

European GNU Radio Days 2019

J.-M Friedt, pour le comité d'organisation de la conférence, 16 juillet 2019

La conférence européenne GNU Radio Days s'est tenue mi-Juin à Besançon. Opportunité de faire se rencontrer académiques, industriels, amateurs et professionnels des agences gouvernementales, les 85 participants ont pu démontrer leur capacité à mettre en pratique les concepts de traitement de signaux échantillonnés en temps discrets sur des problèmes concrets grâce à l'environnement libre GNU Radio, largement développé dans ces pages ces derniers mois pour traiter numériquement des signaux radiofréquences.

1 Introduction

Les deuxièmes GNU Radio Days [1] se sont tenus des 17 au 19 juin 2019 dans les locaux de l'École Supérieure de Mécanique et des Microtechniques (ENSMM) à Besançon. Suite à la première édition à Lyon l'année dernière, le choix a été fait de préfixer le titre de European pour expliciter la portée européenne du rendez-vous, au-delà des frontières nationales qui avaient été perçues l'an passé. La publicité au FOSDEM et à la GRCon américaine ont en particulier permis d'attirer un tiers de



participants européens non-français, venant d'aussi loin que la Finlande, au-delà des deux tiers de participants français qui ont connu la date de l'évènement au travers des réseaux sociaux ou des publicités dans ce journal. La majorité des participants est universitaire, probablement par une culture des conférences mieux ancrée, mais la fraction des organisations gouvernementales (recherche et développement, militaire), industrielles et amateurs est significative pour argumenter d'avoir atteint l'objectif de "bridge the gap between technical/engineering & scientific/research points of view on topics researched with GNU Radio, share experience and development strategies" (Fig. 1). Il nous semble en effet que l'environnement libre de développement GNU Radio est un outil idéal pour mettre en pratique des études plus fondamentales de traitement du signal, radiofréquence en particulier, par l'aisance à remplacer des sources synthétiques par des sources de signaux issus de récepteurs de radio logicielle répondant à toutes les attentes de prix en fonction de la bande passante recherchée.

Les trois journées de conférence se sont organisées en une journée de présentations orales, une journée de tutoriels dédiés à GNU Radio et source de données matérielles, et finalement une journée d'introduction à l'environnement de co-développement CPU/FPGA OscImpDigital. Nous ne survolerons que ce dernier point ajouté en cours d'organisation pour mentionner qu'il vise à répondre à une problématique récurrente pour le développeur de savoir où placer le curseur entre traitement en flux tendu selon des chemin physiquement parallèles des données radiofréquences – dans le FPGA nécessairement programmé dans un langage de définition du matériel (VHDL ou Verilog) pour implémenter des traitements en *pipelines* sans FIFO entre les blocs de traitement – et le processeur généraliste, plus lent mais configuré par des langages de haut niveau d'abstraction (C, C++) qui permettent de s'affranchir d'un certain nombre des affres du développement sur FPGA (respect des latences, nombre de bits significatifs, gestion fine des ressources). Ainsi, OscImpDigital vise, sous l'impulsion de Armadeus Systems (maintenant Oposom) qui a financé une thèse sur ce sujet, à fournir un environnement de développement dans lequel des blocs optimisés sont assemblés (méthode du *skeleton*) et fournissant les pilotes pour le noyau Linux ainsi que les interfaces vers l'espace utilisateur pour échanger les informations. La conférence fut l'occasion de remettre à plat l'ensemble des IPs et pilotes ainsi qu'un certain nombre de documentations.

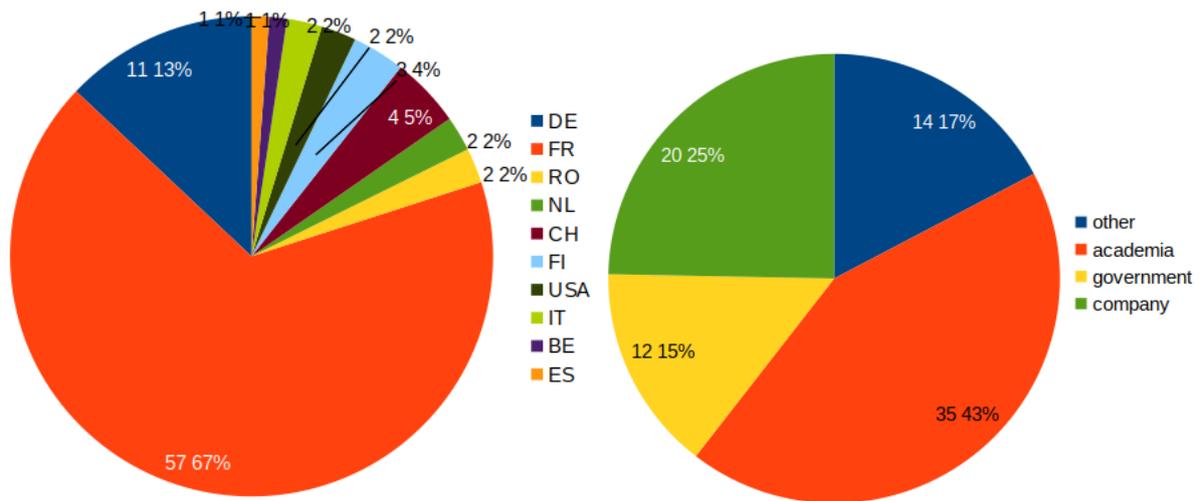


FIGURE 1 – Statistique sur les nationalités (gauche) et l’affiliation (droite) des participants à la conférence. Plus du tiers des participants est européen hors-France, avec une distribution entre académiques (majoritaires), agences gouvernementales, industriels et “autres” incluant radio-amateurs et affiliations professionnelles qui ne sont pas liées à une activité radiofréquence évidente.

Quel outil simple et efficace permet de générer 85 badges à partir d’un fichier ASCII de champs séparés par des virgules (CSV) en insérant le nom, prénom et logo de la conférence? `glabels` propose une interface sobre et simple pour atteindre ce résultat. On lui pardonnera de ne pas avoir d’interface en ligne de commande, la conception d’un badge étant une activité intrinsèquement graphique. Parmi nos objectifs, ne pas mettre en évidence l’affiliation des participants, afin de ne pas différencier académiques, amateurs, industriels et gouvernementaux. Pour ce faire, l’adresse de courrier électronique est encodé sous forme de QRCode : l’interlocuteur désireux de se faire connaître pourra proposer de faire scanner son adresse (en évitant faute d’orthographe et pattes de mouche illisibles sur un bout de papier posé sur un genou). Les divers champs sont identifiés dans le badge par le classique `#{NOM}` mais nous avons eu quelques soucis avec le QRCode pour lequel le champ ne doit être défini *que* dans `Data` mais garder le `Style` en noir et *ne pas lui affecter la valeur d’un champ* (avec le recul, ça semble évident ...). Par ailleurs, `glabels` ne propose pas de grille sur laquelle aligner les divers champs, mais comme les coordonnées exactes des différents objets ou sommets peuvent être entrées manuellement, il suffit d’avoir un brouillon sur papier (le truc qui prend feu quand on le chauffe, issu des fibres de bois ...) pour se rappeler des coordonnées des divers éléments graphiques.

On notera la particularité fort peu amusante que le site de gestion de la conférence `sciencesconf.org` n’inscrit *pas* les présentateurs comme participants à la conférence, une lacune fort désagréable lorsqu’il s’agit de distribuer les paquets de bienvenue aux orateurs.

+

Jean Michel Friedt

European GNU Radio Days 2019



+

2 Conférences

Les présentations orales se sont regroupées en trois grandes classes remarquables : le traitement collaboratif de données acquises selon un réseau spatialement distribué, des aspects de métrologie des oscillateurs qui restent, même si analogiques, la source de cadencement des flux de données et donc imposant les limites sur la gigue de la date d'acquisition des échantillons – définissant donc les performances de toute la chaîne de traitement numérique qui suivra – et des présentations pédagogiques ou sur la sécurité des liaisons radiofréquences. Ce dernier point est resté surprenamment peu développé, sujet pourtant à la mode avec la prolifération des protocoles de communication sans fil laissant chacun sonder le canal de propagation pour en analyser le contenu.

Parmi les points remarquables abordés sur le premier sujet, après une présentation sur le suivi de trajectoires de projectiles balistiques (?!) par un réseau d'antennes et la réception de signaux satellites par le réseau SATNOGS, nous avons découvert KiwiSDR (Fig. 3) qui semble tout a fait passionnant pas sa capacité à dater les trames sur un signal de synchronisation global (1 PPS) du GPS. Cette fonctionnalité répond à un besoin qui a souvent fait défaut lors du traitement de signaux acquis par *websdr* dont l'absence de marqueur de syn-



Figure 3: La KiwiSDR permet la datation sur référence GPS des trames acquises par les récepteurs distribués dans le monde (note : aucun rapport avec l'émetteur Kiwi du CNES conçu pour Planète Science!).

chronisation interdit toute mesure interférométrique cohérente en phase (direction d'arrivée ou temps de vol). Parmi les illustrations des traitements de signaux acquis par réseaux de KiwiSDR, la localisation d'un émetteur inconnu dans les Caraïbes ou la démonstration sur DCF77 de la trilatération du signal à Mainflingen. De telles fonctionnalités ouvrent des perspectives fascinantes pour les suivis de trajectoires de satellites et l'identification de leurs paramètres orbitaux, au-delà de l'ambition d'un réseau mondial de récepteurs de signaux de satellites en orbite basse dont les États-Unis avaient dû se munir lors des premiers vols spatiaux, avant la disponibilité de TDRS. Cette session s'est conclue avec les derniers de résultats de Paul Boven sur la synchronisation de radiotélescopes néerlandais par réseau fibré White Rabbit, une considération qui dépasse quelque peu les activités accessibles par l'amateur mais dans laquelle GNU Radio et les interfaces matérielles qu'il contrôle ont une place centrale. Nous reverrons ce lien recherche-logiciel libre avec les activités de l'Observatoire de Paris sur les horloges optiques. La seconde grande classe de présentations, au vu de l'activité du laboratoire organisateur de cette édition de la conférence, porte sur l'importance du bruit de phase sur les performances d'un récepteur radiofréquence. Tout système numérique synchrone a besoin d'une horloge de référence. L'hypothèse de l'analyse spectrale est que les échantillons sont acquis à intervalle de temps régulier, la période d'échantillonnage. Si la fréquence de l'horloge de référence fluctue (alimentations, dérive thermique, accélération/vibrations), cette hypothèse n'est plus vérifiée et introduit une phase aléatoire sur les échantillons acquis. En nous rappelant qu'à peu près tous les modes récents de communication exploitent une modulation de phase [2], ou qu'une mesure de temps de vol τ d'une impulsion RADAR centrée sur une fréquence f est en pratique mesurée comme une phase $\varphi = 2\pi f\tau$, toute fluctuation de la phase de l'oscillateur local qui sert de référence aura un impact sur la chaîne de mesure qui suit. Lorsque le circuit numérique est cadencé par un oscillateur dédié, la caractérisation de cet oscillateur permet de prédire les planchers de bruits



Figure 2: L'École Supérieure de Mécatronique et des Microtechniques (ENSMM) de Besançon a accueilli les deuxièmes GNU Radio Days pour réunir les utilisateurs et développeurs de GNU Radio.

de la chaîne de mesure, mais si cet oscillateur est issu d'une multiplication par boucle à verrouillage de phase, et en particulier intégré dans des composants aussi complexes qu'un FPGA ou un SOC tel que le Zynq de Xilinx, le problème devient beaucoup moins simple.

Une présentation est ressortie comme particulièrement inspiratrice pour les enseignants et étudiants présents dans l'auditoire : Antoine Blais, de l'ENAC, a présenté comment le système de navigation d'avions par le système VOR est analysé, simulé et décodé par ses étudiants dans le but de mettre en pratique les connaissances acquises en traitement du signal, en particulier radiofréquence. L'étude de signaux interagissant avec les aéronefs est toujours source d'inspiration, en particulier du fait de la portée des signaux émis de quelques milliers de mètres d'altitude tout en restant plus simples à capter que les signaux issus de satellite (dans ce cas particulier, VOR est émis depuis le sol et ne vérifie donc pas ce pré-requis).

Marcus Müller, que les lecteurs de la liste de diffusion `discuss-gnuradio` reconnaîtront peut être comme le contributeur le plus prolifique, a été invité à présenter l'ordonnanceur de GNU Radio et la gestion des flux de données. Architecte du projet, sa présentation est allée bien plus loin que ces considérations techniques en abordant les évolutions attendues dans GNU Radio 3.8 [3] (au-delà des aspects cosmétiques de la migration Python 2.7 → 3 ou vers Qt5 [4]) et futures. Il est toujours intéressant de voir un architecte remettre en cause ses choix ou ceux des prédécesseurs qui avaient conçu un logiciel selon des objectifs ou des contraintes matérielles qui ont largement évolué depuis les 18 années de vie de GNU Radio. Cette première journée studieuse s'est conclue par un barbecue sur la pelouse de l'ENSMM, opportunité pour les participants de faire connaissance ou de poursuivre les discussions engagées dans la journée autour des posters et des démonstrations pendant les pauses café et repas.

3 Démonstrations et posters

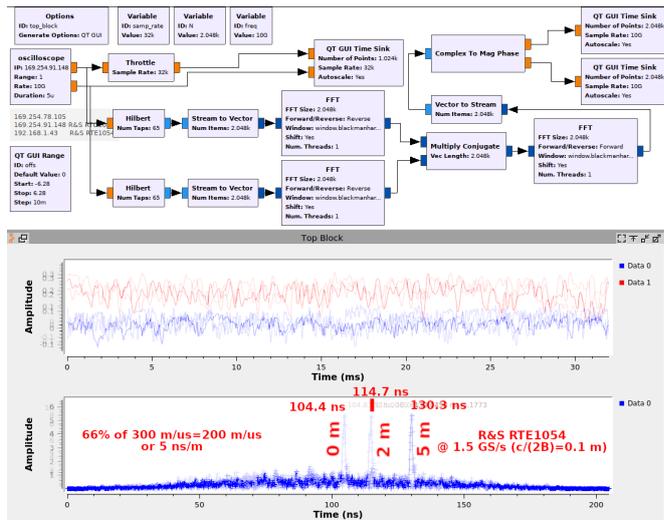


Figure 4: Mesure par corrélation de la distance entre une source de bruit large bande – diode Zener polarisée en inverse – et les divers canaux d'un oscilloscope numérique.

opportunité de commencer à analyser les traitements de réseaux d'antennes pour estimer une direction d'arrivée du signal. Le RADAR passif qui avait été abordé dans ces pages [5] était présent sous forme d'un système de sécurité pour protéger les aéronefs survolant Clermont Ferrand des tirs lasers de LiDAR, que nous prévoyons d'améliorer en remplaçant les récepteurs de télévision terrestre par une Ettus Research B210.

4 Tutoriels

La seconde journée de tutoriels, centrée sur la plateforme PlutoSDR généreusement offerte à tous les participants inscrits avant la date limite officielle par Analog Devices, a tenté d'allier présentations introductives avec la découverte de quelques modes de communication sur le spectre radiofréquence (Leo Cardoso et Matthieu Cunche de l'INSA Lyon, Fig. 5) et l'infrastructure logicielle autour de la PlutoSDR (Travis Collins, Analog Devices), avec des concepts plus avancés tels que la rédaction de blocs dédiés à

Les démonstrations ont été l'occasion de mettre en pratique un certain nombre de concepts développés au cours des exposés. Des extensions aux USRPs de Ettus Research pour permettre une liaison micro-onde (60 GHz) ou infrarouge ont été démontrées, ainsi que l'analyse du protocole de communication du radiocommande de modélisme. Nous avons pour notre part présenté une expérience pédagogique d'utilisation d'un oscilloscope numérique radiofréquence comme source de données pour la datation précise de temps de vol (Fig. 4) dans un contexte de mesure RADAR (les 5 GHz de bande passante d'un oscilloscope Rohde & Schwarz RTE1054 permettent une résolution de 3 cm même si sa bande passante de 500 MHz nous limite à 30 cm en théorie) avec le gros avantage d'acquérir simultanément 4 voies, une oppor-

un traitement particulier sur le processeur respectant l'API GNU Radio (Tanguy Risset) ou une analyse complète de la liaison radiofréquence mise en œuvre par les Nordic NRF24x et NRF52x par Hervé Boeglen. Cette dernière présentation semble avoir achevé l'auditoire au vu des retours par questionnaire, et mérite un développement approfondi dans ces pages tant la diversité des compétences abordées est vaste : en attendant, le lecteur est encouragé à reprendre le tutorial à tête reposée en s'appuyant sur les documents [6] et la vidéo du tutorial pour avancer à son rythme. Il s'agit là d'un point récurrent dans toutes les présentations avancées exploitant GNU Radio : l'environnement de développement libre facilite le prototypage rapide et la mise en œuvre d'algorithmes sur des signaux synthétiques ou réels, mais ne retire rien aux subtilités du traitement numérique de signaux échantillonnés en temps discret, aux théories de codages de signaux transmis sur canal bruité, ou stratégie pour retrouver ces informations après corruption. GNU Radio rend juste ces concepts beaucoup plus ludiques et pratiques qu'un cours formel au tableau.

Finalement, la dernière journée de tutoriels s'est focalisée sur le co-design CPU/FPGA sur PlutoSDR dans lequel l'environnement OscImpDigital permet d'ajouter ses propres fonctionnalités au bitstream de configuration du FPGA du Zynq (*Programmable Logic* – PL) afin de prétraiter les informations avant de les fournir au processeur généraliste (*Processing System* – PS), de bande passante inférieure mais supportant des niveaux d'abstraction supérieurs, sans prétention de calcul en *pipeline* puisque tous les blocs sont munis de tampons en entrée et sortie tel que le requiert l'ordonnanceur GNU Radio. Le premier exemple a porté sur un transfert PL vers PS – une tâche ardue si abordée au niveau du VHDL – en fournissant le flux de données issu d'un oscillateur numérique (NCO) vers la RAM liant PL et PS. Ce même NCO est ensuite mélangé au flux de données issues du récepteur radiofréquence AD9363 de la PlutoSDR pour une transposition de fréquence. Alors que cet exemple peut sembler sans intérêt pratique puisque l'AD9363 possède déjà son oscillateur matériel pour transposer de la bande radiofréquence à la bande de base, le NCO a le bon goût de ne pas nécessiter de temps de reconfiguration vers une nouvelle fréquence, contrairement à l'oscillateur matériel. Ainsi, les sauts de fréquences sont aussi rapides que le temps de reprogrammation du registre de comparateur de l'accumulateur du NCO. Notre ambition était de démontrer toute une chaîne d'acquisition du signal GPS avec la PlutoSDR – nécessitant notamment le balayage du NCO pour compenser le décalage Doppler de chaque satellite alors qu'il avance sur son orbite – mais cette finalité était juste trop ambitieuse pour les 7 heures de travaux pratiques. Bénéfice de l'open-source : moins d'une semaine après le tutorial, les premiers rapports d'erreurs arrivent déjà.



FIGURE 5 – Tutoriel d'analyse du spectre radiofréquence dans une salle informatique de l'ENSMCM qui n'avait pas vu autant de pingouins depuis longtemps.

5 Conclusion

La radio logicielle est un sujet en vogue – malgré la compétition du *deep learning* et autres algorithmes d'intelligence artificielle – et l'absence de conférence dédiée à l'environnement libre de développement GNU Radio a été comblée pour la seconde fois par les GNU Radio Days. Une différence fondamentale entre ra-

dio logicielle et deep learning ou traitement d'images sur GPU – le premier travail sur des flux de données continus en flux tendu, alors que le deuxième et troisième se contentent de remplir une mémoire pour un traitement aussi rapide que possible et rendre le résultat dans une mémoire (FIFO en entrée et sortie) – n'est pas anodine quant à la disponibilité des outils de développements. Nos tentatives d'utiliser les niveaux s'abstractions tels que High Level Synthesis (HLS) ou Matlab HDL Coder ont permis d'aborder les dernières applications mais pas la première. Nous tentons de pallier cette déficience en fournissant l'environnement OscImpDigital, mais l'avenir des outils de synthèse propriétaires des fondeurs de FPGA risquent de ne pas nous laisser cette liberté longtemps. L'avènement des Zynq avec interfaces RF (RFSoc) renversera peut être, au moins partiellement la tendance.

La conférence que nous proposons n'a aucune prétention de valoriser sa prestation ou ajouter une ligne de plus sans intérêt à son curriculum-vitae : nous désirons promouvoir l'échange entre passionnés du traitement du signal désireux de mettre en pratique leur connaissance sur des cas concrets. Au lieu de faire payer une somme faramineuse comme une société qui se veut savante mais est avant tout lucrative avec un acronyme en quatre lettres commençant par I et finissant par E, nous nous sommes proposés de mettre en valeur la cohésion et les échanges entre participants par une soirée conviviale de barbecue qui a promu les échanges et discussions entre interlocuteurs. Cette option fut un succès auquel le brasseur local CUC ne fut pas étranger (Fig. 7). Une nouvelle édition de la conférence sera réitérée l'année prochaine pour en discuter, en espérant un nombre croissant de participants pour démontrer le dynamisme des sujets liés à la radio logicielle en Europe. Cette prochaine session sera localisée à **Poitiers ou Limoges** (laboratoire XLIM).



Figure 6: Marcus Müller, architecte de GNU Radio et contributeur prolifique, voir encyclopédique, à la liste de diffusion *discuss-gnuradio*, présente les subtilités de son ordonnanceur.



FIGURE 7 – Soirée conviviale autour d'un barbecue arrosée par le brasseur CUC.

Remerciements

Olivier Testault (association Sequanux à Besançon) a dessiné le logo. Olivier Boeglen (Université de Haute Alsace) a dessiné l'affiche de la conférence. Les services technique, informatiques et multimedia (STAMP) de l'ENSM ont soutenu la manifestation par la mise en œuvre des ressources nécessaires (pilotes PlutoSDR, GNU Radio sur MS-Windows et VirtualBox avec autorisation d'accès au matériel, infrastructures et enregistrements video/audio) et la mise en forme et diffusion des vidéos des conférences disponibles à [7, 8, 9, 10]. La conférence a été soutenue financièrement par Rohde & Schwarz, Oposom (Armadeus Systems), le GDR RSD, le labex FIRST-TF, et la Ville de Besançon. Les cadeaux aux participants ont été offerts par Analog Devices.

Références

- [1] gnuradio-fr-19.sciencesconf.org/ et en particulier le programme fournit le lien vers les transparents projetés en support de chaque présentation
- [2] H. Boeglen & L. Mura, *Les bases des communications numériques avec GNU Radio*, GNU/Linux Magazine **225** (2019)
- [3] Annonce du gel de la version 3.8 de GNU Radio après plus de 6 années d'évolutions de 3.7 : www.mail-archive.com/discuss-gnuradio@gnu.org/msg69033.html
- [4] M. Müller, *GNU Radio in 2019 : Facts and Plans*, FOSDEM 2019 à fosdem.org/2019/schedule/event/gnuradio/
- [5] J.-M Friedt, *RADAR passif par intercorrélation de signaux acquis par deux récepteurs de télévision numérique terrestre*, GNU/Linux Magazine France **212** pp.36- (2018)
- [6] Documents de support du tutoriel à herve.boeglen.free.fr/actualites/eur_gr_days_2019.htm
- [7] Marcus Müller présente le fonctionnement interne de GNU Radio et de son ordonnanceur à mediacenter.ens2m.fr/videos/?video=MEDIA190710141050023
- [8] Première demie-journée de conférences à mediacenter.ens2m.fr/videos/?video=MEDIA190710142940080
- [9] Deuxième demie-journée de conférences à mediacenter.ens2m.fr/videos/?video=MEDIA190710152118683
- [10] Troisième demie-journée de tutoriels (Leo Cardoso & Mathieu Cunche puis Tanguy Risset) à mediacenter.ens2m.fr/videos/?video=MEDIA190710161134144