

RAZZies

Maandblad van de
Radio Amateurs
Zoetermeer



Mei 2025

Met in dit nummer:

- 3D ontwerpen met OpenSCAD
- Opa Vonk: Flux of soldeer pasta
- Pititico 2 40m QRP transceiver
- Een eenvoudige TDR/testsignaalinjector
- PA3CNO's Blog
- Afdelingsnieuws



Colofon

RAZZies is een uitgave van de Radio Amateurs Zoetermeer. Bijeenkomsten van de Radio Amateurs Zoetermeer vinden plaats op elke tweede en vierde woensdag van de maanden september - juni om 20:00 uur in buurthuis 't Span, Sullivanlijn 31 Zoetermeer.

Website:

<https://www.pi4raz.nl>

Redactie:

Frank Waarsenburg
PA3CNO
pa3cno@pi4raz.nl

Eindredactie:

Robert de Kok
PA2RDK
pa2rdk@pi4raz.nl

Informatie:

info@pi4raz.nl

Kopij en op- of
aanmerkingen kunnen
verstuurd worden naar
razzies@pi4raz.nl

Nieuwsbrief:

[https://pi4raz.eo.page/
83stt](https://pi4raz.eo.page/83stt)

Van de redactie

Ik weet niet wat jullie ervaring is, maar de condities waren de afgelopen maand erg wisselend. Er waren dagen dat de 100km MUF niet boven de 6MHz uitkwam en de 3000km MUF bleef soms steken op 16MHz. De k-index lag dan zo rond de 4 à 5 en dat was te merken. Het is altijd wel mogelijk om EEN verbinding te maken, maar het wordt lastig om een SPECIFIEKE verbinding te maken. Normaal gaat Spanje op 20m altijd wel, maar van de Spaanse *IARU stations hoorde ik er niet een, terwijl ik ze toch op het reverse beacon netwerk zag. Verbindingen op 17m met Roemenië lukten dan weer wel, evenals een verbinding op 10m met Oman. Met zwakke signalen, maar het lukte. Er is geen

peil op te trekken wat er nou precies wanneer te werken valt. Maar dat maakt de hobby voor mij nou precies zo uitdagend. Bij een hoge k-index worden de signalen bij mij zwakker, maar het lijkt of de QRM dat ook wordt. Heb ik normaal op een zonnige dag op 20m een permanente S7 aan rotzooi staan, bij een hogere k-index zakt dat 1 tot 2 S-punten, wat het maken van verbindingen dan weer makkelijker maakt. Maar er zijn ook situaties waarbij de verstoring van de atmosfeer dusdanig is, dat ik me serieus afvraag of de antenne er nog wel in zit... De komende tijd komt het sporadische E-seizoen er weer aan, met leuke mogelijkheden op 6 en 4m. Tijd om de historische FT101 met FTV-650 transverter weer eens aan te slingeren...

3D ontwerpen met OpenSCAD

Het moeilijkste van een project is om het mechanisch óók mooi te maken. Tenminste, bij mij is dat zo. Het in elkaar zetten van een schakeling is doorgaans geen probleem, maar het dan ook mechanisch een beetje aantrekkelijk maken is een ander verhaal. Ik ga al bloeden als ik naar gereedschap kijk en als ik dan toch mijn gereedschap pak, pakt de XYL alvast de verbandtrommel. Heb ik dan met bloed (letterlijk), zweet en tranen een behuizing gemaakt, dan ziet het er doorgaans niet heel professioneel uit. Vooral voor dingen die je in het zicht van de XYL wil plaatsen (ons weerstation b.v.) is een beetje fraai uiterlijk wel een

must. Maar tegenwoordig is 3D printen de standaard, dus waarom zouden we daar geen gebruik van maken? Nou hoor ik een hoop lezers al denken: je hebt makkelijk praten, maar ik heb geen 3D printer en er een kopen ligt niet binnen mijn financiële mogelijkheden. Maar wist je dat je bij [JLCPCB](#) (ja, die printenboer) je 3D ontwerpen gewoon uit kunt laten printen? En dat tegen heel schappelijke prijzen. Kwestie van je .stl file opsturen en een week of 2 later ligt je design in de bus. Keurig geprint, mooi gezandstraald en dus zo vlak als een spiegel. En dan wordt het aantrekkelijk om zelf je ontwerpen te maken. Daar is uiteraard een programma voor nodig, en mijn keuze is gevallen op [OpenSCAD](#).

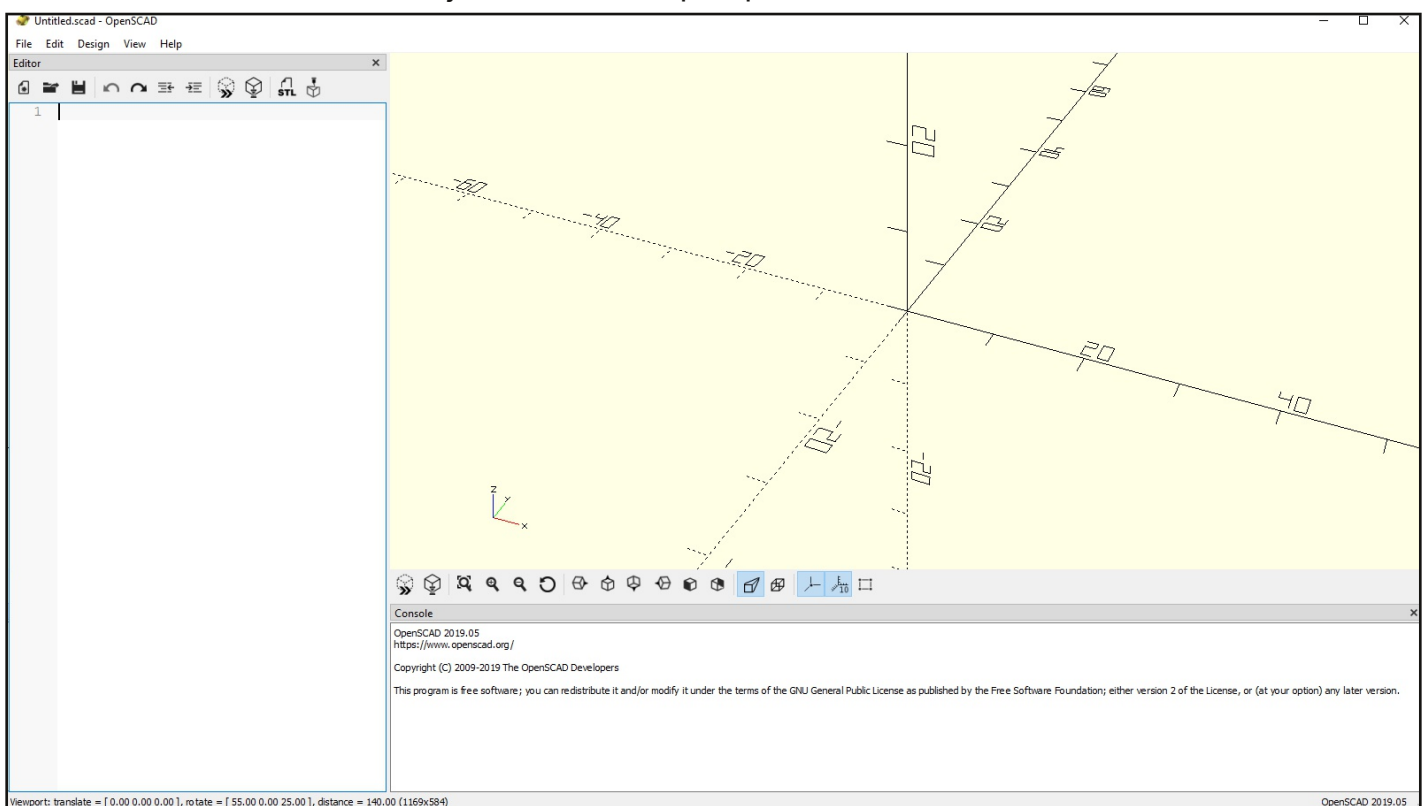
OpenSCAD is gratis, en beschikbaar voor Linux, Windows en Mac OS X. Heb je het programma geïnstalleerd, dan kan je het starten. Je scherm ziet er dan uit zoals de afbeelding onder aan deze pagina. Ik toon nu nog even het hele scherm, maar dat vult natuurlijk aardig de ruimte op de pagina. In het vervolg van dit betoog zal ik de code en het resultaat apart tonen.

Wat zie je op het scherm. De linker kolom, waar de cursor staat, is de plek waar je de code van je ontwerp schrijft. Ja, code. Net als bij Arduino. Maar niet schrikken, het valt allemaal erg mee. Mijn kleinzoon van 10 leert op school 3D ontwerpen met Tinkercad, en dat schijnt wat gebruikersvriendelijker te zijn, maar minder flexibel (volgens hem). Hij is inmiddels net zo handig met OpenSCAD als met Tinkercad. Het grote, wat gelige vlak is waar je ontwerp verschijnt. De assen zijn gelabelled in mm. Links onder in het vlak zie je de oriëntatie van de verschillende assen: X, Y en Z. Je kunt je ontwerp namelijk draaien om het van alle kanten te kunnen bekijken. Dat doe je door op het gele vlak te klikken met de linker muistoets en deze vast te houden. Nu kan je het ontwerp alle richtingen op draaien. Hou je de rechter muistoets vast, dan kan je het ontwerp

verslepen. En met het muiswiel kan je in- en uitzoomen. Dat zijn de basisfuncties voor het bekijken van je ontwerp.

Het vak rechtsonder is het resultatenscherm. Heb je een stuk code getypt, dan moet je dat eerst renderen. Dat betekent het omrekenen van de code naar een vorm. Tijdens het renderen zie je een voortgangsbalk en in het venster kan je zien wat er gebeurt en daar verschijnen ook eventuele foutmeldingen. Het renderen wordt gestart met functietoets F6. Na het indrukken van die toets wordt het gele vlak leeg (als het dat niet al was) en na enige tijd verschijnt je ontwerp in dat vlak. De rendertijd hangt van een aantal factoren af. Uiteraard van de rekenkracht van je computer (mijn wat oude laptop heeft het soms erg zwaar), het aantal objecten dat berekend moet worden maar ook van de nauwkeurigheid waarmee je de objecten laat berekenen. Hoe je daar invloed op kunt uitoefenen, zal ik je zo laten zien.

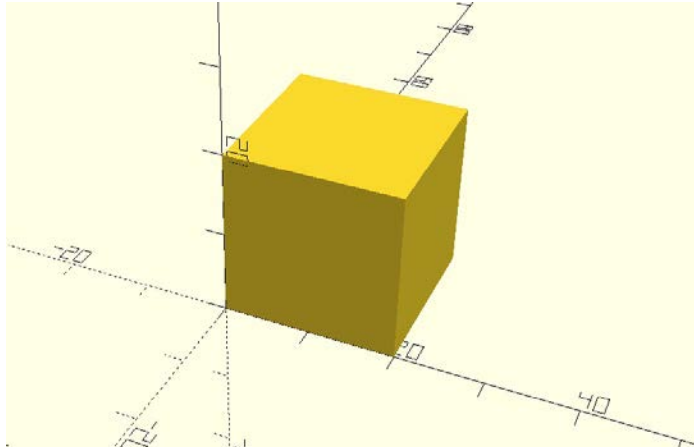
Laten we eens beginnen met een paar eenvoudige vormen te maken. Een goede hulp bij het zoeken naar de juiste objecten is dit [Cheatsheet](#), waar de meeste functies en vormen op beschreven staan. Bookmark het!



We beginnen met een kubus met een ribbe van 20mm. De code daarvoor is:

```
1 cube(20);
```

Druk je daarna op F6, dan krijg je het volgende resultaat:



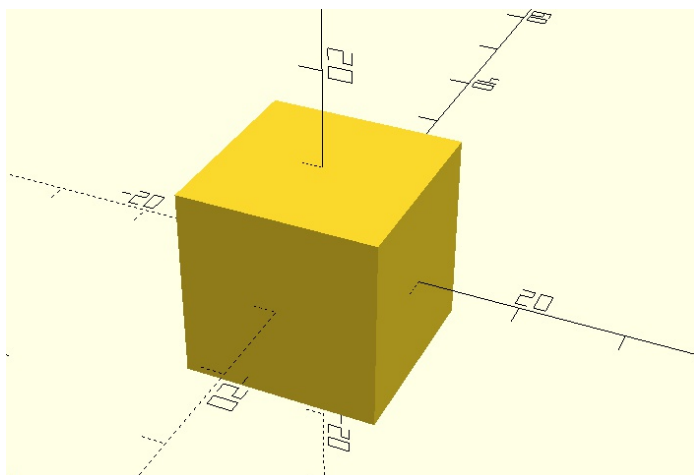
Merk op dat de ribben van de kubus beginnen op het punt $(x,y,z)=(0,0,0)$ en zich uitstrekken langs de positieve kant van het assenstelsel. Deze code is hetzelfde als:

```
1 cube(20, center = false);
```

Is er dan ook een True? Jazeker:

```
1 cube(20, center = true);
```

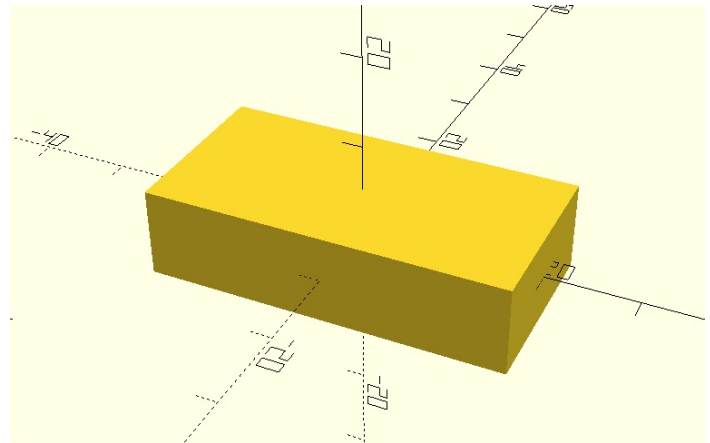
Met als resultaat:



Zoals je ziet, wordt de kubus nu gecentreerd rond het nulpunt getekend. Let er ook op, dat - net zoals bij programmeren in C - de functies afgesloten worden met een puntkomma (;).

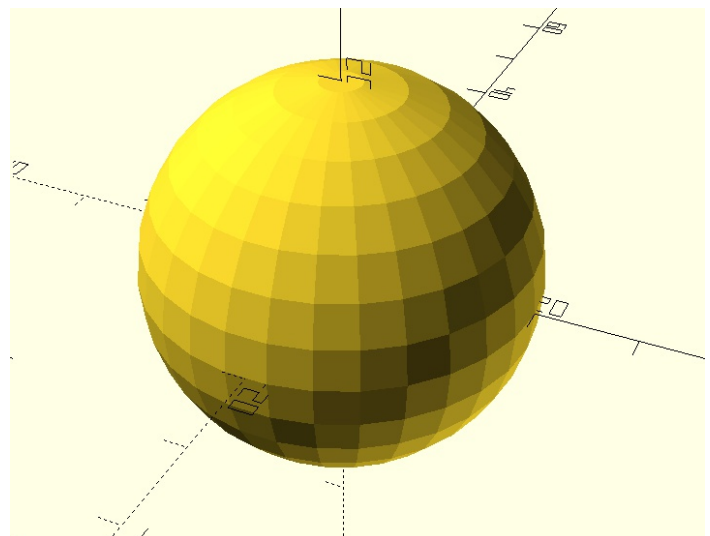
En als je nou een rechthoek wil tekenen? Dan gebruik je nog steeds 'cube' als functie, maar nu gaan we breedte, diepte en hoogte apart opgeven. Dat doen we tussen brackets ([]). Bijvoorbeeld een rechthoek van 40 breed, 20 diep en 10 hoog, gecentreerd rond het midden:

```
1 cube([40,20,10], center = true);
```

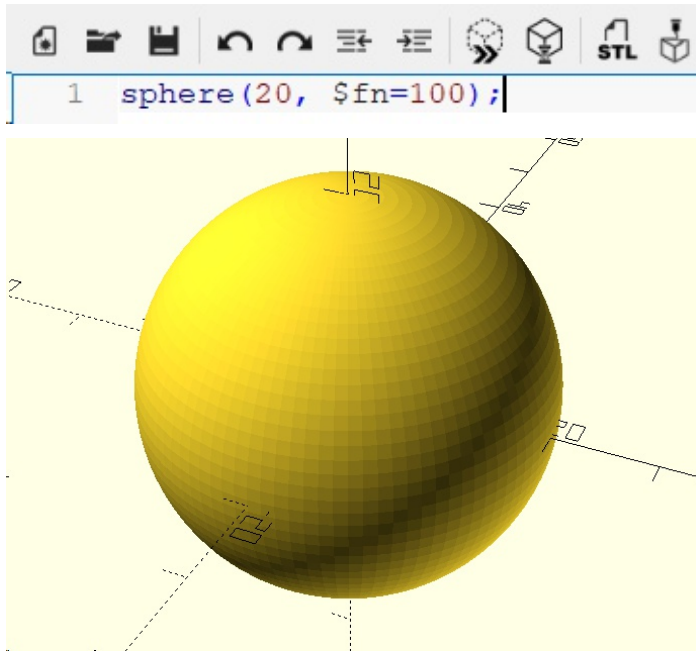


Niet al te ingewikkeld, toch? Met 'cube' kan je dus allerlei rechte vormen maken. Maar ronde vormen kan ook. Bijvoorbeeld een bol maken, met de functie 'sphere':

```
1 sphere(20);
```



De 20 binnen sphere(20) is dus de straal van de bol. Maar wacht even. Dit ziet er toch niet uit? De bol lijkt wel opgebouwd uit allemaal vlakjes! En dat is ook zo. Dat kan beter, en wel met de variabele \$fn. Die bepaalt hoe glad het oppervlak wordt. Dat ziet er zo uit:



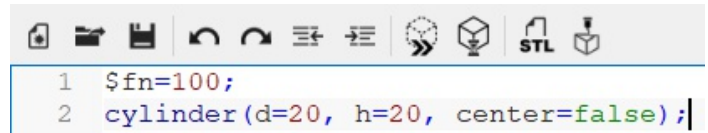
Ik heb de \$fn=100 nu binnen de functie gezet, en dan geldt hij alleen voor de functie. Maar je kunt 'm ook 'global' maken, wat betekent dat hij voor alle functies geldt die daarna aangeroepen worden:



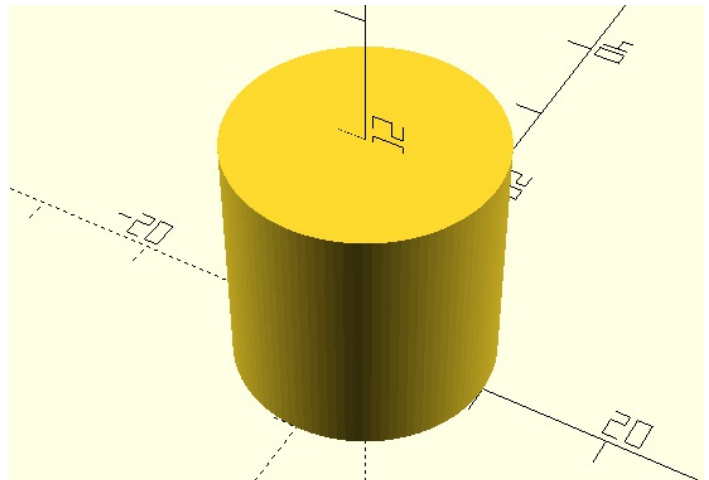
Dit geeft hetzelfde resultaat voor de bol, alleen geldt \$fn=100 nu voor alle functies die daarna volgen. Hoe hoger \$fn, hoe gladder het oppervlak. Maar ook: hoe langer je computer staat te renderen en hoe groter het uiteindelijke .stl bestand wordt dat naar de printer moet. Tijdens het ontwerpen zet ik \$fn ergens rond de 30 tot 50 zodat het renderen wat sneller gaat, en uiteindelijk tussen de 100 en 200 voordat er geprint gaat worden.

Bij de functie 'sphere' is er geen verschil tussen center=true en center=false: de bol wordt altijd getekend rond zijn middelpunt.

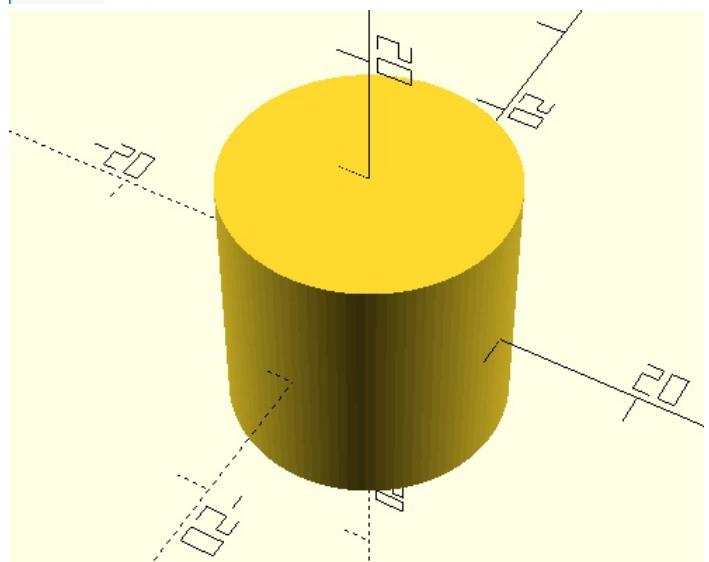
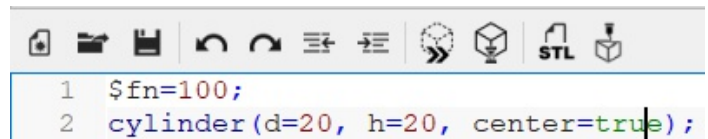
De volgende vorm die belangrijk is bij het ontwerpen, is de cylinder. Die wordt veel gebruikt voor b.v. het maken van pootjes onder een kastje of om gaten te maken voor doorvoeren. De syntax van de cylinder kan ook weer op verschillende manieren gebruikt worden. Een gewone cylinder teken je als volgt:



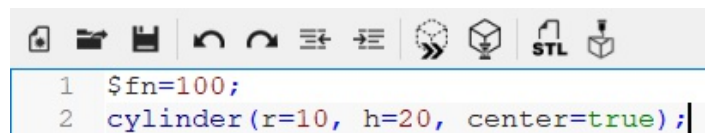
Een druk op F6 geeft dan het volgende plaatje:



Als je 'center' niet specificeert, is deze default 'false'. Ik zet het er altijd bij, zodat ik weet wat de functie gaat doen. De 'd' geeft de diameter van de cylinder en de 'h' geeft de hoogte. Omdat center=false, staat hij op het x-y assenstelsel en gaat daar h de hoogte in (de z-as). Bij het specificeren van center=true komt de cylinder symmetrisch rond het assenstelsel te liggen.



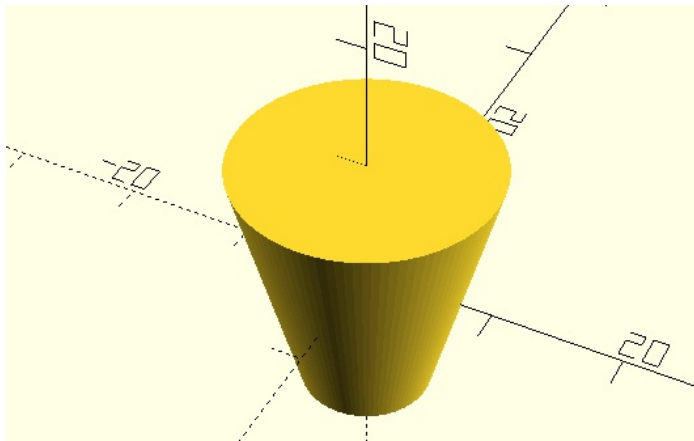
In plaats van de diameter mag je ook de straal opgeven. Dit geeft hetzelfde resultaat als boven:



Ik schreef al even "gewone cylinder". Dat doet vermoeden dat er ook ongewone cylinders zijn, en dat is ook zo. Net zoals je bij de kubus apart de breedte, diepte en hoogte kon opgeven (waardoor het eigenlijk geen kubus maar is, maar dat terzijde), kan je bij de cylinder apart de straal van de bovenkant en de onderkant opgeven. Bijvoorbeeld:

```
1 $fn=100;
2 cylinder(r1=5, r2=10, h=20, center
```

Let op het carriage-return tekenkje aan het eind van regel 2. Dat gebeurt helemaal automatisch; het programma doet zelf een word wrap. Nu specificieer je twee keer een straal: r1 is de straal aan de onderkant en r2 is de straal aan de bovenkant. Dat levert het volgende plaatje op:



Formeel is dit geen cylinder meer maar een kegel, maar alweer: een kniesoor die daar op let. Ook in dit geval mag je in plaats van de straal de diameter opgeven, dus d1=10 en d2=20:

```
1 $fn=100;
2 cylinder(d1=10, d2=20, h=20,
```

Dit geeft hetzelfde resultaat. Probeer het maar eens.

Dit zijn de belangrijkste vormen waar je objecten mee kunt maken. Er zijn nog een paar functies die belangrijk zijn om objecten mee te maken zonder dat het vormen zijn. De eerste functie is de translate(). We gaan terug naar de eerste kubus. Het is mooi dat de kubus op het

assenstelsel wordt geplaatst, maar waarschijnlijk wil je 'm daar helemaal niet hebben. Om de kubus op zijn plek te zetten, gebruik je de translate() functie.

```
1 translate([10,10,0])
2 cube(20, center=false);
```

Wat er nu gebeurt is het volgende:



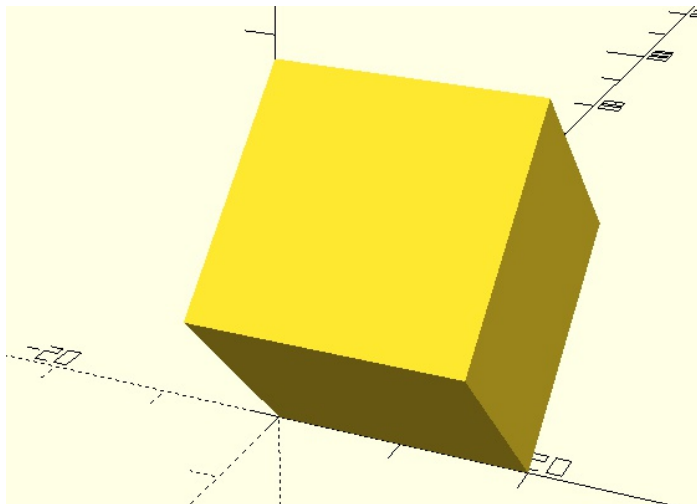
De kubus is 10mm langs de x-as en de y-as opgeschoven. Let op de syntax van de functie: 'translate([10,10,0])'

Je ziet dat coördinaten weer worden geplaatst tussen brackets ([]), maar let ook op het ontbreken van de puntkomma achter de functie. Zolang er niets getekend is, worden er geen puntkomma's geplaatst. Met de translate functie is het dus mogelijk om objecten te verplaatsen.

De volgende functie die je vaak zal gebruiken bij het maken van een ontwerp, is rotate(). Zoals de naam doet vermoeden, draait dat het ontwerp om een aangegeven as. Laten we de kubus nu eens 45 graden draaien om de x-as (dat is de horizontale as). Ik ga eerst uit van een gewone kubus met center=true:

```
1 rotate([45,0,0])
2 cube(20, center=false);
```

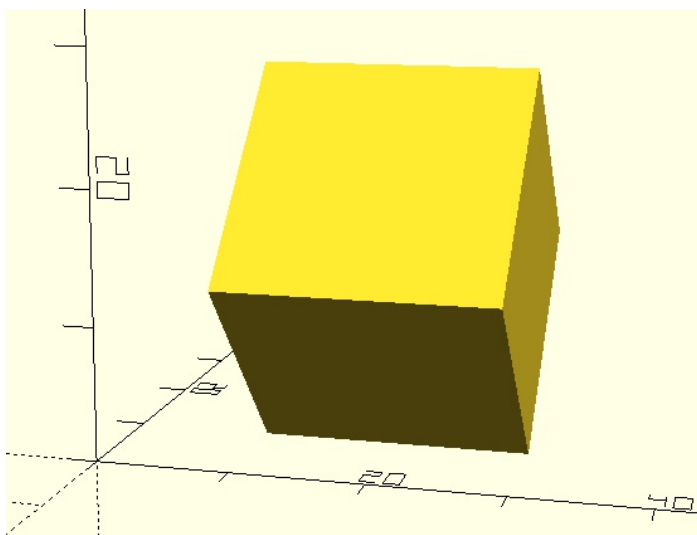
De kubus komt er nu als volgt uit te zien:



Merk op dat ook achter 'rotate()' geen puntkomma komt. Moet een object zowel verplaatst als gedraaid worden, dan kan je beide routines gebruiken:

```
1 translate([10,10,0])
2 rotate([45,0,0])
3 cube(20, center=false);
```

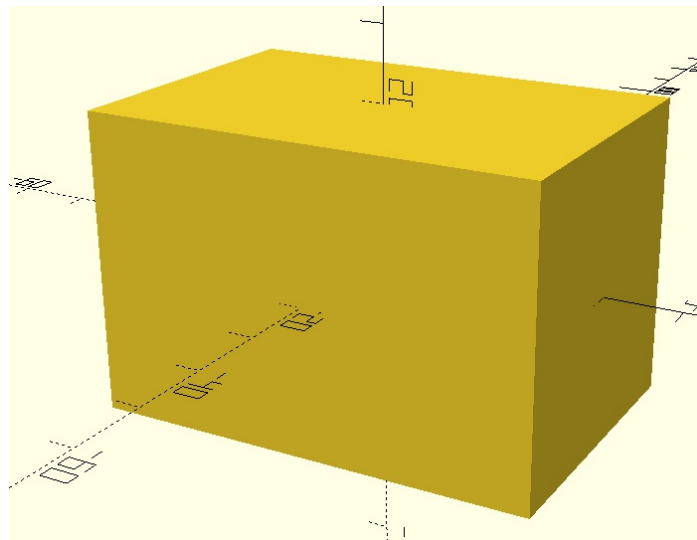
Dus met translate() verschuif je het object 10 naar rechts en 10 naar achteren, met rotate draai je het object vervolgens 45 graden om de x-as en cube() tekent uiteindelijk het object. Let erop dat zowel achter translate() als achter rotate() geen puntkomma volgt. Die komt pas na cube(), waarmee de hele bewerking bestaande uit regel 1 t/m 3 afgesloten wordt.



Goed. Nu kunnen we rechthoekige vormen, bollen en cylinders maken. Maar hoe maak je een kastje? Dat kan je doen door vormen van elkaar af te trekken. De functie waarmee je dat

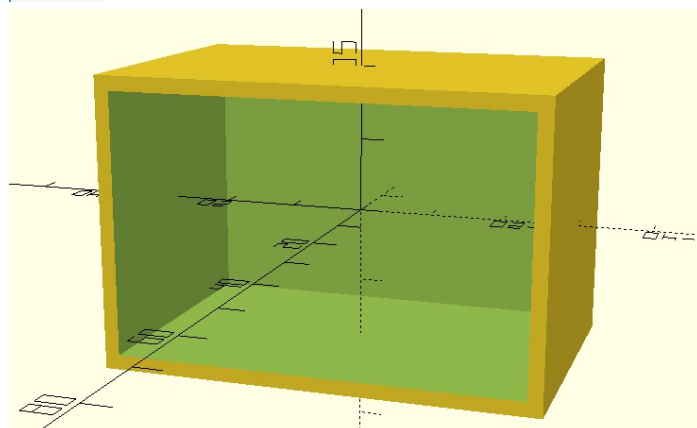
kunt doen heet difference(). Verschil, dus. Binnen die functie wordt elke vorm die na de eerste vorm komt, van die 1^e vorm afgetrokken. Voorbeeld. Ik begin met een doos te maken.

```
1 cube([60,40,40], center=true);
```

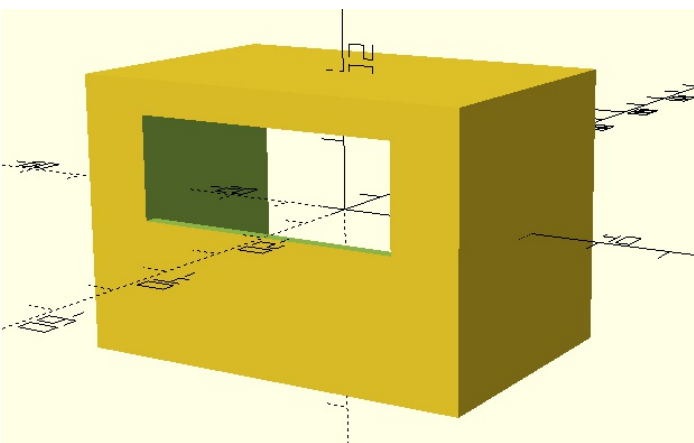


Vervolgens ga ik de doos "uithollen", door er een kleinere kubus vanaf te trekken. Als ik de wanden van het kastje 2mm dik wil hebben, moet de kubus in de x en de z richting 4mm kleiner zijn (2mm aan elke kant). De y-richting maakt niet uit, omdat ik die naar achter uit kan laten steken om voldoende materiaal weg te halen. Om de uitholling precies te kunnen plaatsen, maak ik center=false en plaats de tweede kubus op zijn plek met translate().

```
1 difference() {
2   cube([60,40,40], center=true);
3   translate([-28, -18, -18])
4   cube([56,40,36], center=false);
5 }
```



Je ziet de doos nu van de achterkant (ik heb 'm met de linker muistoets om zijn as gedraaid). Nog even stap voor stap door de code: ik begin met het maken van de 'doos', zullen we maar zeggen. Afmetingen 60 breed, 40 diep en 40 hoog. Met de `difference()` functie ga ik daar een tweede kubus vanaf trekken. Let op de syntax van de functie: `difference() { <code> }`. Wat `difference` moet doen, staat tussen gekrulde haken. `cube([60,40,40], center=true)` tekent de doos die we willen hebben. Ik wil een wanddikte van 2mm dus maak ik een tweede kubus die in x- en z-richting 4mm korter is. Naar de achterkant mag hij uitsteken: daar zie je niets van dus die blijft 40mm. Omdat ik voor de tweede kubus `center=false` gebruik, weet ik hoeveel ik 'm moet verschuiven, namelijk 28 naar links (de helft van 60 min 2mm wanddikte), 18mm naar voren (de helft van 40mm min 2mm wanddikte) en 18mm naar beneden (ook de helft van 40mm min 2mm wanddikte). Vandaar `translate([-28, -18, -18])`. De kubus die er vanaf getrokken wordt is dan 56mm breed, nog steeds 40 diep maar 36mm hoog. En nu hebben we een kastje. Stel dat je een uitsparing wil maken voor een display van 40 x 15mm. Dan moeten we weer een kubus definiëren die we dwars door de front heen laten steken, zodat daar een opening ontstaat.



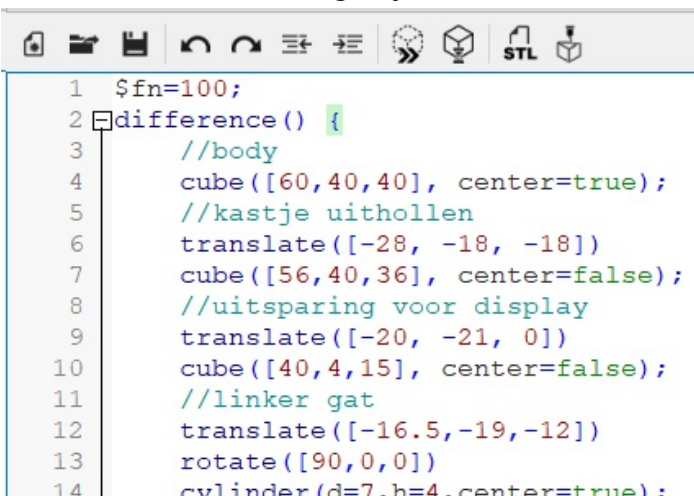
```

1 difference() {
2     cube([60,40,40], center=true);
3     translate([-28, -18, -18])
4     cube([56,40,36], center=false);
5     translate([-20, -21, 0])
6     cube([40,4,15], center=false);
7 }

```

De kubus maak ik 40mm breed, 15mm hoog en 4mm dik, om de 2mm van het kastje weg te halen. Daarvoor verplaats ik 'm 20mm naar links, 21 naar voren (het kastje begint bij 20, dus ik haal alles tussen 21 en 17mm weg, de dikte van het kastje+1) en ik vind de plaats vanaf het midden naar boven wel mooi, dus `z=0`. Er komt dus een tweede `translate` te staan (regel 5) en een tweede kubus (regel 6) die ons display moet voorstellen en die we dus eveneens van het kastje gaan aftrekken (alles na de eerste vorm wordt er immers vanaf getrokken).

En we hebben een uitsparing voor een display! Nu nog een paar 7mm gaatjes maken voor schakelaartjes of drukknoppen. Daarvoor gebruiken we cylinders van weer 4mm hoog die we op geschikte plekken van het kastje aftrekken. Ik maak vier gaatjes:



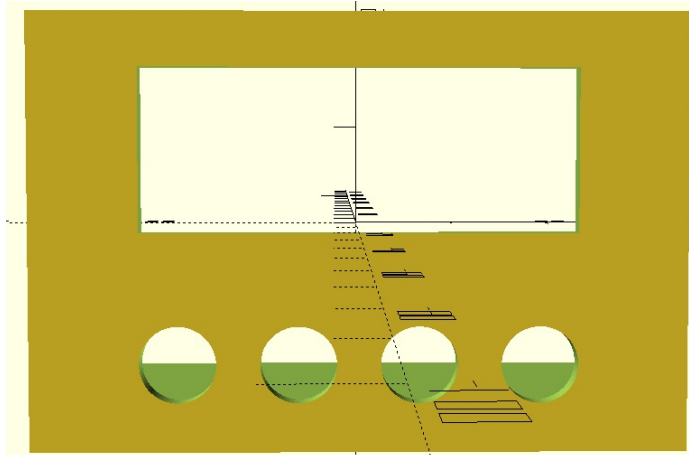
```

1 $fn=100;
2 difference() {
3     //body
4     cube([60,40,40], center=true);
5     //kastje uithollen
6     translate([-28, -18, -18])
7     cube([56,40,36], center=false);
8     //uitsparing voor display
9     translate([-20, -21, 0])
10    cube([40,4,15], center=false);
11    //linker gat
12    translate([-16.5,-19,-12])
13    rotate([90,0,0])
14    cylinder(d=7,h=4,center=true);
15    //2e gat
16    translate([-5.5,-19,-12])
17    rotate([90,0,0])
18    cylinder(d=7,h=4,center=true);
19    //3e gat
20    translate([5.5,-19,-12])
21    rotate([90,0,0])
22    cylinder(d=7,h=4,center=true);
23    //rechter gat
24    translate([16.5,-19,-12])
25    rotate([90,0,0])
26    cylinder(d=7,h=4,center=true);
27 }

```

Ik heb de blokken van commentaar voorzien (voorafgegaan door twee schuine strepen) zodat je kunt zien waar wat gebeurt. De gaten maak ik met een cylinder, maar die staat standaard rechtop. Die moet dus 90 graden om de x-as gedraaid worden. Daarnaast zet ik 'm op 19mm van de y-as, waardoor door de dikte van 4mm

alles tussen 17 en 21mm weggehaald wordt. En dus krijg je een gat. Dat komt er dan als volgt uit te zien:



De x-coördinaten van de gaten heb ik als volgt berekend: ik wilde de randen van de gaten laten samenvallen met de buitenranden van de display opening. Aangezien de gaten 7mm doorsnede zijn, moeten die op 3,5mm van de rand komen. De rand ligt op 20mm van het midden, dus dat is $20 - 3,5 = 16,5$ mm. Er zitten 3 afstanden tussen de 4 gaten, en aangezien er 33mm tussen de gaten zit ($40\text{mm} - 2 \times 3,5$) zit er tussen elk gat 11 mm. Dan zitten ze netjes op gelijke afstanden van elkaar. Nou wil ik nog pootjes onder het kastje. Dat doe ik met een paar extra cylinders. Zie de code hiernaast vanaf regel 28. Deze code is buiten de difference() functie geplaatst, dus de cylinders worden niet van het kastje afgetrokken, maar er bij opgeteld. Ik plaats ze 23mm naar beneden en omdat ze 3mm hoog zijn, komen ze tegen het kastje aan en worden daar onderdeel van. Voor het dichtmaken van het kastje kan je een deksel maken met opstaande randen, dat precies op de achterzijde past.

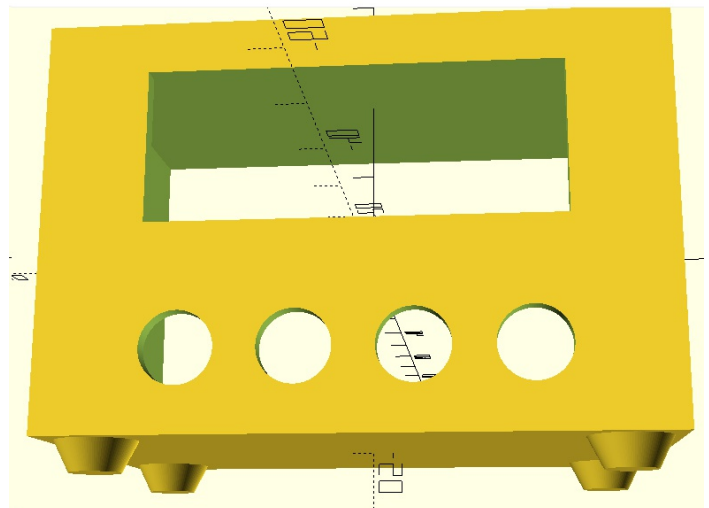
Met de kennis die je nu hebt, kan je zelf kastjes maken. Je moet een beetje driedimensionaal leren denken: vormen bij elkaar optellen en/of van elkaar aftrekken. Ben je tevreden, druk dan op functietoets F7. Je kunt dan een .stl bestand aanmaken dat je naar een 3D printer kunt sturen (of het laten printen door b.v. JLCPCB). Gebruik de hier getoonde code om er zelf mee te experimenteren: zie hoe het kastje blok voor blok ontstaat. Tot slot een praktijkvoorbeeld.

```

1 $fn=100;
2 difference() {
3     //body
4     cube([60,40,40], center=true);
5     //kastje uithollen
6     translate([-28, -18, -18])
7     cube([56,40,36], center=false);
8     //uitsparing voor display
9     translate([-20, -21, 0])
10    cube([40,4,15], center=false);
11    //linker gat
12    translate([-16.5,-19,-12])
13    rotate([90,0,0])
14    cylinder(d=7,h=4,center=true);
15    //2e gat
16    translate([-5.5,-19,-12])
17    rotate([90,0,0])
18    cylinder(d=7,h=4,center=true);
19    //3e gat
20    translate([5.5,-19,-12])
21    rotate([90,0,0])
22    cylinder(d=7,h=4,center=true);
23    //rechter gat
24    translate([16.5,-19,-12])
25    rotate([90,0,0])
26    cylinder(d=7,h=4,center=true);
27 }
28 translate([25,15,-23])
29 cylinder(r1=2.5, r2=4, h=3,
30         center=false);
31 translate([-25,15,-23])
32 cylinder(r1=2.5, r2=4, h=3,
33         center=false);
34 translate([25,-15,-23])
35 cylinder(r1=2.5, r2=4, h=3,
36         center=false);
37 translate([-25,-15,-23])
38 cylinder(r1=2.5, r2=4, h=3,
39         center=false);

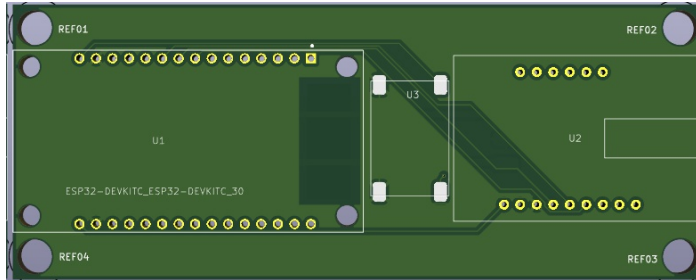
```

De complete code voor het kastje

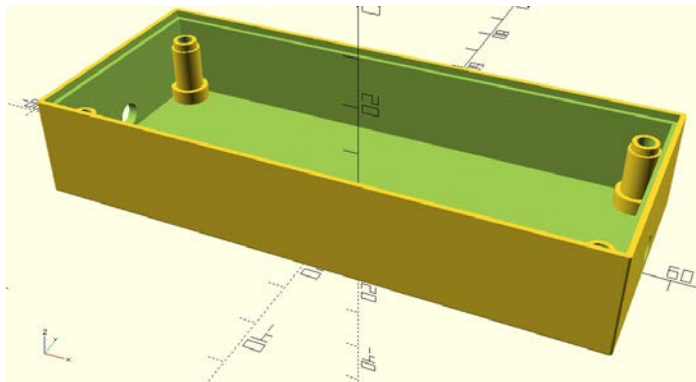


Het uiteindelijke resultaat

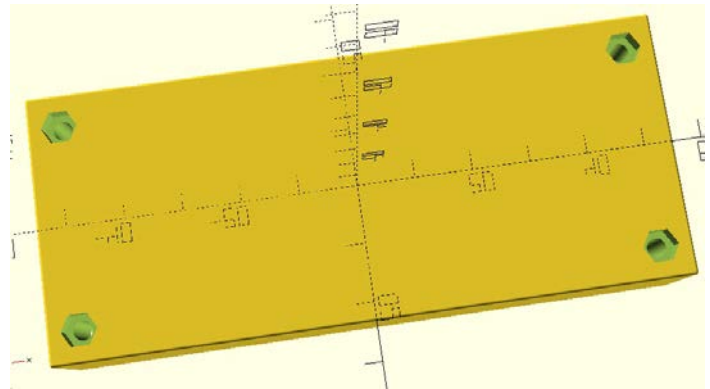
Voor mijn Home Assistant automatiserings-systeem wilde ik een muzikspeler maken met een ESP32. De details zal ik je besparen, maar er moest natuurlijk een printje van gemaakt dat ik in een fraai kastje wilde bouwen. Op de hoeken van de print had ik gaten gemaakt van 5mm doorsnee:



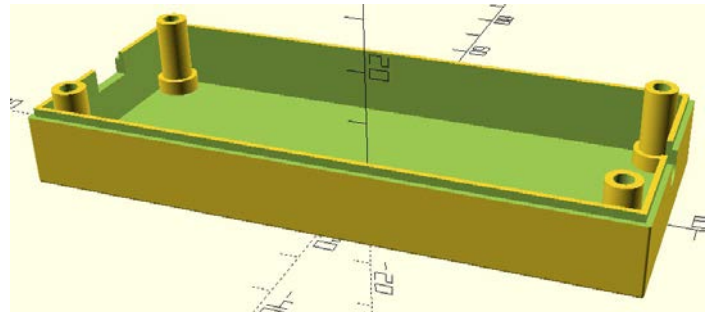
Op de print zit een ESP32, een buck converter en een D/A converterprint voor de audio. De USB connector van de ESP32 steekt uit, evenals de 3,5mm jack van de A/D converter. Het idee was om steuntjes te maken met een rand van 5mm waar je de print op kunt leggen, waar een 3mm bout doorheen gaat om de boel dicht te maken. Tevens moeten de boutjes en moertjes verzonken worden. Wat details:



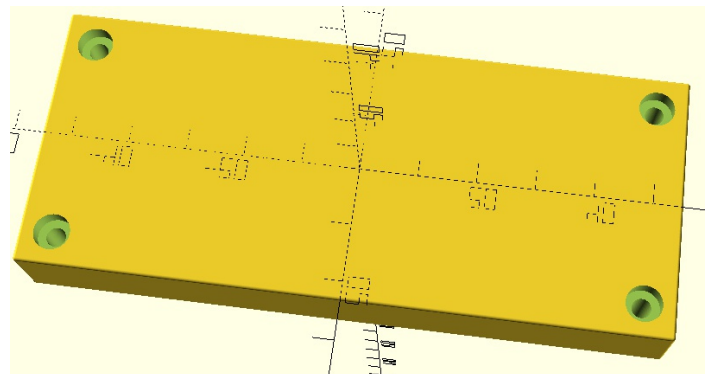
Zie de verdikking aan de onderzijde van de steunen: die zijn bedoeld om de moeren te kunnen laten verzinken aan de andere kant. Door de versmalling aan de bovenzijde van de steunen kan ik de print er zo opleggen. De bodem ziet er uit zoals op het plaatje rechtsboven. Let op de mooie uitsparing voor de moertjes: die heb ik gemaakt door een cylinder te gebruiken met een $\phi=6$. Daardoor wordt de cylinder zeskantig... Uiteraard is er een deksel bij gemaakt, die ook weer steunen heeft die de print op zijn plaats drukt. In de bovenkant van het deksel zijn weer uitsparingen gemaakt voor de boutjes. Ook die zijn weer gemaakt met een cylinder, maar dan met een $\phi=100$.



De bodem van het kastje.



Het deksel met de uitsparingen voor de USB en 3,5mm jack



De bovenkant met de uitsparingen voor de boutjes

Natuurlijk kan je een kastje ook maken door alle zijden apart te definiëren en tegen elkaar aan te zetten, maar persoonlijk vind ik het uithollen van een vorm makkelijker. Er zijn nog veel meer mogelijkheden, zoals bijvoorbeeld het afronden van de vormen, het maken van ringen etc. Met de in dit artikel beschreven vormen heb je al voldoende mogelijkheden om de in de hobby meest voorkomende objecten te maken. Ik heb b.v. kegels gebruikt als pootjes voor apparaten, cylinders voor steunen en uitsparingen en rechthoekige vormen voor kastjes e.d. Probeer zelf de voorbeelden in dit artikel uit, en experimenteer met de vormen en de functies. En verder kun je googlen als je iets specifiek zoekt. Op internet zijn veel voorbeelden te vinden!



Pim snoof bij het binnenkomen van Opa's piephok, de bijnaam die de shack gekregen had vanwege de morse-signalen die daar regelmatig uit opstegen, de aldaar hem tegemoetkomende dampen op. "Wat bent u in vredesnaam aan het doen, een bunzing aan het cremen?" vroeg Pim. Opa schoof zijn leesbril naar zijn voorhoofd en antwoordde: "Nee, ik ben wat SMD aan het solderen met soldeerpasta, en wat je ruikt zijn de oplosmiddelen". "Oh, flux dus", zei Pim. Opa schudde van nee. "Dat is niet hetzelfde. Je hebt flux, en je hebt soldeerpasta. Dit is soldeer-pasta", zei Opa. "En wat is het verschil?" vroeg Pim. Opa ging er maar weer eens voor zitten. "Flux heeft meerdere toepassingen; er zijn twee redenen om het te gebruiken. De belangrijkste reden is dat het ervoor zorgt dat tin beter vloeit en mooier op de contactpunten gaat zitten. De andere reden, die deels de oorzaak is van deze reden, is dat het oxidatie tegengaat en wegneemt. Het reinigt het contactoppervlak door de eigenschappen van de stof en de verhitting. Flux is bij de meeste soldeertin in de kern verwerkt. De tin is dus bij voorbaat al vermengd met flux om het solderen te vergemakkelijken. Dus daarom hebben de meeste amateurs die met gewone onderdelen werken, flux nooit apart gebruikt. Maar bij SMD soldeerwerk is het meestal zo dat de aanwezige flux niet meer voldoende is. Daarom is het vaak aan te raden om vooraf al flux toe te voegen.

Flux heeft een de-oxiderende werking. Het zorgt ervoor dat de oxidatierestanten die op de contactvlakken van componenten en printen aanwezig zijn, gereinigd worden. Hierdoor hecht de soldeertin veel beter en kan het zowel elektrisch beter geleiden als mechanisch meer kracht aan. Twee redenen om dus voldoende flux te gebruiken voor het soldeerwerk. De meeste flux bij solderen wordt gebruikt is van organische oorsprong, deze flux heeft vaak 4

componenten. Activators, katalysatoren, oplosmiddelen en toevoegingen.

Activators

Deze stoffen zorgen voor oplossende werking van oxidatie op het metalen oppervlak. Ze hebben als doel om voor een on-geoxideerde makkelijk soldeerbare ondergrond te zorgen. Vaak zijn deze stoffen afhankelijk van hogere temperaturen en werken ze nog niet op kamertemperatuur.

De manier waarop activators werken is door de metaaloxides op de contacten om te zetten in zout en water. Deze reststoffen lossen vaak vrij eenvoudig op in de andere stoffen zoals de katalysatoren. De werking van de activators verbetert vaak bij hogere temperaturen tot aan een maximum waarbij ze ontbinden.

Katalysatoren

Deze chemicaliën zijn in de regel heel goed bestand tegen hoge temperaturen en vaak niet ontvlambaar. Het doel van deze stoffen is om een barrière te vormen tegen zuurstof om de versnelde oxidatie op hoge temperatuur tegen te gaan. Een katalysator is een stof die een proces versnelt zonder er zelf aan deel te nemen.

Oplosmiddelen

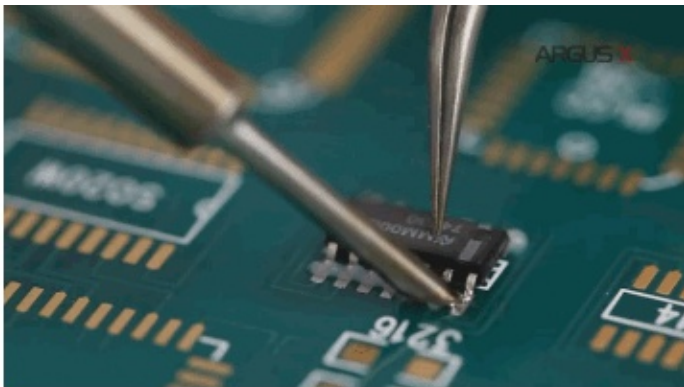
Deze stoffen zorgen voor een betere verdeling van de werkzame stoffen over het soldeeroppervlak. De oplosmiddelen verdampen vaak laat in de voorverwarmings (pre-heat) fase.

Additieven

De additieven zorgen er elk op hun eigen manier voor dat de eigenschappen van de flux worden aangepast. Ze kunnen bijvoorbeeld zorgen voor een lagere oppervlaktetenspanning, corrosie tegengaan, werken als verdikkingsmiddel en andere eigenschappen. Flux zorgt er dus voor dat de tin beter hecht en sterkere verbindingen vormt.

NEUTRALISEER JE ONDERDELEN WANNEER JE FLUX GEBRUIKT. Het is een zuur en zal componenten uiteindelijk aantasten als je de flux erop laat zitten, dus gebruik 99% isopropylalcohol en een tandenborstel op je werkgebied nadat je je soldeerwerk hebt voltooid. No-clean fluxen zijn een bijzonder type, ze zijn zo ontwikkeld dat de restanten gewoon kunnen blijven zitten op de print omdat deze onschadelijk zijn in de meeste situaties. Voor bijna alle toepassingen kunnen deze dus na het solderen achtergelaten worden. De belangrijkste reden om het er vanaf te halen is dat het er minder mooi uit ziet dan een schone print. No-clean flux is vaak enkel te reinigen met isopropylalcohol en niet water oplosbaar.

Soldeerpasta is in wezen poedermetaal soldeer opgelost in een dik medium dat flux wordt genoemd (hee, daar is het weer). Flux wordt toegevoegd om te dienen als een tijdelijke kleefstof, die de componenten vasthoudt totdat het soldeerproces het soldeer smelt en de onderdelen aan elkaar smelt. De pasta is een grijs, deegachtig materiaal. De samenstelling van de soldeerpasta varieert, afhankelijk van het beoogde gebruik. Dus in dit geval voeg je geen soldeer meer toe, de soldeerpasta is de soldeer.



Zie het plaatje hierboven. Als je een SMD IC moet solderen, druk je 'm op zijn plaats. Trek er een streep soldeerpasta over (de grijze streep) en verhit dat met de soldeerbout (of een SMD werkstation, als je daarover beschikt). Je zal zien dat de pasta zich keurig samentrekt op de pootjes van het IC en dan zit hij vast.

De meeste soldeerpasta's moeten koel bewaard worden en hebben ook in ongeopende toestand

maar een zeer beperkte levensduur. Een potje dat 6 maanden staat, kan eigenlijk niet meer goed gebruikt worden voor kwaliteits-soldeerwerk en dient dus afgevoerd te worden. Een belangrijk bestanddeel van de flux in de soldeerpasta zijn namelijk de activators die oxides te lijf gaan. Hierbij gaat het niet alleen om de oxides op de te solderen oppervlaktes, maar ook om de oxides die zich bevinden op het oppervlak van het soldeer zelf. Deze activators werken beter bij hogere temperaturen, maar ook bij kamertemperatuur zijn ze actief. Aangezien in een soldeerpasta flux en soldeer gemengd zijn, werken de activators dus constant in op het soldeer en alleen door de temperatuur zeer laag te houden, wordt het chemische proces waarin de activators betrokken zijn dusdanig vertraagd dat de levensduur van de pasta werkbaar lang is. Wordt de temperatuur echter te hoog, dan gaat zelfs in de afgesloten pot het chemische proces dusdanig snel, dat in no time de pasta niet meer bruikbaar is. De flux is dan uitgewerkt en verwijdt niet meer (goed genoeg) de oxides op de print en de onderdelen met als gevolg dat de soldeerverbinding zeer slecht wordt. De soldeerpasta moest dan ook bewaard worden in de koelkast en mocht pas vlak voor gebruik opgewarmd worden tot kamertemperatuur. Gelukkig is dat tegenwoordig verbeterd. Kijk bijvoorbeeld eens naar [deze webpagina](#), waar zowel loodvrije als loodhoudende soldeerpasta aangeboden wordt, ook als no-clean. Dus die techniek is een stuk beter geworden de laatste jaren. Voor het solderen van IC's met veel pootjes geef ik de voorkeur aan soldeerpasta, dat is wel zo makkelijk", besloot Opa. Pim knikte dat hij het begrepen had. "Maar dan had ik toch gelijk dat ik flux rook, want dat is dus een bestanddeel van de soldeerpasta. Is het al van dat nieuwe spul?" vroeg Pim. "Jazeker", zei Opa. "Ik had het dus ook met alleen flux kunnen doen, maar zoals ik zei, IC's met veel pootjes solderen makkelijker met soldeerpasta. En het type wat ik gebruik, hoeft niet koel bewaard te worden. Desondanks laat ik het geen maanden staan. Ik combineer SMD projecten zoveel mogelijk vanwege de kosten". "Het is me weer helemaal helder, bedankt Opa", zei Pim.

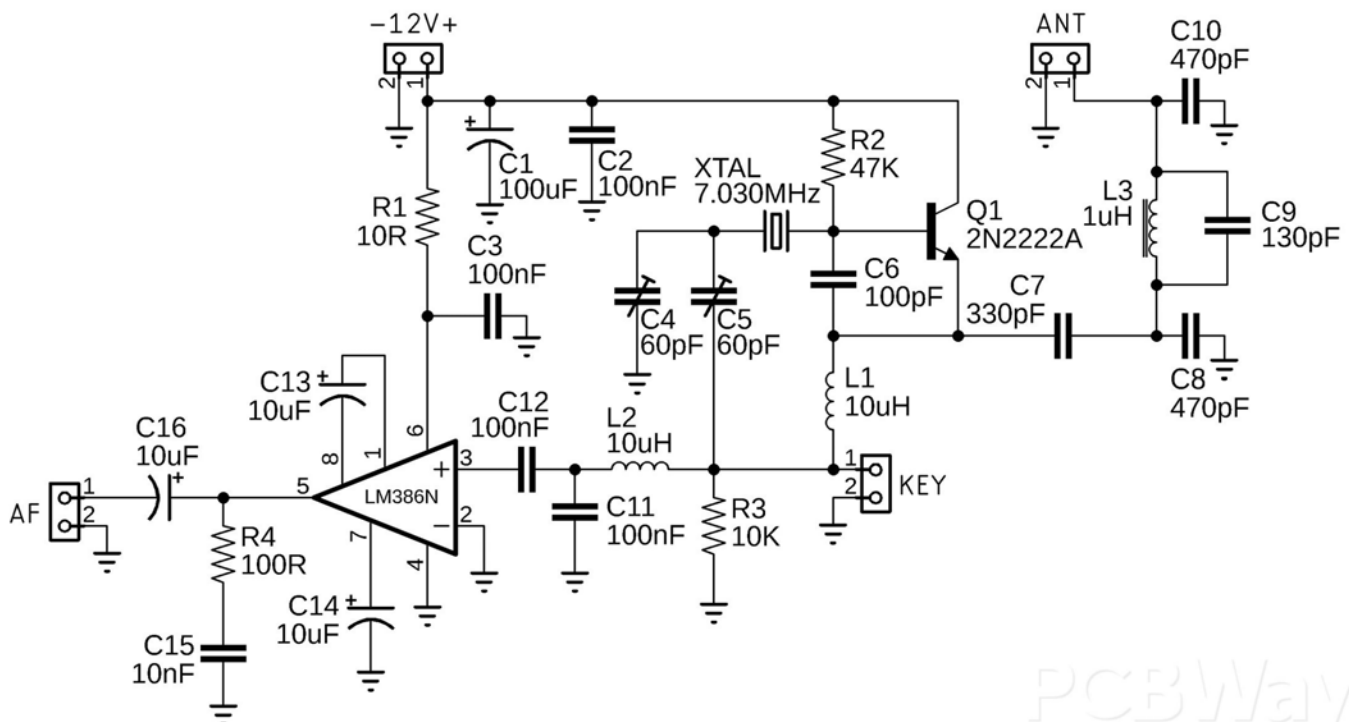
Pititico 2 40m QRP CW Transceiver

Deze QRP CW transceiver is een variant op de Pititico transceiver die oorspronkelijk ontworpen is door PY2OHH. De hele transceiver bestaat uit slechts 1 transistor en 1 IC. Het uitgangsvermogen is ongeveer 700mW bij 12V, voldoende om leuke verbindingen mee te kunnen maken (met wel een fatsoenlijke antenne). De gemodificeerde versie maakt instellen van de TX frequentie en de offset frequentie (voor de instelling van de ontvangstfrequentie) eenvoudiger. Daarnaast is het laagdoorlaatfilter verbeterd voor een betere onderdrukking van de harmonischen, en is de schakeling verbeterd zodat de meeste AM omroep storingen verdwenen zijn.

Het schema van de transceiver zie je hieronder. De frequentie wordt bepaald door het kristal, in dit geval een 7.030kHz type (de centrale QRP frequentie). In de stand ontvangst (key-up) ligt de emitter van Q1 via L1 en R3 aan massa. R3 is relatief groot, waardoor Q1 maar lichtjes staat te oscilleren. Het antennesignaal komt via het laagdoorlaatfilter C8/C9/C10/L3 op de emitter

van Q1 terecht en mixt daar met het oscillator-signaal. Door de spoelen L1 en L2 komt alleen het laagfrequent deel van het mengproduct op de ingang van de LM386 eindversterker terecht. Kijk of je een LM386-4 kunt krijgen, want die levert een hoger uitgangsvermogen dan de standaard LM386-1 typen (1W i.p.v. 325mW) en kan tegen een hogere voedingsspanning (18V i.p.v. 12V). De transceiver heeft geen AGC en ook geen volumeregelaar: de neembaarheid van de signalen hangt geheel af van het dynamisch bereik van je gehoor(apparaat). Met C4 kan je de ontvanger op een prettige ontvangsttoon zetten (700Hz b.v.) bij 7.030kHz. Er wordt voldoende audio geproduceerd voor een hoofdtelefoon.

Tijdens zenden doet de key een aantal dingen: hij sluit de ingang van de LF versterker kort om je gehoor te sparen, hij trekt C5 naar massa waardoor deze parallel aan C4 komt te staan en waarmee je de RX/TX shift kunt instellen - wat wil zetten dat je TX frequentie onder de RX frequentie ligt. Daarnaast wordt R3 kortgesloten



waardoor Q1 nu op vol vermogen kan gaan werken, en zijn vermogen via de emitter afgeeft aan de antenne. Normaal zie je een uitkoppeling via de collector met een of andere trafo, maar door de emitter te gebruiken is geen impedantie aanpassing nodig omdat de emitter van zichzelf laagohmig is. Dat heeft meteen als voordeel dat de antenne weinig terugwerking op de oscillator heeft (waardoor de belasting van de antenne de frequentie zou kunnen beïnvloeden).

De afregeling zou ik als volgt doen: voer met een meetzender een signaal toe van 7.030,7kHz (dus 700Hz boven de zendfrequentie) en regel C4 op zero beat. Dan levert ontvangen op 7.030 een toon van 700Hz op (7.031,4 trouwens ook). Regel daarna met key down C5 af op een zendfrequentie van 7.030kHz en je ben klaar om verbindingen te maken.

Opbouwen kan op een stuk experimenteerprint, maar mocht je een echte print willen, dan is deze te koop bij [pcbway](https://www.pcbway.com/). Je kunt alleen de print

of de hele assembly kopen. Hoef je ook niet naar onderdelen te zoeken.

Onderschat niet wat je met 700mW kunt doen. Ik heb diverse QRP transceivers (waaronder een Pixie) die eveneens vermogens in die orde grootte hebben, en daar heb ik heel Europa mee gewerkt. Zeker als je nog nooit een transceiver gebouwd hebt, is dit een leuk ontwerp om mee te spelen (mits je CW spreekt natuurlijk). Advies: niet in het weekend proberen. Dan is de band vergeven van de testers die schijt hebben aan bandplannen waardoor 7.030 met aan zekerheid grenzende waarschijnlijkheid bezet is, met ook nog eens om de 300Hz een ander teststation met 1kW+. De selectiviteit van deze transceiver is niet om over naar huis te schrijven dus dan kan je verbindingen wel vergeten. Neem dus een doordeweekse dag (waarop er geen testen zijn; tegenwoordig gaat die ellende door de week gewoon door) dan heb je de meeste kans. Succes met dit leuke ontwerp!

Een eenvoudige TDR/testsignaalinjector

Donald Cantrell, ND6T

TDR staat voor Time Domain Reflectometer. Een instrument dat een signaal via een transmissielijn (zoals een coaxkabel) stuurt en vervolgens de reflectie analyseert. Zoek een kabelfout onder de grond, zodat je weet waar je moet graven! Boven je hoofd, in de lucht, voordat je hem verwijderd! Is hij beschadigd in een muur? Is er een nietje doorheen gegaan? Wees de held en ga direct op onderzoek! Met een 100 MHz oscilloscoop maakt deze adapter nauwkeurige rudimentaire coaxkabeltests mogelijk op kabels tot enkele honderden meters lengte. Hiermee kan de gebruiker niet alleen een fout lokaliseren, maar ook een aanwijzing krijgen over de aard van de fout. En er is meer! Dit eenvoudige apparaat genereert ook een toon die kan worden gebruikt om problemen met ontvangers en audiocircuits op te lossen. Het beste van alles is dat het weinig kost om te bouwen en slechts een paar

minuten tijd kost om het te realiseren!



Toen ik op zoek was naar een generator met snelle stijgtijd, ontdekte ik dat mijn testapparatuur gewoon niet had wat ik nodig had. Een zoektocht in mijn junkbox leverde een paar 74AC04 hex inverter logic chips op die aan de eisen leken te voldoen. Ze zijn nieuw te koop

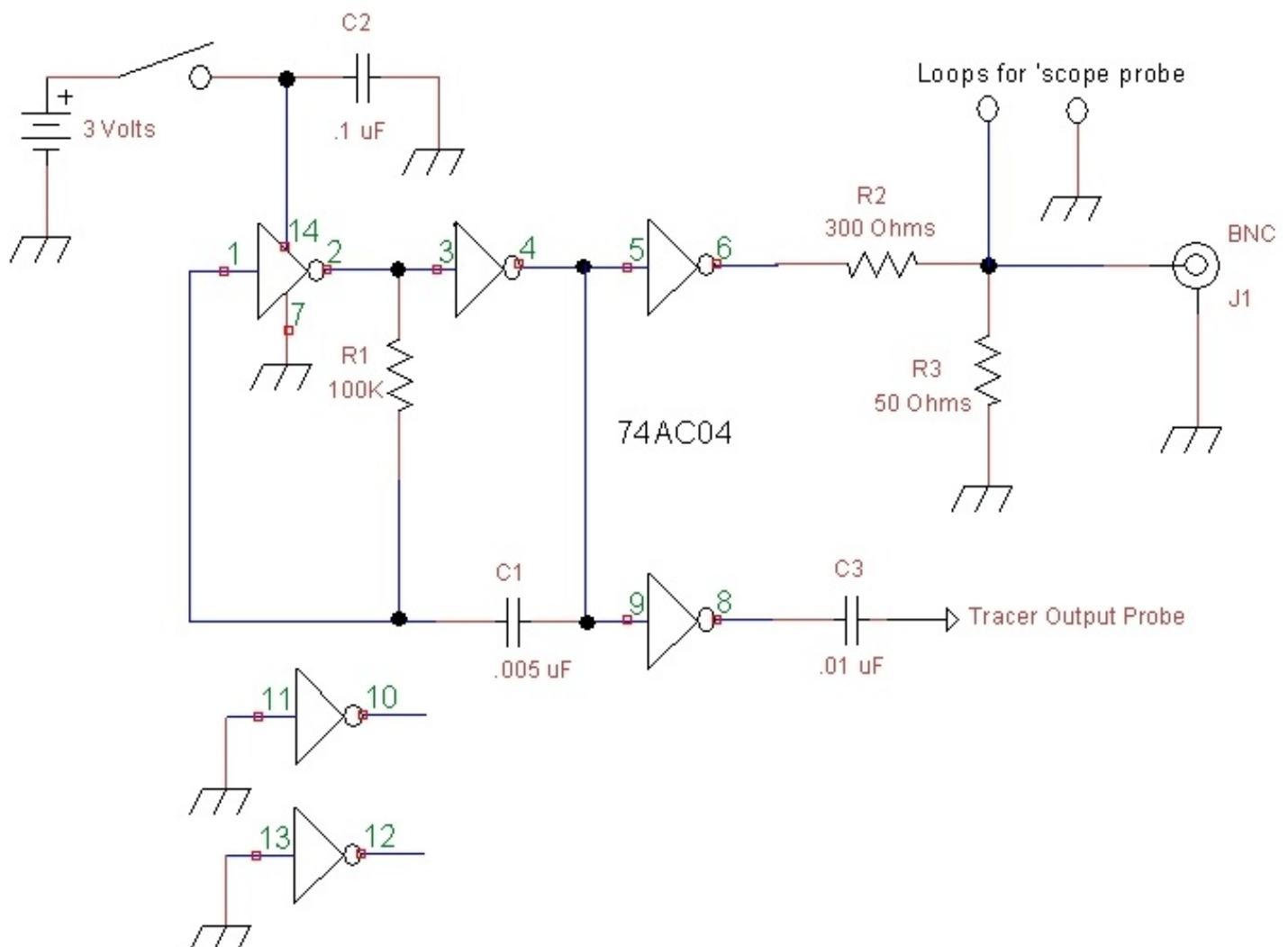
voor 25 cent per stuk als je ze nodig hebt. Handige onderdelen om te hebben. De eerste twee secties vormen een oscillator met R1 en C1 die de frequentie instellen op ongeveer 1 kHz. Dit voedt twee extra secties om de oscillator te bufferen en te isoleren. Door de verbindingen kort en de componenten klein te houden, kunnen de stijgtijden van de uitgang onder de 2 nanoseconden blijven, sneller dan een 100 MHz oscilloscoop kan meten.

De bouw

Een klein stukje printplaat, ongeveer zo groot als een flinke postzegel, was net iets groter dan de chip, met genoeg ruimte om de chip op zijn rug te leggen volgens de dode kever methode, en pin 7 terug te buigen en aan het massavlak te solderen om hem te verankeren. Omdat je alle ongebruikte ingangen aan massa moet leggen, heb ik hetzelfde gedaan met pin 11 en

13. Dit maakt lijm overbodig. Onthoud dat je met de chip op zijn rug de pinnen met de klok mee moet tellen. Omdat pin 7 op een hoek zit en aan massa ligt, gebruik je deze als referentie: pin 8 zit aan dezelfde kant ertegenover, enz.

Ik heb pinnen 2 en 3 naar elkaar toe gebogen en hetzelfde gedaan met pinnen 4 en 5. Ik heb een BNC-aansluiting zo dicht mogelijk aan de kant van pin 7 en 8 van de chip gemonteerd (je moet de aansluitingen heel kort houden). Je kunt de aansluiting eenvoudig installeren met behulp van een 3/8" krimp oog (ring terminal). Verwijder de mantel van het krimp deel, maak het krimp gedeelte plat, buig het zo ver mogelijk van het oog af (90 graden) en soldeer het op de printplaat. Soldeer een kleine weerstand van 50 ohm (R3) van de middelste pin van de BNC connector naar massa. Deze weerstand bepaalt de impedantie van de TDR-test. Als je een kabel van 75 ohm wilt testen, gebruik je een



weerstand van 75 ohm. Zo simpel is het!

Soldeer ook een weerstand van 300 ohm vanaf de middelste pin van de BNC om de inverter te ontlasten en binnen de 25 milliampère uitgangscapaciteit te houden. maak een lusje van de overlengte van de weerstandsdraad aan de BNC-kant, zodat je een plek hebt om je oscilloscoopprobe aan te sluiten. Het andere uiteinde van de weerstand soldeer je gewoon aan pin 6. Soldeer een weerstand van 100k van pin 1 naar de verbinding van pin 2 en 3. Soldeer een condensator van 0,005 μ F (of daaromtrent) van pin 1 naar de verbinding van pin 4 en 5 en buig de resterende draad om naar pin 9 en soldeer alle drie de verbindingen.

De batterij kan vrijwel alles zijn wat je wilt gebruiken, van 2 tot 5 Volt. Ik had een kleine AA-batterijbehuizing met twee cellen en een ingebouwde schakelaar, dus die heb ik voor de mijne gebruikt. Ik heb de printplaat gewoon op de behuizing gelijmd. De schakeling trekt slechts ongeveer 6 milliampère. Soldeer de minpool van de voeding aan het massavlak. De pluspool van de voeding gaat naar pin 14. Misschien niet noodzakelijk, maar wel een goede gewoonte, is om een condensator van 0,1 microfarad van pin 14 naar massa te solderen als ontkoppeling van de voeding.

Een 10nF condensator van pin 8 naar een soort probestructuur maakt het apparaatje compleet. Dit heb ik pas later bedacht en wordt niet gebruikt als tijdsdomeinreflectometer. De snelle stijgtijd van de 1 kHz blok golf produceert een toon van audio tot en met HF. Mooi en krachtig bij audio, omdat je daar een hoog signaalniveau nodig hebt, lager bij middenfrequente frequenties, en nog steeds bruikbaar tot 180 MHz (S8-sigitaal op AM). Ik raad aan om een probe te monteren, geïsoleerd behalve de punt (ik gebruikte AWG#12 installatiedraad), aan het uiteinde van de printplaat tegenover de BNC-aansluiting. Gebruik een vijl om de probepunt te slijpen. Als je een 1 kV keramische condensator gebruikt, beschermt dit de schakeling tegen de hoge spanningen die vaak voorkomen in

buizenapparatuur. Terwijl je naar de toon luistert via de luidspreker van een radio, kun je met de probe door de verschillende trappen lopen om te bepalen waar het probleem zit. Handcapaciteit zorgt voor voldoende signaalretour.

Om als TDR te gebruiken, sluit je de probe van je 10x oscilloscoop aan op het lusje bij de BNC en de aardklem op de nabijgelegen massa. Met de probe met lange punt en de normale aardingsklem krijg je behoorlijk wat overshoot aan de voorflank van de puls. Je kunt de kleine aardingsklem met veer gebruiken om deze overshoot te verminderen, maar daar is eigenlijk geen reden voor. Die overshoot heeft geen invloed op de TDR-meting. Je moet hiervoor je 10x oscilloscoopprobe gebruiken. Zonder deze probe is de bandbreedte te laag en niet gecompenseerd.

Schakel de spanning in en je zou een mooie blok golf moeten zien. Verhoog de scansnelheid (horizontale afbuiging) om de voorflank van de puls in detail te bekijken. Sluit een 50 ohm kabel aan op de BNC-aansluiting en let op de verandering. Als de kabel enkele meters lang is en het uiteinde open is, zie je een extra stap verschijnen. De tijd tussen de eerste puls en deze nieuwe stap (die net verscheen) geeft de tijd weer die het signaal nodig had om zich door de kabel voort te planten, terug te kaatsen van het open uiteinde en terug te keren naar de probe. [1]

Als de discontinuïteit (in dit geval het open uiteinde) een hogere impedantie heeft dan die van de kabel, dan is de retourpuls additief en heeft deze een hogere amplitude dan het eerste deel van het display. Als je het verre uiteinde kortsluit, is deze retourpuls subtractief en wordt deze weergegeven als een stap lager dan de amplitude van de vorige stap. De amplitude van de retourstap wordt beïnvloed door zowel het kabelverlies als de verandering in impedantie. Een afgeklemd kabel zal een dip in de golfvorm laten zien, omdat dit een discontinuïteit met een lagere impedantie creëert.

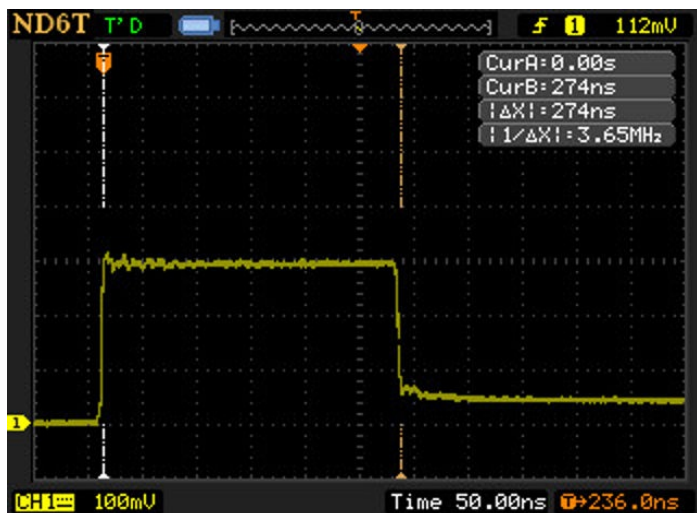


Foto 1. Kortgesloten kabel

Sluit de kabel af met de juiste weerstand en hij lijkt oneindig lang voor de TDR. Geen reflectie. [2]

Bij lange kabels zullen de details van het gereflecteerde signaal minder worden vanwege de hoge frequentieverliezen van de coaxkabel. Toch kun je met een verrassende nauwkeurigheid enkele duizenden meters meten. Niet dat je dat ooit nodig zal hebben. Na duizenden meters coax is je vermogen sowieso weg...

Omdat de lichtsnelheid in vacuüm ongeveer 299.792 kilometer per seconde bedraagt, zal het licht iets minder dan 30 centimeter per nanoseconde afleggen. Als we de afstand tot een kabelbreuk meten, moeten we er rekening mee houden dat het signaal heen en weer naar de breuk moet reizen. Zo kunnen we de vertraging van het terugkerende signaal in nanoseconden meten en vermenigvuldigen met 14,99 om de afstand tot de kabelbreuk te berekenen (in centimeters)... dat wil zeggen als het signaal zich in de vrije ruimte zou voortplanten, wat niet het geval is. We moeten rekening houden met de lagere voortplantings-snelheid door het diëlektricum van de kabel! Hier komt de Velocity Factor (VF) om de hoek kijken. Deze wordt gepubliceerd voor de meeste kabelontwerpen, maar als je een stuk van de kabel hebt om mee te experimenteren, kan je een nauwkeurigere waarde krijgen. Vermenigvuldig met de VF-waarde om de juiste afstand te krijgen.

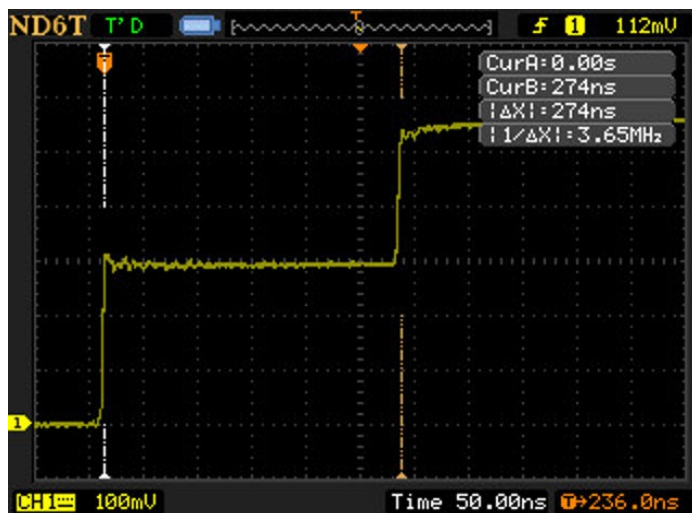


Foto 2. Open kabel

De kabel in foto 1 (kortgesloten) en 2 (open) had een gemeten vertraging van 274 nanoseconden. De VF is 0,75 voor dit type coaxkabel, dus elke nanoseconde vertegenwoordigt 11,24cm ($0,75 \times 14,99$). De afstand tot de fout (in dit geval het uiteinde van de kabel) is dus $274 \times 11,24 = 3079,76\text{cm}$ of 30,80 meter. De haspel zou 30 meter moeten zijn, wat dichtbij genoeg is, aangezien de gepubliceerde VF mogelijk niet zo nauwkeurig is. 1% is een goede nauwkeurigheid.

Maar wacht! Er is nog meer!

Door de uitgangsstijgtijd van dit apparaatje te meten, kan je de werkelijke 3 dB-bandbreedte van je oscilloscoop en probe benaderen. Deel 0,35 door de gemeten stijgtijd en het resultaat is die bandbreedte. Handig.

Discrete logic chips zijn zo gangbaar en goedkoop dat we ze vaak negeren. Voor een paar centen krijgen we hoogwaardige versterkers en schakelaars die verkrijgbaar zijn in plastic behuizingen die robuust, gebruiksvriendelijk en ideaal zijn voor batterijvoeding. Overweeg er een paar in een la te leggen, klaar voor eenvoudige projecten.

Opmerkingen:

[1] Het maakt niet uit of je het begin of het maximum van de puls gebruikt, zolang je maar het corresponderende punt gebruikt voor zowel

de start- als de eindmeting. De meeste digitale oscilloscopen hebben een automatische cursorfunctie. Gebruik die als je die hebt.

[2] Merk op dat ik "weerstand" zei in plaats van "impedantie". Dit is een breedbandtest, van

gelijkspanning tot buiten de bandbreedte van je oscilloscoop, en de afsluiting moet onafhankelijk zijn voor frequenties binnen dat bereik. Antennes worden weergegeven als open of kortgesloten, afhankelijk van hun voedingsmethode.

PA3CNO's Blog

In de vorige RAZZies schreef ik over de portable JPC-12 antenne; een makkelijk op te bouwen vertical die in een kleine tas geleverd wordt en ideaal lijkt voor portable gebruik. Daarop ontving ik een email van Eric PA7FK met een hoop aanvullende informatie die ik jullie niet wil onthouden. Voor deze antenne zijn diverse modificaties en aanvullingen beschikbaar. Één ervan betreft het vervangen van de RVS windingen van de spoel door verzilverd koperdraad, wat de efficiency van de antenne aanzienlijk zou verbeteren. Ik heb daar eens naar gekeken, en de soortelijke weerstand (hoofdstuk 3 van de VRZA N-cursus!) van RVS is $0,72 \cdot 10^{-6}$. Dat van zilver is $0,016 \cdot 10^{-6}$. Dat scheelt een factor 45 in weerstand! De hoeveelheid draad op de spoel wordt geschat op 4,5m, wat betekent dat de weerstand van de RVS spoel die met 1mm diameter draad gewikkeld is, op ongeveer $4,2\Omega$ komt, behoorlijk in de buurt van wat gemeten is. Vervang je de wikkeling door 1mm diameter verzilverd koperdraad, dan kom ik op ongeveer $0,09\Omega$. Dat scheelt nogal wat. Men heeft ook thermische foto's gemaakt van de antenne met vermogen erop, en dan zie



je de warmte t.o.v. de omgeving, zie de foto links onder. (Ik vraag me af hoeveel vermogen ze erin hebben gepompt om dat plaatje te krijgen...) Andere uitbreidingen van de antenne zijn een capacitieve hoed voor de lage banden:



Dit kan je bovenop de antenne plaatsen waardoor je minder inductie nodig hebt om op de lagere banden te kunnen werken. En minder spoel betekent minder verlies. Deze uitbreiding is bij AliExpress te vinden onder "PAC-12 Capacitive Cap" voor rond de €11.

Andere discussies over deze antenne gaan over de bijgeleverde radialen. Ook daar is weer een uitbreiding voor bedacht: de "PAC-12 network disk" (daar kom je toch niet op) waar je weer extra radialen aan kunt maken. Te koop van €12 tot €30 euro bij alweer AliExpress.



Bij deze de links naar de artikelen.

Antenne algemeen:

<https://ka7oei.blogspot.com/2024/05/observations-analysis-and-modifications.html>

Verliezen in en modificatie van spoel:

<https://ka7oei.blogspot.com/2024/07/rewinding-stainless-steel-for-silver.html>

In een tweede mail schreef Eric over baluns en transformatoren die hij toegepast heeft. Één ervan herkende ik: de Triple Radio Switched Balun zoals die bij de Buddipole gebruikt werd. Voor wie 'm niet kent: de Buddipole was een dipool antenne op een standaard, die eveneens gebruikt werd voor portable opstellingen. Echter, een dipool zo kort boven de grond zorgt voor een lage impedantie en dat is waar de Triple Ratio balun zijn werk kan doen. Maar voor de JPC-12 antenne lijkt me dat overbodig. Je kunt de impedantie immers aanpassen met de slider op de spoel. Een mantelstroomspoel snap ik, maar waarom een balun? Er is niets gebalanceerd aan de vertical en de meeste sets zijn eveneens ongebalanceerd (want voorzien van een BNC of UHF connector). Dus die zou ik er niet tussen zetten. Wil je weten hoe het ook al

weer zat met die Triple Ratio Baluns, lees er dan de volgende artikelen op onze website nog eens op na:

<https://pi4raz.nl/2007/08/09/triple-ratio-balun/>

<https://pi4raz.nl/2007/08/21/resultaten-trsb/>

<https://pi4raz.nl/2009/03/16/vervolg-triple-ratio-balun/>

Regelmatig moet ik met amateurs de discussie aan over het werken met QRP. Ik hou ervan: niets zo leuk als met een paar Watt een verbinding maken waarbij je tegenstation van zijn stoel valt van verbazing als ze horen met wat voor vermogen je werkt. Ja, hoor ik dan, maar met CW werkt dat. Met SSB gaat dat gewoon niet. En dat is echt onzin. Laten we eens uitgaan van wat in zowel SSB als CW QRP genoemd wordt: max 5 Watt. De meeste amateurs gebruiken het maximum van hun set, meestal 100W. Dat is een verschil van een factor 20, en in dB's is dat 13 dB. (een factor 10 is 10dB en een factor 2 is 3dB. Dus is een factor 20 gelijk aan 13dB). Hoeveel was een S-punt ook al weer? Juist, 6dB. Die factor 20 is dus iets meer dan twee S-punten. Met andere woorden: een station dat bij jou S9 binnenkomt (echte S-punten, dat dan weer wel), moet jou dus nog altijd met een S7 kunnen horen. Natuurlijk kom je in stedelijke omgevingen dan al gauw op de ruisvloer terecht, maar hee, hoe vaak geef je geen S9+20 aan je tegenstation? Dus 20dB over 9? Dan moet hij jou nog altijd 7dB over 9 horen. Randvoorwaarde is dan wel dat je een fatsoenlijke antenne gebruikt. Maar dan nog. Mijn loop met een omtrek van 5m geeft volgens 66pacific.com op 40m een demping van 6,3dB t.o.v. een full size dipool. Met andere woorden: ik raak nog een extra S-punt kwijt; in totaal 19,3dB. Maar dan ben ik nog steeds S9 bij het tegenstation dat ik S9+20 gaf... Overigens is dat hetzelfde als 1W in een full size dipool. Met andere woorden: QRP is zo gek nog niet. Het lukt je heus wel om met wat minder vermogen en een niet optimale antenne verbindingen te maken. Ook in SSB. Uiteraard moet je dat doen als de band een beetje rustig is. Met een band vol conteststations ben je kansloos. Maar ik daag je uit: laat me eens weten wat je heb gewerkt met 5W!



Afdelingsnieuws

Op 23 april j.l. heeft de jaarvergadering van de afdeling A64 (Zoetermeer) van de VERON plaatsgevonden. Na het voorlezen van het financiële jaarverslag heeft het zittende bestuur te kennen gegeven hun posities ter beschikking te stellen en zich niet herkiesbaar te stellen. Als nieuwe bestuurskandidaten zijn voorgesteld:

Voorzitter: Frank Waarsenburg, PA3CNO

Penningmeester: Wim Schilperoort, PE1PWR

Secretaris: Bart Weerstand, PA3HEA

Het oude bestuur heeft besloten een vierde, algemeen bestuurslid aan het bestuur toe te voegen: Gert Baak, PE0MGB, de voormalig secretaris. Gert beheert al enige tijd de logistiek van de vereniging en zal dat voortzetten als algemeen lid van het bestuur. Tijdens de vergadering hebben zich geen andere kandidaten aangemeld en er zijn geen

bezwaren ingediend tegen de voorgestelde nieuwe bestuursleden, waarmee deze officieel benoemd zijn. Wij bedanken de voormalige bestuursleden voor hun jarenlange inzet en wensen het nieuwe bestuur veel succes in hun nieuwe functie.

In de maand mei zijn de afdelingsbijeenkomsten op de woensdagen 14 en 28 mei (later kan niet). Woensdag 14 mei is de eerste bijeenkomst van de maand en zal ijs en weder dienende de QSL-manager aanwezig zijn voor het uitwisselen van de kaarten. Vanaf 20:00 is iedereen met interesse in onze hobby weer welkom in buurthuis 't Span, Sullivanlijn 31 in Zoetermeer. Houd er rekening mee dat in het buurthuis niet gepind kan worden, alleen cash wordt geaccepteerd! Geld opnemen kan eventueel in de nabijgelegen Albert Heijn.

