

Onderwerp: Radio Data System

Leerdoelen

Aan het einde van dit onderwerp ben je in staat om:

- Bitrate en samplerate te begrijpen en te berekenen
- De basisprincipes van digitale modulatie te begrijpen
- RDS signaal van een FM zender te kunnen decoderen

Vorbereiding op de les

- Lees de hoofdstukken van PySDR zoals vermeld in de planning op de wiki
- Bekijk de introductie videos op Brightspace.
- **Vind een ASK apparaat om te analyseren.**

Beoordeling

Maak als groep een screencast van maximaal 10 minuten waarin jullie de uitwerking van elke opdracht van dit onderwerp onderbouwen. Zorg ervoor dat beide personen aan woord komen en dat je aan de volgende punten aandacht besteedt:

1. Leg uit wat je in de wiskunde hebt gedaan.
2. Onderbouw voor de GNU Radio Companion schemas met name de instellingen van de blokken die je hebt gebruikt. Waarom heb je deze instellingen gekozen?
3. Laat zien dat het werkt, dit mag een korte opname met je telefoon zijn waarin in duidelijk is dat jouw schema worden gebruikt en dat je muziek hoort.

De screencast wordt beoordeeld volgens de rubric en beschrijving die op brightspace is geplaatst. Hou de deadlines van de cursushandleiding aan.

Opdrachten

27 pt

Opdracht 1: Bitrate en samplerate

In de video's is uitgelegd hoe je van bytes naar meerdere samples per bit kunt komen, en hoe je van meerdere samples per bit terug kunt gaan naar bytes. Maak 1 flowgraph waarbij je jouw naam omzet naar een continue bitstream met een bitrate van 1 kbit/s. Gebruik voor het genereren van jouw naam een [Vector Source](#) met `[ord(k) for k in "naam\n"]`^{ab} als vector. Voeg aan deze bistream wat ruis ([Noise Source](#)), gain en een offset toe. Zorg er nu voor in dezelfde flowgraph dat je deze ruis, gain en offset weer kunt verwijderen en dat je jouw naam weer terugkrijgt in byte-samples. Je kunt de uiteindelijke bytes terugzien door een [TCP Sink](#) in te stellen als server en hier de byte-samples in te stoppen. Nu kun je met een terminal programma zoals putty (aangeraden! [Download hier.](#)), teraterm, coolterm oid. verbinding maken met localhost op poort 2000 (via telnet) waarna je jouw naam terugziet. Tip: Bouw het schema van buiten naar binnen op, kijk bijvoorbeeld eerst of je jouw naam ziet als je de [Vector Source](#) direct op de [TCP Sink](#) aansluit.

^a In de video wordt `()` gebruikt ipv `[]`, dit werkt niet onder de Windows versie

^b De slash-n is een newline wat werkt als een *enter*

Opdracht 2: ASK modulatie

[4 pt]

[Amplitude Shift Keying \(ASK\)](#) is gelijk aan AM met draaggolf, maar dan met digitale informatie. Maak gebruik van de kennis die je hebt opgedaan in de AM labs en [opdracht 1](#) om een ASK zender en ontvanger te bouwen waarbij je jouw naam kunt uitzenden en ontvangen.

Opdracht 3: Eigen apparaat aansturen

[8 pt]

Goedkope apparaten zoals afstandsbedieningen en deurbellen maken vaak gebruik van ASK modulatie. Deze kun je herkennen door de frequentie waarop ze zenden. Als je een apparaat hebt dat op 433 MHz zendt dan is de kans groot dat het een ASK signaal is. Probeer een apparaat in jouw omgeving te vinden dat op 433 MHz zendt en achterhaal de binaire code die het apparaat gebruikt. Bouw vervolgens een zender

die deze code kan uitzenden zodat je het apparaat kunt aansturen. Onderbouw de procedure die je hebt gevolgd om de code te achterhalen en hoe je deze code hebt omgezet naar een zender in de video.

Opdracht 4: PSK modulatie

[5 pt]

Phase Shift Keying (PSK) is gelijk aan AM zonder draaggolf^a, maar dan met digitale informatie. Dupliceer de schakelingen van **opdracht 2** en bedenk een manier om dit PSK te maken. Zend je eigen woord of naam en zorg ervoor dat je deze kunt ontvangen. Let op: PSK correct ontvangen komt erg nauw, en door drift van de klokken zul je regelmatig moeten corrigeren. In het laatste onderwerp zullen we synchronisatieblokken gebruiken die frequentie en faseverschuivingen kunnen corrigeren, maar voor nu zul je dit handmatig moeten doen. We zullen wel differentiële codering toepassen, zoals uitgelegd in PySDR, zodat we geen last hebben van een faseverschuiving van 180 graden. Hiervoor zul je de bits^b in een **Differential Encoder** moeten stoppen voordat je deze gaat moduleren, en aan de ontvanger kant een **Differential Decoder** moeten toepassen nadat je hebt gedemoduleerd.

^a Dus we kunnen geen top-detector meer gebruiken en komt het afstellen nauw

^b Dus de plek in je schema waarbij je samples 0 of 1 als waarde hebben

Opdracht 5: FM Ontvanger

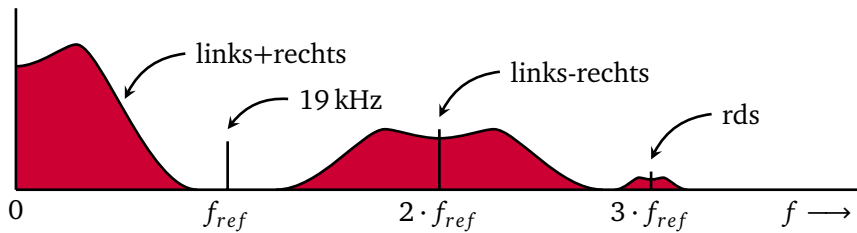
[4 pt]

In PySDR wordt uitgelegd hoe je een FM signaal kunt demoduleren, en welke signalen in het resulterende spectrum zitten. Bouw een FM ontvanger die een FM signaal kan ontvangen en demoduleren. Radio-538 (102,7 MHz) heeft het sterkste signaal in de omgeving van de school, dus probeer deze te ontvangen. Zodra je muziek hoort op de ontvanger, kun je door naar de volgende opdracht.

Opdracht 6: Radio Data System Stereo FM

[6 pt]

Het lab heet RDS; dit was het originele plan, maar dit wordt in het laatste onderwerp behandeld vanwege slecht ontvangst in het lokaal wat geavanceerdere technieken vereist.



Figuur 1: Inhoud FM kanaal na demodulatie

Wat we hiervoor in de plaats doen is het bouwen van een stereo FM ontvanger. In [figuur 1](#) is te zien dat er naast de mono band (L+R) ook een L-R band en een rds band in het FM signaal zitten. Bouw een schema dat door som/verschil van de L+R en L-R banden een stereo signaal kan maken. Je kunt de [Audio Sink](#) blok instellen op 2 inputs voor een stereo signaal. Weinig muziek heeft een duidelijk stereo effect, dus afhankelijk van het nummer wat op dat moment afgespeeld wordt is het effect al dan niet merkbaar.

Woordenlijst

ASK Amplitude Shift Keying. [2](#)

GRC GNU Radio Companion. [1](#)

PSK Phase Shift Keying. [3](#)