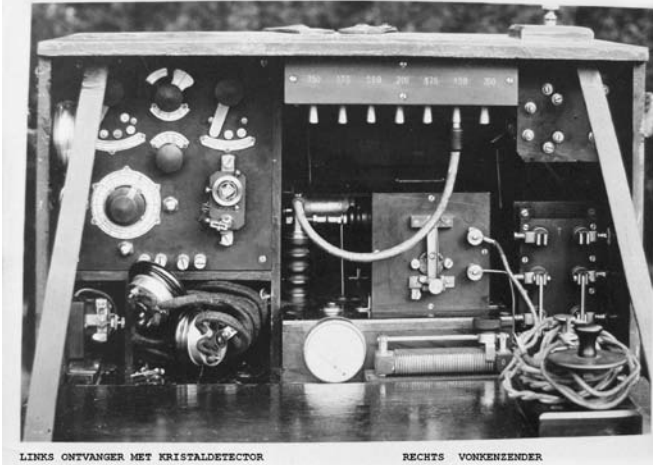
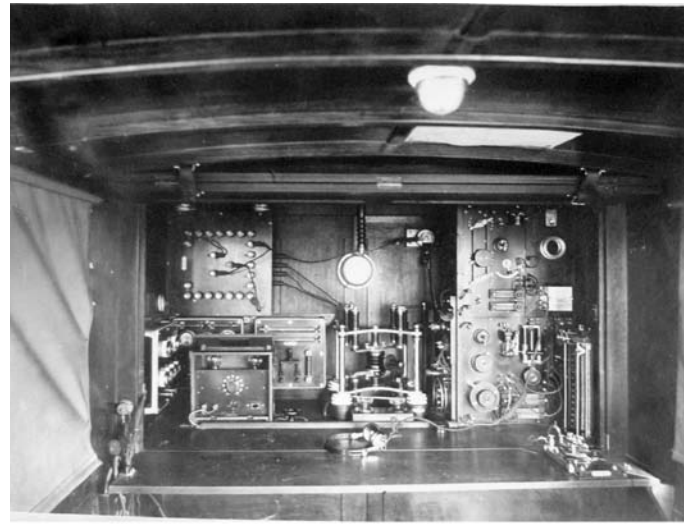


Appareil de radio, Hoogstade 1916
Poste type T.S.F - B, 1916



Een van de radiotoestellen die door MN 7 gebruikt werd, was het Franse toestel A3ter, uitgerust met 3 stuks TM-lampen. Dit toestel kon op tweeërlei manieren gebruikt worden: (1) als laagfrequentversterker om het signaal van de kristalontvanger te versterken en (2) als ontvanger mits toevoeging van een afstembare kring als detector plus tweemaal L.F. OV2. Het toestel had een gewicht van ca. 5,250 kg.



INTERIEUR 1,5 KW ZENDER MARCONI IN MOBIELE POST F

Bronvermeldingen

"MN 7 Het militaire radio luister- en zendstation in Baarle-Hertog 1915-1919" door Jacques Boone, uitgegeven door Heemkundekring "Amalia van Solms" in Baarle-Hertog-Nassau, hoofdredacteur Herman Janssen.

Caractéristiques d'un signal D-STAR Kenmerken van een D-STAR signaal

Par/door ON1ARF – Traduit par ON3ZL

Une des choses sur laquelle la communauté radioamateur à Ostende ne peut certainement pas se plaindre est le nombre de relais: relais analogiques FM sur toutes sortes de bandes, connectés ou non à l'internet, node-APRS, relais ATV et ... deux transpondeurs du relais D-STAR ON00S. Celui qui a déjà connecté son transceiver FM sur 439,500, ou sur 1298,100 MHz pour qui dispose d'un appareil 23cm, a là déjà entendu le son typique d'un relais D-STAR. Le son peut difficilement être décrit, mais une chose est sûre: avec un appareil FM normal on ne peut rien en faire. Ecoute-t-on cependant avec un appareil digital D-STAR (ou via le flux internet), alors entend-t-on miraculeusement une conversation intelligible.

Le but de cet article est d'expliquer comment cette communication vocale est stockée dans le "bruit blob" d'un signal D-SRTAR

Cet article est une traduction de l'anglais vers le néerlandais d'un article du blog sur mon site. L'article web comporte aussi un nombre d'exemples audio qui pour des raisons logiques ne peuvent pas être reprise dans un article écrit. L'article initial est consultable à l'adresse: villazebries.krbonne.net/hamstuff/?p=106.

Récepteur GMSK

La base de cet article est le 'récepteur GMSK', un programme qui peut être utilisé pour recevoir au moyen d'une radio FM normale le signal d'un relais D-STAR, le démoduler et en tirer les informations D-STAR.

Notez que ce récepteur GMSK ne peut recomposer la communication vocale initiale, vu que D-STAR utilise pour cela une technologie brevetée disponible uniquement en hardware (lisez: un processeur). On peut seulement decoder ledit flux vocal avec un soi-disant 'DV-dongle'. Le récepteur GMSK (et le correspondant émetteur) peut bien être utilisé par des applications D-STAR où aucune conversation vocale n'est utilisée, par exemple D-RATS (transfert de fichiers), DPRS (envoi et réception d'information APRS via D-STAR) ou l'envoi ou la réception de petit message via D-STAR.

Eén van de dingen waarover de radioamateurgemeenschap in Oostende zeker niet kan klagen, is het aantal repeaters: analoge FM-repeaters op allerlei banden, al dan niet verbonden met het internet, APRS-node, ATV-repeaters en... twee transponders van de D-STAR repeater ON00S.

Wie ooit zijn FM-transceiver afgestemd heeft op 439,500, of op 1298,100 MHz voor wie over een 23cm-toestel beschikt, heeft daar al het typische geluid gehoord van een D-STAR repeater. Het geluid kan moeilijk beschreven worden, maar één ding is zeker: met een gewoon FM-toestel is er totaal niets van te maken. Luistert men echter met een digitaal D-STAR toestel (of via de internetstream), dan komt daar wonder boven wonder toch een verstaanbaar gesprek uit.

De bedoeling van dit artikel is om uit te leggen hoe die gesproken stemcommunicatie in die 'blob ruis' van een D-STAR signaal opgeslagen wordt. Dit artikel is een vertaling van het Engels naar het Nederlands van een blogartikel op mijn webstek. Het webartikel bevat ook een aantal audio-samples die om logische redenen niet kunnen opgenomen worden in een geschreven artikel. Het oorspronkelijk artikel is terug te vinden op villazebries.krbonne.net/hamstuff/?p=106.

GMSK-ontvanger

De basis van dit artikel is 'GMSK-ontvanger', een computerprogramma dat kan gebruikt worden om door middel van een gewone FM-radio het signaal van een D-STAR repeater te ontvangen, te demoduleren en daar de D-STAR informatie uit te halen.

Merk op dat GMSK-ontvanger geen oorspronkelijk stemcommunicatie kan hersamenstellen, aangezien D-STAR daarvoor gebruik maakt van gepatenteerde technologie die enkel in hardware (lees: een chip) beschikbaar is. Men kan het gesproken stembericht enkel decoderen met een zgn. 'DV-dongle'. GMSK-ontvanger (en bijhorende GMSK-zender) kan wel gebruikt worden voor D-STAR toepassingen waar geen gesproken stem aan te pas komt, bijvoorbeeld D-RATS (overdracht van bestanden), DPRS (sturen en ontvangen van APRS-informatie via D-STAR) of het versturen en ontvangen van korte berichten over D-STAR.

C'est difficile à croire, mais concevoir un récepteur GMSK de nos jours n'est pas si difficile. Tout ce qui doit être fait est dans les quatre étapes suivantes:

- Capturer le son du relais D-STAR via une radio FM et la carte son d'un ordinateur;
- GMSK démodule ce signal, c'est à dire transforme ce son en un train de bits (zéro/un);
- Extrait le signal raw D-STAR de ce flux de bits;
- Travaille quelques parties de ce signal D-STAR: correction d'erreur, descrambling et deinterleaving.

Le résultat final: un fichier '.dvtool' contenant un flux D-STAR. That's it!

Sur l'importance de l'open source

Les mots 'de nos jours' ne sont pas ici sans importance. Une des plus grandes évolutions dans le monde D-STAR dans les deux années précédentes est la survenance de software open source. Ce sont des programmes qui non seulement peuvent être téléchargés et utilisés gratuitement, mais dont l'accès au code source est libre. Pour quelqu'un qui est intéressé de connaître la structure, l'open source est une source importante d'information. Plus encore, c'est la combinaison de ces deux sources d'information qui donne un bon aperçu de comment D-STAR fonctionne véritablement et comment le système D-STAR garantit que le signal vocal à partir d'une radio (ou d'un relais) arrive à bon port sur une autre radio D-STAR.

Les programmes les plus importants qui délivrent des informations pour cet article sont:

- Le projet "pcrepeatercontroller" de Jonathan G4KLX
- Le projet DV-RPTR de Jan DO1FJN, Torsten DG1HT et DH2YBE

Les deux forment la base du code-programme du software 'récepteur GMSK'.

La base des flux GMSK et D-STAR

Normalement, le premier pas de la recherche sur une nouvelle technologie se compose de la recherche d'information sur internet. Il semble cependant que pour D-STAR cela ne sera pas une simple affaire. L'information de base de D-STAR et GMSK est disséminée entre différents documents, écrit par différentes personnes et issue de diverses sources. Il est facile de perdre son chemin dans toute cette information, d'autant plus qu'une grande partie de cette information n'est pas nécessaire pour concevoir un récepteur GMSK.

La base de toute l'information sur D-STAR est un document (de seulement 11 pages): le document 'shogen'. C'est une traduction anglaise d'une documentation du JARL sur D-STAR. Ce document contient seulement des spécifications et est très difficile à comprendre par quelqu'un qui n'est pas expérimenté dans le monde des communications digitales.

Pour ces raisons cet article partira d'un autre point de vue: 'Comment se présente un signal D-STAR pour un programme qui doit le travailler?'. L'intention n'est pas d'expliquer D-STAR dans chaque détail. Certains composants seront proposés comme 'boîtes magiques'. L'intention est de donner une aperçu global de comment un signal D-STAR se présente. Celui qui souhaite plus d'information sur des composants spécifiques pourra trouver d'autres articles.

Dix choses que vous devez connaître sur D-STAR et GMSK

Quand vous cherchez sur internet des informations sur D-STAR, vous êtes rapidement submergé sous divers termes techniques et abréviations. Beaucoup de ces termes sont compréhensibles pour les experts, mais comme profane ils ne vous disent pas grand-chose. Il y a cependant un certain nombre de choses qui sont importantes à savoir sur D-STAR si vous voulez comprendre cet article. Ce sont les suivantes:

Het klinkt misschien vreemd, maar een GMSK-ontvanger maken is vandaag de dag helemaal niet zo moeilijk. Alles wat moet gebeuren zijn de volgende vier stappen:

- onderschep het geluid van de D-STAR repeater via een FM-radio en de geluidskaart van de computer;
- GMSK-demoduleer dit signaal, d.w.z. zet dit geluid om in een stroom van bits (nulletjes/eentjes);
- puur het ruwe D-STAR signaal uit deze ongeordende stroom van bitjes;
- bewerk bepaalde delen van het D-STAR signaal: foutcorrectie, descrambling en deinterleaving.

Het eindresultaat: een '.dvtool' bestand met daarin een D-STAR bestand. That's it!

Over het belang van open source

De woorden 'vandaag de dag' hiervoor zijn niet zonder belang. Eén van de belangrijkste evoluties in de D-STAR wereld in de voorbije twee jaren is de opkomst van open source (open bron) software. Dit zijn programma's die niet enkel gratis gebruikt en gedownload mogen worden, maar waarvan bovendien de broncode vrij toegankelijk is. Voor iemand die geïnteresseerd is om te weten hoe de structuur van een D-STAR signaal is, is open source een belangrijke bron van informatie. Meer nog, het is de combinatie van deze twee informatiebronnen die een goed overzicht geeft hoe D-STAR eigenlijk werkt en hoe het D-STAR systeem ervoor zorgt dat het stemsignaal vanuit een radio (of repeater) op een andere D-STAR radio terecht komt.

De belangrijkste softwarepakketten die informatie leverden voor dit artikel zijn:

- Het "pcrepeatercontroller" project van Jonathan G4KLX
- Het DV-RPTR project van Jan DO1FJN, Torsten DG1HT and DH2YBE

Beiden vormen de basis van de programmacode van de GMSK-ontvanger software.

De basis van GMSK en D-STAR signalen

Normaal is de eerste stap bij het onderzoeken van een nieuwe technologie het opzoeken van de informatie die daarover beschikbaar is op het internet. Raar genoeg blijkt dit voor D-STAR geen simpele zaak te zijn. De basisinformatie van D-STAR en GMSK is verspreid over verschillende documenten, geschreven door verschillende personen en afkomstig van verschillende bronnen. Het is gemakkelijk om verloren te lopen in al deze informatie, vooral doordat het grootste gedeelte van deze informatie niet nodig is om een GMSK-ontvanger te maken.

Dé basis voor alle informatie over D-STAR is één document (van slechts 11 bladzijden): het 'shogen' document. Dit is een Engelse vertaling van de documentatie van de JARL over D-STAR. Dit document bevat echter alleen droge specificaties en is heel moeilijk te begrijpen door iemand die niet thuis is in de wereld van digitale telecommunicatie.

Om deze reden zal dit artikel vertrekken vanuit een ander standpunt, namelijk "hoe ziet een GMSK D-STAR signaal er uit voor een programma dat het moet verwerken?".

Het is niet de bedoeling om D-STAR in elk detail uit te leggen. Bepaalde onderdelen zullen als 'magische dozen' voorgesteld worden. De bedoeling is om een globaal overzicht te geven hoe een D-STAR signaal er uitziet. Wie meer informatie wenst over specifieke onderdelen kan meer daarover vinden in andere artikels.

Tien dingen die je moet weten over D-STAR en GMSK

Wanneer je op het internet informatie opzoekt over D-STAR, word je snel bedolven onder allerlei technische termen en afkortingen. Veel van deze termen zijn bedoeld voor experts, maar als leek heb je er niet veel aan. Toch zijn er een aantal zaken over D-STAR die belangrijk zijn om weten als je dit artikel wilt begrijpen. Hier zijn ze:

- Digital:** D-STAR est un système de communication digital. Cela veut dire que la parole **n'est pas** transportée par une modification d'un signal analogique (comme avec FM, AM of SSB), mais comme un train de zéro et de uns successifs (bits). Ce 'train de bit' sera plus loin dans cet article indiqué comme le 'flux' GMSK de D-STAR.
- GMSK:** bien que le signal D-STAR comporte un signal composé de 0 et de 1, nous utilisons cependant pour l'émission un appareil radio analogique. Pour cela les bits sont convertis en tonalités qui peuvent être transportées par une radio analogique. Il existe plusieurs systèmes de modulation semblable: FSK, PSK, ASK et autres. D-STAR utilise un système nommé 'GMSK', une variante du FSK qui est par exemple utilisé par la RTTY.
- FM-modulé:** comme chaque radioamateur le sait, on peut émettre une signal analogique via différentes méthodes: AM, SSB, FM. Comme certains autres système digitaux (par ex. packet radio et APRS), GMSK utilise la modulation FM. Ainsi peut-on envoyer des signaux GMSK D-STAR via une radio FM normale.
- 9k6 port data:** une des choses dans lequel un modem GMSK diffère d'une communication FM normale et du packet-radio et de l'APRS, est l'utilisation du port data '9k6' d'une radio FM au lieu du port normal 'micro/haut-parleur'. La différence entre ces deux port est que le port 9k6 n'utilise pas la commutation préemphasis/déemphasis du modulateur FM, autrement cette commutation déforme les 'tonalités' du signal GMSK et par là aussi les bits qui sont stockés dans ces tonalités.
- 4800:** il s'agit du nombre de bits par seconde envoyé par une radio D-STAR. Ces bits peuvent être divisés en trois groupes: 2400 bits/seconde: parole, 1200 bits/seconde: correction d'erreur (pour corriger d'éventuelles erreurs dans la communication) et 1200 bits/seconde: 'slow data' (par exemple le nom et QTH de l'opérateur).
- Structurée:** bien qu'on ne le dirait pas à l'écoute d'un signal GMSK, les bits dans un signal GMSK possède bien une structure fixe et pré-déterminée. En résumé il y a un 'header' de 328 bits qui contient des informations comme le callsign, etc., suivi par un nombre indéterminé de 'frames' de 12 bytes (96 bits) qui contiennent chacune 20 millisecondes de parole et 'slow data'.
- Synchronisation:** à côté de la structure sus-mentionnée un signal D-STAR GMSK contient encore aussi des bits de synchronisation. Ceux-ci servent à synchroniser le récepteur avec l'émetteur, ce qui est nécessaire pour que, dans le récepteur, le signal D-STAR soit bien décomposé. Ces séquences de synchronisation se place au début du signal D-STAR et ensuite après chaque 21 frames (c.à.d. chaque 420 millisecondes).
- 20 ms:** un QSO peut durer plusieurs minutes. Ceci ne peut être traité en un morceau par un système digital. C'est pourquoi l'ensemble est divisé en paquets d'une durée de 20 millisecondes. De cette manière la voix peut être travaillée par un 'vocoder' (voyez plus loin) et converti en 9 bytes (6 bytes parole et 3 bytes correction d'erreur). Ajouter alors à cela 3 bytes 'slow data' et vous arrivez à 12 bytes (96 bits) par 20 milliseconde, ou encore 4800 bits par seconde ($20 \text{ ms} = 1/50 \text{ d'une seconde}; 96 \text{ bits} * 50 = 4800 \text{ bits / seconde}$).
- AMBE:** 20 millisecondes de paroles non traitée prend 160 bytes. D-STAR réserve seulement 6 bytes par 20 ms. Afin de quand même insérer la parole dans cet espace limité, on utilise une 'codec' (codeur/décodeur) ou 'vocoder' (contraction de 'voice coder'). Le vocoder utilisé par D-STAR se nomme 'AMBE'.
- DVdongle:** le fonctionnement exact du AMBE-vocoder est muni d'un copyright et non public. Les AMBE-vocoders peuvent seulement être acheté sous forme d'un IC (un circuit intégré). Chaque radio D-STAR est munie de cet IC. Pour pouvoir encoder ou décode AMBE sur une ordinateur on peut utiliser un 'Dvdongle', un USB-dongle avec un circuit intégré comprenant le codeur/décodeur AMBE.
- Digitaal:** D-STAR is een digitaal communicatiesysteem. Dit betekent dat spraak **niet** wordt overgebracht door een verandering van een analoog signaal (zoals bij FM, AM of SSB), maar als een trein van opeenvolgende nullen en enen (bits). Die 'bittrein' zal verder in dit artikel aangeduid worden als de D-STAR of GMSK 'stream'.
- GMSK:** ook al bestaat een D-STAR signaal uit 0-en en 1-en, toch maken we voor het uitzenden ervan gebruik van analoge radioapparatuur. Om dit te kunnen doen worden de bits omgezet in toontjes die kunnen overgedragen worden door een analoge radio. Er bestaan meerdere zulke modulatiesystemen: FSK, PSK, ASK en andere. D-STAR gebruikt een systeem genaamd 'GMSK', een variant van FSK dat bijvoorbeeld gebruikt wordt bij RTTY.
- FM-gemoduleerd:** zoals elke radioamateur hoort te weten, kan men een analoog signaal uitzenden via verschillende methodes: AM, SSB, FM. Net zoals bepaalde andere digitale systemen (bvb. packet radio en APRS) gebruikt GMSK FM-modulatie. Daardoor kan men GMSK D-STAR signalen uitzenden via een gewone FM-radio.
- 9k6 datapoort:** één van de zaken waarin een GMSK-modem verschilt van normale FM-communicatie en van packet-radio en APRS, is dat het gebruik maakt van de '9k6'-datapoort van een FM-radio i.p.v. de normale 'microfoon/luidspreker' poorten. Het verschil tussen deze twee poorten is dat de 9k6 poort de preëmphasis/deëmphasis-schakeling van de FM-modulator niet gebruikt, zoniet zou deze schakeling de 'toontjes' van het GMSK-signaal vervormen en daardoor ook de bits die opgeslagen liggen in die toontjes.
- 4800:** dit is het aantal bits dat per seconde uitgestuurd wordt door een D-STAR radio. Deze bits kunnen ingedeeld worden in drie groepen: 2400 bits/seconde spraakinformatie, 1200 bits/second foutcorrectie-informatie (om eventuele fouten in de communicatie te corrigeren) en 1200 bits/seconde 'slow data' (bijvoorbeeld naam en QTH van de operator).
- Gestructureerd:** alhoewel je het niet zou zeggen als je een GMSK beluistert, hebben de bits in een GMSK-signaal wel degelijk een vaste, voorafbepaalde structuur. In het kort uitgelegd bestaat die uit een 'header' van 328 bits die informatie bevat zoals callsign, enz., gevolgd door een onbepaald aantal 'frames' van 12 bytes (96 bits) die elk 20 milliseconden spraakinformatie en slow data bevatten.
- Synchronisatie:** naast de hierbovenvermelde structuur bevat een D-STAR GMSK signaal ook nog synchronisatiebits. Deze dienen om de ontvanger in de pas te doen lopen met de zender, hetgeen nodig is om in de ontvanger het ontlede van het D-STAR signaal goed te doen verlopen. Deze synchronisatiesequenties bevinden zich in het begin van het D-STAR signaal en na elke 21 frames (d.w.z. elke 420 milliseconden).
- 20 ms:** een QSO kan meerdere minuten duren. Dit kan echter niet in één stuk door een digitaal systeem verwerkt worden. Daarom wordt het opgedeeld in pakketjes die elk 20 milliseconden lang zijn. Op deze manier kan de stem verwerkt worden door de 'vocoder' (zie verder) en omgezet worden in 9 bytes (6 bytes spraakdata en 3 bytes foutcorrectie-informatie). Voeg daarbij dan 3 bytes 'slow data' en je komt aan 12 bytes (96 bits) per 20 milliseconde, ofwel 4800 bits per seconde ($20 \text{ ms} = 1/50 \text{ van een seconde}; 96 \text{ bits} * 50 = 4800 \text{ bits / seconde}$).
- AMBE:** 20 milliseconden onbewerkte spraakinformatie neemt 160 bytes in beslag. D-STAR biedt slechts plaats voor 6 bytes per 20 ms. Om toch spraakinformatie in deze beperkte ruimte onder te brengen, maakt men gebruik van een 'codec' (coder/decoder) of 'vocoder' (kort voor 'voice coder'). De vocoder gebruikt door D-STAR heet 'AMBE'.
- DVdongle:** de exacte werking van de AMBE-vocoder is gepatenteerd en niet publiek. AMBE-vocoders kan je enkel aankopen in de vorm van een IC (een chip). Elke D-STAR radio beschikt over zo'n IC. Om AMBE te kunnen encoder of decoderen op een computer kan men gebruik maken van een 'Dvdongle', een USB-dongle met een ingebouwde AMBE encoder/decoder IC.

Hardware setup

Le software est une chose, mais finalement on a également besoin de matériel pour faire un modem GMSK. Dans cet exemple le setup est plutôt limité (voyez la photo).

Les composants matériels suivants ont été utilisés:

- Un transceiver FM avec un port data 9k6 (dans ce cas un Yaesu FT857D)
- Un ordinateur: un PC (desktop/laptop), ou une ‘development board’. Dans le setup sur la Photo on a utilisé une ‘pandaboard’, mais d’autres boards moins puissantes (comme la friendlyarm mini2440, visible sur la photo) sont disponibles.
- Une façon de lire l’audio du transceiver avec l’ordinateur. Cela peut être le port audio de l’ordinateur/development board ou un USB audio-dongle.
- Un câble pour faire la liaison entre le transceiver et l’ordinateur. Bien qu’absent dans la préparation utilisée dans ce test, il est en pratique nécessaire de placer un condensateur de (par ex.) 10 μ F en série dans le circuit pour arrêter la composante DC du signal audio. Dans la configuration de test, la liaison entre le transceiver et le modem a été réalisée avec un câble non blindé. Dans un setup opérationnel cela ne se peut parce qu’il n’y a aucune protection contre diverses émissions RF. En plus, dans la préparation de test aucun condensateur n’a été placé, de telle sorte que le signal sera perturbé par la composante DC du signal audio. Pour un montage expérimental cela peut encore être négligé.

Enregistrement et analyse d’un signal audio

Comme déjà mentionné ci-avant, cet article est écrit du point de vue d’un logiciel qui doit décoder un signal GMSK D-STAR: qu’entend un ordinateur sur le port audio 9k6 d’une transceiver? La reproduction d’un enregistrement dans un article imprimé n’est naturellement pas possible. Les enregistrements sont disponibles à l’écoute via mon blog.

Audio sample 1

Écoutez exemple 1 sur

villazeebries.krbonne.net/files/gmsk/audio_1.wav (6 s).

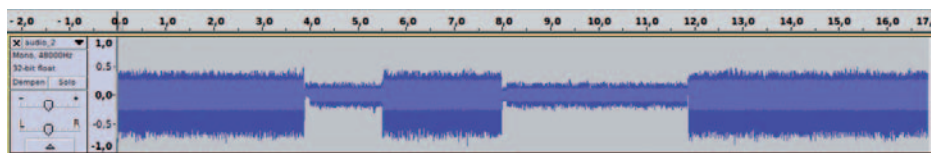
Qu’entendons-nous? Du bruit. C’est le premier fait intéressant. Le port 9k6 du récepteur prend le signal du démodulateur FM avant le discriminateur FM. Le squelch audio d’un récepteur FM se trouve tout à fait derrière dans la commutation. Par conséquent, il est vrai que, au moment où aucun signal n’est présent, sur le port 9k6 aucun silence n’est à attendre comme on pourrait l’imaginer, mais plutôt un fort signal de bruit.

Audio sample 2

Exemple suivant: villazeebries.krbonne.net/files/gmsk/audio_2.wav (16 s).

Dans ce fragment audio suivant on entend deux émissions GMSK, entourées par des moments de bruit.

“A picture says more than a thousand words”, voyez l’impression visuelle de ce fragment audio, réalisé avec l’éditeur audio Audacity.



Hardware setup



Software is één zaak, maar uiteindelijk heeft men ook hardware nodig om een GMSK-modem te maken. In dit voorbeeld is de set-up vrij beperkt (zie de foto).

De volgende hardwarecomponenten worden gebruikt:

- een FM-transceiver met 9k6-dataport (in dit geval een Yaesu FT857D)
- een computer: een PC (desktop/laptop), of een ‘development board’. In de setup op de foto werd gebruik gemaakt van een ‘pandaboard’, maar ook minder krachtige boards (zoals de friendlyarm mini2440, ook te zien op de foto) zijn bruikbaar.
- een manier om het audio van de transceiver in de computer in te lezen. Dit kan zowel de audioport van computer/development board zelf zijn, of een externe USB audio-dongle.
- een kabel om de verbinding te maken tussen de transceiver en de computer. Alhoewel afwezig in de opstelling gebruikt in deze test, is het in praktijk wel nodig om een condensator van (bvb.) 10 μ F serie in het circuit aan te brengen om zo de DC-component van het audiosignaal te stoppen. In de testopstelling is de verbinding tussen transceiver en modem gemaakt met een onafgeschermde kabel. In een operationele setup volstaat dit niet omdat er geen afscherming is tegen allerlei RF-storingen. Bovendien is er in de testopstelling geen condensator aanwezig, waardoor het signaal enigszins gaat zweven door de DC-component van het audiosignaal. Voor een experimentele setup kan dit nog door de vingers worden gezien.

Opnemen en analyseren van een audiosignaal

Zoals hierboven reeds aangegeven, is dit artikel geschreven vanuit het standpunt van een computerprogramma dat een GMSK D-STAR signaal moet decoderen: wat hoort een computer op de 9k6 audioport van een transceiver? Een geluidsopname afspelen in een afgedrukt artikel gaat natuurlijk niet. De opnames zijn wel te beluisteren via mijn blog.

Audio sample 1

Beluister sample 1 op villazeebries.krbonne.net/files/gmsk/audio_1.wav (6 s).

Wat horen we? Ruis. Dit is een eerste interessant feit. De 9k6 dataport van de ontvanger takt het signaal af van de FM-ontvangerschakeling voor de FM-discriminator. De audiosquelch van een FM-ontvanger bevindt zich helemaal achteraan in de schakeling. Daarom is het zo dat, op het moment dat er geen signaal binnenkomt, op de 9k6 poort er geen stilte te horen is zoals men zou verwachten, maar een sterk ruissignaal.

Audio sample 2

Volgende sample: villazeebries.krbonne.net/files/gmsk/audio_2.wav (16 s).

In dit geluidsfragment hoort men twee GMSK-uitzendingen, omgeven door momenten van ruis.

“A picture says more than a thousand words”, zie de visuele afdrruk van dit audiofragment, aangemaakt met de audio-editor Audacity.

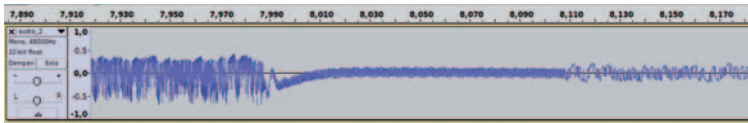
Encore une fois, nous pouvons observer ce qui suit:

- deus émissions GMSK;
- du bruit au moment où ont ne capte rien;

Intéressant est le niveau audio du signal reçu. Au moment où il n'y a pas d'émissions, le niveau du son reçu est **plus haut** qu'au moment où un signal est présent. Ce fait est utilisé par le programme de réception GMSK pour déterminer quand une émission débute ou s'arrête.

Un close-up du début du signal GMSK

Dans l'image suivante vous voyez un close-up de l'émission GMSK dans le fragment audio. Comme vous pouvez le voir sur l'échelle de temps, c'est le début de la deuxième émission GMSK.



On peut clairement distinguer trois zones différentes:

1. Jusqu'au marqueur 7,985 il n'y a pas de signal, seulement du bruit
2. Entre les marqueurs 7,99 et 8,105 nous voyons une séquence de rappel
3. A partir du marqueur 8,105 la séquence varie

L'explication de ceci peut être retrouvé dans la sixième et septième règle de base du GMSK, décrites ci-dessus. Un signal D-STAR n'est pas simplement une séquence aléatoire de bits. Il y a un canevas fixe. Plus précisément se trouve au début d'une émission GMSK la 'séquence de synchronisation'. Ce n'est pas seulement une séquence fixe mais aussi répétitive.

Un autre élément intéressant dans le fragment audio est le petit 'hick up' au début de la séquence de synchronisation. Il est la conséquence des condensateurs dans le circuit de réception en parallèle avec l'impédance de l'entrée. Si vous placez des extra condensateurs en série avec le circuit d'interface (e.a. pour bloquer la composante DC), alors ce hickup deviendra un signal sinusoïdal étouffé. Ceci montre dans tout les cas pourquoi il y a un signal de synchronisation et qu'il doit avoir une longueur minimale spécifique: le récepteur a besoin d'un certain temps pour se synchroniser avec le signal reçu.

Démodulation du début d'une émission GMSK

Les règles du journalisme d'investigation commande qu'il doit y avoir au moins deux sources indépendantes pour faire un récit crédible. Donc nous regardons cela aussi d'un autre œil. Que voyons-nous en fait quand nous démodulons ce signal, c.à d. convertir le signal audio en une série de 0 et 1? Attendu que le récepteur GMSK a une option '-dd' ('dump more') avons-nous la possibilité de voir ce que le software fait de cela.

La procédure pour traiter cette série de bits est la suivante:

1. Un composant du software mesure la valeur de l'extrait audio (à l'entrée du dongle-audio). Ceci 48000 fois par seconde.
2. Cette valeur est envoyée vers le démodulateur GMSK. C'est une boîte magique et elle renvoie une valeur 0 ou 1.
3. Comme indiqué ci-dessus le signal audio est mesuré 48.000 fois par seconde. Mais le signal D-STAR varie seulement 4.800 fois par seconde (une facteur 10 en moins donc). Encore une autre morceau de 'magie' met ce signal à 48000 bits/seconde à la bonne vitesse.

Retournons à la séquence audio. Ci-dessous vous trouvez le résultat de la démodulation GMSK du premier fragment, le bruit:

```
00111000 10011100 00000010 00001000 00110111 11011111
00000000 10000001 00000000 00101101 10011111 10111111
11000001 11010010 01111010 01100001 00101000 00110100
00000000 01110100 10011010 00110000 00011000 10011011
00011101 10000111 10000111 11100111 10100011 11110101
11100000 11110000 11100110 11000001 01111101 10100001
(...)
```

Opnieuw kunnen we de volgende zaken waarnemen:

- twee GMSK uitzendingen;
- ruis op het moment dat er niets ontvangen wordt;

Interessant is het audioniveau van het ontvangen signaal. Op het moment dat er geen uitzendingen zijn is het niveau van het ontvangen geluid **hoger** dan op het moment dat er wel signaal aanwezig is. Dit feit wordt door de GMSK-ontvangerssoftware gebruikt om te bepalen wanneer een uitzending begint of stopt.

Een close-up van het begin van het GMSK signaal

In de volgende afbeelding zie je een close-up van de GMSK-uitzending in het geluidsfragment. Zoals je ook kan zien op de tijdmarkeringen, is dit het begin van de 2de GMSK-uitzending.

Men kan duidelijk drie verschillende zones onderscheiden:

1. tot tijdsmarkering 7,985 is er geen signaal, enkel ruis
2. tussen markeringen 7,99 en 8,105 zien we een herhalingspatroon
3. vanaf markering 8,105 wordt het patroon gevarieerder

De uitleg hiervoor kan men terugvinden in de zesde en zevende GMSK-basisregels, hiervoor beschreven. Een D-STAR signaal is niet zomaar een willekeurig patroon van bits. Er zit een vast stramien in. Meer bepaald bevindt zich in het begin van een GMSK-uitzending het 'synchronisatiepatroon'. Dit is niet enkel een vast, maar ook een zich herhalend patroon.

Een ander interessant element in het geluidsfragment is de kleine 'hick up' in het begin van de synchronisatiepatroon. Deze is het gevolg van de condensatoren in het ontvangercircuit, parallel met deingangsimpedantieweerstand. Indien je extra condensatoren bijvoegt in serie van het interfacecircuit (o.a. voor het blokkeren van de DC-component), dan wordt deze hickup zelfs een klein uitdempend sinussignaal. Dit toont in elk geval aan waarom het synchronisatiesignaal er is en een bepaalde minimumlengte moet hebben: de ontvanger heeft een bepaalde tijd nodig om zich te synchroniseren met het ontvangen signaal.

Demodulatie van het begin van een GMSK-uitzending

De regels van de onderzoeksjournalistiek gebieden dat er minstens twee onafhankelijke bronnen moet en bestaan om een verhaal geloofwaardig te maken. Dus bekijken we dit ook eens vanuit een andere hoek. Wat zien we eigenlijk als we dit signaal demoduleren, d.w.z. het geluidssignaal omzetten naar een reeks van 0- en 1-en? Vermits GMSK-receiver een optie '-dd' heeft ('dump more') hebben we de mogelijkheid om te zien wat de software hiervan maakt.

Het proces om deze bitreeks aan te maken werkt als volgt:

1. Een onderdeel van de software meet de waarde van het audiosample (aan de ingang van de audio-dongle). Dit gebeurt 48000 keer per seconde.
2. Deze waarde wordt naar de GMSK-demodulator gestuurd. Dit is een magische doos en keert terug met een waarde 0 of 1.
3. Zoals hierboven vermeld wordt het geluidssignaal 48.000 keer per seconde gemeten. Maar het D-STAR signaal varieert maar 4.800 keer per seconde (een factor 10 minder dus). Nog een ander stuk 'magie' zet dit 48000 bits/seconde signaal om naar de juiste snelheid.

Terug naar het geluidsfragment. Hieronder vind je het resultaat van de GMSK-demodulatie van het eerste onderdeel, de ruis:

```
00111000 10011100 00000010 00001000 00110111 11011111
00000000 10000001 00000000 00101101 10011111 10111111
11000001 11010010 01111010 01100001 00101000 00110100
00000000 01110100 10011010 00110000 00011000 10011011
00011101 10000111 10000111 11100111 10100011 11110101
11100000 11110000 11100110 11000001 01111101 10100001
(...)
```

La répartition en groupes de 8 bits sert seulement à la lisibilité et n'a pas d'autre signification.

Conclusion: étant donné que le bruit n'est pas autre chose qu'une suite de niveaux de tension arbitraires, on reçoit de même après démodulation rien d'autre qu'une ligne de zéros et uns aléatoires. Rien de mal. Mais une peu plus loin dans l'enregistrement audio cela devient intéressant:

```
(...)  
10000110 00010000 00000001 00010111 11010110 10010100  
00111111 10100111 00100011 00111101 00111100 10111111  
11101000 10000001 11111101 11101010 11101101 10111111  
00010000 11101011 11110000 00000000 11111110 10000000  
00001000 11000001 01111100 11111000 10111101 11111111  
11111100 00011111 11111111 11111111 11111111 11111111  
11111111 11111111 11111111 11111111 11111111 11111111  
11111111 11011101 01111101 01010101 01010101 01010101  
01010101 01010101 01010101 01010101 01010101 01010101  
01010101 01010101 01010101 01010101 01010101 01010101  
01010101 01010101 01010101 01010101 01010101 01010101  
01010101 01010101 01010101 01010101 01010101 01010101  
01010101 01010101 01010101 01010101 01010101 01010101  
01010101 01010101 01010101 01010101 01010101 01010101  
01010101 01010101 01010101 01010101 01010101 01010101  
01010101 01010101 01010101 01010101 01010101 010  
...11101 10010100 00
```

Une tendance commence à émerger. Premier seulement "1" et alors une répétition fixe de "1010". Regardons-en exactement la signification dans les spécifications. Voir page 3 du document "shogen":

2.1 Wireless Communication Packet

2.1.1 Frame structure of a packet: The explanation of the data frame structure the Radio Header follows.

(1) Bit Syn. (Bit synchronization): Repeated standard 64-bit synchronization pattern (for GMSK 1010). (...)

Traduction:

2.1.1 Structure de la séquence d'un paquet. L'explication de la structure de la séquence data suit:

(1) Bit de synchronisation: répétition standard 64-bit séquence de synchronisation. (pour GMSK: 1010)...

Ce qui correspond exactement avec ce que nous avons observé: le début d'un signal D-STAR. Premièrement le récepteur a besoin d'un certain temps pour pouvoir se synchroniser avec le signal entrant, mais après 20 ms apparaît la séquence fixe 1010. Cela ne correspond pas aux spécifications, mais est en accord avec le fragment audio. Une ligne horizontale dans le bitdump revient avec 20 millisecondes. Le processus de synchronisation dans ce fragment prend 120 millisecondes, ce qui correspond parfaitement avec les 6 lignes dans le bitdump. Le hiccup même au début du signal est parfaitement visible. La série de 1-bits successif est en fait une démodulation du signal hiccup. Etant donné que toutes les valeurs de cette séquence audio se situe au-dessus de zéro, Elles deviennent après démodulation toutes 1.

Une preuve supplémentaire est donc trouvée.

Regardons encore une fois les spécifications:

2.1 Wireless Communication Packet

2.1.1 Frame structure of a packet: The explanation of the data frame structure the Radio Header follows.

(...)

(2) Frame Syn. (Frame synchronization): 15bit pattern (111011001010000).

(...)

Traduit:

(2) Séquence de synchronisation: 15-bit (111011001010000)

De verdeling in groepjes van 8 bits dient enkel de leesbaarheid en is verder zonder betekenis.

Conclusie: aangezien ruis niets anders is dan een opeenvolging van willekeurige spanningsniveaus krijgt men na demodulatie eveneens niets anders dan een rij willekeurige nullen en enen. Niets aan de hand. Maar een beetje verder in het geluidsopname wordt het interessanter:

```
(...)  
10000110 00010000 00000001 00010111 11010110 10010100  
00111111 10100111 00100011 00111101 00111100 10111111  
11101000 10000001 11111101 11101010 11101101 10111111  
00010000 11101011 11110000 00000000 11111110 10000000  
00001000 11000001 01111100 11111000 10111101 11111111  
11111100 00011111 11111111 11111111 11111111 11111111  
11111111 11111111 11111111 11111111 11111111 11111111  
11111111 11011101 01111101 01010101 01010101 01010101  
01010101 01010101 01010101 01010101 01010101 01010101  
01010101 01010101 01010101 01010101 01010101 01010101  
01010101 01010101 01010101 01010101 01010101 01010101  
01010101 01010101 01010101 01010101 01010101 01010101  
01010101 01010101 01010101 01010101 01010101 01010101  
01010101 01010101 01010101 01010101 01010101 01010101  
01010101 01010101 01010101 01010101 01010101 010  
...11101 10010100 00
```

Er begint zich een vast patroon af te tekenen. Eerst allemaal "1" en dan een vaste herhaling van "1010". Laten we eens in de specificaties kijken wat dit precies betekent. Zie pagina 3 van het "shogen" document:

2.1 Wireless Communication Packet

2.1.1 Frame structure of a packet: The explanation of the data frame structure the Radio Header follows.

(1) Bit Syn. (Bit synchronization): Repeated standard 64-bit synchronization pattern (for GMSK 1010). (...)

Vertaald:

2.1.1 Frame structuur van een pakket. De dataframe-opbouw van de radio header is als volgt:

(1) Bitsynchronisatie: herhaald standard 64-bit synchronisatiepatroon. (voor GMSK: 1010)...

Dit stemt exact overeen met wat we waarnemen: het begin van een D-STAR signaal. Eerst heeft de ontvanger enige tijd nodig om zichzelf te kunnen synchroniseren met het inkomende signaal, maar na 20 ms verschijnt het vast 1010-patroon. Dit klopt niet enkel met de specificatie, maar komt ook goed overeen met het geluidsfragment. En horizontale lijn in de bitdump komt overeen met 20 milliseconden. Het synchronisatieproces in dit fragment neemt 120 milliseconden in beslag, wat mooi overeenstemt met de 6 lijnen in de bitdump. Zelfs de hiccup bij de start van het signaal is mooi zichtbaar. De reeks opeenvolgende 1-bits is eigenlijk een demodulatie van het hiccup-signaal. Aangezien alle waarden van deze audiosamples zich boven nul bevinden, worden zij na demodulatie allemaal 1.

Bijkomend bewijs dus gevonden.

Laten we nog eens de specificaties verder bekijken:

2.1 Wireless Communication Packet

2.1.1 Frame structure of a packet: The explanation of the data frame structure the Radio Header follows.

(...)

(2) Frame Syn. (Frame synchronization): 15bit pattern (111011001010000).

(...)

Vertaald:

(2) Framesyncronisatie: 15-bitpatroon (111011001010000)

En effet, les 15 bits suivant dans le bitdump correspondent exactement avec les spécifications. Cette séquence de synchronisation n'est rien d'autre qu'une marque qui dit "ici s'arrête la synchronisation et suit la partie suivante du flux D-STAR: le header D-STAR".

Le header D-STAR

Comme mentionné précédemment, le header D-STAR est la partie qui suit la séquence de synchronisation. Regardons une peu ce que le programme de démodulation GMSK en dit.

Ceci est l'information stockée dans le header de ce fragment GMSK spécifique (suivant le programme du récepteur GMSK):

```
HEADER DUMP:
FLAG1: 00 - FLAG2: 00 - FLAG3: 00
RPT 2: DIRECT
RPT 1: DIRECT
YOUR: CQCQCQ
MY: ON1ARF /KRIS
Check sum = E441 (OK)
```

Comme nous pouvons le voir le header D-STAR contient la même information que ce que nous devons introduire sur la radio D-STAR radio, avec, en plus, 3 bytes 'drapeau' et un checksum (2 bytes). De ceci on peut conclure que le header D-STAR contient 328 bits (41 bytes x 8 bits), mais ce n'est pas le cas. Le header fait au total 660 bits. Pour comprendre cela, nous pouvons nous plonger dans le code source du GMSK-decoder. Ci-après suit une vue simplifiée du morceau de programme qui travaille les 660 bits après la séquence de synchronisation:

```
scramble(audioheaderbuffer_in,audioheaderbuffer_temp1);
deinterleave(audioheaderbuffer_temp1,audioheaderbuffer_temp2);
length=FECdecoder(audioheaderbuffer_temp3,audioheaderbuffer_out);
```

Les bits reçus du header D-STAR sont travaillés par 3 fonctions successives: scrambling, deinterleaving et FEC-decoding. Les raisons de ceci peuvent être mieux comprises si nous observons le processus pour fabriquer une signal D-STAR. Ce qui comprend les mêmes fonctions que le décodeur, mais dans l'ordre inverse: premièrement l'encodage FEC, ensuite l'interleaving et enfin le scrambling. La raison d'être de ces trois processus est de rendre le header plus robuste contre les fautes durant la transmission du signal.

Etape 0: insérer deux bits

La toute première étape dans ce processus semble un peu étrange: pour le traitement du header il est inséré deux 2 bits à la fin du header. Ces deux bit ne contiennent aucune information (ils sont tous les deux nuls) mais ils sont nécessaires pour l'étape suivante (FEC). Dans tous les cas ils augmentent la longueur du header de 328 à 330 bits.

Etape 1: FEC

C'est la première étape dans le traitement du header D-STAR, le 'FEC encodage'. FEC signifie 'Forward Error Correction' et sert – comme son nom l'indique – à corriger les erreurs dans les signaux radios digitaux. Cela se passe avant d'ajouter des extras bits au signal, par lesquels le système peut utiliser cette information pour corriger d'éventuelles erreurs.

Le système de correction d'erreur FEC forme une assez grande matière complexe et tombe hors de la portée de cet article. Bien important à savoir est qu'à l'application de la FEC, la longueur du header D-STAR double: le header initial van 330 bits contient maintenant 660 bits.

Etape 2: interleaving

L'étape suivante se nomme 'interleaving'. Quel beau mot pour 'bits jetés pele-mêle'. Ce que fait l'interleaving c'est de modifier l'ordre des bits. Si un header devrait contenir 4 bits - bits 1, 2, 3 en 4 - alors l'interleaving modifierait cela en 1, 3, 2, 4. Cela a à voir avec le système de correction d'erreur FEC. A côté du fait qu'un système FEC rend les bits dans le

Inderdaad, de volgende 15 bits in de bitdump stemmen exact overeen met het patroon in de specificaties. Dit framesynchronisatiepatroon is niets anders dan een markering die zegt "hier stopt de bitsynchronisatie en volgt het volgende deel van de D-STAR stream: de D-STAR header".

De D-STAR header

Zoals eerder vermeld, is de D-STAR header het onderdeel dat volgt op de framesynchronisatie. Laten we eens kijken wat de GMSK demodulatiesoftware daarover zegt.

Dit is de informatie opgeslagen in de header van dit specifiek GMSK-fragment (volgens de GMSK-receiver software):

```
HEADER DUMP:
FLAG1: 00 - FLAG2: 00 - FLAG3: 00
RPT 2: DIRECT
RPT 1: DIRECT
YOUR: CQCQCQ
MY: ON1ARF /KRIS
Check sum = E441 (OK)
```

Zoals we kunnen zien bevat de D-STAR header dezelfde informatie als wat je moet instellen op de D-STAR radio, met nog iets extra: 3 'vlag' bytes en een checksum (2 bytes). Je zou hierdoor kunnen aannemen dat de D-STAR header 328 bits (41 bytes x 8 bits) bevat, maar dat is niet het geval. De header is in totaal 660 bits. Om dit te begrijpen kunnen we het best in de broncode duiken van de GMSK-decoder. Hierna volgt een iets vereenvoudigd overzicht van het programmagedeelte dat de 660 bits na het framesynchronisatiepatroon verwerkt:

```
scramble(audioheaderbuffer_in,audioheaderbuffer_temp1);
deinterleave(audioheaderbuffer_temp1,audioheaderbuffer_temp2);
length=FECdecoder(audioheaderbuffer_temp3,audioheaderbuffer_out);
```

De ontvangen bits van de D-STAR header worden verwerkt door 3 opeenvolgende functies: scrambling, deinterleaving en FEC-decoding. De reden hiervoor kan beter begrepen worden als we het proces bekijken om een D-STAR signaal aan te maken. Dit bevat net dezelfde functies als de decoder, maar dan in omgekeerde volgorde: eerst de FEC-encoding, dan interleaving en dan scrambling. De bedoeling van deze drie processen is om de header robuuster te maken voor fouten tijdens de signaaloverdracht.

Stap 0: twee bits toevoegen

De allereerste stap in dit proces lijkt een beetje vreemd: voor het verwerken van de header worden er 2 bits toegevoegd op het einde van de header. Beide bits bevatten geen informatie (ze zijn beiden nul) maar zijn nodig voor de volgende stap (FEC). In ieder geval doen zij de lengte van de header toenemen van 328 naar 330 bits.

Stap 1: FEC

De eerste eerste echte stap in het verwerken van de D-STAR header is de 'FEC encodage'. FEC staat voor 'Forward Error Correction' en dient – zoals de naam aangeeft – om fouten te corrigeren in digitale radiosignalen. Het doet dit door **vooraf** extra bits toe te voegen aan het signaal, waardoor het systeem bij eventuele fouten deze extra informatie kan gebruiken om die te corrigeren.

FEC foutcorrectiesystemen vormen een tamelijk complexe materie en vallen buiten het bestek van dit artikel. Wel belangrijk om weten is dat door het toepassen van FEC de lengte van de D-STAR header verdubbelt: de oorspronkelijk header van 330 bits bevat nu 660 bits.

Stap 2: interleaving

De volgende stap heet 'interleaving' wat eigenlijk een mooi woord is voor 'bitjes dooreen gooien'. Wat interleaving doet is de volgorde van de bits veranderen. Indien een header 4 bits zou bevatten - bits 1, 2, 3 en 4 - dan zal het interleaving proces dit veranderen in 1, 3, 2, 4. Dit heeft te maken met het FEC foutcorrectiesysteem. Naast het feit dat een FEC

header plus résistants aux erreurs en ajoutant des bits de corrections d'erreur, a-t-il encore une importante caractéristique. Les algorithmes d'un système FEC fonctionnent sur base d'une suite de bits. La valeur d'un bit à la sortie du modem FEC n'est pas seulement déterminée par un bit unique à l'entrée, mais aussi par le deuxième bit qui a été reçu pour cela. Si plusieurs bits sont effacé par interférence (impuls-noise), alors le résultat du système FEC sera également incorrect. C'est cela maintenant que l'interleaving essaie de diminuer. Par mélange des bits avant émission et – après réception – par remise dans l'ordre correct, les éventuelles erreurs durant l'émission radio sont répartie sur l'ensemble du header. De cette manière la chance pour que deux ou plusieurs bits consécutifs soient perdus est minimisée par rapport à l'envoi de bits consécutifs.

Etape 3: scrambling

'Scrambling' est surtout utilisé pour mettre le secret ou rendre illisible des notes, mais il a ici une tout autre signification. Il faut ici comprendre une technique pour le récepteur pour être mieux synchronisé avec l'émetteur. D-STAR utilise une vitesse de 4800 bits par seconde. Ce qui veut dire que le récepteur et l'émetteur travaille avec une vitesse fixe de 4800 'coups' par seconde. Le problème est qu'il y a toujours une petite différence dans la vitesse d'horloge du récepteur par rapport à l'émetteur. Ce qui donnera – après un certain temps – des erreurs dans la réception parce que le récepteur lira au mauvais moment dans le signal reçu. Un truc pour lutter contre cela est que le récepteur utilise le passage de 0 vers 1 et de 1 vers 0 pour se maintenir synchrone.

Le scrambling aide à améliorer cela. Même après la et l'interleaving il y a encore un risque de transmettre trop peu de passages 0-1 ou 1-0, par lesquels pour le récepteur des possibilités de problèmes de synchronisation peuvent survenir. Pour ces raisons, à des places prédéterminées, les bits sont placés de 0-vers-1 et inversement et pour augmenter les changement de valeurs des bits dans le signal.

Remarquons que le processus de scrambling est exactement le même tant du côté émission que réception. Il n'y a pas de fonction unscrambling.

Pour tout résumer:

1. L' header D-STAR contient toutes les informations nécessaires pour pouvoir router un QSO D-STAR sur le réseau D-STAR, particulièrement les callsigns.
2. Pour rendre le header D-STAR moins sensible aux perturbations il est appliqué trois traitements différents sur le header préalablement et après l'expédition. Le header en est doublé en taille.
3. Heureusement il y a des modules open source pour apporter ces fonctions, lesquelles sont considérées comme des boîtes noires magiques pour cet article.
4. A 4800 bits par seconde l'envoi du header dure 137,5 millisecondes.

Enfin: la parole

Regardons encore une fois les spécifications pour voir ce qui suit. La page cinq du document donne la réponse: parole et données basse vitesse. La structure de ce morceau d'un flux D-STAR est simple. Dans les 20 millisecondes 96 bits (12 bytes) sont envoyés:

- 72 bits (9 bytes) pour la communication parlée, encodée par le vocoder audio AMBE
- 24 bits (3 bytes) pour des données basse vitesse

Si nous jetons un coup d'œil sur le dump du flux de bits, alors, à première vue il n'y a pas grand chose à voir:

système de bits in de header resisterter maakt voor fouten door extra foutcorrectiebits bij te voegen, heeft het nog een belangrijke eigenschap. De algoritmes van een FEC-systeem werken op basis van opeenvolgende bits. De waarde van een bit aan de uitgang van de FEC-modem wordt niet enkel bepaald door één enkele bit aan de ingang, maar ook door de twee bits die daarvoor ontvangen werden. Indien meerder bits na elkaar uitgewist worden door storingen (impuls-noise), dan zal het resultaat van het FEC-systeem ook verkeerd zijn. Dit is nu net wat interleaving probeert te vermijden. Door de bits eerst 'door elkaar te schudden' voor het uitzenden en – na ontvangst – terug in de juiste orde te plaatsen, zullen eventuele fouten tijdens de radiouitzending gespreid worden over de ganse header. Op die manier is de kans dat twee of meerdere opeenvolgende bits verloren gaan veel kleiner dan in het geval dat de bits gewoon na elkaar worden verstuurd.

Stap 3: scrambling

'Scrambling' wordt meestal gebruikt voor het geheim of onleesbaar maken van berichten, maar heeft hier een andere betekenis. Hier wordt een techniek bedoeld om de ontvanger beter synchroon te houden t.o.v. de zender. D-STAR werkt met een snelheid van 4800 bits per seconde. Dat wil zeggen dat zowel de zender als de ontvanger met een vaste klok werken van 4800 'slagen' per seconde. Het probleem is dat er altijd wel een klein verschil zit in de snelheid van de klok van de ontvanger t.o.v. de zender. Daar zal die – na een bepaalde tijd – fouten geven in de ontvangst omdat de ontvanger op het verkeerde moment de bits leest in het ontvangen signaal. Een truc om dit tegen te gaan is dat de ontvanger de overgang van 0 naar 1 en van 1 naar 0 gebruikt om zichzelf synchroon te houden.

Scrambling helpt om dit te verbeteren. Zelfs na de FEC en de interleaving zijn er bepaalde stukken waar het risico bestaat dat er weinig 0-1 of 1-0 overgangen uitgezonden worden, waardoor er bij de ontvanger mogelijks synchronisatieproblemen kunnen optreden. Om die reden worden in voorafbepaalde plaatsen de bits omgedraaid van 0-naar-1 en omgekeerd en om zo meer bitovergangen te produceren in het signaal.

Merk op dat het scramblingproces exact hetzelfde is aan de zenderzijde en de ontvangstkant. Er bestaat geen afzonderlijke unscrambling-functie.

Om alles samen te vatten:

1. De D-STAR header bevat alle informatie die nodig is om een D-STAR QSO te kunnen routeren over het D-STAR network, vooral callsigns.
2. Om de D-STAR header minder gevoelig te maken voor storingen worden er drie verschillende bewerkingen uitgevoerd op de header voor en na het versturen. Daardoor wordt de header wel dubbel zo groot.
3. Gelukkig bestaan er open source modules om deze functies uit te voeren zodat ze voor dit artikel kunnen beschouwd worden als magische zwarte dozen.
4. Aan 4800 bits per seconde duurt het versturen van de header 137,5 milliseconden.

Eindelijk: spraak

Laten we nog eens in de specificaties kijken wat er nu volgt. Pagina vijf van het document geeft het antwoord: spraak en lage snelheidsdata. De structuur van dit gedeel van een D-STAR stream is eenvoudig. Om de 20 milliseconden worden 96 bits (12 bytes) verstuurd:

- 72 bits (9 bytes) voor de spraakcommunicatie, geëncodeerd met de AMBE audio vocoder
- 24 bits (3 bytes) voor slow speed data (lage snelheidsdata)

Als we een blik werpen op de bitstream dump, dan is er op het eerste zicht niet veel te zien:


```

11011111 11010000 00011000 00011001 11100110 10010100
10110111 11111111 10011101 10101010 10110100 01101000

01111101 01010000 01100000 00011111 10100100 11011100
10001001 01100010 10100111 00001100 00100000 10000011

01011111 10010001 01010000 00000011 10100100 11111100
10011111 11100000 10100101 10011100 00111000 11100011
( ... )

```

Chaque séquence (paire de ligne de chaque 12 bytes dans ce dump) dure 20 millisecondes. Les premiers 9 bytes contiennent de la parole + FEC. Les derniers bytes contiennent des données basse vitesse.

Dans ces données basse vitesse peuvent encore être mentionné ce qui suit:

- Les trois bytes de données basse vitesse de la première séquence du flux, et pour chaque 21ème séquence suivante, ne contiennent pas de données utilisateur. En lieu et place se trouve une séquence fixe (10101010 10110100 01101000) comme on peut le voir dans la première ligne dans l'exemple ci-dessus. Cette séquence tient lieu d'information de synchronisation supplémentaire. De cette façon le récepteur peut maintenir le tempo avec l'émetteur.
- Pour éviter que des longues séquences de 0 seul ou 1 seul dans les données basse vitesse mettent en danger la synchronisation du récepteur avec l'émetteur, le scrambling est aussi ici appliqué.

La fin

Tout a une fin, un QSO également. Un des avantages de la communication digitale est que non seulement le début et les données d'un QSO Digital Voice est inventorié. Egalement la fin du QSO est marquée à l'intérieur du flux. Cela rend beaucoup plus facile de déterminer la fin d'un flux, indépendamment de la disparition de la porteuse. Surtout pour les fluxs qui sont envoyés par le réseau D-STAR vers un autre repeater c'est un avantage.

Consultons encore pour la dernière fois les spécifications. La page 6 mentionne:

The last data frame, which requires a means of terminating the transmission, is a unique synchronizing signal (32 bit + 15bit "000100110101111" + "0", making 48 bits) as defined by the modulation type.

La dernière séquence de données, nécessaire pour déterminer la fin d'une émission, est un signal unique de synchronisation (32 bits + 15 bits "000100110101111" + "0", 48 bits au total), tel que déterminé par le type de modulation.

Clairement libellé donc... Heureusement il y a encore le code source qui peut donner plus d'explication. De cela il ressort que le 'signal de synchronisation unique' est au fond comme suit:

- Les 3 derniers bytes d'une séquence contiennent '10101010 10101010 10101010'
- Les 3 premiers bytes de la séquence suivante contiennent '10101010 00010011 01011110'

Nous reprenons la sortie du programme récepteur GMSK:

```

11011001 10110000 00101101 00001101 11000011 11000110
00111111 01100110 01010110 10000100 00110000 01000011

00111011 10010011 00000101 10001000 01101011 01111111
11110001 11001100 10010111 11111110 10101010 10101010

10101010 00010011 01011110 END.

```

Si vous prêtez attention, vous voyez que la séquence dans le dump ne correspond pas à 100 % avec les spécifications. Le dixième byte de la der-

```

11011111 11010000 00011000 00011001 11100110 10010100
10110111 11111111 10011101 10101010 10110100 01101000

01111101 01010000 01100000 00011111 10100100 11011100
10001001 01100010 10100111 00001100 00100000 10000011

01011111 10010001 01010000 00000011 10100100 11111100
10011111 11100000 10100101 10011100 00111000 11100011
( ... )

```

Elk frame (lijnenpaar van elk 12 bytes in deze dump) is 20 milliseconden. De eerste 9 bytes bevatten spraak + FEC. De laatste bytes bevatten lage snelheidsdata.

Over deze trage snelheidsdata kan nog het volgende vermeld worden:

- De drie lagesnelheidsdatabytes van het eerste frame van de stream, en van elk 21ste frame daarna, bevatten geen gebruikersdata. In plaats daarvan bevindt er zich een vast bitpatroon (10101010 10110100 01101000) zoals ook kan waargenomen worden in de eerste lijn in het voorbeeld hierboven. Dit patroon dient als extra synchronisatie-informatie. Op die manier kan de ontvanger zichzelf in tempo houden met de zender.
- Om te vermijden dat lange patronen van allemaal-0 of allemaal-1 in de lage snelheidsdata de synchronisatie van de ontvanger t.o.v. de zender in gevaar zouden brengen, wordt ook hier scrambling toegepast.

Het einde

Aan alles komt een einde, ook aan een QSO. Eén van de voordelen van digitale communicatie is dat niet alleen het begin en de data van een Digital Voice QSO beschreven is. Ook het einde van de QSO wordt binnen de stream gemarkeerd. Dit maakt het veel gemakkelijker om mooi het einde van een stream te bepalen, onafhankelijk van het wegvallen van de draaggolf. Vooral bij streams die over het D-STAR netwerk naar een andere repeater gestuurd worden is dit een voordeel.

Laten we voor de laatste maal nog eens de specificaties raadplegen. Pagina 6 vermeldt:

The last data frame, which requires a means of terminating the transmission, is a unique synchronizing signal (32 bit + 15bit "000100110101111" + "0", making 48 bits) as defined by the modulation type.

Het laatste dataframe, nodig om het einde van een uitzending te bepalen, is een uniek synchronisatiesignaal (32 bits + 15 bits "000100110101111" + "0", 48 bits in totaal), zoals bepaald door het modulatietype.

Duidelijke bewoording dus... Gelukkig is er nog de broncode die meer uitleg kan geven. Daaruit blijkt dat het 'unique synchronisation signal' eigenlijk het volgende is:

- De laatste 3 bytes van een frame bevat het patroon 10101010 10101010 10101010
- De eerste 3 bytes van het volgende frame bevat het patroon 10101010 00010011 01011110

We nemen terug de uitvoer van de GMSK-receiver software erbij:

```

11011001 10110000 00101101 00001101 11000011 11000110
00111111 01100110 01010110 10000100 00110000 01000011

00111011 10010011 00000101 10001000 01101011 01111111
11110001 11001100 10010111 11111110 10101010 10101010

10101010 00010011 01011110 END.

```

Als je goed oplet zie je dat het patroon in de dump niet voor 100 % overeenkomt met de specificaties. De tiende byte van de laatste lijn zou

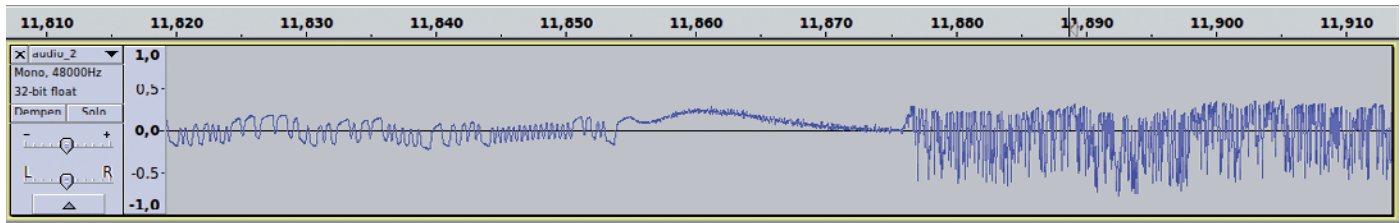
nière ligne doit être 10101010, mais contient 11111110. Cette séquence est quand même reconnue comme un marqueur de fin de flux valable. La raison en est que le programme est spécialement modifié pour gérer les éventuelles erreurs dans le marqueur de fin. Si cela n'était pas le cas, le récepteur manquerait la fin du flux.

L'image ci-dessous montre comment a été reçu le signal par la development board ARM:

10101010 moeten zijn, maar bevat 11111110. Toch wordt dit patroon herkend als een geldige einde-stream marker.

De reden daarvoor is dat de software speciaal aangepast is om met eventuele fouten in de end-marker om te gaan. Moest dat niet het geval zijn, dan zou de ontvanger het einde van de stream missen.

De figuur hieronder toont hoe het signaal door het ARM development board ontvangen wordt:



Le marqueur de fin se situe entre 11,844 en 11,850: les 16 fois 01 sont bien visible dans l'image. On peut aussi voir pourquoi le début de la marque de fin n'est pas correctement décodé: une petite déviation du signal reçu fait en sorte que le signal se place juste un peu au-dessus de la ligne zéro, d'où la traduction des cette séquence de bit comme un 1.

En fin de compte ceci montre précisément la difficulté qu'a le programme décodeur GMSK: interpréter correctement les signaux, également si des erreurs y sont présentes.

73,
Kristoff - ON1ARF

De EINDE-markering bevindt zich tussen tijdmarkeringen 11,844 en 11,850: de 16 maal 01 is mooi te zien in de figuur. Er is ook te zien waarom het begin van de eindmarkering niet juist gedecodeerd wordt: een kleine drift van het ontvangen signaal zorgt ervoor dat het signaal nog net boven de nullijn uitkomt, waardoor deze bitsequentie vertaald wordt als een 1.

Uiteindelijk toont dit aan wat precies de moeilijkheid is voor GMSK-decodersoftware: signalen juist interpreteren, ook als er fouten aanwezig zijn.

73,
Kristoff - ON1ARF



ar benelux








NIEUW!

Rigol DSA815 9 kHz - 1.5 GHz Spectrum Analyzer, 100 Hz RBW

Join the wave!



Prijzen (incl. BTW):

DSA815 € 1259,=

DSA815-TG € 1453,=

(Inclusief Tracking Generator)

Overige opties op aanvraag:

- EMI-Filter & Quasi peak detector
- Advanced Measurements
- VSWR Measurements

AR Benelux is de officiële hoofddistributeur van Rigol in de Benelux met lokale support en 3 jaar garantie!

oscilloscopes • ac / dc power • test & measurement • emc • rf & μwave • services & specials

ar benelux bv info@arbenelux.com • www.arbenelux.com • +31 (0)172-423 000