekentijd van de DFT neemt kwadratisch toe met het aantal samples N . Het kan dus behoorlijk lang duren bij een grotere N . Metingen met de Raspberry Pi Pico leidden tot de benadering $T_{\text{DFT}} \approx 9 N^2 \mu\text{s}$. Als N een macht van twee is, kun je de snelle Fourier-transformatie (FFT) gebruiken in plaats van de DFT, omdat die dezelfde waarden veel sneller berekent. Op het Pico-board bedraagt de rekentijd dan $T_{\text{FFT}} \approx 4 N \log_2(N)$. **Tabel 1** toont de rekentijden voor enkele praktische waarden van N .

Voor een eenvoudige spectrumanalyse wordt vaak $N = 1024$ gebruikt. De rekentijd van de DFT zou hier met ongeveer 9 s nog acceptabel zijn, maar de FFT is met 40 ms veel sneller. Soms wordt ook $N = 16.384$ gebruikt, waarvoor de DFT al ruim 40 minuten zou duren – onaantvaardbaar traag. De FFT heeft hier echter minder dan een seconde voor nodig! N wordt daarom altijd als een macht van twee gekozen om de voordelen van de FFT te genieten.

RMS-waarde en dBm

In de communicatietechniek worden grootheden vaak gespecificeerd als effectieve waarden. Omdat hiervoor de tijdsafhankelijke grootheid wordt gekwadeerd, gemiddeld en dan de wortel wordt genomen, is de term RMS (root mean square) hiervoor gebruikelijk geworden. Voor een discreet periodiek signaal met een periode-lengte N (zoals bij onze DFT-signalen) wordt de RMS-waarde als volgt berekend:

$$\text{RMS}_s = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} |s_n|^2}$$

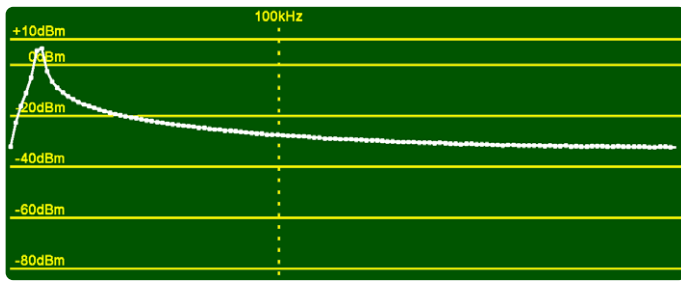
Het is ook mogelijk om de RMS-waarde van de spectrumwaarden te berekenen. We krijgen dan:

$$\text{RMS}_Z = \sqrt{\frac{1}{N} \sum_{k=0}^{N-1} |Z_k|^2} \quad \text{en} \quad \text{RMS}_W = \sqrt{\frac{1}{N} \sum_{k=0}^{N-1} |W_k|^2}$$

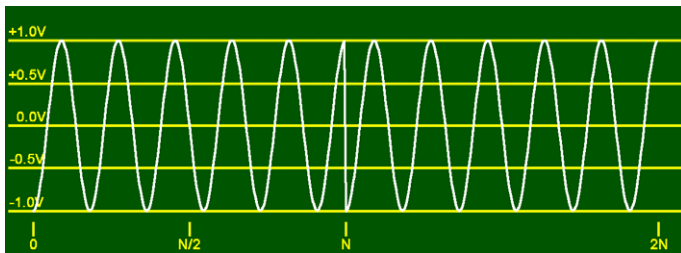
Interessant is dat nu het volgende geldt:

$$\text{RMS}_s = \text{RMS}_Z / \sqrt{N} \quad \text{of} \quad \text{RMS}_s = \sqrt{N} \cdot \text{RMS}_W / 2$$

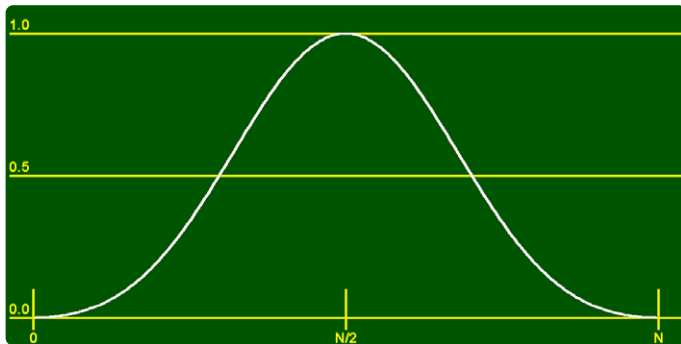
Dit betekent dat de RMS-waarde ook kan worden berekend in het frequentiedomein in plaats van in het tijddomein. Hiervoor hoeft alleen de coëfficiënt te worden bekeken. Deze relatie wordt ook wel de *gelijkheid van Parseval* genoemd [1]. De energie in het tijdsdomein wordt dus nauwkeurig weergegeven door de som van de energieën van de individuele trillingen in het frequentiedomein.



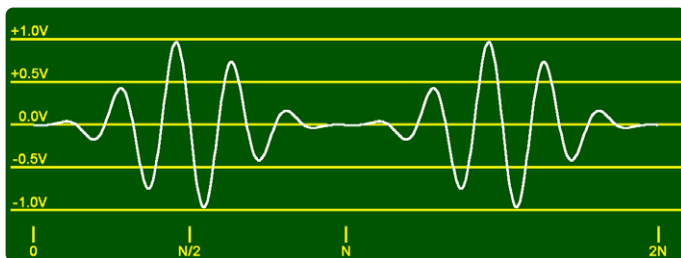
Figuur 3. Slechte spectrumweergave bij niet-heeltallige frequentie.



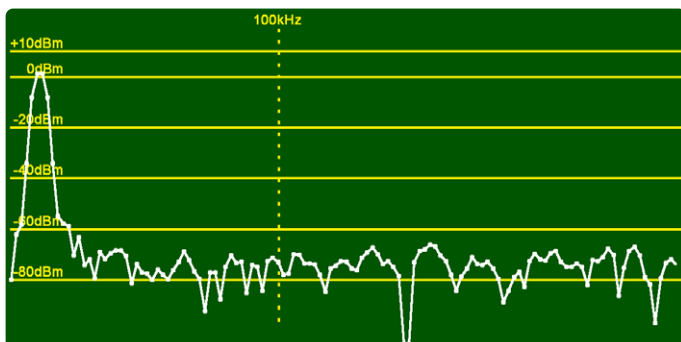
Figuur 4. Bij sample N treedt een amplitudesprong op als het signaal periodiek wordt voortgezet.



Figuur 5. Blackman-vensterfunctie.



Figuur 6. Het signaal vermenigvuldigd met de vensterfunctie verloopt continu aan de randen.



Figuur 7. Spectrum na toepassing van de vensterfunctie.

Energie- of vermogensverhoudingen worden in de communicatietechniek vaak uitgedrukt in dB. Voor de vermogensverhouding $v = P_1 / P_0$ krijgen we $v_{dB} = 10 \log_{10}(P_1 / P_0)$. Voor absolute niveaus is het referentieniveau $P_0 = 1 \text{ mW}$ gebruikelijk. We krijgen dan $P_{dBm} = 10 \log_{10}(P / 1 \text{ mW})$. Als we spanningen U in dBm willen specificeren, gebruiken we het vermogen P in een weerstand van $R = 50 \Omega$, waarop de spanning U_{RMS} wordt aangelegd. Aangezien het door de weerstand gedissipeerde vermogen $P = U^2 / R$ is, krijgen we $U_{dBm} = 10 \log_{10}(U_{RMS}^2 / (R \cdot 1 \text{ mW}))$.

Voor de spectrumanalyzer die we hier beschrijven, wordt de Y-as geschaald in dBm, de gebruikelijke logaritmische weergave voor spectrumanalyzers, maar let op: in de geluidstechniek wordt met dBm vaak een weerstand van 600Ω bedoeld.

Signalen en vensters

Tot nu toe hebben we ingangssignalen beschouwd die precies in een venster van N waarden passen. De frequentie m van het signaal $s_n = A \cos(2 \pi m n / N)$ was daarom een geheel getal. **Figuur 3** toont het spectrum bij $m = 5,5$ en $N = 256$.

Eigenlijk zou je een lijnvormig spectrum verwachten omdat het signaal maar één frequentie bevat. In plaats daarvan krijg je een spectrum dat langzaam afneemt aan beide kanten van het maximum. Dit is natuurlijk nutteloos voor een spectrum analyzer. De reden is dat het periodiek doorlopende signaal (periodeduur N) aan de randen een sprong vertoont. Dit is duidelijk te zien in het amplitude/tijddiagram van twee perioden in **figuur 4**: Bij monster N springt het signaal van $+1 \text{ V}$ naar -1 V . Deze sprong genereert spectraal hoogfrequente componenten buiten de signaalfrequentie. Om deze effecten te vermijden, wordt de venstertechniek toegepast, die in principe gelijk is aan een gesuperponeerde omhullende. Hierbij wordt het ingangssignaal vermenigvuldigd met een vensterfunctie, die het signaal voorzichtig terugbrengt tot amplitude 0 aan de venstergrenzen, maar het signaal verder zo min mogelijk aantast. **Figuur 5** toont de gebruikte vensterfunctie.

Figuur 6 toont twee perioden van het signaal na vermenigvuldiging met de vensterfunctie. Duidelijk is te zien hoe het signaal amplitudemoduleerd is, zodat het niet verspringt op de periodegrenzen ($0, N, 2N, \dots$) maar continu doorloopt. Er zijn verschillende vensterfuncties in gebruik, elk met verschillende voor- en nadelen. De volgende C-listing toont hoe het hier gebruikte Blackman-venster (drie termen) wordt berekend:

```
double windowFun(int k) {
    double alpha=0.16;
    double a0=(1-alpha)/2;
    double a1=0.5;
    double a2=alpha/2;
    return a0-a1*cos(2*PI*k/(N-1))+a2*cos(4*PI*k/(N-1));
}
```

De vensterfunctie is samengesteld uit twee cosinusfuncties. Na toepassing van het venster levert de FFT het spectrum van **figuur 7** op. Het spectrale maximum is vier bins breed rond $m = 5,5$ en daalt daarna aanzienlijk. Met grotere waarden voor N is de spectrale verbreding goed acceptabel en worden goede, lijnachtige spectra

verkregen. We moeten er echter rekening mee houden dat de vensterfunctie de signaalamplitude vermindert. De vensterfactor kunnen we benaderen door het rekenkundig gemiddelde van de venster functiewaarden te gebruiken, zoals in de volgende listing.

```
float getWindowFactor() {
    float mean=0;
    for (k=0; k < N ; k++) {
        mean += abs(windowFun(k));
    }
    return mean/N ;
}
```

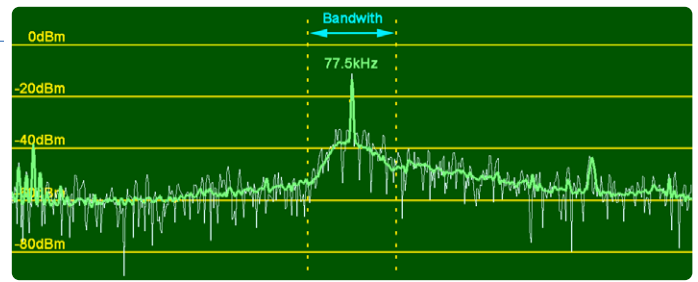
Om de amplitude-afname te compenseren, delen we het spectrum door de vensterfactor nadat we de FFT hebben berekend. Bij gebruik van een vensterfunctie is de gelijkheid van Parseval niet langer van toepassing.

Analyzer-opties

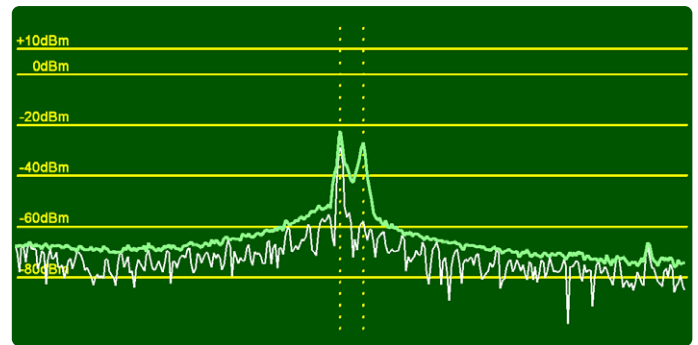
De spectrumanalyzer biedt verschillende configureerbare opties. In het eenvoudigste geval wordt het signaal door de μC bemonsterd, wordt het spectrum berekend, uitgevoerd en eindeloos herhaald. De bemonsteringssnelheid f_s is gelijk aan de samplefrequentie. Tegelijkertijd bepaalt dit het frequentiebereik van het spectrum, want in principe loopt dit van 0 tot $f_s / 2$. Bij 500 kS/s kunnen we dus frequentiecomponenten tot 250 kHz analyseren. De signaallengte N bepaalt hoeveel samples worden gebruikt voor het spectrum en bepaalt dus de resolutie van de FFT. Twee aangrenzende spectraallijnen (bins genoemd) hebben de afstand $\Delta = f_s / N$. Bij 500 kS/s en $N = 1024$ bedraagt de resolutie $\Delta = 500 \text{ kHz} / 1024 \approx 488 \text{ Hz}$. Dat is voldoende om radiosignalen te verwerken.

Bij het analyseren van signalen met ruiscomponenten zijn de individuele spectra vaak erg grillig. Dit kan worden verbeterd door het gemiddelde te nemen van opeenvolgende spectra. Deze procedure wordt geactiveerd door de 'MEAN'-optie van de analyzersoftware, die overigens gratis gedownload kan worden van de projectpagina bij dit artikel [2]. Het spectrum van de DCF77-tijdseinzender kan worden gebruikt om te zien wat het voordeel is. De samplefrequentie is hier 250 kHz en N is 16.384. **Figuur 8** toont het resulterende spectrum in een bereik van $\pm 5 \text{ kHz}$ rond de centrale frequentie van 77,5 kHz. De dunne witte lijn toont een enkel spectrum met veel ruis. De middeling (dikke groene lijn) laat duidelijk zien dat DCF77 een ruisspectrum heeft. Dit komt omdat DCF77 de gegevens verzendt met pseudo-willekeurige ruis met een bitsnelheid van $77,5 \text{ kHz} / 120 \approx 645,8 \text{ Hz}$, naast zeer smalbandige amplitudemodulatie. De bandbreedte van $\pm 645,8 \text{ Hz}$ wordt weergegeven door de twee verticale lijnen.

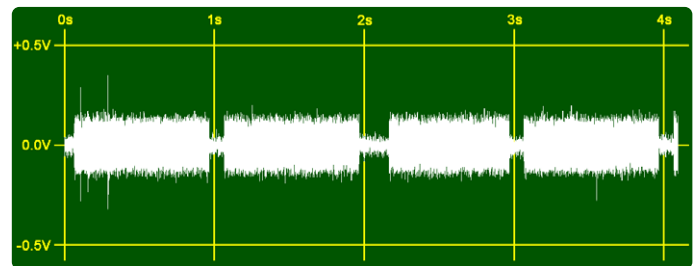
Vervolgens wordt het spectrum van de EFR-zender (129,1 kHz) geanalyseerd. EFR verzendt RTTY-gegevens met een verschuiving van $\pm 170 \text{ Hz}$ en een datasnelheid van 200 Bd. Er zouden dus twee spectraallijnen te zien moeten zijn in het spectrum bij 128,93 kHz en 129,27 kHz. Omdat er echter alleen korte berichten worden verzonden met lange pauzes ertussen, komt de frequentie $\pm 170 \text{ Hz}$ maar zelden voor en is daarom moeilijk te detecteren. Hiervoor biedt de software de functie 'MAXHOLD', waarbij het



Figuur 8. Het 'normale' DCF77-spectrum (wit) en het gemiddelde spectrum (groen).



Figuur 9. Spectrum van DCF49 (EFR op 129,1 kHz) met MAXHOLD-functie.



Figuur 10. Amplitude als functie van de tijd voor het DCF77-sigitaal.

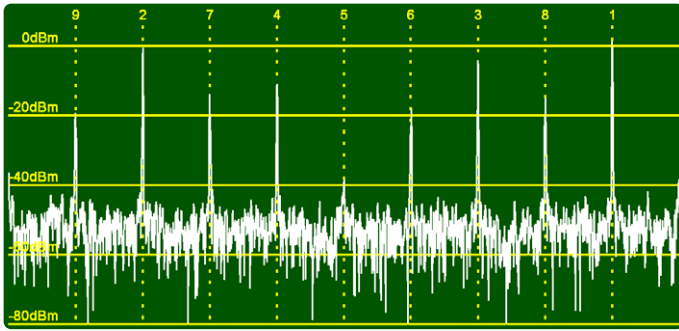
maximum wordt berekend uit vele spectra. Op die manier komen de spectrale componenten die kort voorkomen bij 129,27 kHz beter tot uiting (zie **figuur 9**).

De dunne witte lijn is een enkel spectrum. Aangezien er momenteel geen bericht wordt verzonden, is alleen de piek bij 128,93 kHz zichtbaar. Het MAXHOLD-spectrum wordt weergegeven door de dickere groene lijn. De twee pieken bij $129,1 \text{ kHz} \pm 170 \text{ Hz}$ (bij de verticale stippellijnen) zijn duidelijk zichtbaar. Om zo'n spectrum met hoog oplossend vermogen te verkrijgen, zijn veel samples nodig met $N = 16.384$. De samplesnelheid is 500 kHz en de resolutie is dus $\Delta = 500 \text{ kHz} / 16.384 \approx 30,5 \text{ Hz}$. De twee spectraallijnen zijn dus slechts $340 \text{ Hz} / 30,5 \text{ Hz} \approx 11$ frequentiesamples van elkaar verwijderd.

Tijddomein (oscilloscoopfunctie)

Natuurlijk is het signaal in het tijdsdomein ook informatief, al was het maar om te controleren hoe ver het acquisitiebereik van de ADC benut wordt. De software kan dit ook doen en geeft daartoe het bemonsterde signaal weer. **Figuur 10** toont een soort oscillogram van het DCF77 signaal op 77,5 kHz.

Omdat de draaggolfdips die elke seconde optreden hier van bijzonder belang zijn, is het raadzaam een lagere samplefrequentie en een groter aantal samples te kiezen. Voor de weergave in figuur 10



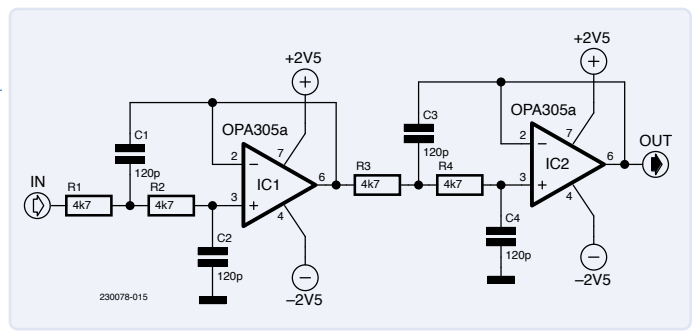
Figuur 11. Subsampling van een 225kHz-blokgolf met $f_s = 500$ kHz. Een aantal spectraallijnen verschijnt bij $f_k = 225 \text{ kHz} * k$ met $k = 1, 2, 3, \dots$

is $f_s = 4$ kHz en $N = 16.384$ gekozen. Het tijdsvenster heeft dus een grootte van $T_s = N/f_s \approx 4$ s, wat resulteert in vier draaggolfdips in figuur 10. Het 77,5kHz-signaal is onvoldoende bemonsterd, maar dat is geen probleem omdat hier alleen de amplitude van belang is.

Bemonsteringstheorema

Volgens het bemonsteringstheorema mogen de te analyseren signalen alleen frequenties bevatten tot maximaal de helft van de samplefrequentie. Ofwel nemen we dit in acht ofwel voegen we een geschikt laagdoorlaatfilter toe vóór de ADC. Tot nu toe is er geen geschikt filter gepresenteerd, omdat dit altijd afhangt van de specifieke toepassing. Verder is het ook interessant om de omstandigheden te bestuderen bij het analyseren van hogere frequenties. Als we samples nemen van signalen met oneindig korte (Dirac-)pulsen, dan krijgen we een periodiek spectrum met periodebreedte = samplesnelheid. De spectraallijnen in het signaal worden dus periodiek herhaald met een interval f_s . Dienovereenkomstig verschijnen frequenties groter dan f_s ook in het spectrum. Ze worden als het ware naar beneden 'gevouwen' in het fundamentele interval. Als de sample-afstand groter is, worden hogere frequenties sterker verzwakt. De relaties worden geïllustreerd in het volgende voorbeeld: de samplefrequentie f_s is 500 kHz. We analyseren een blokgolf met een frequentie van 225 kHz en een duty cycle van 10%.

In het spectrum van **figuur 11** staat de grondgolf op 225 kHz het verst naar rechts en heeft het hoogste niveau. De eerste harmonische ($k = 2$) heeft de frequentie $2 * 225 \text{ kHz} = 450 \text{ kHz}$, wat al voorbij $f_s / 2$ is. Vanwege de symmetrie verschijnt hij daarom op $f = 500 \text{ kHz} - 450 \text{ kHz} = 50 \text{ kHz}$ in het spectrum onder de stippellijn met het label '2' met een nog steeds aanzienlijke amplitude. De frequentie voor q is gelijk aan $225 \text{ kHz} * 9 = 2025 \text{ kHz}$. Omdat dit gelijk is aan $4 * 500 \text{ kHz} + 25 \text{ kHz}$, verschijnt deze piek bij 25 kHz helemaal links in het spectrum onder cursorlijn '9'. De cursor-indices $k = 1 \dots 9$ staan voor de theoretisch verwachte posities van de harmonischen van het geanalyseerde signaal met frequenties van $k * 225 \text{ kHz}$, bekend als 'aliasfrequenties'. Hoewel de frequenties tot meer dan 2 MHz gaan, verschijnen er tien pieken in het venster dat doorloopt tot 250 kHz. Dit betekent dat de ADC zeker hogere frequenties aankan dan $f_s / 2$. Dit effect wordt gebruikt in wat bekend staat als 'undersampling'. Voor het nemen van een monster kan het signaal worden gefilterd door een banddoorlaatfilter met een doorlaatband van $f_s / 2$, zodanig dat alleen de relevante delen van het spectrum worden vastgelegd. Hiervoor zou een snelle sample-and-hold functie aan de opstelling kunnen worden toegevoegd om toch de hoge frequenties met de ADC te kunnen verwerken. Als dit effect echter verstoord wordt, wordt het signaal gefilterd met een laagdoorlaatfilter (antialiasing-filter) om



Figuur 12. Anti-aliasing filter met twee tweede-orde Sallen-Key filters met een afsnijfrequentie van 280 kHz.

signaalcomponenten met een hogere frequentie te onderdrukken voordat een sample van het signaal wordt genomen.

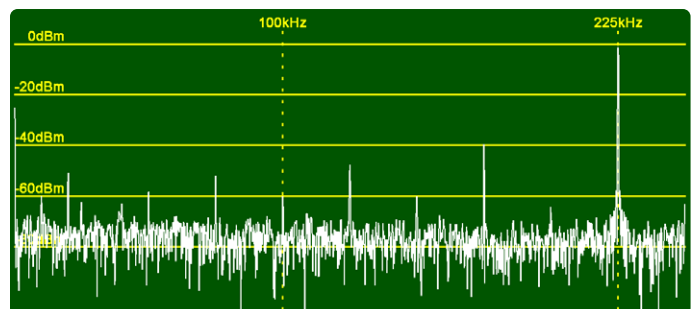
Anti-aliasing filter

Hieronder demonstreren we het effect van een antialiasing-laagdoorlaatfilter. Hier wordt een symmetrische blokgolf van 225 kHz en een duty cycle van 50% vóór de ADC door het laagdoorlaatfilter van **figuur 12** gefilterd. Het filter is gesimuleerd met LTspice [3] en gebouwd met opamps van het type OPA2350.

In het spectrum van het gefilterde signaal (**figuur 13**) is duidelijk te zien dat de harmonischen sterk verzwakt zijn en dat alleen de 225kHz-grondgolf duidelijk boven de -40 dBm uitkomt. Ons filter verzwakt echter al vrij sterk bij 225 kHz. Een steiler filter zou daarom beter zijn.

Ruis en ENOB

Tot nu toe zijn alleen signalen onderzocht die bestaan uit individuele sinustrillingen met verschillende frequentie en amplitude. In de praktijk bestaan er echter ook ruissignalen met verschillende bandbreedte. Hun behandeling in termen van communicatietechniek is niet bepaald eenvoudig, daarom beschrijven we hier alleen eenvoudige gevallen. Als voorbeeld worden ruiswaarden n_k met een amplitude A gegenereerd voor binaire ruis met een ruisfrequentie f_n . Er wordt telkens op toevalsbasis bepaald of $s_k = +A$ of $s_k = -A$ moet gelden voor de volgende waarde. Aanvankelijk geldt $f_n = 1$ MHz, zodat de waarden sneller worden gegenereerd dan de ADC ze kan bemonsteren. De samples zijn dus onafhankelijk van elkaar en de waarde is telkens $\pm A$. Bijgevolg is het ruissignaal bijna 'wit', wat betekent dat alle frequenties in gelijke mate voorkomen in het spectrum. Het signaal wordt gegenereerd door software op de Raspberry Pi Pico. Om deze taak parallel te laten verlopen aan de spectrumanalyzer-software, wordt de tweede kern van de RP2040-CPU gebruikt. Het is geïmplementeerd met behulp van een PIO-toestandsmachine, waarmee maximaal 125 Mb/s haalbaar is.



Figuur 13. Spectrum van het blokgolf na anti-aliasing.

In de regel wordt de RMS-waarde van de ruisspanning gebruikt. Omdat het signaal niet periodiek is, mogen we de RMS-formule niet gebruiken. Voor grote aantallen samples M kan de RMS-waarde echter benaderd worden met:

$$RMS_n \approx \sqrt{\frac{1}{M} \sum_{k=0}^{M-1} n_k^2}$$

Voor binaire ruis kan de RMS-waarde dus toch eenvoudig berekend worden. Omdat elk monster de absolute waarde A heeft, krijgen we de eenvoudige vergelijking $RMS_n = A$. In de testopstelling geldt $A = 750$ mV. **Figuur 14** toont een voorbeeld van dit signaal en **figuur 15** geeft het bijbehorende spectrum. De dunne rode lijn is een enkel spectrum. Het is erg grillig omdat het slechts één enkel spectrum is. De dikke groene lijn is het resultaat van middeling over vele spectra (MEAN-functie), en maakt duidelijker dat dit witte ruis is met een bepaald niveau.

Als we kijken naar ruisspectra met verschillende signaallengtes N , dan zien we dat het weergegeven niveau ook afhangt van N . De volgende relatie bestaat tussen de RMS-waarde, het weergegeven niveau en N : voor een grote N geldt $RMS_s \approx RMS_n$. Vanwege de gelijkheid van Parseval (zie hierboven) geldt ook

$$RMS_w = 2 RMS_s / \sqrt{N}$$

De energie in

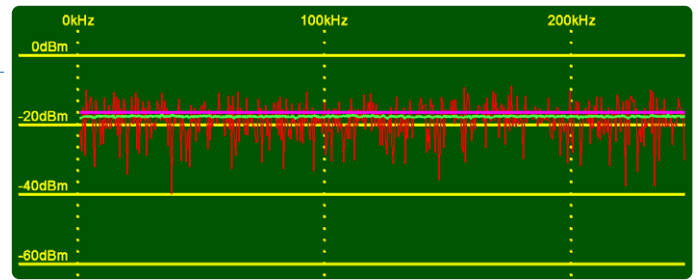
$$RMS_w = \sqrt{\frac{1}{N} \sum_{k=0}^{N-1} |w_k|^2}$$

wordt gelijkmatig verdeeld worden over w_k . Voor de eenvoud wordt aangenomen dat alle w_k even groot zijn en gelijk aan w . Als we dit alles bij elkaar optellen, krijgen we $w^2 = 4 RMS_n^2 / N$. Hier is w precies de gezochte sterkte van het ruissignaal in het spectrum. In **figuur 15** is dit niveau te zien als een dikke paarse lijn. Het is eigenlijk een vrij goede voorspelling van het echte ruisniveau. Het wordt ook duidelijk waarom dit niveau afneemt met grotere N : dat komt omdat de energie van de ruis over meer spectraallijnen wordt verdeeld.

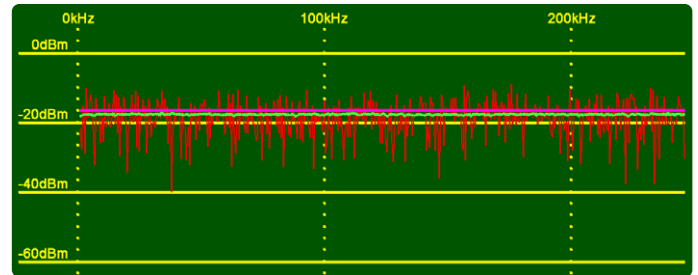
In het volgende voorbeeld wordt de bitsnelheid van de ruisbron gereduceerd tot $f_n = 25$ kHz. Daarom zijn 20 opeenvolgende samples altijd gelijk, wat gelijk staat aan een soort laagdoorlaatfilter. Het resulterende spectrum is te zien in **figuur 16**.

De dunne rode lijn toont opnieuw één enkel monster. Het gemiddelde spectrum is in groen weergegeven. Het eerste nulpunt van het spectrum ligt bij 25 kHz. Het spectrum is evenredig met $H(f) = \text{si}(\pi f / f_n)$ waarbij $\text{si}(x) = \sin(x) / x$ de zogenaamde gap-functie is. Vereenvoudigd kan het niveau rond 0 als volgt worden berekend: de bandbreedte is ongeveer $f_{BW} = 12,5$ kHz, wat overeenkomt met $k_{BW} = N f_{BW} / f_s = 25$ spectraallijnen.

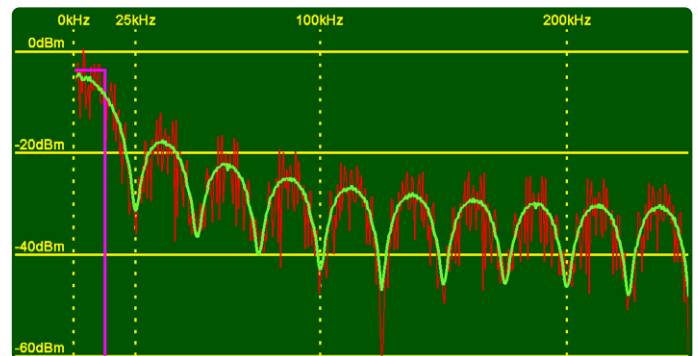
Als het spectrum een constante bandbreedte w heeft, wordt de vergelijking $RMS_w^2 = 2 k_{BW} w^2 / N$ verkregen voor de RMS-waarden door de spectraallijnen binnen de bandbreedte bij elkaar op te tellen. De factor 2 is nodig omdat de FFT een symmetrisch spectrum oplevert met de spectraallijnen $N - k_{BW}$ tot N . Met behulp



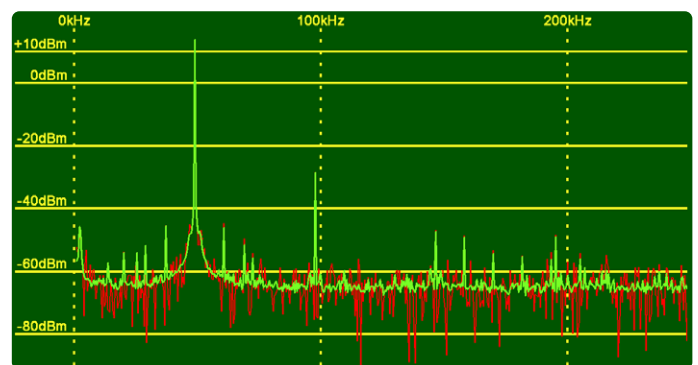
Figuur 14. Binaire ruis als een tijddomein-signaal met een amplitude van ± 750 mV.



Figuur 15. Spectrum van de binaire ruis van figuur 14.



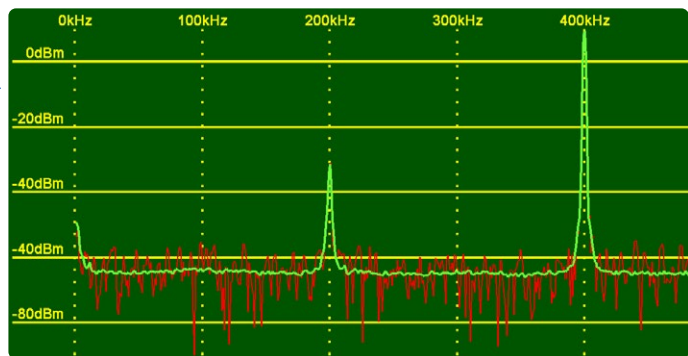
Figuur 16. Spectrum van de ruis bij een bitsnelheid van 25 kHz.



Figuur 17. Spectrum van een sinusmet 48,828 kHz bij een samplefrequentie van 500 kHz.

van de gelijkheid van Parseval wordt spectraal vermogen w als volgt berekend: $w^2 = 2 RMS_n^2 / k_{BW}$. Dit wordt in figuur 16 weergegeven door de dikke paarse lijn en komt goed overeen met het huidige spectrum.

Het volgende voorbeeld is iets praktischer: de ADC krijgt nu het signaal aangeboden van een sinusgenerator met een amplitude van $3 V_{SS}$ (ongeveer 14 dBm). De signaalfrequentie $f_s = 48,828$ kHz komt precies overeen met de 100ste spectraallijn. **Figuur 17** toont het resulterende spectrum.



Figuur 20. Spectrum van een sinus van 400 kHz bij een samplefrequentie van 1 MHz.

LCD

Omdat fijne details vaak van belang zijn in spectra, werd het MAR3502 Arduino Shield gekozen vanwege de relatief hoge resolutie van 320×480 pixels. Het wordt parallel aangesloten op het Pico-board zoals getoond in **figuur 21**. Hierdoor kan een spectrumanalyzer worden opgezet als een stand-alone apparaat – maar de bedieningselementen ontbreken nog.

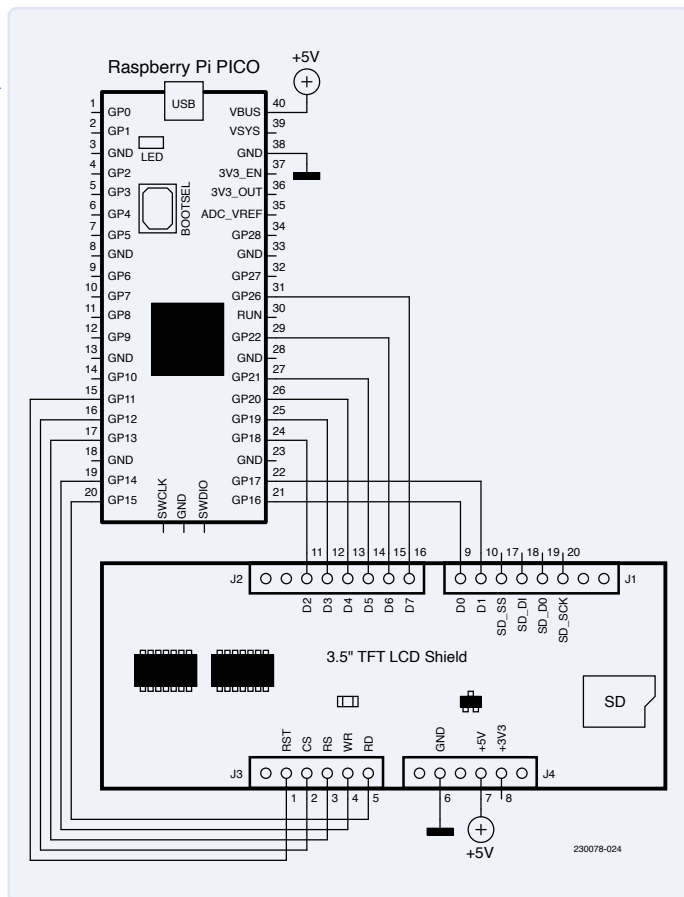
Figuur 20 toont het spectrum van een FM multiplex-audiosignaal op het display. Het monosignaal (L+R) is duidelijk te zien in het bereik tot 17 kHz. Bij 19 kHz verschijnt de spectraallijn van de stereopiloottoon. Bij 38 kHz is het spectrum van het stereo-verschilsignaal (L–R) zichtbaar. Bij 57 kHz is het spectrum van het RDS-signaal te zien, dat bestaat uit twee zijbanden rond 57 kHz.

12-bit, 50 MS/s ADC

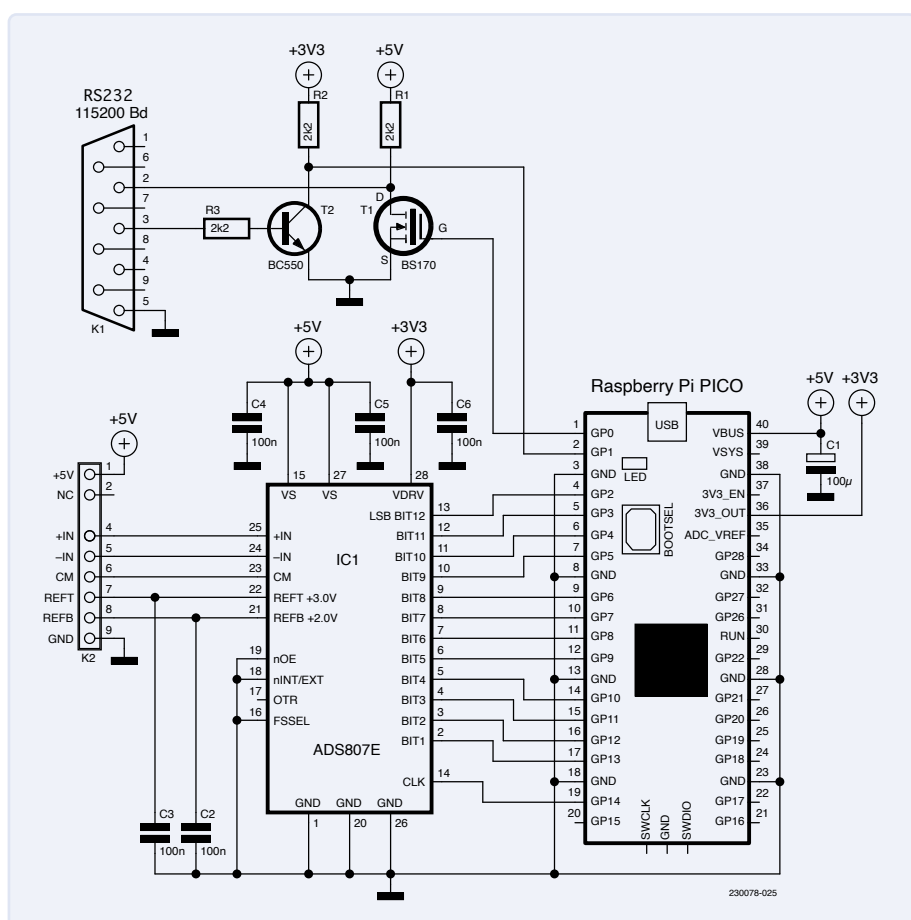
Voor lagere frequenties is de 500kHz-samplefrequentie van de ADC in de RP2040 voldoende, maar er zijn ook interessante spectra in het RF-bereik. Hiervoor kan het Pico-board worden uitgebreid met een snelle externe ADC met de ADS807E-chip. Met deze 12bit-ADC is tot 53 MS/s mogelijk. Voor de spectra heb je veel rekenkracht nodig en dus wat trucs om dit voor elkaar te krijgen. Handig is dat de ADS807E een ingebouwde referentiespanning heeft. De ADC wordt aangesloten op het Pico-board zoals getoond in **figuur 22**.

Door de parallelle aansluiting worden veel GPIO-pinnen in beslag genomen, dus je kunt helaas niet tegelijkertijd het LC-display aansluiten. Daarom krijgt de ADS807 een tweede Pico-board. Tijdens de eerste experimenten hiermee wordt de data verwerkt en weergegeven met de PC zoals in het begin. Op K2 kunnen verschillende analoge front-ends worden aangesloten, die hieronder worden besproken. **Figuur 23** toont de testopstelling van een Pico-board met een ADS807 en een 50Ω-transformator front-end.

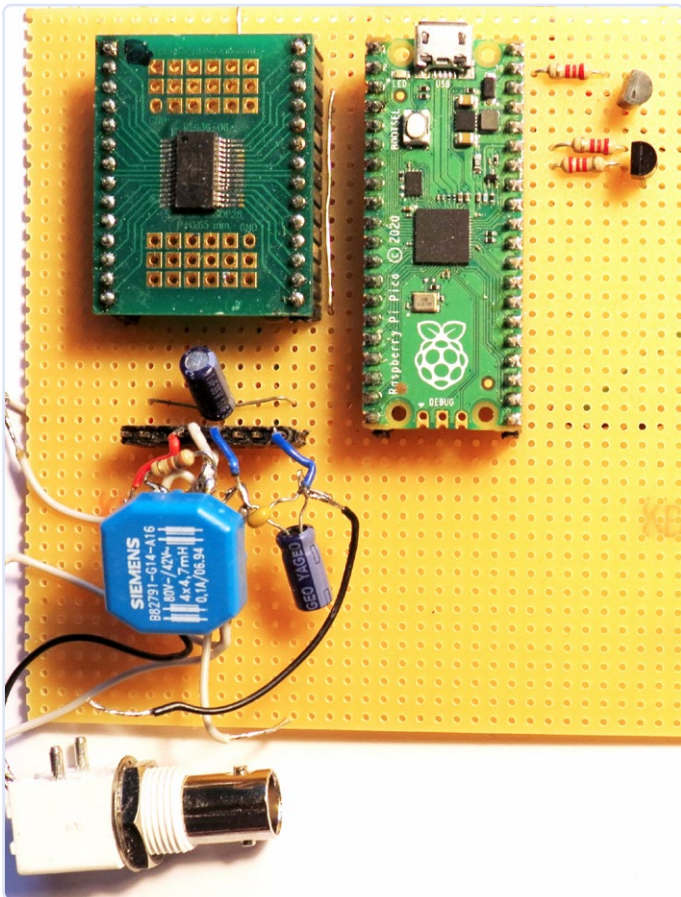
Overigens: als de ADS807 te duur is, of je hebt zijn 12bit-resolutie niet nodig, dan kun je in plaats daarvan een goedkope 8bit-ADC gebruiken, zoals de ADS830E. Deze biedt samplefrequenties tot 60 MS/s en vereist alleen een aangepaste sampleroutine.



Figuur 21. Parallelle aansluiting van een 3,5" 480×320-pixel LCD op het Pico-board.



Figuur 22. Aansluiting van de externe ADC ADS807 op het Pico-board.



Figuur 23. Extra Pico-board met ADS807 en front-end met 50Ω-transformator.

Snelle sampling via PIO, FIFO en DMA

De RP2040-CPU is normaal geklokt op 125 MHz. De volgende listing toont de minimale lus om N gegevenswaarden mee te verwerven:

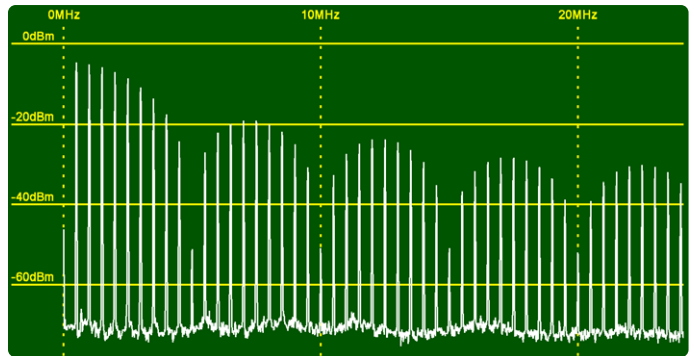
```
for (k = 0; k < N; k++) {
    sampleBuffer[k] = gpio_get_all();
    gpio_put(ADCClock, 0);
    gpio_put(ADCClock, 1);
}
```

Volgens mijn meting haalt de RP2040-CPU maximaal 15 MS/s – nog steeds ver verwijderd van de 53 MS/s van een ADS807. De CPU heeft echter als speciale eigenschap een peripheral input/output-unit (PIO), die uit acht programmeerbare toestandsmachines bestaat. Deze kunnen worden aangestuurd met de klok van de CPU en worden geprogrammeerd met eenvoudige commando's. Hieronder worden enkele aspecten van de sample-toepassing belicht. De volledige details zijn vrij complex, dus raadpleeg hiervoor de documentatie van de RP2040 [4].

Het PIO-programma bestaat uit slechts twee commando's:

```
.wrap_target
    in pins,12    side 0b0
    nop          side 0b1
.wrap
```

De sleutelwoorden `.wrap_target` en `.wrap` zorgen ervoor dat de twee instructies ertussen eindeloos na elkaar worden uitgevoerd. De lus zelf kost geen overhead in het proces. De `in`-instructie zorgt ervoor dat 12 bits worden verplaatst van de ADC naar het

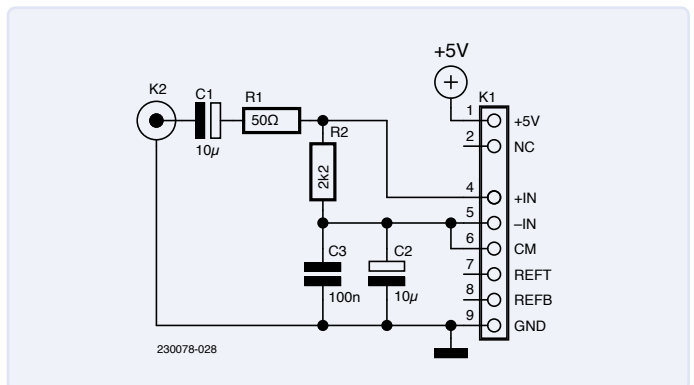


Figuur 24. Spectrum van een 500kHz-blok golf met een duty cycle van 10%.

input shift register (ISR). Zoals we weten, doet de `nop`-instructie niets. Na elke instructie wordt de `side`-optie gebruikt. Hiermee kunnen GPIO-pinnen parallel aan het eigenlijke PIO-commando worden beïnvloed. Op deze manier wordt de ADC-klok gegenereerd op GPIO14. Aangezien elke instructie één cyclus vereist, kan dit programma in principe 125 / 2 MS/s verwerken. Dit is voldoende voor de ADS807. Naast het eigenlijke programma speelt de configuratie van de toestandsmachine een belangrijke rol. Deze bepaalt bijvoorbeeld welke pinnen worden beïnvloed door `side-set`. Daarnaast is hij zo geconfigureerd dat na elke twee samples (als er 24 bits in de ISR zijn aangekomen) de ISR-waarde naar een FIFO-uitgang wordt geschreven. De gegevens worden snel naar het geheugen geschreven via directe geheugentoegang (DMA). De eigenlijke bemonstering verloopt dus volledig autonoom. De CPU hoeft alleen de waarden uit de buffer op te halen na elke DMA-overdracht.

In principe kan dezelfde software die gebruikt is voor de ADS807-ADC, worden benut voor de geïntegreerde ADC. Alleen de sample-routine moet worden aangepast. Om bijvoorbeeld de maximale samplesnelheid van 53 MHz te bereiken, kun je een CPU-klok van 106 MHz instellen en de PIO-prescaler op 1 zetten. Nu draait het PIO-programma op 106 MHz. Aangezien voor elke sample twee klokpulsen nodig zijn, is de samplefrequentie precies 53 MHz. Helaas is de CPU dan ruim 15% langzamer geklokt dan normaal. Dit is echter voldoende om de taken zonder problemen uit te voeren.

Figuur 24 toont het spectrum van een blok golf met een duty cycle van 10% bij een samplefrequentie van 50 MHz. De harmonischen tot 25 MHz zijn duidelijk zichtbaar. Ten gevolge van de duty cycle van 10% ontbreekt elke 10de harmonische. Via subsampling kun je met de ADS807 nog steeds probleemloos signaalcomponenten tot ongeveer 200 MHz analyseren.



Figuur 25. Enkelzijdig front-end.

Analoge front-ends

De differentiële ingangen +IN en -IN van de ADS807E hebben een grote signaal/ruis-verhouding, zelfs bij hogere frequenties. De chip is echter iets lastiger aan te sturen dan ADC's met een differentiële ingang. Mogelijke front-ends die kunnen worden aangesloten op K2 in **figuur 22** worden hieronder beschreven. Het dynamisch bereik van de ADS807 is $2 V_{SS}$, wat resulteert in een resolutie van $2 V / 4096 = 480 \mu V$.

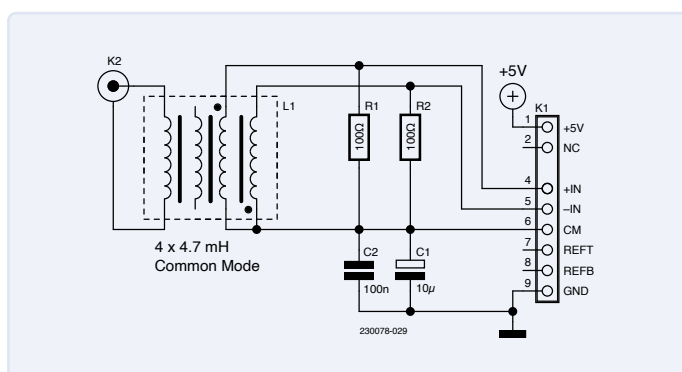
Front-end 1: enkelzijdig

In het eenvoudigste geval negeer je de differentiële ingang, zet je -IN op CM (common mode) en voer je het signaal gewoon naar +IN. Je werkt dan met AC-koppeling en de DC-offset van 2,5 V wordt geleverd door CM (**figuur 25**). Als je eeningangsimpedantie van 50Ω , kun je eenvoudig een 50Ω -weerstand parallel schakelen aan ingang K2.

Front-end 2: transformatorkoppeling met 50Ω

Met behulp van een RF-transformator met drie nauw gekoppelde windingen met hetzelfde aantal windingen kan de differentiële ingang eenvoudig worden gebruikt. Het prototype gebruikte een 4,7 mH common-mode smoorspoel met vier wikkelingen, waarvan er slechts drie zijn gebruikt (**figuur 26**). De blauwe smoorspoel van Siemens is duidelijk te zien in figuur 23.

Telkens één wikkeling voedt de +IN- en -IN-ingangen met tegengestelde fase. De wikkelingen hebben elk een parallelle weerstand van 100Ω . De 1:1-koppeling transformeert deze weerstanden in twee parallel geschakelde 100Ω -weerstand aan de primaire kant, wat resulteert in eeningangsimpedantie van 50Ω . Aangezien de ADC twee keer deingangsspanning ziet, resulteert dit in een versterking van tweemaal. Tegelijkertijd is de ingang elektrisch geïsoleerd van de ADC, wat common-mode-problemen voorkomt. De schakeling is zeer geschikt voor metingen aan 50Ω -systemen. Als je transformator L1 zelf wilt wikkelen, moet je de drie wikkelingen trifilair wikkelen (dat wil zeggen één wikkeling met drie draden naast elkaar) om een goede koppeling te krijgen, zelfs bij hoge frequenties. De kern moet een hoge permeabiliteit hebben, ook bij frequenties tot 25 MHz.

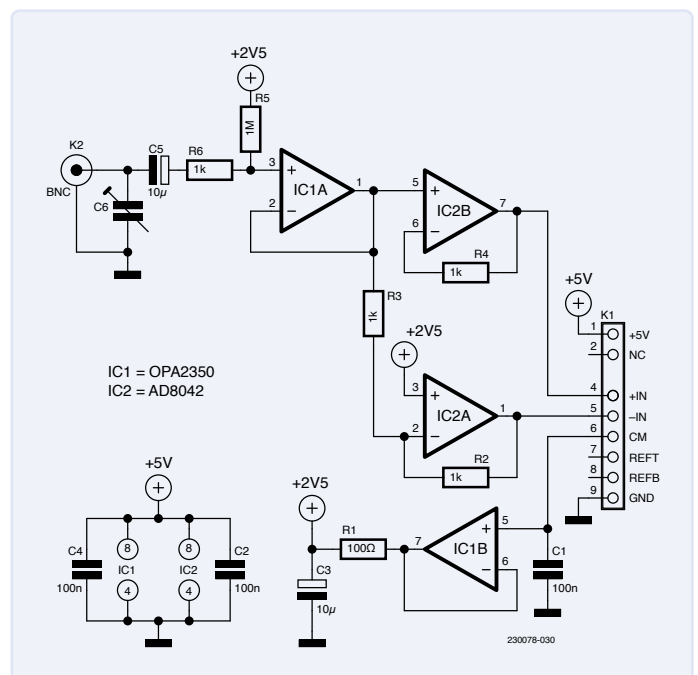


Figuur 26: Transformatorgekoppeld 50Ω front-end.

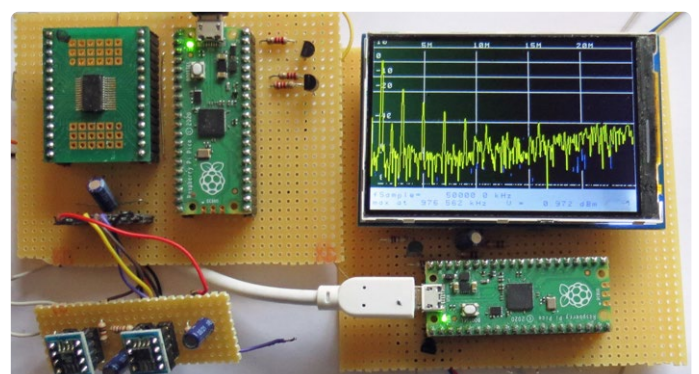
Front-end 3: 25MHz-voorversterker met $1 M\Omega$

Voor bepaalde metingen heeft zelfs een snelle spectrumanalyzer een ingang met hoge impedantie nodig. Dan kunnen de gebruikelijke 10:1-oscilloscoop-probes worden gebruikt. **Figuur 27** toont het schema van een geschikte voorversterker.

Opamp IC1A dient als impedantietransformator met een ingangsweerstand van $1 M\Omega$ dankzij R5. Het type OPA2350 is niet alleen snel, maar heeft ook een geringe biasstroom en ingangswaars. IC2A en IC2B dienen als bipolaire drivers voor de ADC. Omdat de unipolaire ingang wordt omgezet in een differentieel signaal, is het resultaat een versterkingsfactor van twee. IC2 is geselecteerd vanwege de grote bandbreedte, zodat het faseverschil tussen +IN en -IN zo klein mogelijk blijft. R2 en R4 verbeteren de stabiliteit van deze opamps. IC1B buffert de offsetspanning van 2,5 V. **Figuur 28** toont (onder andere) het front-end dat op gaatjesprint is opgebouwd.



Figuur 27: RF-voorversterker met eeningangsimpedantie van $1 M\Omega$.



Figuur 28: Stand-alone spectrumanalyzer met twee Pico-boards.

Ruis

Bepaalde ruisbronnen vergallen het leven van een ontwerper. Bij R5 bijvoorbeeld treedt een thermische ruis spanning op van $U_{N,RMS} = \sqrt{4 k_B T B R}$, waarbij k_B de constante van Boltzmann is, T de absolute temperatuur (in kelvin), B de bandbreedte (hier 25 MHz) en R de waarde van R5 (1 MΩ). Dit resulteert in een effectieve ruis spanning van 6,3 mV.

Dit is enkele malen de resolutie van de ADC en dit ruisniveau is sterk merkbaar bij breedbandige metingen met hoge impedantie. Dit verklaart ook waarom een laagohmig systeem van bijvoorbeeld 50 Ω beter is voor RF. De ingangsstroom van een OPA2350 heeft een ruisdichtheid van 4 fA / √Hz. Dit resulteert in een verwaarloosbare effectieve ruis spanning van 20 μV over R5 bij een bandbreedte van 25 MHz. Voor de ingangsisruis wordt een dichtheid van 7 V / √Hz opgegeven. Dit resulteert in een ruis spanning van 30 μV bij 25 MHz, die ook verwaarloosbaar is. Het is echter duidelijk hoe de ruisparameters van de opamp het ontwerp beïnvloeden. De AD8042 heeft bijvoorbeeld een stroomruis van 500 fA / √Hz en is daarom ongeschikt als impedantietransformator aan de ingang.

Grafisch display

Zoals al opgemerkt is het niet mogelijk om de externe ADC en het LCD tegelijkertijd aan te sluiten door het gebrek aan GPIO-pinnen. Voor een stand alone-apparaat wordt deze tekortkoming gecompenseerd door twee aparte Pico-boards te gebruiken: één voor de data-acquisitie van de externe ADC en één voor het LC-display. Deze laatste dient als een grafische terminal die zelfstandig verschillende basisfuncties kan uitvoeren, zoals het tekenen van lijnen en rechthoeken en tekstuitvoer. Op het Pico-board met de ADS807 zijn de spectrumanalyzer-functies geïmplementeerd. Voor grafische uitvoer stuurt dit onderdeel overeenkomstige commando's naar de grafische terminal via de seriële interface. Bij een prijs van een paar dollar voor een Pico-board is dit ontwerp economisch verantwoord.

Figuur 28 toont hoe deze toepassing eruit zou kunnen zien met twee Pico-boards. Het display toont het spectrum van een 1MHz-blokgolf bij een samplesnelheid van 50 MS/s. De harmonischen worden snel kleiner met toenemende orde. ◀

vertaling: Hans Adams — 230078-03

Vragen of opmerkingen

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

Over de auteur

Martin Ossmann begon op 12-jarige leeftijd Elektor te lezen – en te experimenteren natuurlijk. Na zijn studie elektrotechniek en een aantal jaren als ontwikkeltechnicus te hebben gewerkt, was hij professor aan de afdeling elektrotechniek en informatietechnologie van de FH Aachen University of Applied Sciences. Hij is niet alleen auteur van wetenschappelijke artikelen, maar publiceert ook al meer dan drie decennia regelmatig schakelingen en softwareprojecten met veel technische knowhow in Elektor.



Gerelateerde producten

- > **Joy-IT JDS6600 Signal Generator & Frequency Counter**
www.elektor.nl/18714
- > **OWON HDS242 2-channel Oscilloscope (40 MHz) + Multimeter**
www.elektor.nl/20415
- > **Raspberry Pi Pico RP2040**
www.elektor.nl/19562



WEBLINKS

- [1] Gelijkheid van Parseval: https://nl.wikipedia.org/wiki/Gelijkheid_van_Parseval
- [2] Projectpagina bij dit artikel: <http://www.elektormagazine.nl/230078-03>
- [3] LTspice: <https://tinyurl.com/3zpwzk4y>
- [4] RP2040-datasheet: <https://tinyurl.com/2sf4uvfm>

Word lid van de Elektor Community



Neem **nu** een
lidmaatschap!



- ✓ Een compleet web-archief t/m 1980!
- ✓ 8x Elektor Magazine (Print)
- ✓ 8x digitaal (PDF)
- ✓ 10% korting in onze webshop, en exclusieve aanbiedingen
- ✓ Toegang tot meer dan 5000 Gerberfiles



Ook verkrijgbaar

Het digitale
lidmaatschap!



- ✓ Toegang tot ons web-archief
- ✓ 10% korting in onze webshop
- ✓ 8x Elektor Magazine (PDF)
- ✓ Toegang tot meer dan 5000 Gerberfiles



www.elektormagazine.nl/Abonnement