

Prototypage de traitements numériques de signaux au moyen d'un outil libre : GNURadio

G. Matten, J.-M Friedt, É. Carry
FEMTO-ST, UMR CNRS 6174, Univ. Franche-Comté, Besançon, France
jmfriedt@femto-st.fr, <http://jmfriedt.free.fr>

RÉSUMÉ : la disponibilité à faible coût d'interfaces d'acquisitions – carte son d'ordinateur ou récepteur de télévision numérique terrestre – couplée à un logiciel libre implémentant divers outils de prototypage de traitement du signal numérique, sont décrits pour offrir une solution attractive pour introduire les notions de traitement numérique du signal. La capacité de s'approprier les méthodes de traitement en analysant les codes sources du logiciel garantit une compréhension des algorithmes mis en œuvre et fournit les bases pour adapter le logiciel initialement orienté vers la démodulation de signaux radiofréquences vers une vaste gamme de domaines d'applications. Nous mentionnerons ici quelques applications allant de la caractérisation des fonctions de transfert de dispositifs audiofréquences jusqu'à la réception de modes numériques de communication radiofréquences en passant par le simple oscilloscope.

Mots clés : logiciel libre, traitement du signal, radio définie par logiciel, GNURadio

1 Introduction

Le traitement numérique du signal devient un domaine incontournable dans la caractérisation ou l'asservissement de systèmes physiques. Initialement sélectionné, lors de la transition depuis le traitement analogique de signaux, pour sa flexibilité (possibilité de modifier les caractéristiques de filtres sans souder de composant), sa reconfigurabilité (capacité à faire évoluer les fonctions de transfert en fonction du régime de fonctionnement du système physique) et sa stabilité (absence de dérive des paramètres dans le temps, par exemple du fait des dérives des valeurs de composants analogiques), le traitement numérique du signal est une alternative crédible aux traitements analogiques par la chute des coûts des composants numériques et des microcontrôleurs en particulier.

Le traitement numérique est, au premier abord, un sujet peu attractif s'il reste confiné aux concepts théoriques, incontournables pour une exploitation rationnelle du domaine : seule une mise en pratique permet de réellement s'approprier les concepts aussi simples que la conception d'un filtre numérique ou l'exploitation volontaire du repliement spectral. GNURadio¹ est un ensemble de bibliothèques de traitement numérique du signal incluant filtres, modulateurs et démodulateurs, diverses sources de données (carte son, récepteur de télévision numérique terrestre, émetteur et récepteur dédiés) et puits (interfaces graphiques d'oscilloscope et d'analyseur de spectre, carte son, convertisseur numérique-analogique) ainsi qu'un séquenceur de tâches.

S'approprier un domaine nécessite de pouvoir s'y impliquer en dehors de quelques heures de

cours présentant des sujets formatés, en accord avec les préceptes d'une formation Cursus de Master en Ingénierie (CMI) : la multitude d'instruments numériques qui nous entourent (console de jeu portable, téléphone mobile, ordinateurs) offrent des plateformes matérielles de développement idéales accessibles au plus grand nombre. Les architectures de processeurs ARM qui dominent ces marchés ne sont pas appropriées pour faire fonctionner des outils de traitement du signal propriétaires généralement compilés pour processeur compatible Intel x86 fonctionnant sur un système d'exploitation donné. La disponibilité des codes sources de GNURadio permet d'une part de déployer cet outil sur toute plateforme compatible POSIX – en particulier exécutant le système d'exploitation GNU/Linux, avec ou sans interface graphique – et d'autre part d'apprendre par la lecture des codes sources en C et Python, les méthodes de traitement mises en œuvre par les auteurs de ces outils.

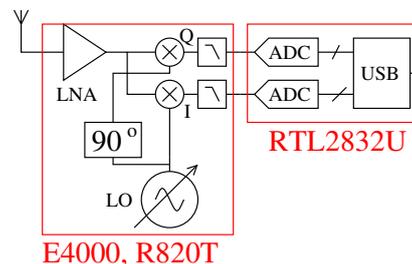


FIGURE 1 – Architecture du récepteur DVB-T : un démodulateur I/Q sans fréquence intermédiaire convertit le signal radiofréquence reçu par l'antenne après amplification (LNA – *Low Noise Amplifier*) en signal basse-fréquence après mélange avec un oscillateur local (LO – *Local Oscillator*) de fréquence configurable numériquement. L'Elonics

1. <http://gnuradio.org/>

E4000 est obsolète, remplacé par le Rafael Micro R820T. Les signaux en identité (I) et quadrature (Q) sont numérisés et transmis sur bus USB par le Realtek RTL2832U.

Un développement original concerne le détournement de récepteurs de télévision numérique terrestre – DVB-T – formés (Fig. 1) d'un premier étage de transposition de signaux radiofréquences (amplificateur - démodulateur I/Q sans fréquence intermédiaire - filtre passe-bas) et d'un convertisseur analogique-numérique avec interface USB (Realtek RTL2832U). Il s'avère que la flexibilité de cette architecture permet de l'utiliser pour des applications bien plus intéressantes que la réception de signaux de télévision² : le démodulateur peut s'ajuster sur la bande de fréquence de l'ordre de 50 à 1800 MHz et le flux de signaux I/Q (identité et quadrature, en vue de représenter le signal complexe dont la magnitude et phase seront exploités pour démoduler le signal) est suffisamment général pour permettre de détecter tout signal dans la bande passante du système d'acquisition, limitée à 2,4 Méchantillons/s par la bande passante du bus USB. Plus classique, la carte son permet de générer et acquérir un signal audiofréquence sur deux canaux simultanément avec un bande passante de l'ordre de 48 kHz. Cette gamme de fréquence inclut en particulier la fréquence de fonctionnement du diapason à quartz ($32768=2^{15}$ Hz), un objet qui mérite une investigation à part entière et l'opportunité d'introduire les concepts électromécaniques liés à la piézoélectricité.

2 Généralités sur GNURadio

GNURadio est un environnement libre de traitement de signaux, fonctionnant sur toutes les plateformes les plus communes (GNU/Linux, *BSD, MacOS, MS-Windows). Sa compilation est rendue quelque peu fastidieuse par l'activité intense des développements et une refonte d'un certain nombre des interfaces avec le passage récent à la date de rédaction de cette prose à la version 3.7. Elle est cependant conseillée afin de garantir une version à jour et une cohérence avec les bibliothèques contre lesquelles les blocs de traitement que nous allons développer seront liés. L'utilisateur qui veut se contenter des blocs de traitement existant et n'envisage pas de développer ses propres fonctions peut se contenter des paquets binaires de leur distribution favorite (la version stable de Debian/wheezy propose encore la version obsolète de GNURadio 3.5,

2. <http://sdr.osmocom.org/trac/wiki/rtl-sdr>

tandis que la version *testing* de Debian est passée à la nouvelle API 3.7).

D'un point de vue utilisateur, une interface graphique `gnuradio-companion` aide l'utilisateur débutant à assembler la séquence de traitements comprenant une source, les blocs de traitement et un ou plusieurs puits de données. Les blocs de traitement doivent être assemblés avec un souci de cohérence du flux de données qui n'est *pas* géré automatiquement : ce point est critique lorsque la source de données et le puits ne produisent et exploitent pas le même débit de données. Ce cas est courant : par exemple, un DVB-T génère un flux de 2,4 Méchantillons/s complexes, tandis qu'une carte son ne consomme que 48 kéchantillons/s. Le rapport de 50 doit être pris en compte dans les décimations successives (filtrage, démodulation) en tenant compte de la bande passante nécessaire à chaque séquence de traitement et aux ressources de calcul associées. L'interface graphique gère la cohérence de la nature des flux échangés entre les blocs (entier, réel, complexe ...).

`gnuradio-companion` n'est en fait qu'un générateur de code Python que l'utilisateur pourra rapidement écrire manuellement. Les données elles-mêmes ne sont pas accessibles depuis ce programme Python, mais les blocs de traitement sont décrits soit en C, soit en Python. Au-delà de la documentation sur le site web du projet GNURadio, les codes sources sont l'inspiration la plus naturelle pour rédiger ses propres fonctions de traitement pour répondre aux besoins plus exotiques que la démodulation de modes de communication numériques ou analogiques radiofréquences. Nous proposons quelques applications ci-dessous pertinentes pour des domaines aussi variés que la caractérisation des modes propres d'un objet, la caractérisation d'un diapason à quartz ou la réception d'un signal numérique modulé en fréquence (FSK) ou en amplitude (AM).

3 Traitement de signaux audiofréquences

Les cartes son sont une interface d'excitation et d'acquisition classiquement disponibles sur la majorité des ordinateurs récents. GNURadio permet de générer un signal depuis la sortie casque et d'acquérir un signal depuis l'entrée microphone, mais surtout Python supporte, si le matériel le permet, une communication full-duplex dans laquelle émission et réception sont simultanées. Ainsi, nous réalisons trivialement un analyseur de réseau au-

diofréquence.

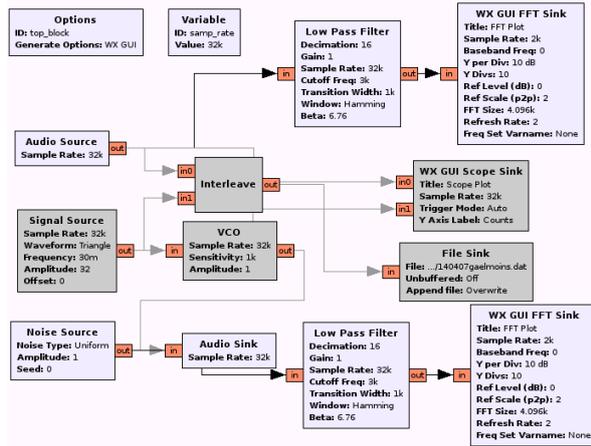


FIGURE 2 – Deux modes de fonctionnement d’analyseurs de réseaux audiofréquences sont présentés dans cette interface : en gris (commenté) un mode par balayage de fréquence dans lequel la réponse aux sollicitations successives par des signaux de fréquences variables est enregistrée ; et en couleur (actif) un mode par transformée de Fourier de la réponse excitée par une source de bruit blanc.

Comme tout analyseur de réseau, deux architectures sont accessibles (Fig. 2) : exciter le dispositif analysé par un bruit rose et tracer la transformée de Fourier du signal acquis pour en analyser la réponse spectrale ; ou exciter le dispositif analysé par un signal périodique de fréquence variable, et enregistrer pour chaque fréquence la réponse en magnitude et phase.

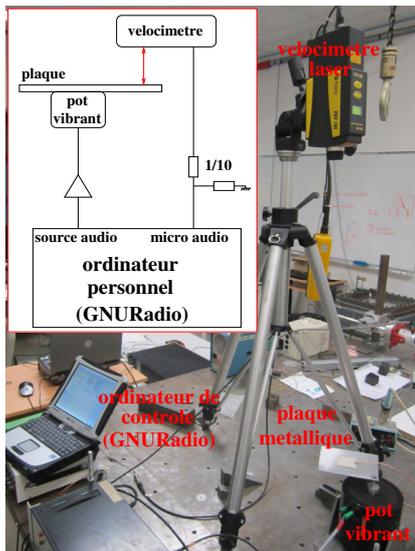


FIGURE 3 – Dispositif expérimental de mesure des modes propres d’une plaque vibrante.

Les deux approches ont été mises en œuvre pour la caractérisation des modes d’une plaque métallique (Fig. 3) : fixée sur un pot vibrant, la vitesse de déplacement en un point de la plaque est observée par un vélocimètre laser. Le signal d’excitation du pot vibrant est généré par la sortie audio d’un ordinateur personnel, et la sortie analogique est acquise par l’entrée audio de cette même carte son (après réduction du signal par un pont diviseur).

D’un point de vue GNURadio, le schéma du traitement se résume en une source (bruit blanc ou source de signal périodique de fréquence définie par un triangle) qui définit la sortie audio. L’entrée audio est quant à elle soit tracée sous forme de composantes de Fourier par l’outil de traitement approprié, soit enregistrée dans un fichier pour tracé ultérieur de l’amplitude du signal acquis en fonction de la fréquence d’excitation (Fig. 2).

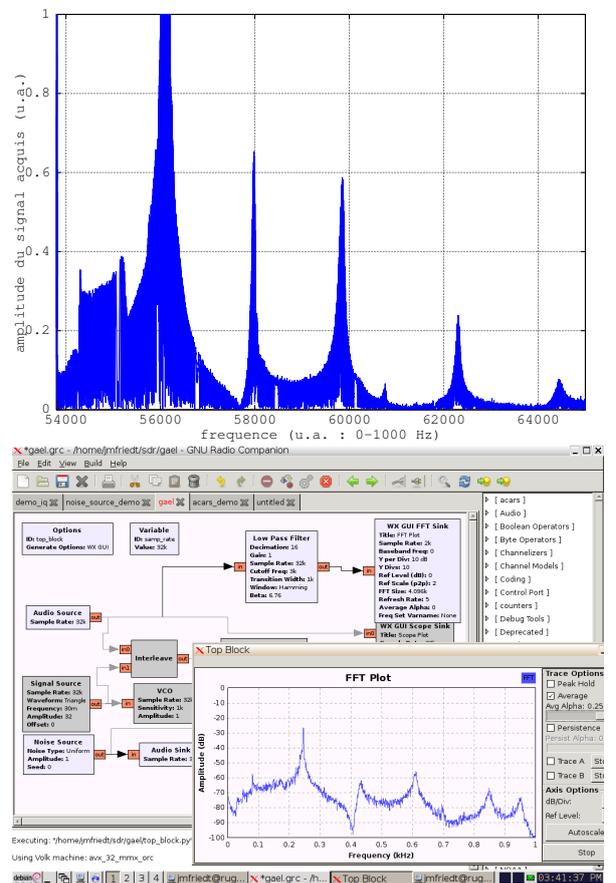


FIGURE 4 – Exemple de mesure acquise par une excitation par un bruit blanc du pot vibrant (en haut) et par balayage de fréquence (en bas). Les divers modes propres entre 0 et 1000 Hz sont clairement visibles.

Un exemple de mesure issu des deux modes d’acquisition est proposé sur la Fig. 4 : la cohérence

entre les deux séquences de données est évidente, avec divers modes propres visibles dans la gamme explorée entre 0 et 1000 Hz.

4 Traitement de signaux haute-fréquence

La caractérisation de dispositifs audiofréquences est certes déjà une première étape pour se familiariser avec la configuration du flux de données et la mise en œuvre des interfaces d'acquisition et de contrôle, mais la bande passante d'analyse est limitée par la fréquence d'échantillonnage de la carte son à 48 ou 96 kéchantillons/s. Largement suffisant pour qualifier de nombreux montages à base d'amplificateurs-opérationnels, cette fréquence d'échantillonnage est tout juste suffisante pour travailler sur le mode fondamental du diapason à quartz [1], objet de fort facteur de qualité et donc utile pour se familiariser avec les concepts des objets à forte pureté spectrale (par exemple la constante de temps de relaxation en Q/π périodes – Q le facteur de qualité de l'objet – est de plusieurs secondes pour un diapason encapsulé sous vide et d'une fraction de seconde pour un diapason décapsulé et exposé à l'air). Une vidéo illustrant ce point est disponible à <http://jmfriedt.free.fr/diapason0.avi>. On notera par ailleurs que diverses méthodes de traitements de signaux radiofréquences visent à ne générer en sortie des mélanges analogiques un signal ne comportant que des composantes dans la bande audiofréquence. En particulier, le RADAR à balayage de fréquence (FMCW) excite une onde aux hyperfréquences mais par un mélange entre le signal émis à un instant et les échos retournés après un délai lié à la distance de la cible, le signal acquis n'est qu'un battement périodique qui, par un choix judicieux des réglages de l'instrument (excursion spectrale, vitesse de balayage), se limite à une fréquence compatible avec la bande passante d'une carte son [2, 3].

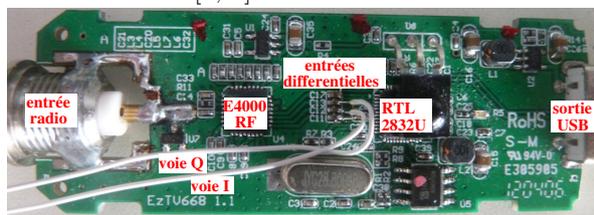


FIGURE 5 – Modifications matérielles sur récepteur DVB-T afin d'exploiter directement le convertisseur analogique-numérique sans passer par l'étage de démodulation radiofréquence.

Afin d'atteindre des bandes passantes dépassant le MHz, le convertisseur analogique-numérique équipant les récepteurs DVB-T est une solution intéressante pour un coût de l'ordre de la dizaine d'euros (Fig. 5). Les deux voies d'acquisitions des deux canaux I et Q fournissent deux voies de mesures simultanées de données réelles codées sur 8 bits [4]. Dans ce cas, les signaux que nous injectons sont directement connectés *après* les condensateurs de découplage en sortie de voies I/Q du récepteur de télévision numérique terrestre et en entrée du convertisseur analogique (Fig. 6). La chaîne d'acquisition, de traitement et d'affichage est triviale (Fig. 6) : le flux complexe interlace sous forme de partie réelle et imaginaire les deux voies d'acquisitions, qui sont séparées au moyen des blocs d'extraction des parties réelle et imaginaire. Dans ce cas, nous affichons l'évolution temporelle du signal, mais ce schéma de base peut être complexifié à souhait en ajoutant les blocs de traitement qui ne sont pas spécifiques à la radiofréquence tels que les filtres ou le passage dans le domaine spectral.

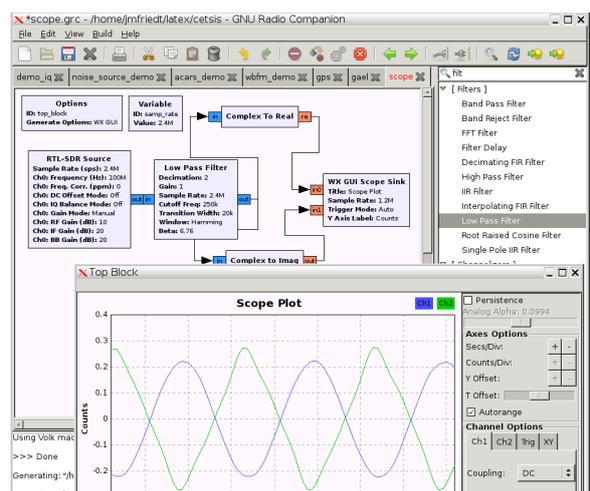


FIGURE 6 – Utilisation du convertisseur analogique-numérique d'un récepteur DVB-T pour échantillonner deux signaux, l'un sinusoïdal d'amplitude 150 mV_{pp}, l'autre triangulaire d'amplitude 200 mV_{pp}, tous deux de fréquence 40 kHz. Le filtre passe-bas a pour vocation d'illustrer la décimation entre la source et le puits.

En particulier, nous sensibilisons avec cette application au problème de la gestion du flux de données : le filtre passe bas que nous avons inséré entre la source et l'oscilloscope décime d'un facteur 2. Nous avons pris soin de manuellement indiquer à l'oscilloscope que sa fréquence d'échantillonnage est moitié de celle de la source : cette opération manuelle reste fastidieuse et source d'erreurs. Les fréquences d'échantillonnage en entrée et en sor-

tie doivent être des fractions rationnelles : les coefficients des blocs décimateurs et interpolateurs doivent être choisis judicieusement pour atteindre cette fraction, tout en maintenant le bande passante de traitement de chaque bloc (e.g. un démodulateur FM dans la bande commerciale a besoin de 150 kHz de bande passante) mais en la limitant afin de réduire le puissance de calcul requise pour effectuer chaque tâche.

5 Traitement de signaux radiofréquences

Le nom du logiciel indique clairement une volonté d'aborder le traitement de signaux radiofréquences. Alors qu'initialement des périphériques très performants mais relativement coûteux étaient nécessaires pour acquérir les signaux radiofréquences, ces dernières années ont vu se multiplier les initiatives de développements de périphériques à coût (et bande passante) réduit, pour finalement aboutir sur le récepteur DVB-T qui permet de démoduler à peu près n'importe quel mode de communication radiofréquence occupant une bande passante inférieure au MHz (notons qu'ainsi les signaux bruts GPS³ ont été traités au moyen de ce récepteur).

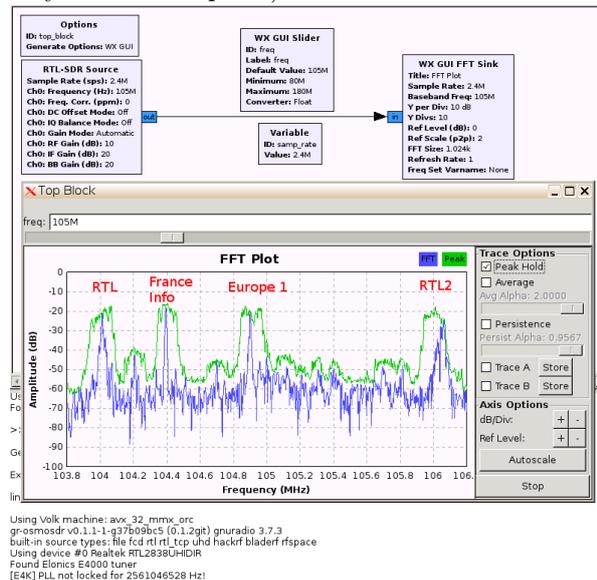


FIGURE 7 – Utilisation du récepteur de télévision numérique terrestre (DVB-T) comme analyseur de spectre dans la bande FM.

Avec un investissement financier de l'ordre de la dizaine d'euros, la seule limitation pour appréhender le domaine de la radio définie par logiciel (*Software Defined Radio* – SDR) est de

3. <http://www.gnss-sdr.org/documentation/gnss-sdr-operation-realtek-rtl2832u-usb-dongle-dvb-t-receiver>

s'approprier le vocabulaire du domaine et identifier un sujet qui n'a pas encore été abordé par la communauté de développeurs GNURadio (<https://www.cgran.org/>). Le périphérique sert trivialement comme instrument de test, par exemple pour explorer les diverses émissions de la bande FM en assemblant simplement deux blocs – une source et un puits de donnée chargé d'effectuer la transformée de Fourier du flux de données I/Q (Fig. 7).

Notre intérêt s'est porté sur ACARS, un protocole de liaison radiofréquence modulé en amplitude autour de 137 MHz pour les liaisons entre les avions et les aéroports. Le protocole est ancien : les documentations sont donc accessibles sur le web. Il ne nécessite que des connaissances modestes sur l'identification des symboles transmis en vue d'interpréter les trames des communications. Un portage récent du logiciel `acarsdec` fait office de référence par rapport auquel nous comparons notre implémentation (Fig. 8). Ce bloc de traitement est écrit en C, basé sur un modèle de la multitude d'exemples disponibles, et s'insère du point de vue de l'utilisateur comme un des divers puits de données disponibles. La séquence de développement, du prototypage à l'implémentation du bloc GNURadio pour le traitement en temps réel, est décrit en détail dans [4], que le lecteur est encouragé à consulter pour approfondir les concepts survolés dans cette prose.

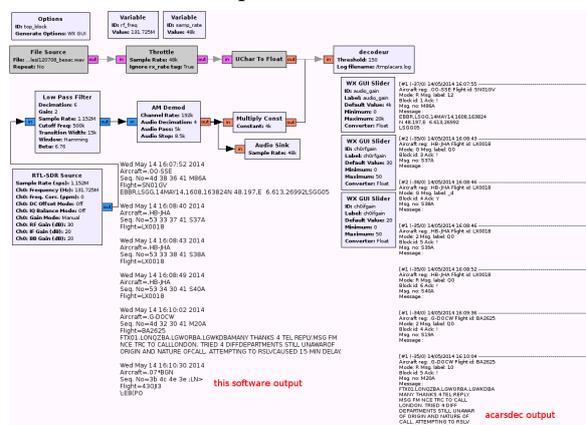


FIGURE 8 – Exemple de traitement de données pour le décodage de trames ACARS : le bloc de démodulation AM de GNURadio est exploité pour fournir un signal au bloc de décodage ACARS qui ne contient plus que l'information encodée en fréquence (FSK) dont il nous reste à identifier les symboles puis les trames. En insert en bas à droite, une comparaison du décodage des trames ACARS par notre code disponible à <https://www.cgran.org/wiki/ACARS>, et `acarsdec` utilisé comme référence.

Un message ACARS comprend l'identifiant de l'avion, l'identifiant du vol, un indicateur de l'ordre du message dans une séquence et finalement le contenu du message lui-même. Alors que la majorité des messages ne contiennent que de la télémétrie complexe à décoder, un message (ici acquis par `acardec`⁴) du type

```
Aircraft reg: .HB-IJR Flight id: LX063J
Mode: R Msg. label: 45
Msg. no: M36A
HDQOLLX,
HALLO
8 PAX FROM LX645 GOING ONTO LX256 WOULD LIKE TO HAVE
KSML ORDERED IS THIS STILL POSSIBLE
THANKS
LX645
```

nous informe que 8 passagers sur l'Airbus A320 immatriculé HB-IJR (Swissair) dans le vol LX063J requièrent des repas kacher (KSML) lors de leur transit entre les vols LX645 (Paris-Zürich) et LX256 (Zürich-Tel Aviv).

La capacité à exécuter le code Python généré par `gnuradio-companion` sur une plateforme sans interface graphique ou tout au moins sans devoir exécuter l'interface utilisateur est un argument en faveur de l'application sur systèmes embarqués. GNU/Linux est le système d'exploitation le plus utilisé sur systèmes embarqués⁵ et la compilation de GNURadio pour les diverses architectures de processeurs équipant les systèmes embarqués est un point clé qui n'est possible que par la disponibilité des codes sources de l'application. Nous avons ainsi été capables d'exécuter GNURadio sur plateformes Zedboard (processeur Zynq⁶) et Armadeus Systems APF51 (processeur iMX [5]). Dans les deux cas, l'affichage est soit déporté vers un écran d'ordinateur personnel, ou simplement inutile si le traitement du signal ne nécessite pas d'interface graphique (démodulation de signaux audiofréquences modulés en fréquence pour une sortie sur carte son).

6 Boucle d'asservissement

Le cas des boucles d'asservissements dans lequel les acquisitions sont rebouclées après traitement vers une sortie pose le problème des latences. Deux sources de retards sont d'une part intrinsèques aux traitements numériques (nombre d'échantillonnages nécessaires à l'application d'un filtre par exemple), et d'autre part liées aux implémentations des outils de traitement (par exemple remplissage de tampons avant d'amorcer le traitement en l'absence d'une architecture pipeline sur le processeur exécutant

GNURadio). Ces diverses sources de déphasage sont à caractériser avec attention si la boucle entre émission et acquisition de données est fermée pour contrôler un système physique.

7 Conclusion

Nous avons proposé une démarche expérimentale visant à s'approprier les concepts de traitement numérique du signal. La démarche consiste d'une part à minimiser l'investissement financier pour s'engager dans cette thématique, et d'autre part à favoriser la capacité de chacun à mettre en œuvre les concepts quelque peu abstraits et en particulier permettant de démontrer les artéfacts liés au traitement en temps discret d'échantillons numérisés. La finalité consiste donc à aider les étudiants à s'approprier des concepts complexes en ayant accès aux outils pour mettre les résultats des enseignements en pratique. En ce sens, l'exploitation de la carte son d'un part, et d'interfaces d'acquisitions rapides d'autre part, compatibles avec un budget réduit et ne nécessitant aucun investissement financier du point de vue logiciel semblent rendre la technologie accessible au plus grand nombre. La disponibilité des codes sources des outils exploités est un point clé pour la diffusion de la connaissance et des compétences liées à la mise en œuvre expérimentale du traitement de signaux numériques.

Ces développements ont été engagés dans le contexte d'une requête d'un groupe d'élèves désireux d'étudier les propriétés des diapasons à quartz mais n'ayant pas réalisé les contraintes de stabilité de la source de fréquence pour sonder un dispositif de facteur de qualité élevé. L'objectif était donc de fournir un environnement de travail accessible sans matériel spécialisé (compteur de fréquence, synthétiseur) tout en fournissant les performances requises. L'outil ne représente cependant qu'une petite fraction des concepts à maîtriser pour une analyse des propriétés spectrales du diapason, qui ne sont cependant pas enseignés dans une formation généraliste. La démarche consistera donc à l'avenir à se supporter sur ce matériel favorisant la capacité de l'étudiant à s'approprier les concepts, pour diffuser les méthodes de travail utilisées sur des instruments spécialisés lors de l'analyse des propriétés spectrales de résonateurs de fort facteur de qualité pour nos activités de recherche. De tels outils ne semblent accessibles qu'aux étudiants visant à maîtriser les concepts d'analyse spectrale des signaux.

4. <http://sourceforge.net/projects/acarsdec/>

5. <http://www.slideshare.net/StephanCadene/embedded-mar1913>

6. <https://github.com/balister/oe-gnuradio-manifest>

Les références bibliographiques sont disponibles à <http://jmfriedt.free.fr>

Références

- [1] J. Marc, C. Canard, A. Vailly, V. Pichery, J.-M. Friedt, *Le diapason à quartz comme capteur : utilisation de la carte son de PC pour l'instrumentation*, Bulletin de l'Union des Physiciens **107** (958), pp.1051-1076 (2013)
- [2] N. Chrétien, J.-M. Friedt, G. Martin, S. Balandras, *Acoustic transducers as passive sensors probed through a wireless radiofrequency link*, Instrumentation, Mesure, Métrologie (I2m), **13** (3-4), pp.159-178 (2013)
- [3] J.- M Friedt, G. Goavec-Mérou, *Traitement du signal sur système embarqué – application au RADAR à onde continue*, GNU/Linux Magazine France **149** (Mai 2012)
- [4] J.-M Friedt, G. Goavec-Mérou, *La réception radiofréquence définie par logiciel (Software Defined Radio – SDR)*, GNU/Linux Magazine France **153**,pp. 4-33 (Octobre 2012),
- [5] G. Goavec-Mérou, J.-M Friedt, *GNURadio as a general purpose digital signal processing environment*, FOSDEM (Bruxelles, 2014)