



ANR, Appel à projets Générique (AAPG 2019)

QCSP Project (ANR-19-CE25-0013-02)

Deliverable D3.1

Demonstrator set-up.

Editor:	Roland Christian
Deliverable Nature:	Internal (scope: Consortium and ANR)
Due Date:	??
Delivery Date:	Monday, September 02, 2024
Version:	1.0
Total number of page:	22 pages
Keywords:	GNU Radio

Abstract: This document presents the method used to generate real QCSP frames with GNU radio. It gives the structured of the transmitted signal as well as the file structure of the measured received frame. It is also a user guideline to the use of GNU Radio hardware.



Monday, September 02, 2024

List of authors

Partner:	Authors
UBS	Christian ROLAND

History

V1.0: May the 24 2021, Organization of real signal capture files; method and analysis.

Outline

1	Excutive summary	4
2	Introduction	5
3	Présentation des matériels et logiciels	6
1	Présentation des répertoires de données	6
2	Présentation du matériel utilisé	6
1	Manipulation avec horloge commune (Câble MIMO)	6
2	Manipulation avec du matériel sans fil	6
3	Présentation des logiciels utilisés	7
1	Gnuradio-companion	7
2	Matlab	7
4	Méthode de création des signaux	8
1	Présentation de la chaîne de transmission	8
	Une ou plusieurs trames sont générées à l'aide du logiciel Matlab puis mise en fichier afin d'être émises à l'aide du logiciel GNuradio-companion (voir fichier Matlab en annexe 16).	8
2	Présentation du fichier de génération de la trame	10
5	méthode d'utilisation des signaux	10
1	Présentation du fichier de génération de la trame	10
2	Corrélation grossière de la trame	12
3	Présentation du fichier d'info	13
6	Conclusion	14
7	Bibliographie	15
8	Annexe	16

1 EXECUTIVE SUMMARY

Work Package 3 (WP3) To ensure the real advantages of combining CCSK modulation with a NBcode, this third package proposes to put into practice the work resulting from WP1 and WP2. The final objective is to obtain a demonstrator (in real time and in software) composed of several embedded systems implementing.

Deliverable 3.1: Report (private), software (private) and database of measurement (private) [M6].

The goal of **Task 3.1** is to define a common experimental process that will be used during the project. The objective is twofold: it will be used to validate the performance of the hardware prototype at the end of the project but it will also provide reference data for WP1 and WP2. The use of Software-Defined Radio (SDR) modules (USRP) for data transmission and reception will be favoured to emulate communications between IoT type modules and a base station. Each transmitter will send coded and modulated frame N times with a decreasing power level until the loss of frame detection at the decoder side. These experimental measurements will make it possible to estimate the channel in different environments.

These parameters will be re-injected into WP1 and WP2 in order to accurately evaluate the algorithmic proposals. The implementation of modulation, demodulation, coding and decoding algorithms will be carried out using off-line software. Methods and software generated during this task will be put in open access to the academic/industrial community.



Figure 1 : Gantt diagram of WP3

This deliverable is written in French. The introduction (section 1) explains the principle of offline transmission. Then section 2 presents the used hardware and the objective. Section 3 described how the transmission signal that contains QCSP frame is constructed. Section 4 described how to use the sampled data and section 5 concludes the deliverable.

Note that most of the content of this deliverable has been share with the QCSP consortium at the very beginning of the project. The formal finalization of this deliverable has been done much later than the useful work.

2 INTRODUCTION

Ce document récapitule le contenu des répertoires contenant des acquisitions de signaux associés au WP3. Dans une première partie nous présentons le matériel et les logiciels utilisés pour générer et stocker les données. Dans la seconde, nous verrons quelles données et canaux sont utilisés dans ces répertoires. Puis pour finir, les éléments pour l'utilisation des données font l'objet du dernier paragraphe.

Il est important de noter ici, que les simulations pour l'instant ne sont pas réalisées complètement en temps réel de façon à pouvoir travailler sur les données et la mise au point du système. À cet effet, le schéma de fonctionnement suivant sert de base au système.

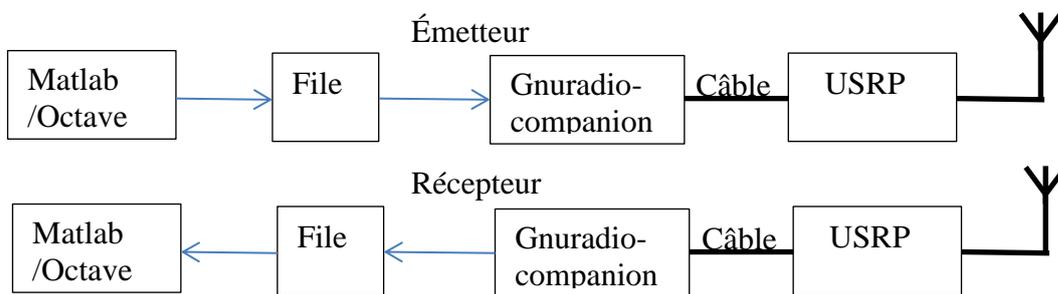


Figure 2 : Principe de fonctionnement matériel / logiciel

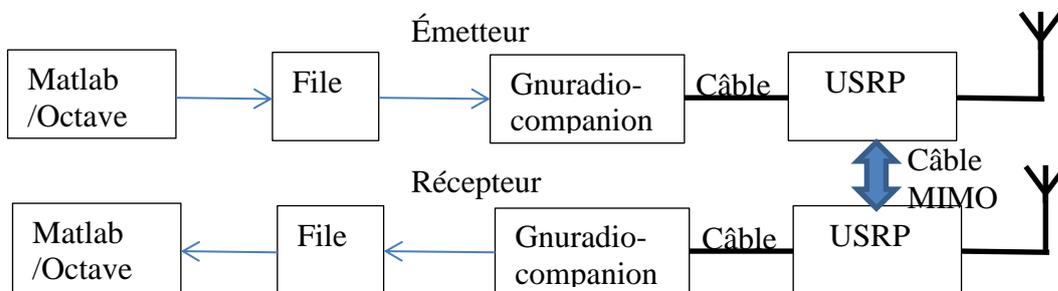


Figure 3 : Principe de fonctionnement avec synchronisation émetteur/récepteur

3 PRESENTATION DES MATERIELS ET LOGICIELS

1 PRESENTATION DES REPERTOIRES DE DONNEES

Le répertoire WP3 contient des sous répertoires de données nommés de la façon suivante : ‘Data_’ suivi d’une date au format ‘yyyymmjj’.

Dans chaque répertoire, se situe un fichier ‘info.pdf’ récapitulant les différents fichiers qui se trouvent dans le répertoire et les conditions de capture des données.

Dans chacun de ces répertoires des données ont été capturées selon les protocoles suivant les explications présentées au paragraphe 4 et 2.

2 PRESENTATION DU MATERIEL UTILISE

Le matériel utilisé pour ces manipulations se compose d’émetteur/récepteur de chez NI/Ettus. Les acquisitions réalisées auraient pu l’être avec d’autres appareils de radio-logicielle. Pour le fonctionnement de ces appareils, le site Web « qcsp-cloud.univ-ubs.fr » fournit les informations utiles.

En effet, 3 types d’appareils sont utilisés :

- NI_USRP_2920 : cet appareil fonctionne avec une carte fille radio de type WBX de chez ETTUS. La communication du PC avec l’appareil se fait via un port Ethernet.
- NI_USRP_2932 / USRP_N210(ETTUS) cet appareil fonctionne avec une carte fille radio de type SBX + une horloge GPSDO.
- NI_USRP_2900 : cet appareil fonctionne sans carte fille. La communication du PC avec l’appareil se fait via un port USB.

1 MANIPULATION AVEC HORLOGE COMMUNE (CABLE MIMO)



Figure 4 : émetteur et récepteur avec horloge MIMO

Dans cette solution l’émetteur et le récepteur ont la même horloge. Ils sont connectés via le câble ‘MIMO’. Ce câble assure une cohérence entre les deux horloges avec un léger décalage constant de 17,6 ns.

2 MANIPULATION AVEC DU MATERIEL SANS FIL

Contrairement à la manipulation précédente, les récepteurs et les émetteurs sont complètement dissociés et diverses conditions peuvent être choisies.



Figure 5 : émetteur Nb-CCSK outdoor et récepteur indoor

3 PRESENTATION DES LOGICIELS UTILISES

1 GNURADIO-COMPANION

Dans la documentation de GnuRadio se trouve les éléments utiles à la génération des chaînes de transmissions. Les fichiers de base sont des fichiers d'extensions '.GRC'.

Les blocs composants la chaîne de transmission sont en général en C++ ou en python, quelquefois en VHDL.

GnuRadio sert ici d'interface entre la partie radio et la partie de génération ou réception des signaux.

2 MATLAB

Matlab est utilisé pour générer les trames et aussi pour analyser celle-ci en réception.

4 METHODE DE CREATION DES SIGNAUX

1 PRESENTATION DE LA CHAINE DE TRANSMISSION

Une ou plusieurs trames sont générées à l'aide du logiciel Matlab puis mise en fichier afin d'être émises à l'aide du logiciel GNuradio-companion (voir fichier Matlab en annexe 17).

Un émetteur envoi des trames répétées régulièrement à la fréquence de 433,92 MHz.

L'émetteur 1 envoi le signal sur le récepteur 2.

3 cas sont possibles :

- Le canal est un câble (pas d'antenne) donc on simule un canal gaussien dans ces conditions.
- Les antennes sont des $\frac{1}{4}$ ondes en visibilité directe donc le canal peut être considéré comme un canal de Rice/Nagasami. Les acquisitions ont été réalisées soit en intérieur soit en extérieur.
- Les antennes sont des $\frac{1}{4}$ ondes sans aucune visibilité directe donc le canal peut être considéré comme un canal de Rayleigh. Les acquisitions ont été réalisées 1 antenne en intérieur et une en extérieure.
- Mêmes conditions que précédemment mais avec du mouvement sur l'antenne extérieure.

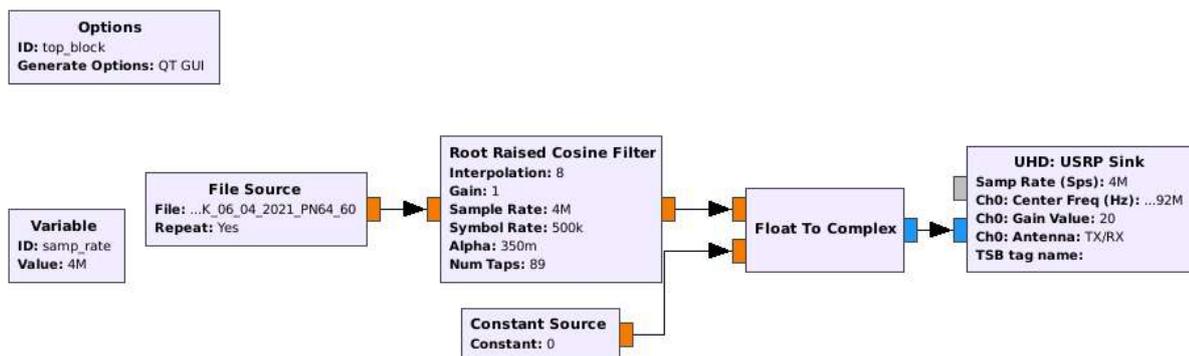


Figure 6 : schéma de l'émetteur

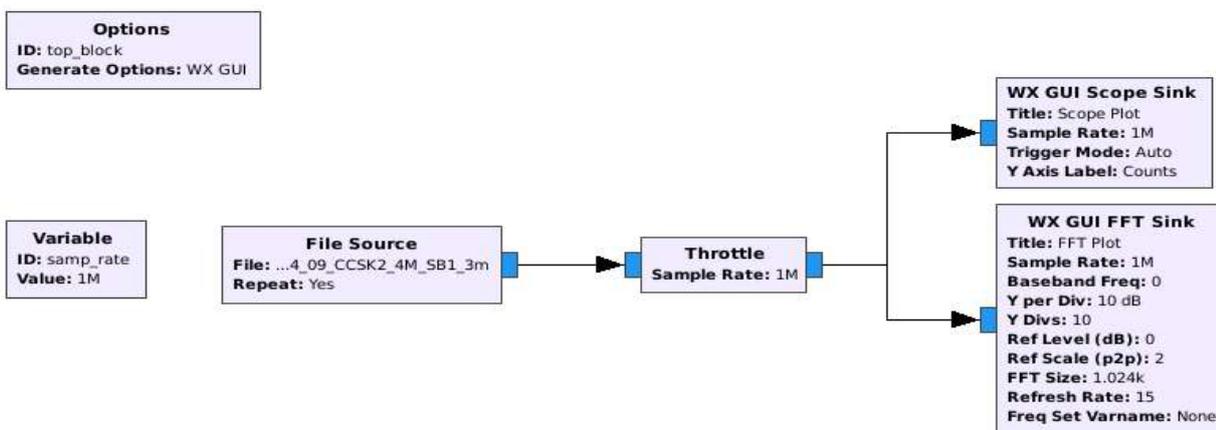


Figure 7 : lecture avec GnuRadio du fichier reçu

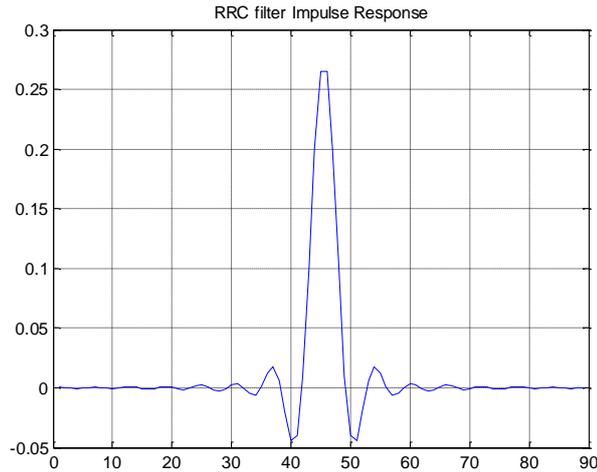


Figure 8 : Réponse impulsionnelle du filtre

Le filtre en cosinus surélevé est réalisé suivant le code GnuRadio en C++ [3] :

Le schéma de la Figure 6 montre comment à partir du fichier généré par Matlab celui-ci est récupéré par le logiciel GnuRadio-Companion pour être émis après filtrage. Le filtrage ici est en filtre en cosinus surélevé. Le nombre de coefficients est de 89, leurs valeurs sont dans l'annexe 1). De façon à être au même format « complexe » en entrée et en sortie une composante nulle en quadrature est ajoutée.

Pour le récepteur, 2 types de fichiers peuvent être générés. La solution la plus simple est un enregistrement direct des données. Pour la seconde, un filtre en cosinus surélevé est ajouté avec un décimateur par 8 ou par 4. Il est préférable de garder un sur-échantillonnage pour retrouver l'instant précis de synchronisation temporelle.

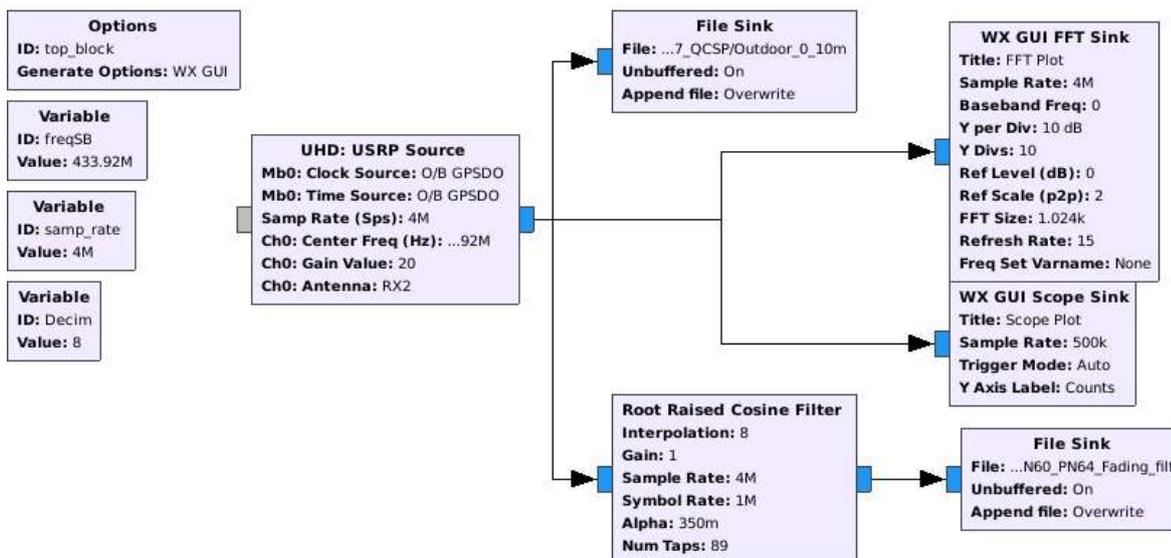


Figure 9 : schéma du récepteur

Pour le filtre de réception il est possible de l'insérer dans Octave ou Matlab grâce au code suivant :

```
Decim = 8;
samp_rate = 4e6; % Hz
sampling_freq = samp_rate;
symbol_rate = 2*samp_rate/Decim;
alpha = 0.35; % roll off
ntaps = 11*Decim+1; %

spb = sampling_freq / symbol_rate;
i = 0 : ntaps - 1;
xindx = i - ntaps / 2;
x1 = pi * xindx / spb;
x2 = 4 * alpha * xindx / spb;
x3 = x2 .* x2 - 1;

num = cos((1 + alpha) * x1) + ...
      sin((1 - alpha) * x1) ./ (4 * alpha * xindx / spb);
den = x3 * pi;
taps = 4 * alpha * num ./ den;
if ntaps/2 == round(ntaps/2)
    num(ntaps/2) = cos((1 + alpha) * x1) + (1 - alpha) * pi / (4 * alpha);
end

taps = taps / sum(taps);
```

2 PRESENTATION DU FICHIER DE GENERATION DE LA TRAME

Le fichier contenant un exemple de génération de trame se situe en Annexe.

Les lignes générant par exemple le fichier « Float32_GenereCCSK_06_05_2020_PN64_60 » sont les suivantes :

```
NomFich = ['Float32_GenereCCSK_' datestr(now,'dd_mm_yyyy') '_' Typefile num2str(Taille) '_'
num2str(LgTrame)];
fid = fopen(NomFich,'w');
fwrite(fid, seq, 'float32');
fclose(fid);
```

Les data sont enregistrées en flottant 32 bits avec une séquence PN de 64 symboles. La trame est de 60 éléments CCSK.

La variable : « *CodewordK60* » contient les séquences de 60 éléments permettant de trouver les symboles CCSK.

Une trame simple peut-être envoyée régulièrement avec le même *CodewordK* ou avec les 10 différents *CodewordK*. L'autre cas possible est l'envoi d'une super trame contenant la même séquence avec des puissances différentes. Celles-ci sont alors émises avec une décroissance régulière de -2,5 dB.

5 METHODE D'UTILISATION DES SIGNAUX

1 PRESENTATION DU FICHIER DE GENERATION DE LA TRAME

Ces signaux peuvent lus où rejouer avec les logiciels Matlab, Octave ou GNuradioCompanion. En fait les fichiers sont des documents binaires en flottant 32 bits dont les données complexes sont entrelacées de la façon suivante : partie réelle, partie imaginaire ...

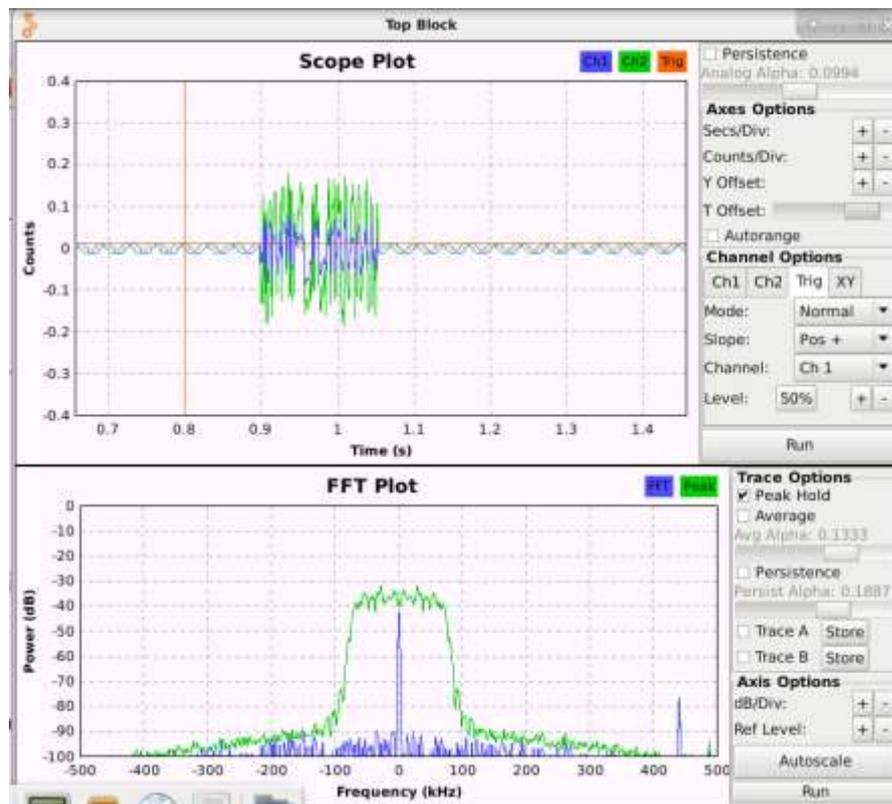


Figure 10 : Nb-CCSK scope and spectral analyser avec GnuRadio-Companion

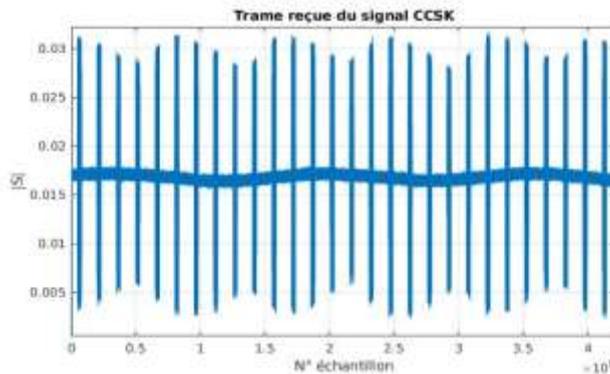


Figure 11 : Nb-CCSK scope avec Matlab

La longueur des trames, l'amplitude de celles-ci sont différentes suivant les répertoires. La durée temporelle d'une trame est de :

$$T_e * N_{b_{symb}} * T_{symb} * SurEch = \frac{1}{4 \times 10^6} 60 \times 64 \times 8 = \frac{30720}{4 \times 10^6} = 7,68ms$$

Son rapport signal à bruit vaut ici environ : 50 dB (lecture directe sur la figure)

2 CORRELATION GROSSIERE DE LA TRAME

Lors de synchronisation avec des appareils de qualité médiocre (par exemple : avec une émission d'un USRP 2900). Un décalage existe alors entre l'émetteur et le récepteur. Il est intéressant de pouvoir réaliser une corrélation grossière sur l'ensemble de la trame pour trouver la position précise et le décalage de fréquence.

$$Rx_D = \left| \sum_{k=1}^N x(k) \times ref(k) \times e^{-i2\pi f_D k / F_s} \right|$$

Soit le code Matlab suivant :

```
n=0;%compteur Df
for fD=-3000:100:3000 %boucle de-3 à 3kHz par pas de 100Hz
    n=n+1;
    for k=1:2*Nt+1 %boucle de décalage temporel
        RxD(k,n) = abs(sum(x(k:k+Nc-1) .* exp(-1i*2*pi*fD*(1/Fs)) .* conj(ref')));
    end
end
```

Nc : taille de la trame suréchantillonnée

Sig : signal reçu suréchantillonné

PM_ups : trame émise suréchantillonnée et filtrée

Fs : Fréquence d'échantillonnage



3 PRESENTATION DU FICHIER D'INFO

Le fichier d'info permet de comprendre comment a été réalisée la transmission.

Exemple de fichier « info.pdf »

%%%%%%%%%%%%
Date : le 06 mai 2020 Liste des fichiers contenus dans le fichier « 2020_05_06.zip » :

- « 2020_05_06_433_SB_N60_PN64_Gauss »
- « 2020_05_06_433_SB_N60_PN64_Rice »
- « 2020_05_06_433_SB_N60_PN64_Rayleigh »
- « 2020_05_06_433_SB_N60_PN64_Fading »
- « 2020_05_06_433_SB_N60_ZC64_Gauss »
- « EmReSBMS_CCSK.grc » : fichier pour la génération avec Gnuradio-companion.
- « 2020_05_06_433_MS_N60_PN64 » : fichier émis par gnuradio avec un filtrage en Racine de Cosinus surélevé et sur-échantillonné par 8.
- « Genere_CCSK_2020_04.m » : fichier servant à la génération du fichier à émettre.
- « Float32_GenereCCSK_06_05_2020_PN64_60 » le fichier généré par Matlab
- « Decode.m » le fichier d'Emmanuel pour tester le décodage.

SB correspond au récepteur et MS à l'émetteur.

N60 à la longueur de la trame.

PN64 pour une séquence PN de 64 chips.

433 Mhz est la fréquence de transmission.

Gauss pour une transmission par câble.

Rice pour une transmission par antenne en Indoor avec une visibilité des antennes.

Rayleigh pour une transmission avec une antenne en Indoor et une autre en Outdoor.

Fading pour une transmission avec au début et à la fin les deux antennes en Indoor puis du mouvement avec sortie d'une des antennes en Outdoor.

Toutes les captures sont faites avec une synchronisation câblée des horloges des postes émetteurs/récepteurs.

%%%%%%%%%%%%



6 CONCLUSION



7 BIBLIOGRAPHIE

- [1] https://wiki.gnuradio.org/index.php/Root_Raised_Cosine_Filter
- [2] [http://kb.ettus.com/Synchronization and MIMO Capability with USRP Devices](http://kb.ettus.com/Synchronization_and_MIMO_Capability_with_USRP_Devices)
- [3] <https://github.com/gnuradio/gnuradio/blob/master/gr-filter/lib/firdes.cc>



8 ANNEXE

1) Liste des 89 coefficients du filtre :

{ 0,000240037391680428 ; 0,000133531150159659 ; -0,000124160279666194 ;
-0,000278428063356302 ; -0,000159917805895280 ; 0,000127652675891689 ;
0,000291945033348371 ; 0,000146141226069830 ; -0,000179409586975112 ;
-0,000343160717735489 ; -0,000137791704896632 ; 0,000267674903625302 ;
0,000461300456709111 ; 0,000200471906413622 ; -0,000304000535914489 ;
-0,000553381818709779 ; -0,000258080704419451 ; 0,000322797718791935 ;
0,000587045036669409 ; 0,000197416895308131 ; -0,000504923711303198 ;
-0,000761629412321887 ; -0,000162788363878417 ; 0,000827191405091155 ;
0,00120051247190656 ; 0,000399149982801001 ; -0,000977118215373021 ;
-0,00157734660219583 ; -0,000620188830936060 ; 0,00110415656446183 ;
0,00172718734184819 ; 0,000143109918812845 ; -0,00244268371151581 ;
-0,00304793147954196 ; 0,000356135720699488 ; 0,00606770960076603 ;
0,00872257786962423 ; 0,00321253909970504 ; -0,00991444462618543 ;
-0,0220302900868710 ; -0,0200723635842911 ; 0,00499168235301790 ;
0,0503253221970904 ; 0,100019315036903 ; 0,132497454580427 ;
0,132497454580427 ; 0,100019315036903 ; 0,0503253221970904 ;
0,00499168235301790 ; -0,0200723635842911 ; -0,0220302900868710 ;
-0,00991444462618543 ; 0,00321253909970504 ; 0,00872257786962423 ;
0,00606770960076603 ; 0,000356135720699488 ; -0,00304793147954196 ;
-0,00244268371151581 ; 0,000143109918812845 ; 0,00172718734184819 ;
0,00110415656446183 ; -0,000620188830936060 ; -0,00157734660219583 ;
-0,000977118215373021 ; 0,000399149982801001 ; 0,00120051247190656 ;
0,000827191405091155 ; -0,000162788363878417 ; -0,000761629412321887 ;
-0,000504923711303198 ; 0,000197416895308131 ; 0,000587045036669409 ;
0,000322797718791935 ; -0,000258080704419451 ; -0,000553381818709779 ;
-0,000304000535914489 ; 0,000200471906413622 ; 0,000461300456709111 ;
0,000267674903625302 ; -0,000137791704896632 ; -0,000343160717735489 ;
-0,000179409586975112 ; 0,000146141226069830 ; 0,000291945033348371 ;
0,000127652675891689 ; -0,000159917805895280 ; -0,000278428063356302 ;
-0,000124160279666194 ; 0,000133531150159659 }



2) Exemple de fichier de génération

```

%%%%%%%%%%

%%=====
=====
% E. Boutillon
% 07/04/2020
% Création de forme d'onde CCSK pour test en grandeur réelle.
%%=====
=====
PN64 = [1 0 0 0 1 1 1 1 1 0 1 0 0 1 1 0 1 0 0 0 1 0 0 0 0 1 0 0 1 0 1 0 0 1 1 1 0 1 1 1 0 0 1 0 0 0 1 0 0
1 1 1 1 1 0 1 0 1 0 0 0 1 1 1];
PN = 2*PN64 - 1;

%%=====
=====
% List of 10 codewords consistent with the alist given here after in the
% end of the file.
% For example, the first check (line 138) is given as 1 0 21 15 41 41
% This mean that variable v(1), v(21) and v(41) verifies this equation
% v(1)*alpha^0 + v(2)*alpha^15 + v(41)*alpha^41 = 0.
% In the first codeword, we have (CodewordK120(1,[1 21 41]) = 31 58 11)
% Thus v(1)=alpha^30, v(2)=alpha^57 and v(3)=alpha^10.
% The first parity is: alpha^30*alpha^0 + alpha^57*alpha^15 + alpha^10*alpha^41
%           = alpha^30 + alpha^9 + alpha^51
%           =0.
%
% The method to construct the GF element is given hereafter:
% GF(1,:) = [0, 0, 0, 0, 0, 0]; % value 0
% GF(2,:) = [1, 0, 0, 0, 0, 0]; %alpha^0
%for i = 2 : 63
%           %alpha^i
%   GF(i+1,:) = mod( [0 GF(i,1:5)] + ([1 1 0 0 0 0] * GF(i,6)),2);
%end
% Thus, in this table, GF(1,:) = 0, GF(2,:) = alpha]^0
% thus the value of alpha^t is at the adress GF(2+t,:).
% we have alpha^30 = GF(30+2,:) = 1 1 0 0 1 1
%   alpha^9 = GF(9+2,:) = 0 0 0 1 1 0
%   alpha^51 = GF(51+2,:) = 1 1 0 1 0 1
%           -----
%   The sum of 3 GF values gives 0 0 0 0 0 0 (it is mod 2).
%%=====
=====

CodewordK60 = [31 28 52 25 49 58 48 53 1 26 23 37 37 20 9 17 51 8 52 55 58 2 43 33
30 62 23 6 35 14 63 8 9 62 25 58 21 44 37 40 11 0 51 8 27 56 6 26 45 30 51 52 44
24 61 44 40 60 13 13
47 1 25 43 40 60 44 54 39 60 5 23 63 30 14 18 62 34 42 46 42 31 17 3 37 26 10 39
19 44 31 35 4 24 0 45 39 36 14 8 4 35 4 33 7 52 36 31 55 10 59 24 61 42 22 62 0 1
42 49

```



```

16 28 23 38 13 44 34 19 5 17 11 23 26 61 38 14 51 49 10 26 53 44 13 28 42 46 26
54 32 55 13 56 50 25 1 17 41 19 0 27 52 1 8 23 5 37 13 44 8 38 55 62 26 3 9 4 56
43 1 27
40 16 45 47 4 37 50 39 20 55 29 55 47 8 26 63 12 14 8 3 40 12 48 5 21 58 19 5 6
32 18 24 52 29 29 19 55 20 42 50 22 18 57 16 58 6 59 31 37 39 31 21 53 54 35 18 25
5 18 23
14 46 39 56 50 52 19 26 10 45 21 44 60 15 56 39 60 5 56 30 60 43 28 16 9 51 11 63
38 13 41 61 42 44 30 30 25 37 27 11 46 13 14 46 10 15 61 0 61 55 56 53 51 25 47 3
30 20 55 35
20 8 36 37 26 51 63 22 47 5 27 11 1 0 61 60 5 44 31 26 2 20 8 4 6 3 15 8 1 2 26
62 57 61 30 6 18 19 40 12 8 9 61 8 42 31 43 39 38 35 61 30 31 49 39 41 26 38 46 54
18 57 47 59 55 37 37 63 61 51 27 18 14 47 29 29 49 41 42 38 50 38 25 38 24 31 11
11 62 20 17 49 58 35 20 21 53 47 11 63 57 35 23 10 1 48 0 59 31 60 24 9 42 13 54
53 1 30 27 61
51 57 27 13 39 38 53 3 61 50 32 53 28 61 18 28 55 61 1 51 17 56 38 24 37 58 16 55
16 38 8 43 60 44 6 33 50 54 22 9 47 20 23 9 58 20 19 41 50 54 0 24 22 57 2 7 58 14
57 63
41 24 14 15 14 47 35 42 37 62 54 23 32 37 11 58 52 29 57 34 29 63 60 61 21 26 24
63 15 24 13 0 49 27 30 55 33 15 20 11 32 51 11 12 45 61 10 11 13 4 35 11 46 58 55
7 21 29 54 6
14 30 42 33 39 14 13 20 61 22 48 29 25 46 39 24 21 37 10 28 7 56 14 57 50 0 52 17
33 26 31 37 20 56 26 44 27 39 5 39 21 59 4 7 35 29 40 50 23 20 1 63 39 20 41 10 45
32 10 7];

```

```

%=====
=====

```

```

% This sequence has been optimized to have good auto-correlation properties.
% It is used to suppress the time ambiguity at CCSK symbol level.

```

```

%=====
=====

```

```

best_60 = '100010011001011110001010101000011011111101000010010000110000' - 48;
best_60 = 2*best_60 - 1;

```

```

%=====
=====

```

```

% Generation of the frame: length 60 x 64 chips = 3840 chips.
% A mega frame is the transmission of 10 different codewords, with -2.5 dB
% of attenuation between two frames.

```

```

%=====
=====

```

```

CodewordK = CodewordK60;
SeqTime = best_60;

```

```

SpreadSeq = PN;
Typefile = 'PN';
att = 1;
seq = zeros(1,5000);

```

```

for k = 1:10 % Number of codewords.
    for i = 1:LgTrame

```




% 6 41 34 15
% 7 41 35 15
% 8 41 36 15
% 9 41 37 15
% 10 41 38 15
% 21 0 36 41
% 22 15 37 0
% 23 15 38 41
% 24 15 39 41
% 25 15 40 41
% 26 15 31 41
% 27 15 32 41
% 28 15 33 41
% 29 15 34 41
% 30 41 35 0
% 1 41 11 0
% 2 0 12 41
% 3 0 13 41
% 4 0 14 41
% 5 0 15 41
% 6 0 16 41
% 7 0 17 41
% 8 0 18 41
% 9 0 19 15
% 10 0 20 41
% 11 41 30 15
% 12 0 21 41
% 13 0 22 41
% 14 0 23 41
% 15 0 24 41
% 16 0 25 41
% 17 0 26 41
% 18 0 27 41
% 19 0 28 41
% 20 0 29 41
%
% 1 0 21 15 41 41
% 2 15 22 41 42 0
% 3 15 23 41 43 0
% 4 15 24 41 44 0
% 5 15 25 41 45 0
% 6 15 26 41 46 0
% 7 15 27 41 47 0
% 8 15 28 41 48 0
% 9 15 29 41 49 0
% 10 15 30 41 50 0
% 11 15 41 0 51 41
% 12 15 42 41 52 0
% 13 15 43 41 53 0
% 14 15 44 41 54 0
% 15 15 45 41 55 0



% 16 15 46 41 56 0
 % 17 15 47 41 57 0
 % 18 15 48 41 58 0
 % 19 41 49 15 59 0
 % 20 15 50 41 60 0
 % 1 15 31 0 52 41
 % 2 0 32 15 53 41
 % 3 0 33 15 54 41
 % 4 0 34 15 55 41
 % 5 0 35 15 56 41
 % 6 0 36 15 57 41
 % 7 0 37 15 58 41
 % 8 0 38 15 59 41
 % 9 0 39 15 60 41
 % 10 0 40 41 51 15
 % 11 0 23 15 36 41
 % 12 0 24 15 37 41
 % 13 0 25 15 38 41
 % 14 0 26 15 39 41
 % 15 41 27 15 40 0
 % 16 0 28 15 31 41
 % 17 41 29 15 32 0
 % 18 0 30 15 33 41
 % 19 15 21 0 34 41
 % 20 0 22 15 35 41
 %%%%%%%%%%

Exemple de fichier « info.pdf »

%%%%%%%%%

Date : le 06 mai 2020 Liste des fichiers contenus dans le fichier « 2020_05_06.zip » :

- « 2020_05_06_433_SB_N60_PN64_Gauss »
- « 2020_05_06_433_SB_N60_PN64_Rice »
- « 2020_05_06_433_SB_N60_PN64_Rayleigh »
- « 2020_05_06_433_SB_N60_PN64_Fading »
- « 2020_05_06_433_SB_N60_ZC64_Gauss »

- « EmReSBMS_CCSK.grc » : fichier pour la génération avec Gnuradio-companion.
- « 2020_05_06_433_MS_N60_PN64 » : fichier émis par gnuradio avec filtrage RC et sur ech par 8.
- « Genere_CCSK_2020_04.m » : fichier servant à la génération du fichier à émettre.
- « Float32_GenereCCSK_06_05_2020_PN64_60 » le fichier généré par Matlab
- « Decode.m » le fichier d’Emmanuel pour tester le décodage.

SB correspond au récepteur et MS à l’émetteur. N60 à la longueur de la trame. PN64 pour une séquence PN de 64 chips. ZC pour une zéquence Zadoff-chu.

433 Mhz est la fréquence de transmission.

Gauss pour une transmission par câble.

Rice pour une transmission par antenne en Indoor avec une visibilité des antennes.

Rayleigh pour une transmission avec une antenne en Indoor et une autre en Outdoor.



Fading pour une transmission avec au début et à la fin les deux antennes en Indoor puis du mouvement avec sortie d'une des antennes en Outdoor.

Toutes les captures sont faites avec une synchronisation câblée des horloges des postes émetteurs/récepteurs.