



Mijn eerste Software Defined Radio

in minder dan 15 minuten klaar

Clemens Valens (Elektor Labs)

Wist je dat je een FM radio-ontvanger kunt bouwen door gewoon een paar blokken en lijnen te tekenen op een virtueel doek? Na het lezen van dit artikel wordt tekenen en schilderen gerarandeerd je nieuwe hobby!



Figuur 1. Het in dit artikel gebruikte SDR-front-end is de HackRF One van Great Scott Gadgets.

Software-gedefinieerde radio (Software Defined Radio, SDR) gaat over het moduleren en demoduleren van radio-signalen in software. In plaats van speciale elektronische schakelingen, zoals mixers of demodulators, gebruikt SDR software. Natuurlijk is er wat hardware nodig om analoge radiosignalen om te zetten in digitale en omgekeerd, maar dat is dan ook alles wat je nodig hebt. Om het eenvoudig te houden, beperken we ons in dit artikel tot de ontvangst van radiosignalen, maar het meeste van het gepresenteerde materiaal geldt ook voor zenden. SDR kan in beide richtingen werken.

SDR kan alles

SDR biedt vele voordelen ten opzichte van analoge radio. Ten eerste begrijpt SDR elke modulatie-techniek die je maar kunt bedenken, en meer. Dit betekent dat je met SDR niet alleen AM- en FM-radio kunt beluisteren, maar ook WiFi of Bluetooth kunt gebruiken, of digitale TV en DAB+ kunt decoderen, of met HAM-radio kan spelen – en nog veel meer.

SDR kan dit omdat het de modulatie- of demodulatie-techniek in software implementeert, en software is gemakkelijk te veranderen. Omdat alles in software gebeurt, kan SDR ook dingen doen die met alleen elektronische componenten zeer moeilijk of zelfs onmogelijk zijn.

Past altijd

Een tweede voordeel van SDR is dat je voor al dit lekkers slechts één stuk hardware nodig hebt. Je hebt geen aparte FM-ontvanger en WiFi-ontvanger en digitale TV-ontvan-

ger en wat al niet meer nodig. SDR gebruikt hetzelfde front-end voor alles.

Dit front-end is een analoog/digitaal-converter (ADC) met een antenne die analoge radiosignalen omzet in digitale signalen die met een computer kunnen worden verwerkt. Voor het zenden is een digitaal/analoog-converter (DAC) nodig. In werkelijkheid is dit front-end méér dan alleen een ADC en/of DAC. Het voert ook enige menging uit om het gewenste RF-sigitaal in het bereik te brengen dat kan worden bemonsterd (en omgekeerd bij zenden). De basisfunctie blijft echter hetzelfde.

SDR-front-ends zijn er in soorten en maten; je moet er een kiezen die bij je budget en wensen past. Hoe duurder de converter, hoe meer je ermee kunt doen. Hier gebruiken we de HackRF One van Great Scott Gadgets [1] (**figuur 1**). Dit is niet het goedkoopste SDR front-end, maar het ondersteunt ontvangen en zenden, en werkt tot 6 GHz. Ook en niet onbelangrijk wordt het ondersteund door vele SDR-softwareplatforms zoals GnuRadio [2], SDR Sharp (SDR#) [3] en SDR++ [4]. Voor het hieronder beschreven project zullen andere (goedkopere) SDR-front-ends waarschijnlijk even goed werken, zolang ze in de FM-radio band (87,5...108 MHz) kunnen functioneren.

Open source en hardware

Een ander leuk aspect van SDR is dat veel ervan open source en open hardware is, gemaakt door een grote

gemeenschap van enthousiastelingen. Zij ontwerpen en publiceren schema's voor front-ends en ontwikkelen softwaretools en algoritmen voor SDR-toepassingen. HackRF One is ook open source, en de ontwerpbestanden zijn beschikbaar op GitHub [5].

Vereisten

Zoals al opgemerkt, heb ik voor dit SDR-project HackRF One gebruikt als front-end, in combinatie met een telescoopantenne van 88 cm. We zullen GNU Radio Companion gebruiken voor het softwaredeel. Eigenlijk gebruiken we GNU Radio Companion oftewel GRC, de grafische interface voor GNU Radio. GNU Radio is beschikbaar voor Linux, Windows en macOS, dus de kans is groot dat jouw computer het ook kan draaien. Het draait zelfs op een Raspberry Pi.

Programmeren = slepen en neerzetten!

GRC is een grafisch programmeertool waarmee je een radiotoepassing kunt maken zonder echt te programmeren. Je sleept gewoon blokken op een 'doek' die je onderling verbindt met virtuele draden. De programmering blijft beperkt tot het configureren van de blokken. GRC wordt geleverd met een grote bibliotheek met blokken voor het bouwen van allerlei modulatie- en demodulatieschema's. We beginnen met een leeg doek. Nou ja – niet helemaal leeg, want het bevat al twee blokken: een *Options*-blok en een *Variable*-blok. Dubbelklikken met de linker muisknop opent de eigenschappen van een blok, zodat je de parameters ervan kunt wijzigen. Met het *Options*-blok kun je enkele low-level opties definiëren, en je kunt een naam voor het doek instellen – zie **figuur 2**. We laten overal de standaardwaarden staan.

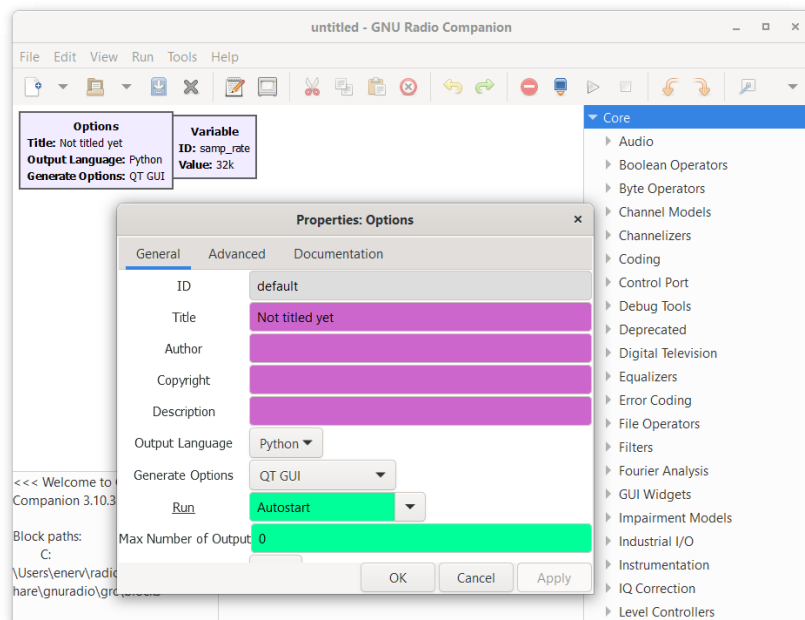
Globale bemonsteringsfrequentie

Het blok *Variable* definieert een globale sample rate-variabele die `samp_rate` heet. Deze wordt gebruikt in andere blokken. Strikt genomen is het niet nodig, maar het is praktisch, en dus laten we het zo. De waarde is echter te laag voor het bemonsteren van FM-signalen; daarom veranderen we hem in 10 miljoen. De maximale waarde voor HackRF One is 20 miljoen. Als je een te hoge waarde neemt, kan je computer het moeilijk bijbenen.

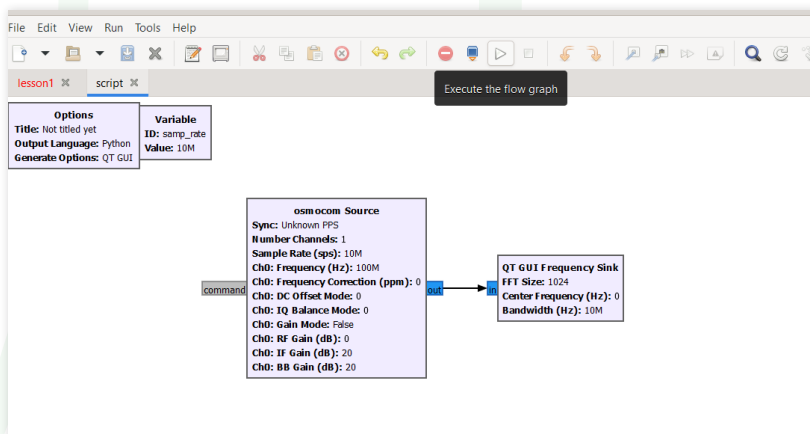
Voeg een RF-source toe

Nu hebben we RF-invoer nodig. In GRC wordt dit een *source* (bron) genoemd. Er is geen speciaal source-blok voor de HackRF One. De *Soapy*-sectie heeft een Soapy HackRF Source, maar die werkte bij mij niet. De *osmocom*-source werkte wel. Het OsmoSDR front-end bestaat niet meer, maar het blok ondersteunt ook andere SDR-hardware, waaronder HackRF One.

Men een dubbelklik open je het eigenschappen-venster, en we zien dat de sample rate is ingesteld op `samp_rate`, de naam van het variable-blok dat GRC ons gaf. We hoeven hier maar één parameter te veranderen, en dat



Figuur 2. Een nieuw project in GNU Radio Companion (GRC) wordt gestart met twee blokken. Dubbelklikken op een blok opent de eigenschappen ervan.



is de RF-versterking van kanaal 0, die te hoog is voor FM-radiosignalen die nogal sterk zijn. We kunnen deze op nul zetten.

Spectrumanalyser

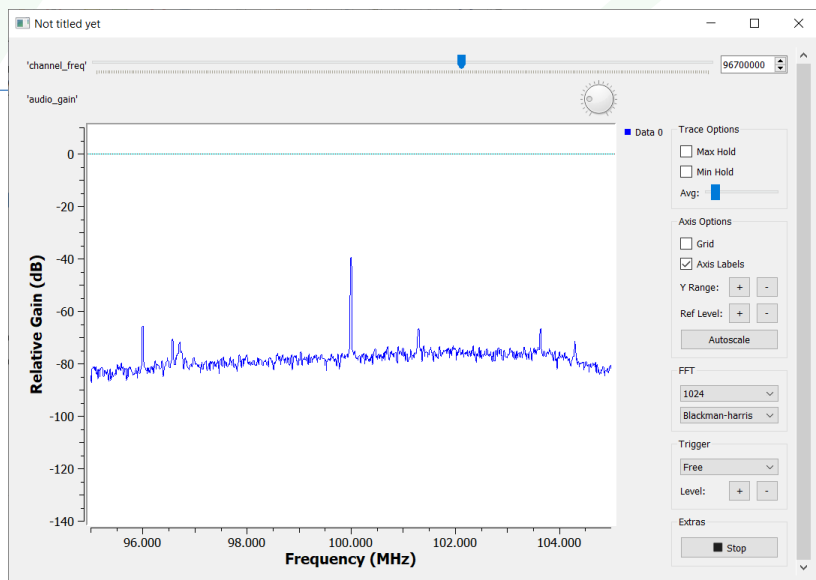
Om snel te kijken of alles werkt, kunnen we een spectrum-analyzer toevoegen aan het doek en die aansluiten op de uitgang van het blok RF-source. Het gaat om de QT GUI Frequency Sink en die is te vinden in de *Instrumentation*-sectie van GRC. De nomenclatuur van deze blokken kan een beetje verwarrend zijn, en dat maakt ze soms moeilijk te vinden.

Om de frequentie-ingang te verbinden met de RF-source, klik je op de ingang en vervolgens op de uitgang van de source. Andersom gaat ook – net als tekenen met ingedrukte linkermuisknop. Merk op dat je alleen in- en uitgangen van dezelfde kleur met elkaar kunt verbinden, blauw in dit geval.

Eerste poging

Je kunt nu op de knop *Play* (die met het driehoekje erop) klikken om het flowdiagram uit te voeren (**figuur 3**). Je moet natuurlijk, als je dat nog niet hebt gedaan, je ontwerp opslaan voordat je verder kunt gaan. Als het goed is,

Figuur 3. Klik op de play-knop om je eerste flowdiagram uit te voeren.



▲
Figuur 4. Na activeren van het Control Panel van de QT GUI Frequency Sink heb je toegang tot verschillende regelaars.

wordt een frequentieplot zichtbaar. Een beetje 'spelen' met de antenne zou iets andere resultaten moeten opleveren, maar dat is misschien moeilijk te zien. Voor een duidelijker beeld moeten we de frequentieplot configureren. Klik op de **Stop**-knop (die met het vierkantje) om de flow-grafiek te stoppen. Dubbelklik op het frequentieblok om de eigenschappen ervan te openen en stel **averaging** (middeling) in op gemiddeld. Stel de middenfrequentie in op dezelfde waarde als de frequentie van Channel 0 van de osmocom-source, in ons geval 100 MHz. Klik vervolgens op het tabblad **Config** en zet **Control Panel** op **Yes**. Sluit het eigenschappenvenster en klik weer op **Play**. Nu heeft de frequentieplot een bedieningspaneel en kun je het vakje **Max Hold** aanvinken (figuur 4). Het spelen met de antenne zou nu beter zichtbaar moeten worden en als dat zo is, weet je dat je RF-signalen kunt ontvangen.

Het is allemaal plug&play!

Het leuke is dat we niets hoefden te doen om het flowdiagram met HackRF One te laten communiceren. Er hoeven geen drivers te worden geïnstalleerd, geen poorten geselecteerd, het is allemaal plug&play.

Variabelen zijn handig

We hoefden alleen de middenfrequentie van de frequentiesink in te stellen op dezelfde waarde als die van de RF-bron, wat betekent dat deze op ten minste twee plaatsen nodig is. Als we deze vervangen door een

globale variabele, zoals GRC deed voor de sample rate, hoeven we hem maar één maal in te stellen.

Kopieer het **sample rate** variabeleblok, open de eigenschappen ervan, en verander de ID in **center_freq**. Stel de waarde in op 100 MHz, stel de Channel 0-frequentie van de osmocom-source in op **center_freq**, en stel de middenfrequentie van de frequency sink in op **center_freq**. De twee parameters zijn nu met elkaar verbonden. Als we de waarde van de variabele **center_freq** veranderen, zal deze ook veranderen in alle andere blokken die er gebruik van maken.

Mengen

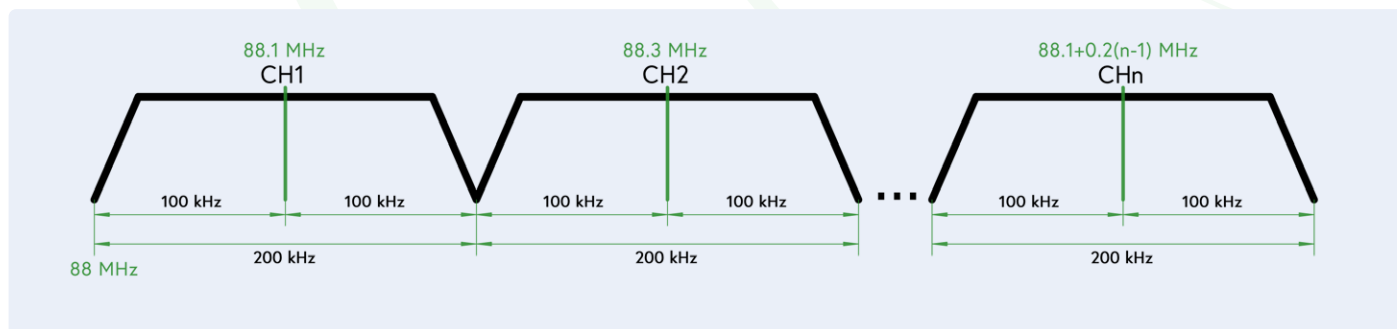
Om iets te bereiken met SDR moet je natuurlijk wel enig idee hebben hoe radio's werken. De algoritmen in SDR doen dezelfde dingen als de elektronische schakelingen in een analoge radio. Om een FM-radiozender te ontvangen moeten we er dus op afstemmen, het signaal demoduleren en filteren en het hoorbaar maken.

Afstemming kan plaatsvinden met een zogenaamde mixer of frequentievermenigvuldiger. Wanneer dit gebeurt met kwadratuur- of complexe signalen, kunnen hiermee frequenties naar boven en beneden worden verschoven. In GRC komen de blauwe in- en uitgangen overeen met complexe floating-point signalen die geschikt zijn om te worden gemengd.

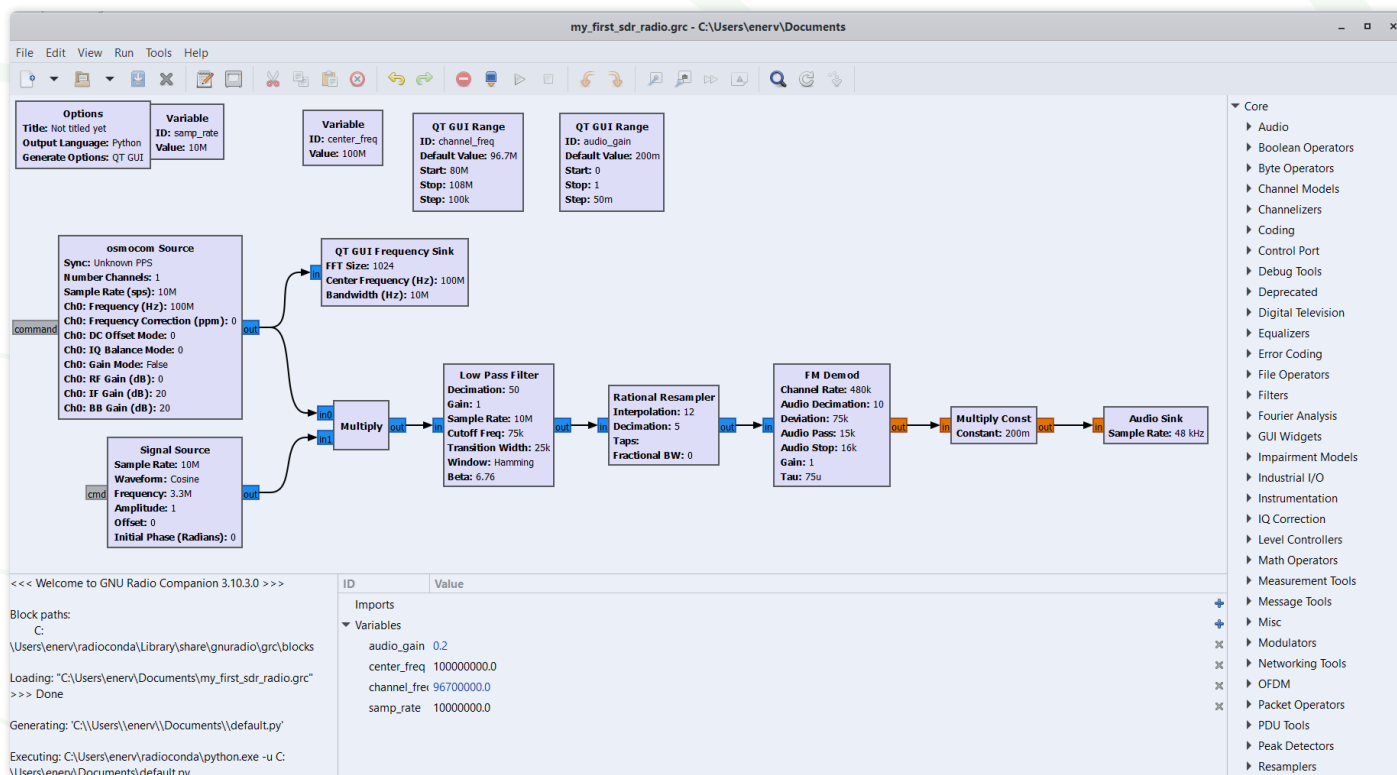
Het doel is om de gewenste frequentie te verschuiven naar het midden van de frequentieplot, waar we hem gemakkelijker uit kunnen halen. Om af te stemmen op een radiostation hebben we dus een vermenigvuldiger nodig en een ander signaal dat de mate van frequentieverschuiving bepaalt. Een **Multiply**-blok is beschikbaar in de sectie **Math Operators**. Voor het afstemsignaal kunnen we het blok **Signal Source** uit de sectie **Waveform Generators** gebruiken.

Afstemmen op een zender

FM-radiozenders hebben een bekende frequentie, die we de kanaalfrequentie zullen noemen. Omdat we deze frequentie gemakkelijk willen kunnen veranderen, maken we er een globale variabele voor aan die we **channel_freq** noemen. Waar ik woon zendt een radiostation op



Figuur 5. De FM-frequentieband is verdeeld in kanalen met een breedte van 200 kHz.



96,7 MHz uit, dus zet ik die waarde in `channel_freq`. De mixer moet de kanaalfrequentie verschuiven naar de middenfrequentie, dus moet de frequentie van de signaalbron het verschil van de twee zijn. Het verschil tussen 100 MHz en 96,7 MHz is 3,3 MHz en dat is inderdaad de frequentie in het signaalbron-blok.

Omlaag converteren

Met het station op de centrale frequentie moeten we het signaal extraheren en omlaag mengen zodat het kan worden gedemoduleerd. Extraheren betekent filteren en omlaag converteren komt in SDR overeen met het reduceren of decimeren van de sample rate. Het **Low Pass Filter** uit de sectie **Filters** combineert beide bewerkingen in één blok.

De FM-band is verdeeld in kanalen. De breedte verschilt per land, maar ligt meestal rond de 200 kHz. Een radiostation zit in het midden van een kanaal, wat betekent dat het aan weerszijden 100 kHz bandbreedte heeft (figuur 5). De minimale bemonsteringsfrequentie die nodig is voor een signaal van 100 kHz is 200 kHz (Nyquist-Shannon theorema). Onze bemonsteringsfrequentie is 10 MHz, en daarom stellen we de deelfactor van het filterblok in op $10 \text{ MHz} / 200 \text{ kHz}$ oftewel 50. Om het gewenste signaal te extraheren, hebben we geen supersteil filter nodig. Een afsnijfrequentie van 75 kHz met een overgangsbreedte van 25 kHz is prima voor een totale bandbreedte van 100 kHz.

Demoduleren

We hebben nu een FM-gemoduleerd kwadratuursignaal met een sample rate van 200 kHz dat we willen demoduleren. Hiervoor kunnen we het blok **FM Demod** uit de sectie **Modulators** gebruiken. Op de uitgang van de demodulator staat een audiosignaal dat we naar een audio-sink kunnen sturen, in ons geval de geluidskaart van de computer.

Problemen met de sample rate

Het probleem waar we nu tegenaan lopen is sample-rate matching. De audio-sink ondersteunt verschillende bemonsteringsfrequenties die afhankelijk zijn van je geluidskaart. Op mijn computer is de verhouding tussen de mogelijke audiosamplesnelheden en de 200 kHz ingangs-bemonsteringsfrequentie in alle gevallen een niet-geheel getal. Maar de FM demodulator kan alleen delen door een geheel getal. Hoe kunnen we dit probleem oplossen?

De oplossing is om een sample rate-converter in te bouwen die fractionele conversie kan uitvoeren. GRC heeft hiervoor twee opties: een fractionele resampler en een rationale resampler. Aangezien onze in- en uitgangsfrequenties gehele waarden zijn, kunnen we de rationale resampler uit de sectie **Resamplers** gebruiken. Met een fractionele resampler gaat het ook, maar dat kost veel meer rekenkracht.

Voor de beste audiokwaliteit stellen we de audio-bemonsteringsfrequentie in op 48 kHz. Als we dan de demodulator laten werken met bijvoorbeeld 10 keer die frequentie, 480 kHz, hebben we een sample rate-conversie nodig van $200 : 480$. Dit kan worden vereenvoudigd met behulp van de grootste gemene deler van beide waarden, die 40 is. $200 : 40 = 5$ en $480 : 40 = 12$, dus als we eerst omhoog converteren of interpoleren met een factor 12 en dan omlaag converteren met een factor 5, wordt een frequentie van 200 kHz omgezet in een frequentie van 480 kHz. We moeten deze snelheid in het **FM Demod**-blok instellen, samen met een deelfactor van tien, om ervoor te zorgen dat de uitgangsfrequentie overeenkomt met de ingangsfrequentie van de audio-sink.

Bedieningselementen toevoegen

Als je dit flowdiagram uitvoert met de kanaalfrequentie vast ingesteld op een bestaand FM-radiostation in de



Figuur 6. Het flowdiagram van de FM-ontvanger met afstemming en volumeregeling. Als je het uitvoert, ziet het eruit als in figuur 4.

buurt, zou je nu het radioprogramma moeten horen. Is dat niet cool?

We kunnen onze radio flink verbeteren door een afstemmogelijkheid toe te voegen, zodat je gemakkelijk verschillende stations kunt ontvangen. Hiervoor kunnen we een *QT GUI Range* blok gebruiken uit de *GUI Widgets*-sectie. Stel zijn ID in op `channel_freq`. Vul de parameters in en zorg ervoor dat ze allemaal binnen het bereik blijven. Merk op dat het geen zin heeft om een (zeer) kleine stapwaarde in te stellen, aangezien de meeste FM-stations op MHz-frequenties met slechts één decimaal zitten; 100 kHz is waarschijnlijk prima.

Je kunt een heleboel vormen voor de bediening kiezen, maar een teller plus schuifregelaar is het meest praktisch, omdat je dan de frequentie als een numerieke waarde kunt zien. Je kunt ook direct een frequentie intypen. Hernoem of verwijder het oude `channel_freq` variabeleblok, omdat het niet langer nodig is en conflicten met de schuifregelaar veroorzaakt. Je kunt de radio nu afstemmen zoals elke andere radio.

Signaalkleuren

Een andere verbetering is de toevoeging van een volumeregelaar. Die is vergelijkbaar met de afstemknop, en je kunt hem kopiëren en de parameters ervan aanpassen. Hernoem de ID naar `audio_gain`. Kies een *Multiply Const* uit de *Math Operators*-sectie en voeg deze in tussen het *FM Demod*-blok en de audio-sink. Merk op dat de verbindingen rood zijn. Dat komt omdat de input en output van de vermenigvuldiger blauw zijn in plaats van oranje. Ze zijn niet van hetzelfde type.

Open de eigenschappen van de vermenigvuldiger en stel *IO Type* in op `float`. Stel ook het veld *Constant* in op `audio_gain` om het te verbinden met de schuifregelaar. Sluit het blok en zie hoe de draden zwart zijn geworden, aangezien alle audio in- en uitgangen nu oranje zijn.

Uitgangsopties

Standaard is de uitvoer van GRC een Python-script met GUI zoals gedefinieerd door de *Generate Options* in het *Options*-blok. Maar er zijn andere mogelijkheden, zoals C++. Als je alle GUI-blokken uit het flowdiagram verwijdert en *Generate Options* instelt op *No GUI*, kan het script buiten GRC worden uitgevoerd. Zo kunt je stand-alone SDR-toepassingen bouwen.

Dat is het. Voer het flowdiagram uit en geniet van je softwaregedefinieerde FM-ontvanger! Om dieper in te gaan op de fascinerende wereld van SDR, raad ik ten eerste aan de uitstekende videocursus van Michael Ossmann [6] te bekijken, die de inspiratie vormde voor dit artikel. 

220659-03

Vragen of opmerkingen?

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



Gerelateerde producten

- > Great Scott Gadgets HackRF One Software Defined Radio (1 MHz to 6 GHz) (SKU 18306) www.elektor.nl/18306
- > Great Scott Gadgets ANT500 Telescopic Antenna (75 MHz to 1 GHz) (SKU 18481) www.elektor.nl/18481
- > Elektor SDR Hands-on Kit (SKU 19041) www.elektor.nl/19041


SDR-gebaseerde
FM-radio-ontvanger op
Elektor.TV



WEB LINKS

- [1] HackRF One bij Great Scott Gadgets: <https://greatscottgadgets.com/hackrf/>
- [2] GNU Radio en GRC: <https://www.gnuradio.org/>
- [3] SDR# ofwel SDR Sharp: <https://airspace.com/download/>
- [4] SDR++: <https://www.sdrpp.org/>
- [5] HackRF One op GitHub: <https://github.com/greatscottgadgets/hackrf>
- [6] Videocursus over SDR door Michael Ossmann: <https://greatscottgadgets.com/sdr/>

