Objectif :

Le codage est nécessaire pour pouvoir transmettre une information numérique c'est pourquoi dans ce TP on va d'abord effectuer le codage d'une trame binaire en appliquant deux types de codage : codage à deux niveaux comme NRZ et codage à trois niveaux comme RZ et Manchester et puis visualiser l'effet de la variation de la période sur le résultat obtenue, ensuite déduire l'utilité de chaque type de codage et la différence entre eux.
De plus on va effectuer le codage inverse de chaque exemple.

Manipulation :

pour réaliser le TP on va se servir du programme de simulation sous Matlab « Simulink » et sa librairie et on va réaliser le montage de chaque code :

1) montage de codage NRZ :

❖ **1er méthode en utilisant « unipolar to bipolar converter »**

- Générer la trame binaire en utilisant le bloc « Repeating Sequence Stair (RSS »
- Le codage RZ par exemple peut s'effectuer en utilisant le bloc « Unipolar to bipolar convertor »
- Visualiser le signal par le scope
- Ajouter un échantillonneur bloqueur avec « zero order holder »
- Observer la densité spectrale par « Spectrum Analyzer »

❖ **2éme méthode en utilisant les opérations mathématiques :**

Cette méthode consiste à effectuer le codage en utilisant les opérations mathématiques selon l'équation suivante Signal_NRZ= (Signal_Binaire -0.5)×2 :

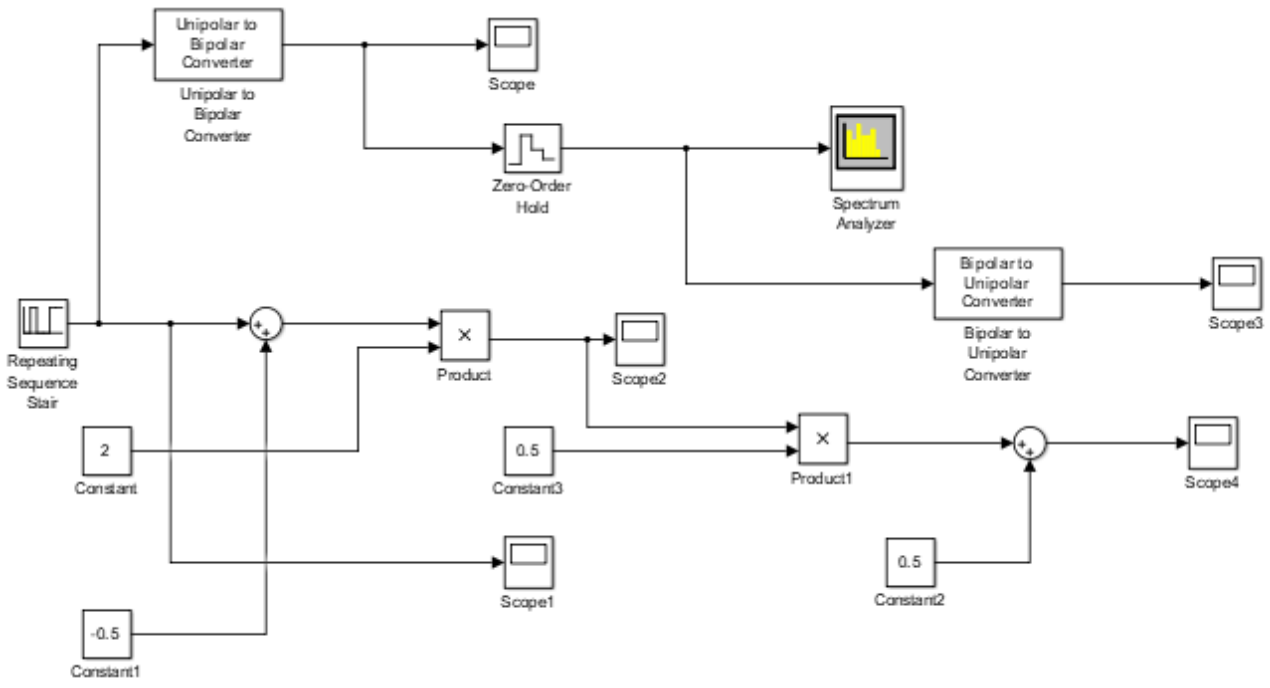


Figure 1 : montage du codage NRZ par deux méthodes et le montage inverse

❖ **Observation du résultat**

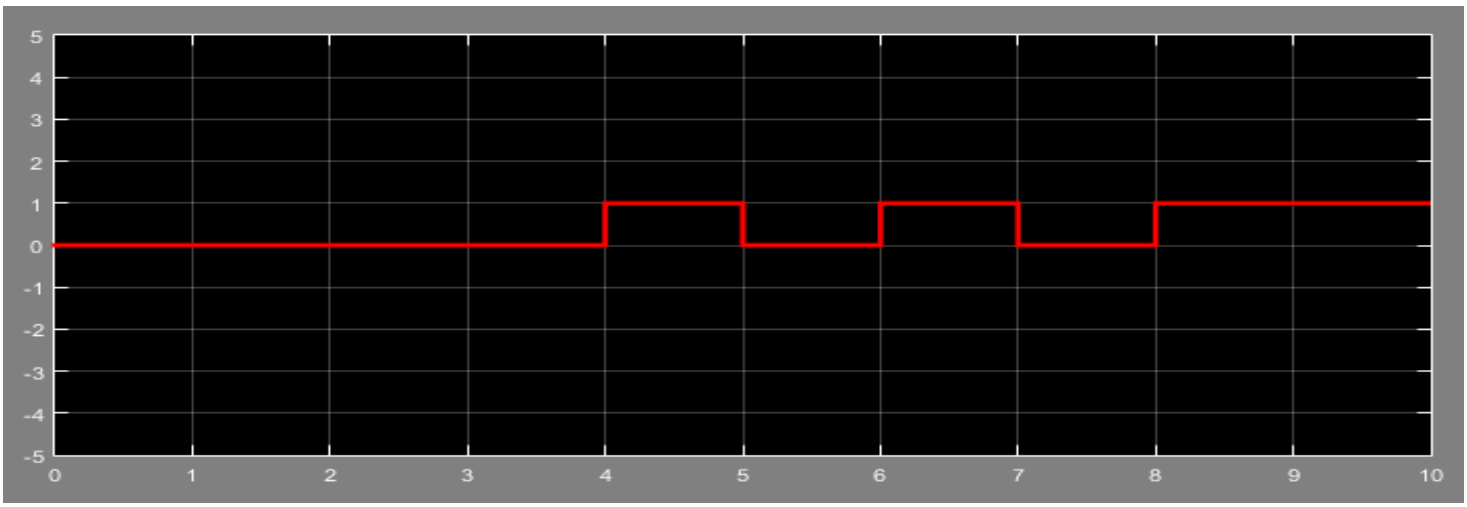


Figure 2 trame binaire 000010101110001 observé sur scope1 avec T=1

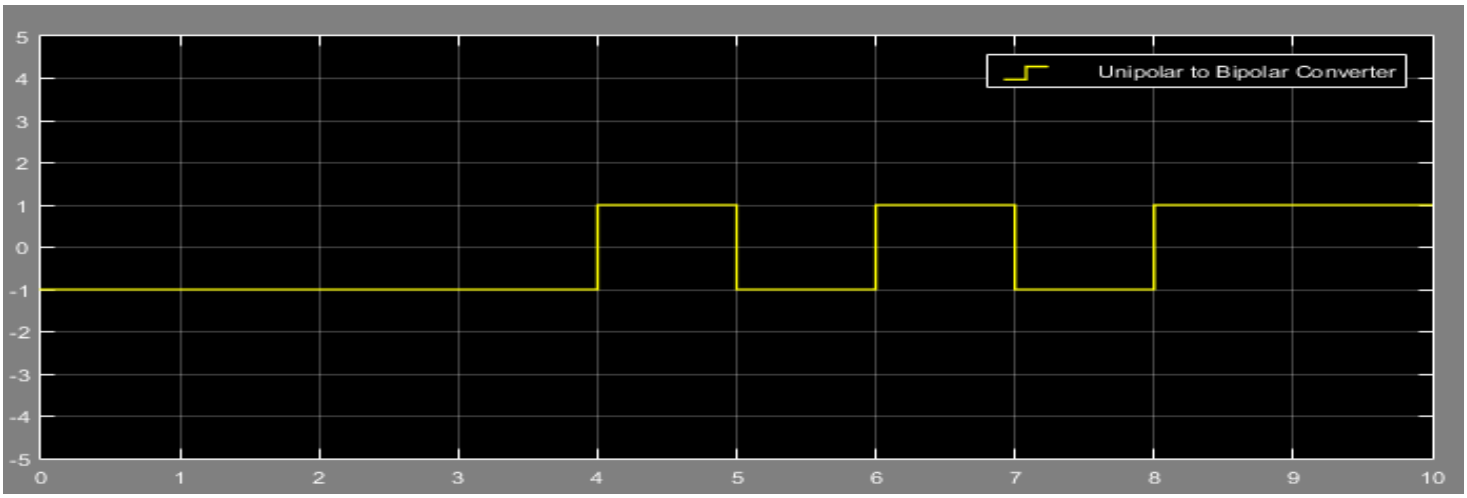


Figure 3 signal codé en NRZ observé sur Scope (méthode 1)

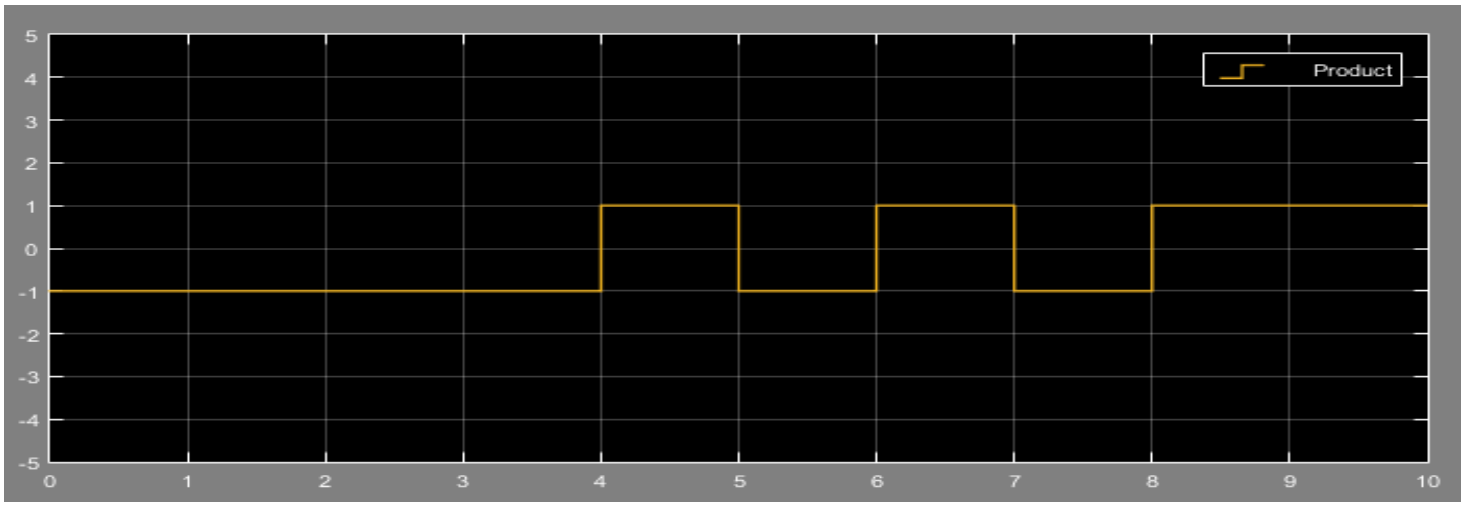


Figure 4 signal codé en NRZ observé sur scope2 (méthode 2)

Commentaire : Le codage Non Return to Zero est à deux états : le signal se trouve dans un état (par exemple à l'état haut) lorsque des 1 logiques sont transmis, et dans l'autre état (à l'état bas dans l'exemple) lorsque des 0 logiques sont transmises. Ces deux états correspondent à deux niveaux de tension symétriques par rapport à 0.

❖ **Spectre du Signal transmis :**

Après avoir ajouté un échantillonneur bloqueur de Te=0.1s, on observe la densité spectrale présenté sur un intervalle de fréquence [-5Hz , 5hz] . On observe donc le résultat ci-dessous :

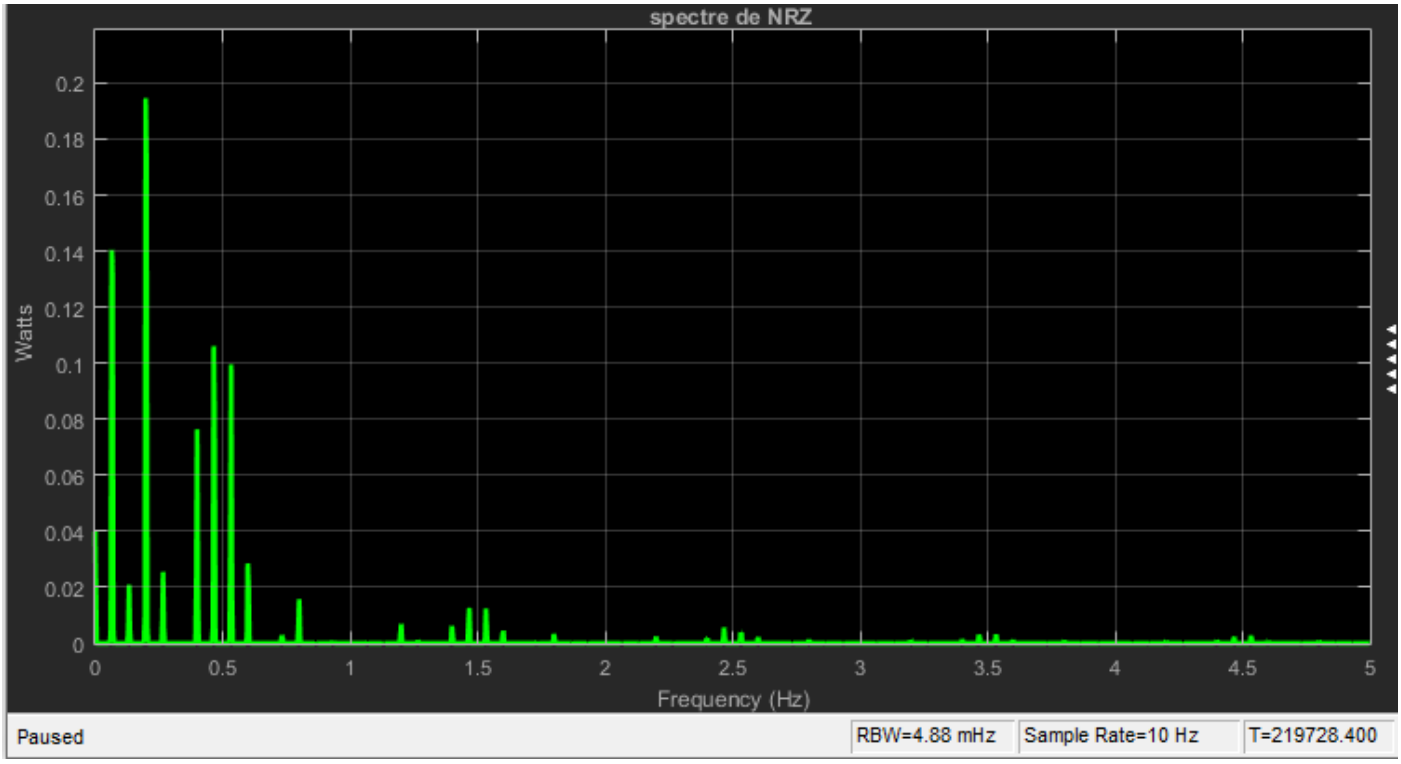


Figure 5 densité spectrale du signal codé NRZ

Commentaire et interprétation : L'allure de La densité spectrale est en forme de sinus cardinal avec 90% de la puissance est situé dans la bande de fréquence [0, 1/T] et qui s'amortie dans les raies suivantes. On remarque aussi que le spectre s'annule pour les valeurs de $\frac{1}{T}, \frac{2}{T} \ldots \frac{k}{T}$

Le fait que le spectre soit nul pour $f = k/T$ rend la synchronisation difficile car il n'y a pas de composante rythmée à la cadence d'émission binaire

❖ **Variation de la période T** :

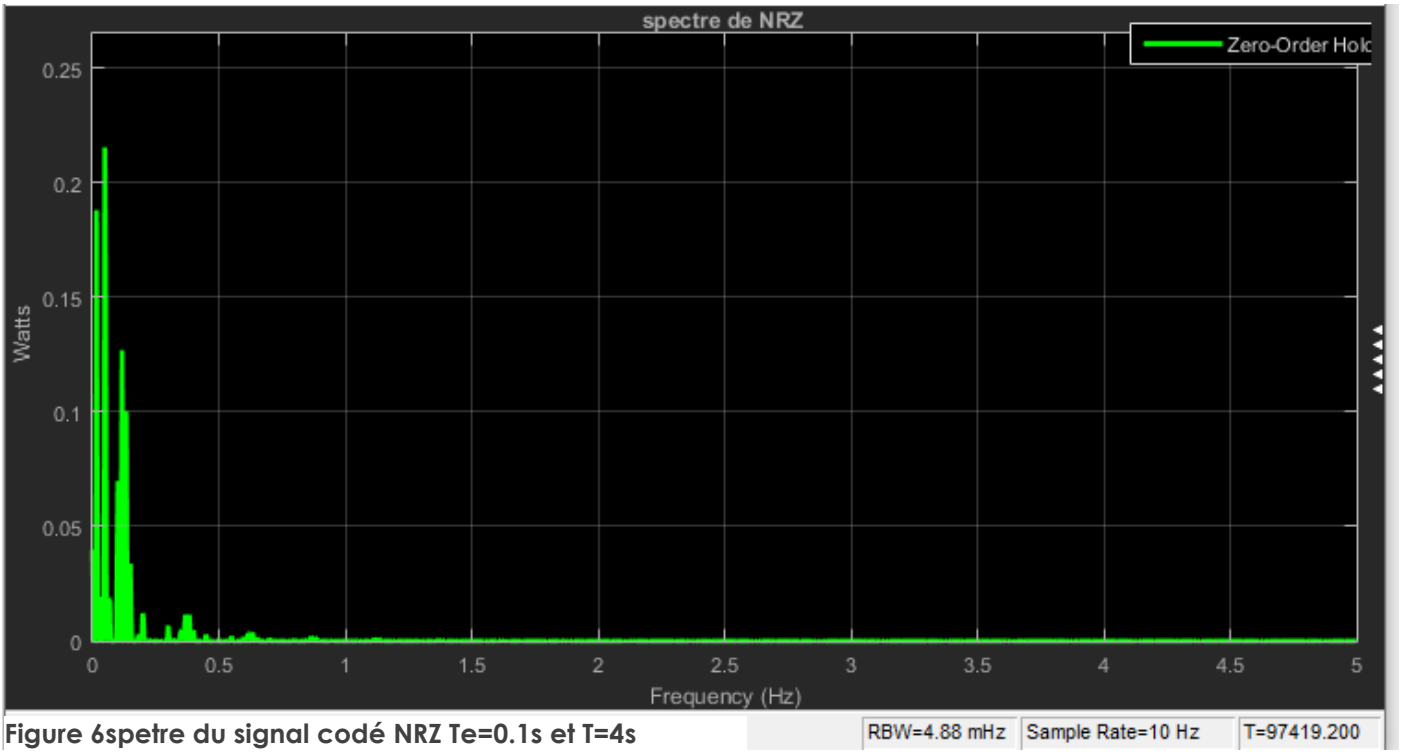


Figure 6spetre du signal codé NRZ Te=0.1s et T=4s

Commentaire et intrépretation ; En variant la période de la trame binaire de 1s à 4s on constate que le spectre est concentré sur l'axe des Y, la première raie est étroite et perte des harmoniques amorties et donc manque d'information transmis

❖ **Montage inverse :**

1ére méthode : on réalise le montage inverse de cette méthode en reliant la sortie de « zero holder order » avec le bloc « Bipolar ti unipolar »

2éme méthode : on réalise le montage selon l'équation mathématique suivante :

signal_binaire= (signal_NRZ×0.5)+0.5

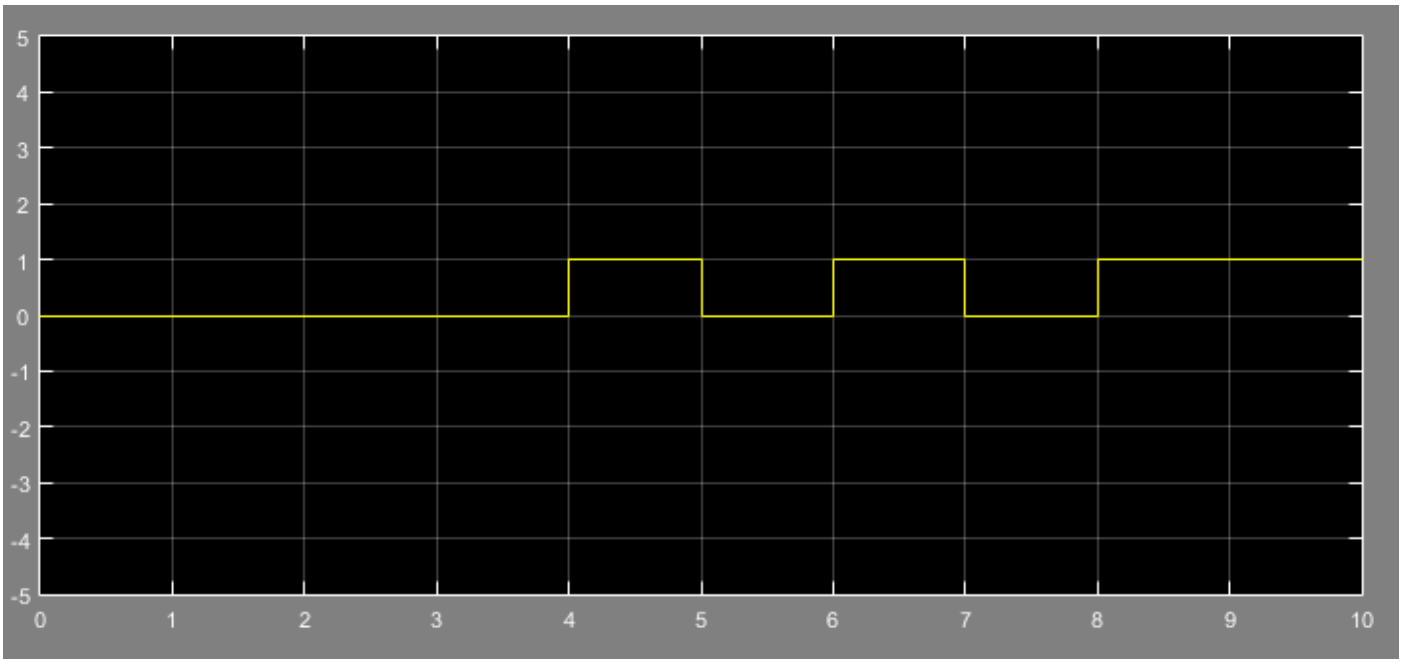


Figure 7 résultat du montage inverse observé sur Scope3

On obtient la même trame binaire résultante du RSS

2) Montage du codage RZ :

Ce code transforme :

- Les 0 en un signal qui vaut -av pendant T/2 puis 0 pendant le reste de la période.
- Les 1 en une transition de +av pendant T/2 vers 0 pendant les T/2 qui reste.

On réalise le montage indiqué dans la fiche de TP :

- Générer la trame binaire « 000010101110001 » en utilisant le bloc « Repeating Sequence Stair (RSS »
- Générer le signal Horloge en utilisant le bloc « Pulse Generator » pour obtenir un signal de 1 sur[0, T/2] et 0 pour[T/2,T]
- Signal_RZ= [(signal_binaire-1)× signal_horloge]+ (signal_horloge× signal_binaire)

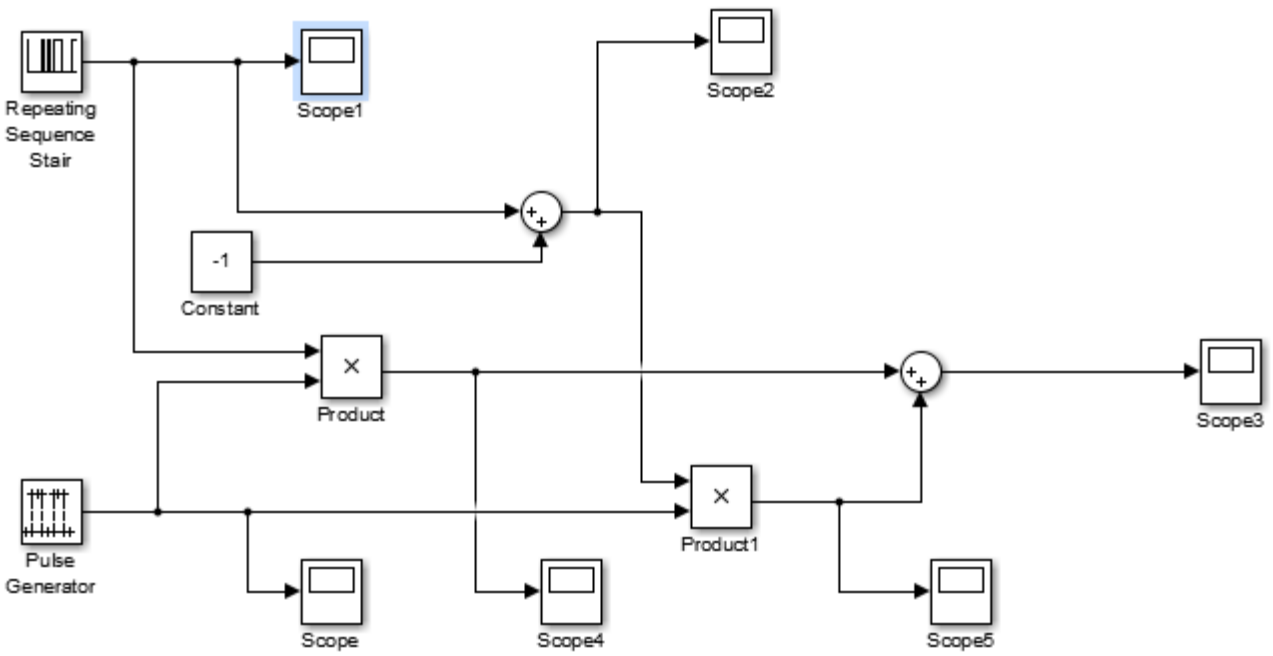


Figure 8 montage RZ

❖ **Observation du résultat :**

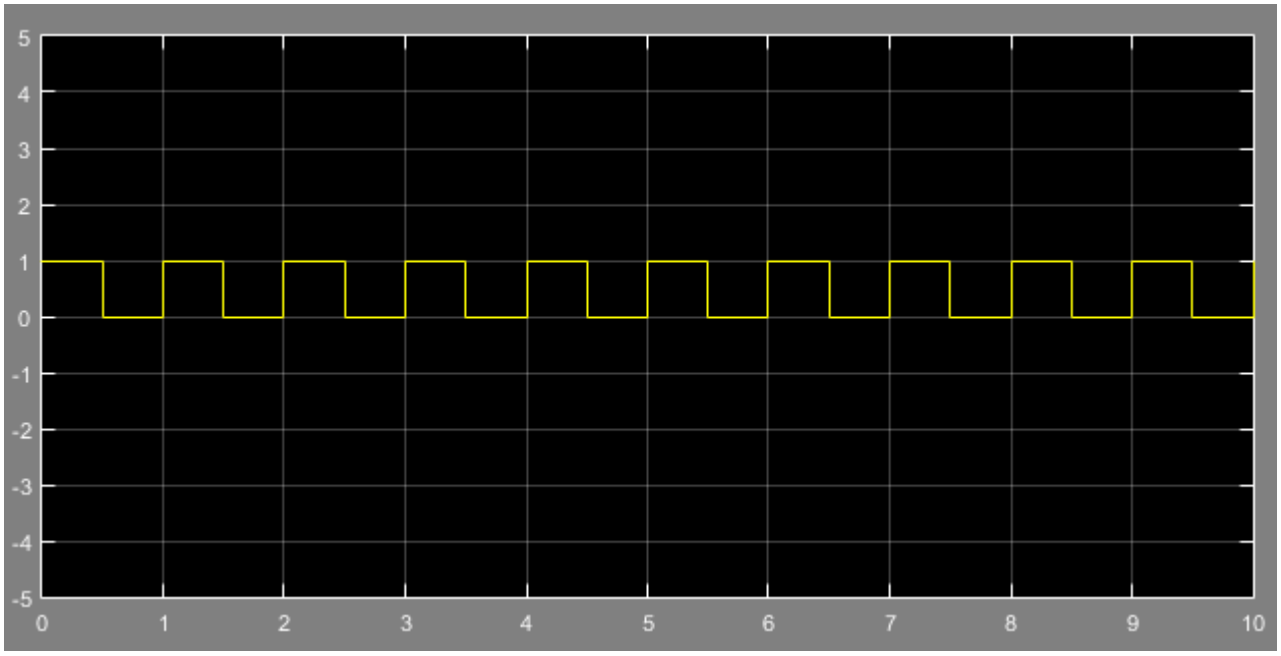


Figure 9 signal Horloge issue de pulse generator

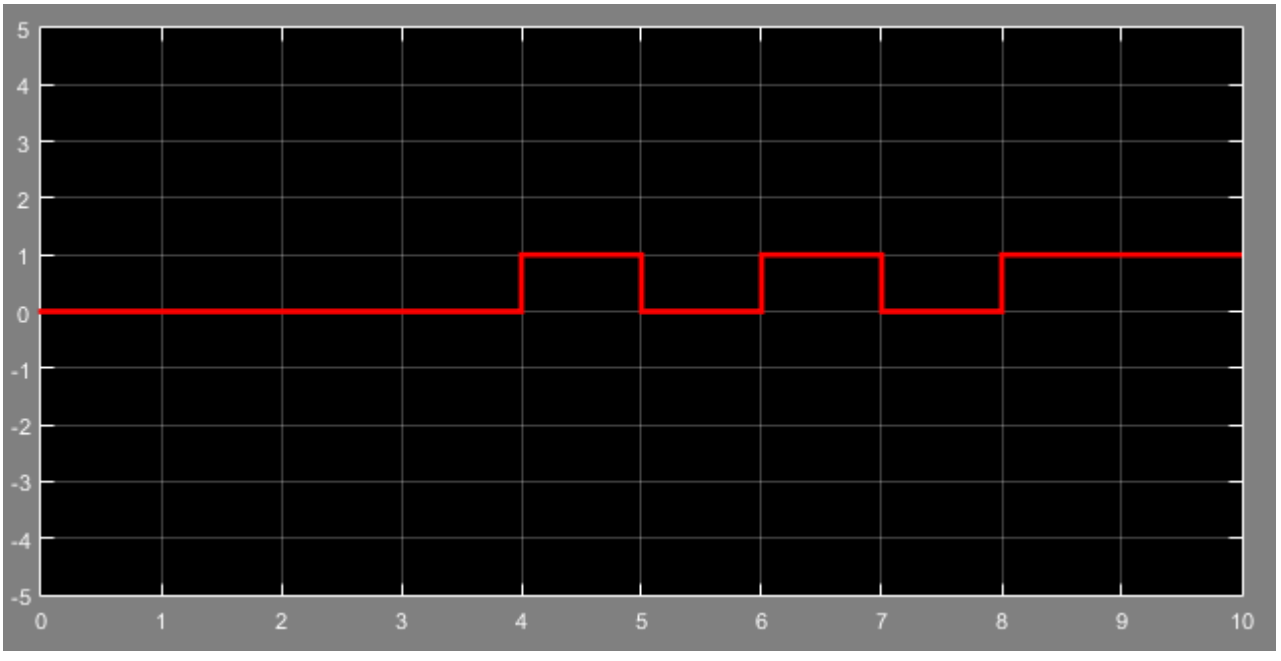


Figure 10 la trame binaire

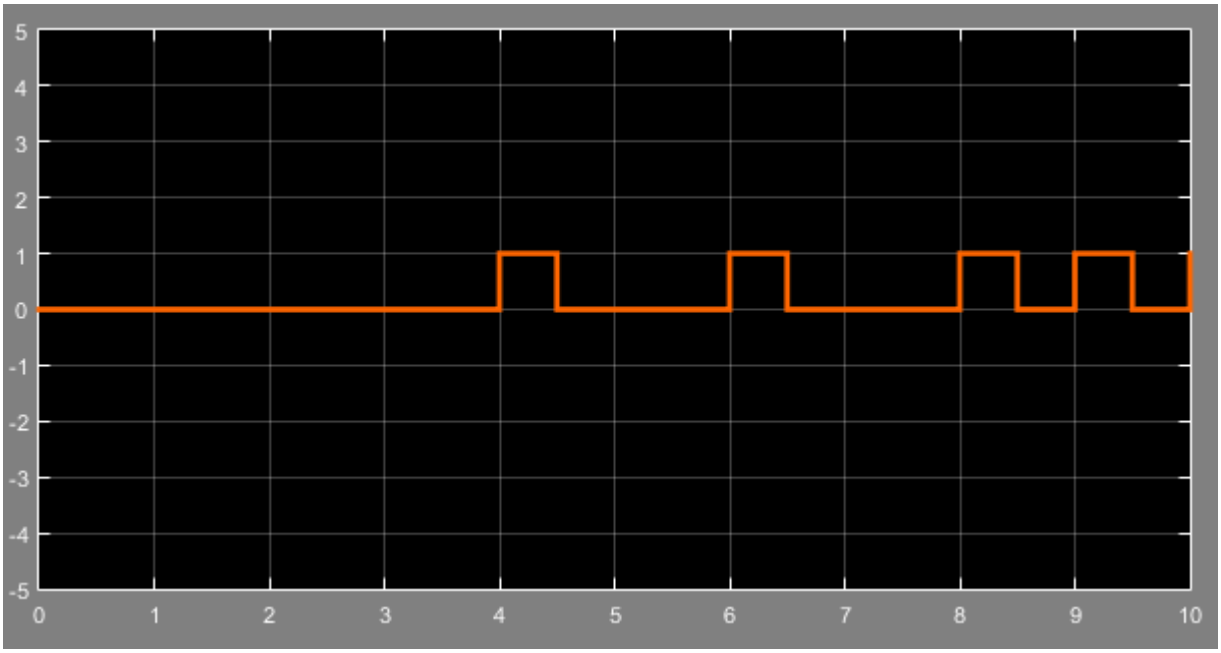


Figure 11 : (signal_horloge× signal_binaire) visualisé sur SCOPE 4

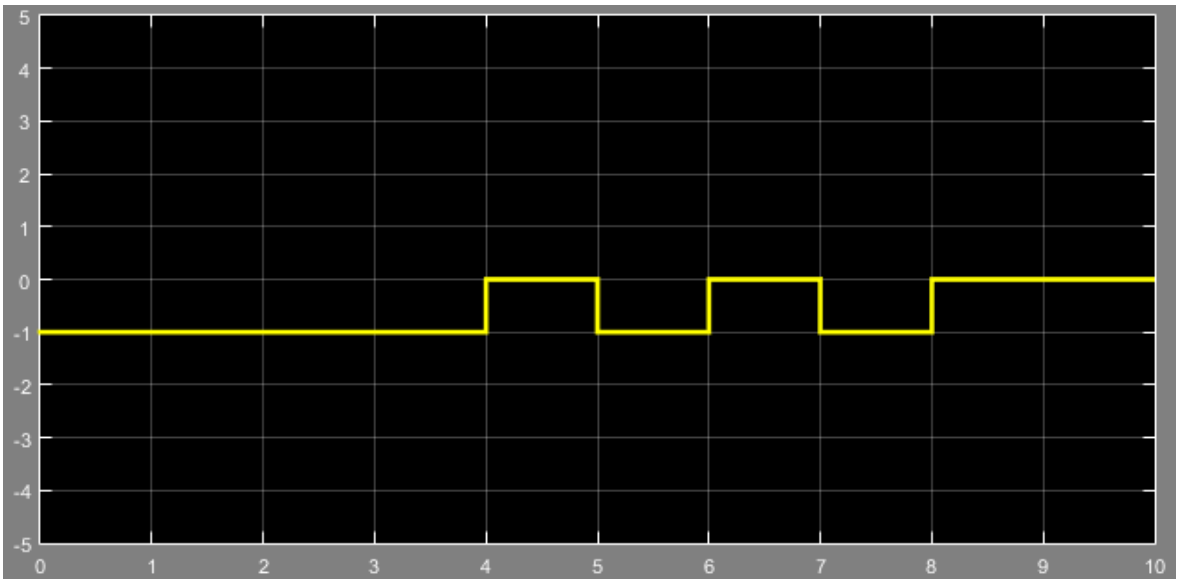


Figure 12 Scope 2 : signal_binaire-1

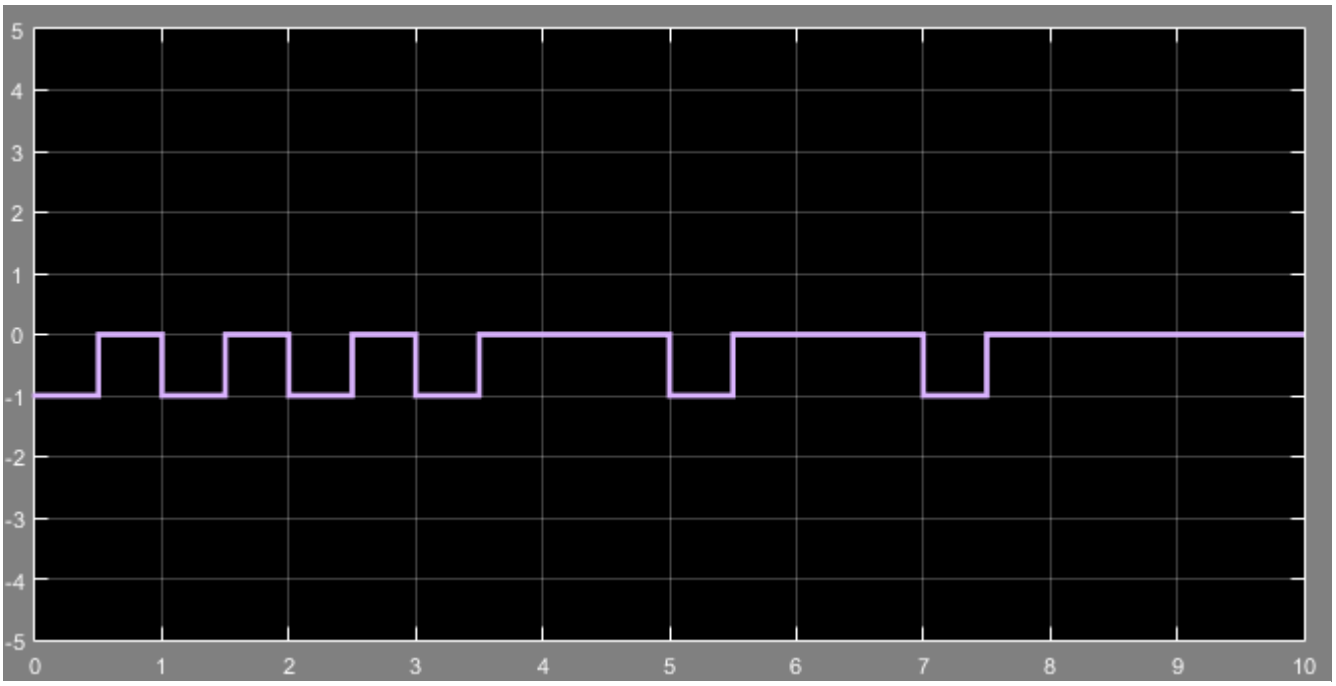


Figure 14: [(signal_binaire-1)× signal_horloge] visualisé sur SCOPE5

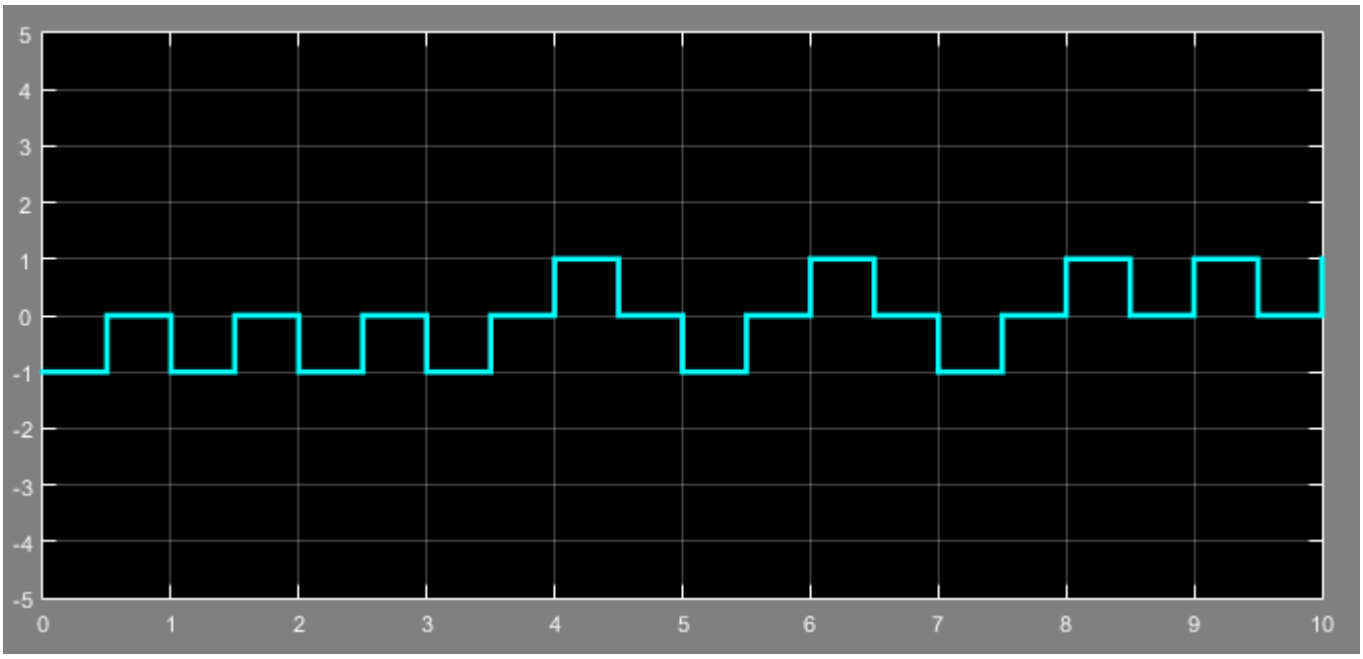


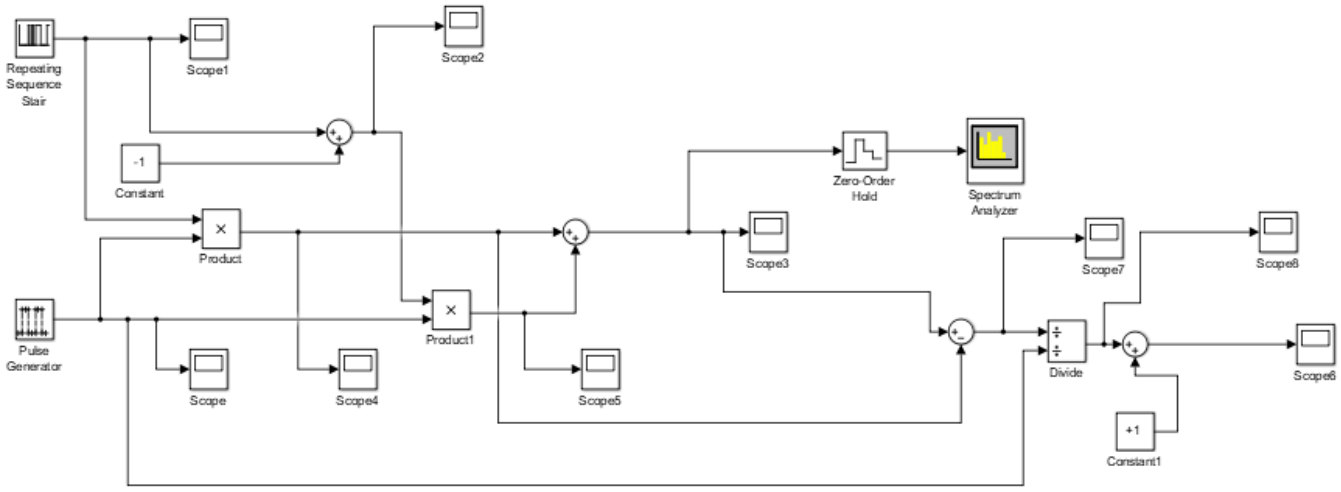
Figure 15: Signal_RZ= [[(signal_binaire-1)× signal_horloge]+ (signal_horloge× signal_binaire] visualisé sur SCOPE3

On constate que Le signal codé en RZ prend trois valeurs -1,0,1 : c'est un codage à 3 niveaux

Avantage de RZ par rapport au NRZ : la perte du signal est moins que celle en NRZ car les crêtes de T/2 permet de récupérer l'information du signal

❖ Représentation spectrale de signal codé RZ :

Pour visualiser le spectre de fréquence on doit ajouter « zero order holder » et « Spectrum Analyzer » au montage



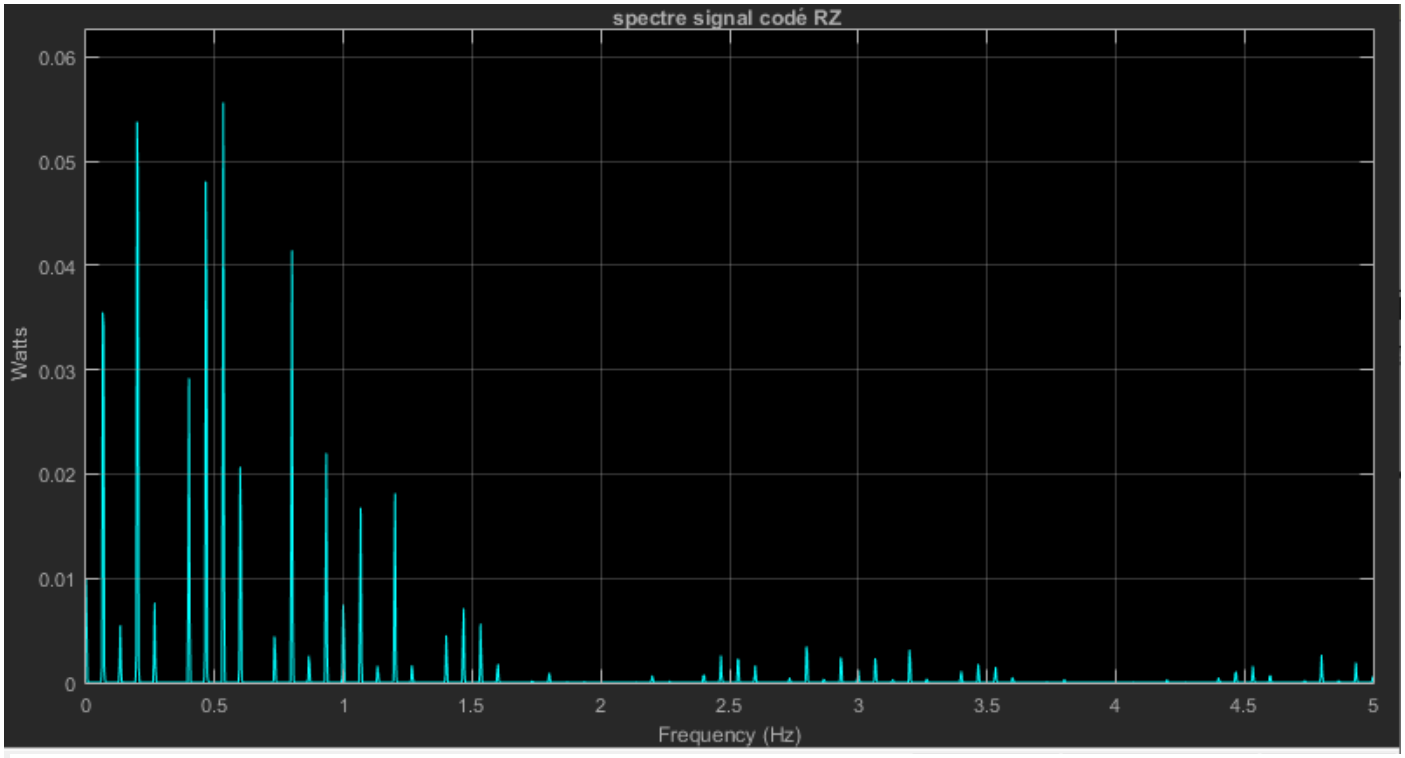


Figure 16: spectre du signal codé RZ

On constate que le spectre de RZ est amorti et de plus on remarque qu' une bande passante [0, 2/T] double par rapport à celle du NRZ, et que le spectre s'annule tous les 2k/T, de plus la puissance est presque un quart de celle de NRZ

- **Montage inverse :**

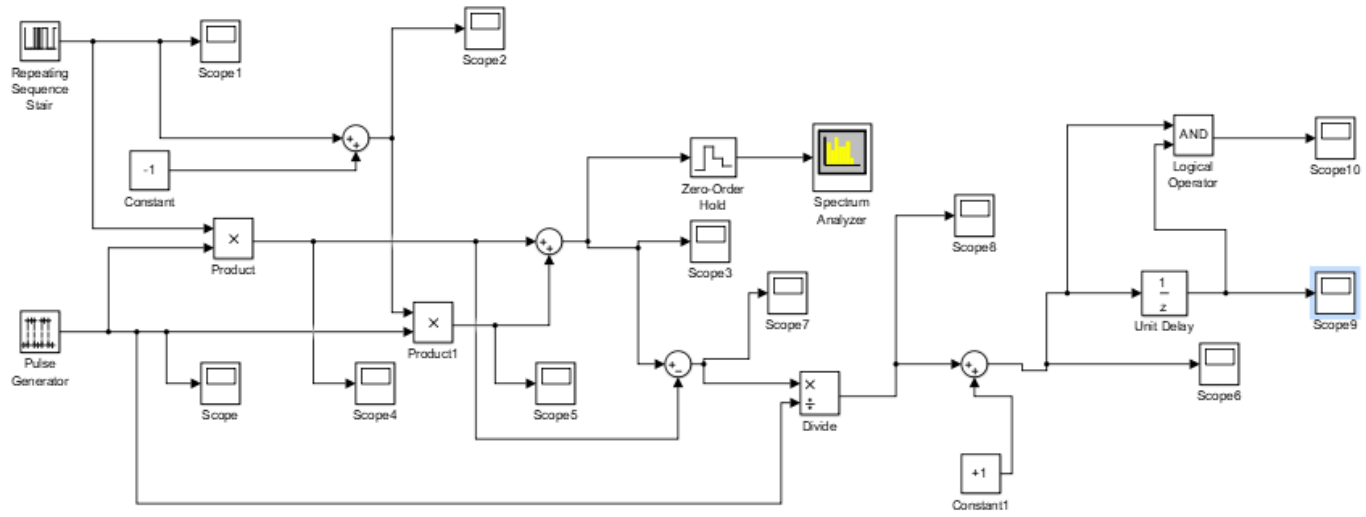


Figure16: montage de décodage RZ

On réalise le montage de décodage selon l'équation mathématique-logique inverse càd

signal_rest = [signal_RZ – (signal_horloge× signal_binaire)] / signal_horloge]+1

Résultat :

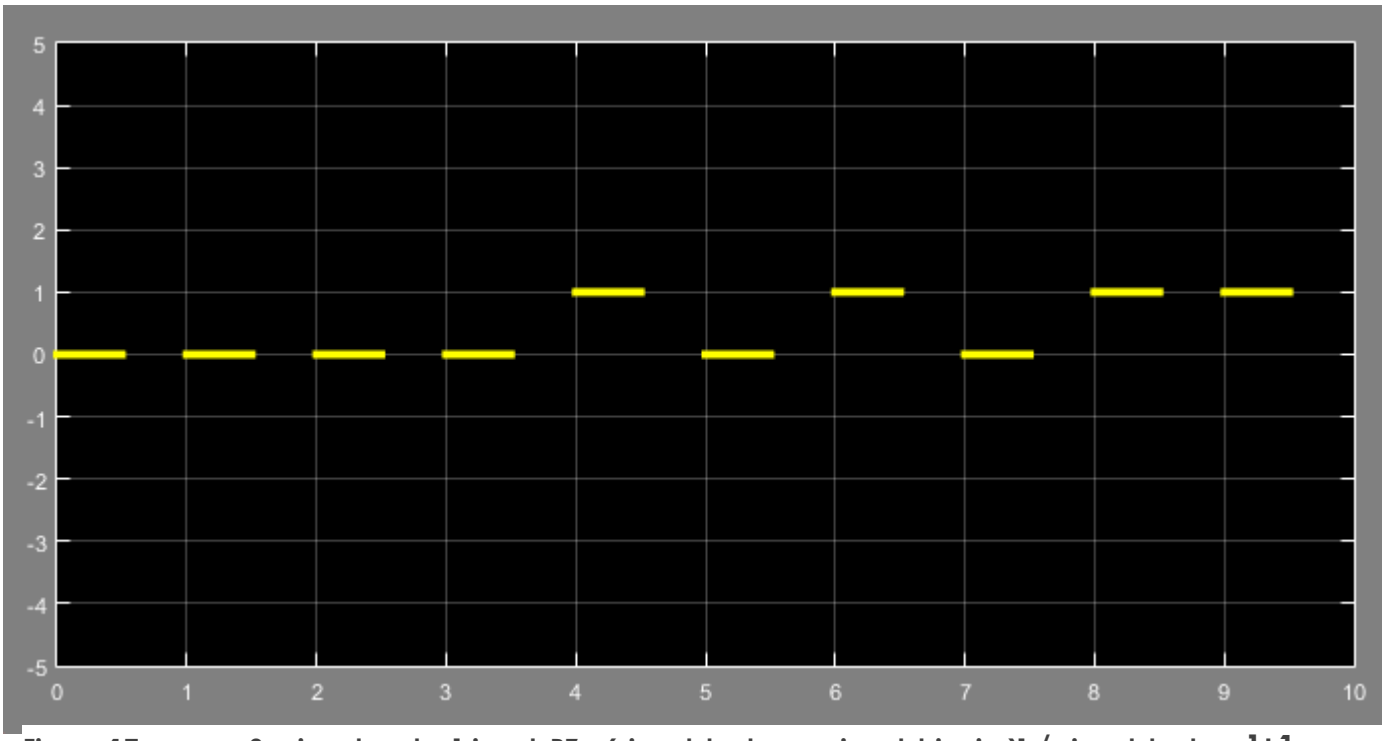


Figure 17 : scope 8 : signal_rest = [signal_RZ – (signal_horloge× signal_binaire)] / signal_horloge]+1

Le résultat obtenu donne un une information partiellement restituer (un pseudo signal discontinue de période T/2. Pour ce là on va décaler le signal obtenu par T/2 en utilisant « Unit Delay » puis on va « sommer » les deux pseudo signaux par l'opération logique « AND » comme le montre le montage dans la figure16

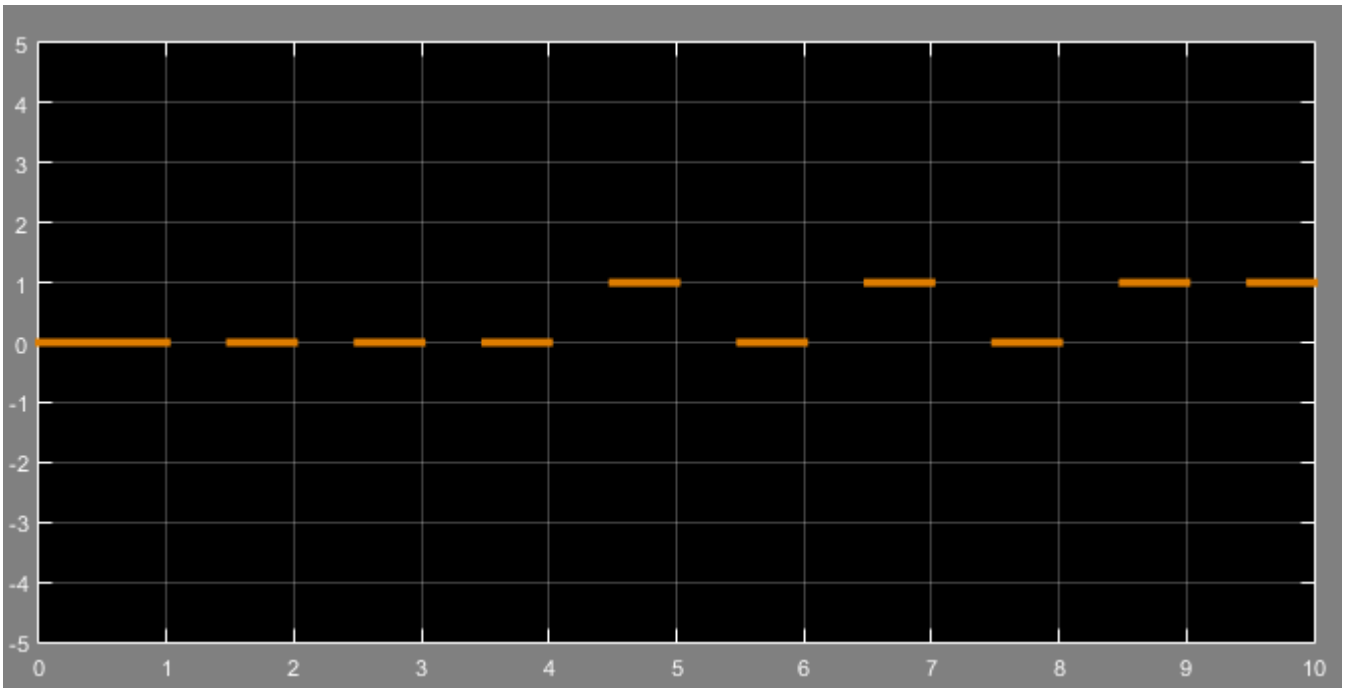


Figure18: Scope9: signal_rest décalé par T/2

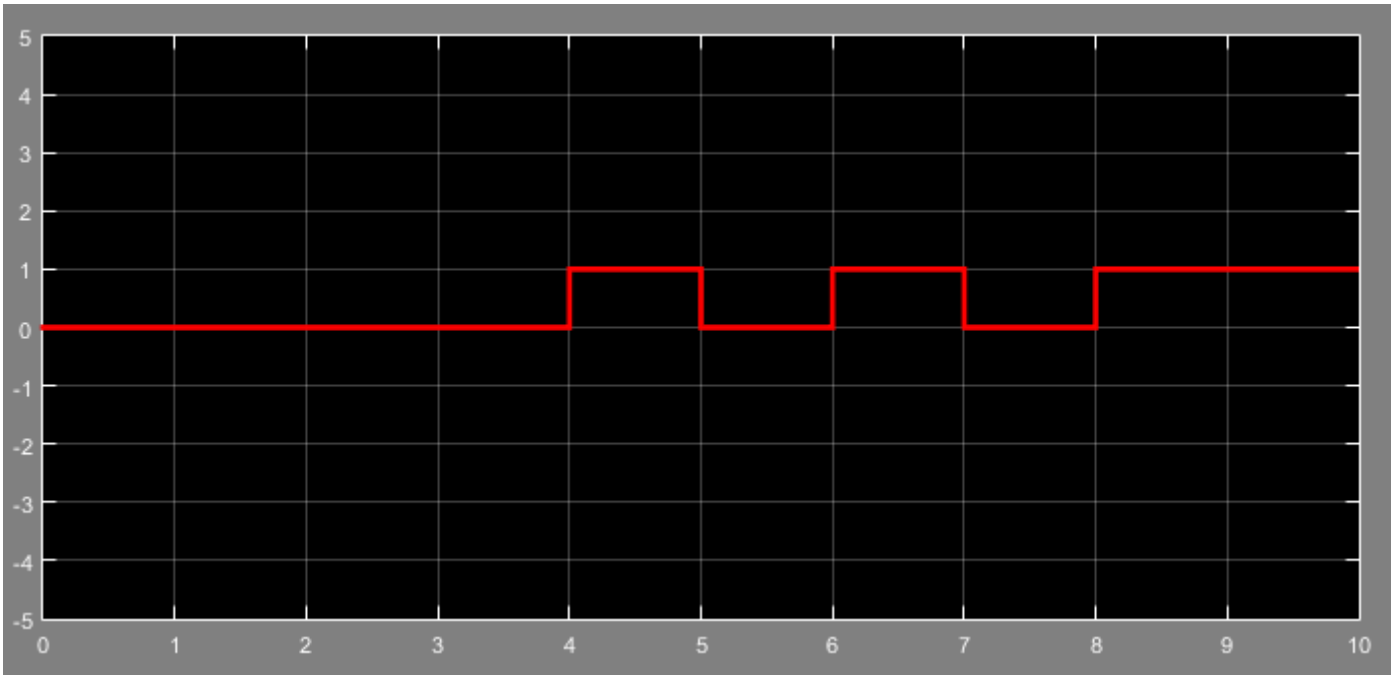


Figure19: Scope 10: signal parfaitement restitué

3) Montage Manchester :

On réalise le montage indiqué dans la fiche TP :

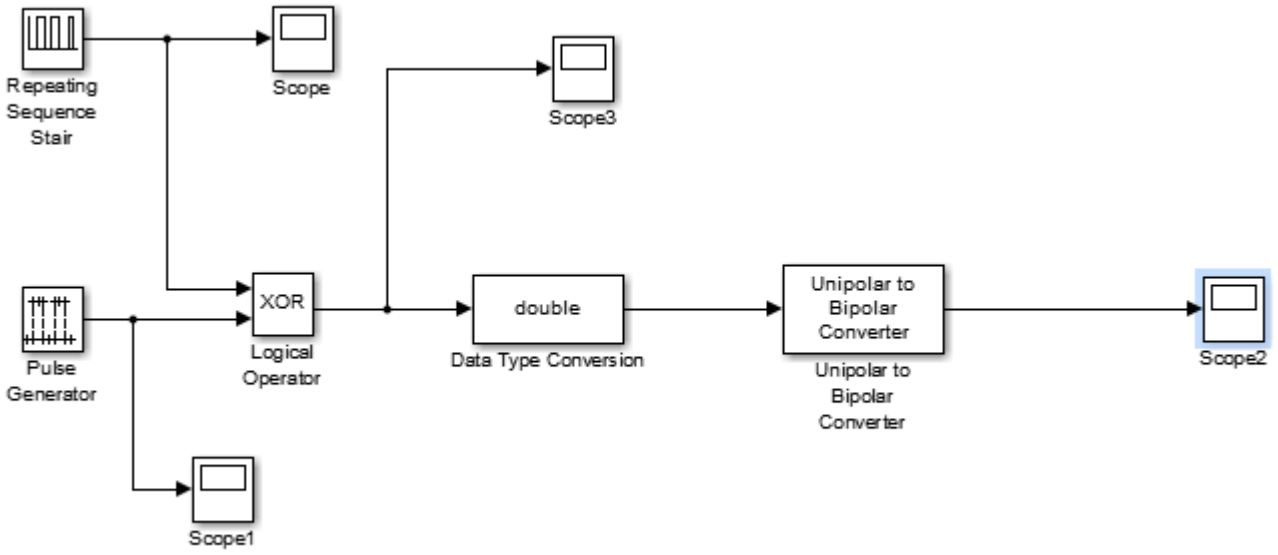


Figure 20 a : montage du codage Manchester

Trame binaire	0		0		1		1		0		0		1		1		0		0		0		1		0	
horloge	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
XOR	1	0	1	0	0	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0	1	0	0	1	1	0

Figure21 table de vérité de XOR

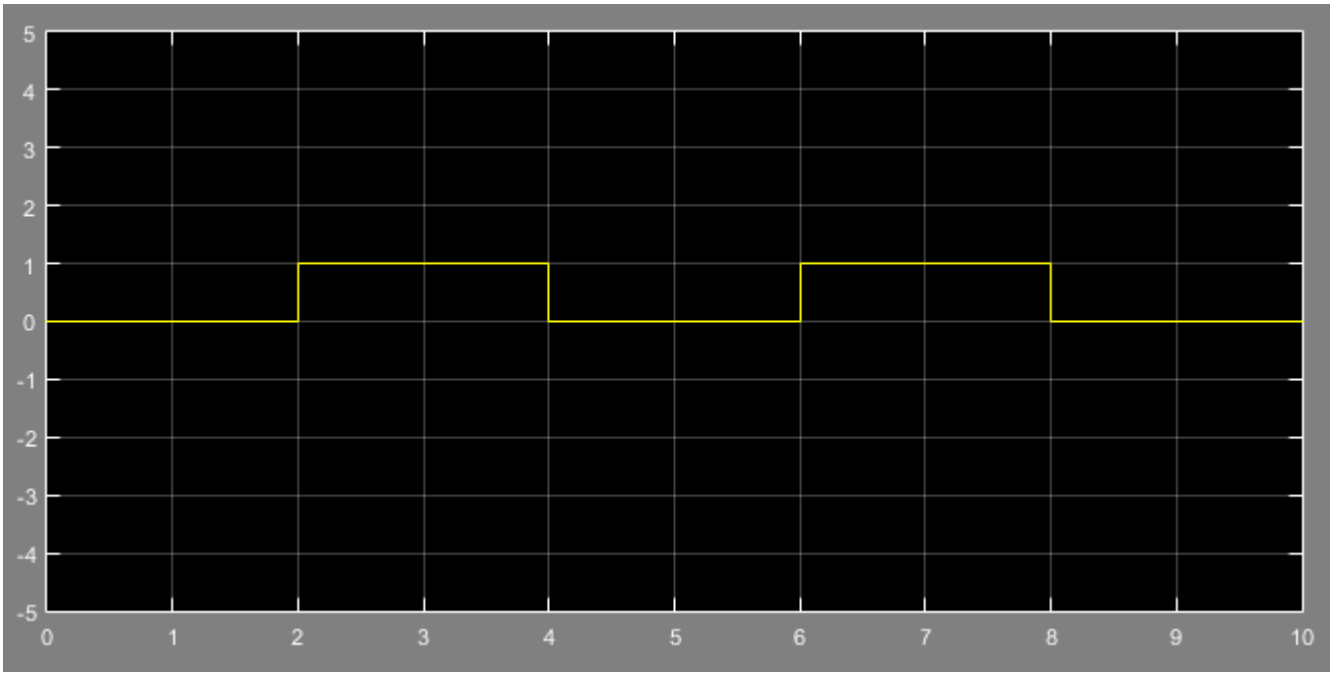


Figure 22 signal binaire

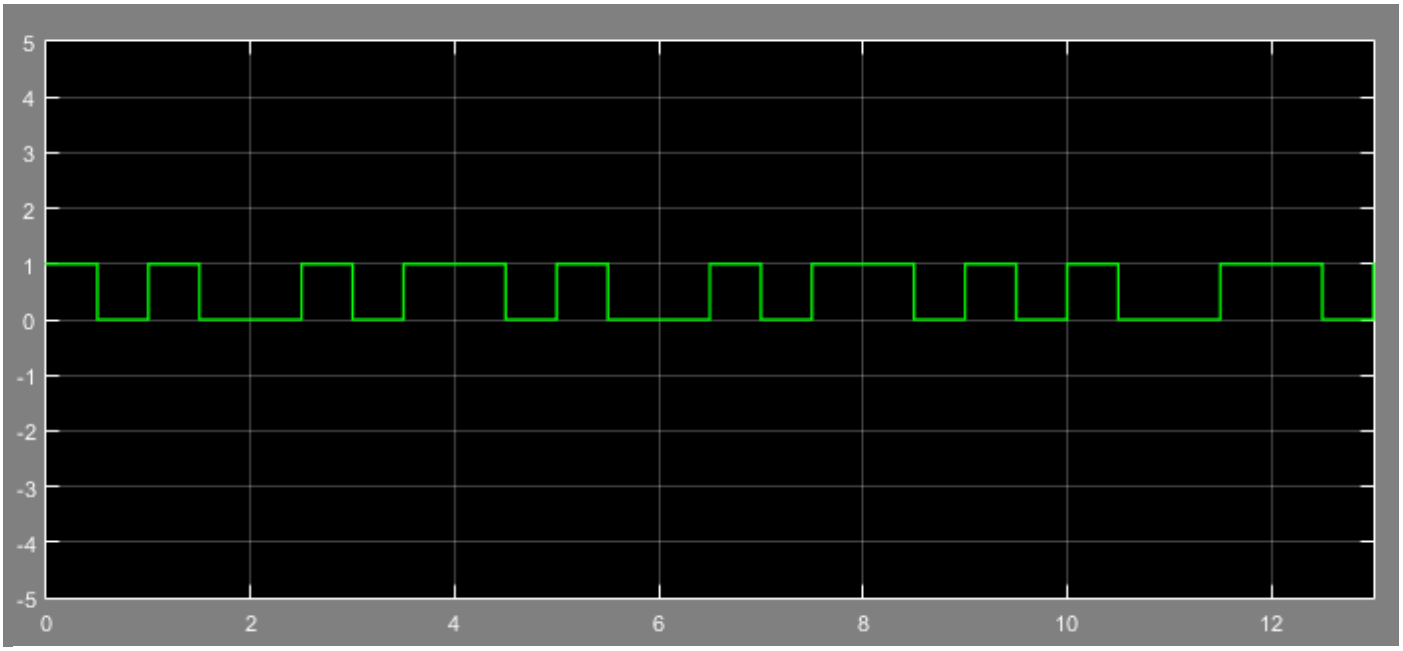


Figure 23 : scope 6 : signal_binaire ⊕ signal horloge (XOR)

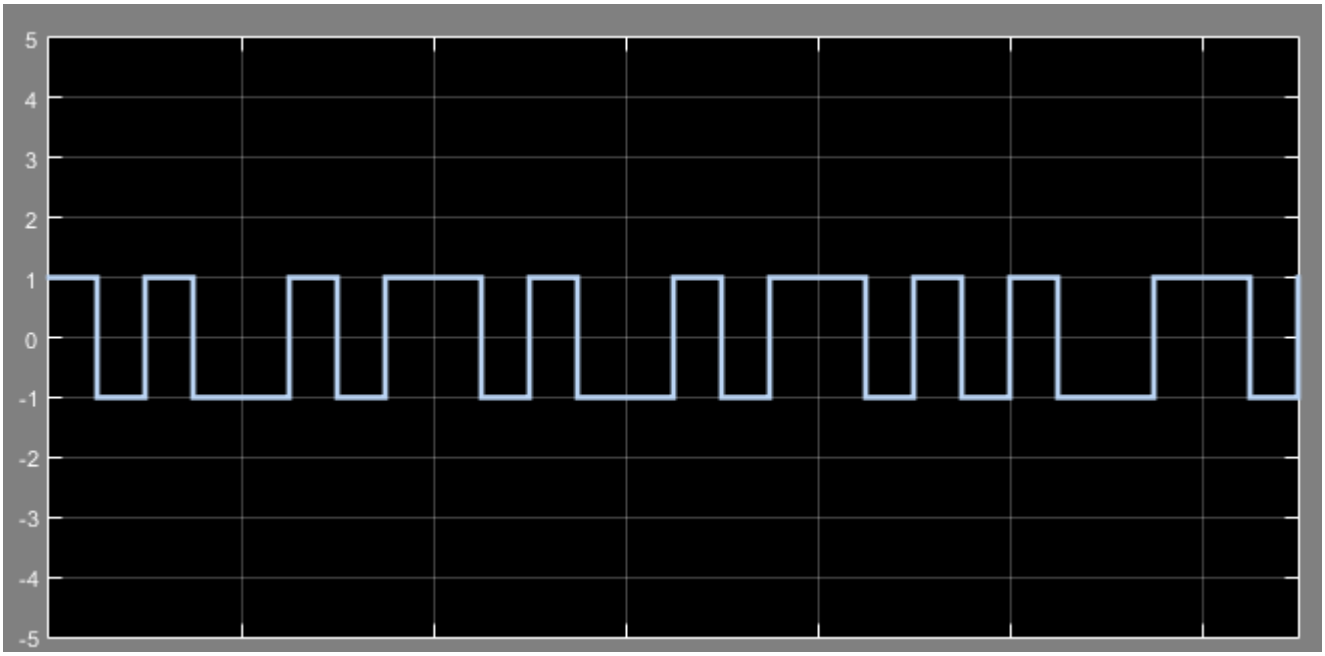


Figure 24 : scope 2 : signal codé Manchester

Avantage Codage Manchester : ça conserve l'horloge, et grâce à la transition au milieu du bit ce codage est plus robuste au bruit pour garantir le décodage : c'est un codage synchrone

❖ Montage Manchester inverse (décodage) :

Après avoir obtenue le signal codé Manchester on le relie à la boite « bipolar to unipolar » puis on le passe sur l'opération logique « XOR » selon le schéma suivant :

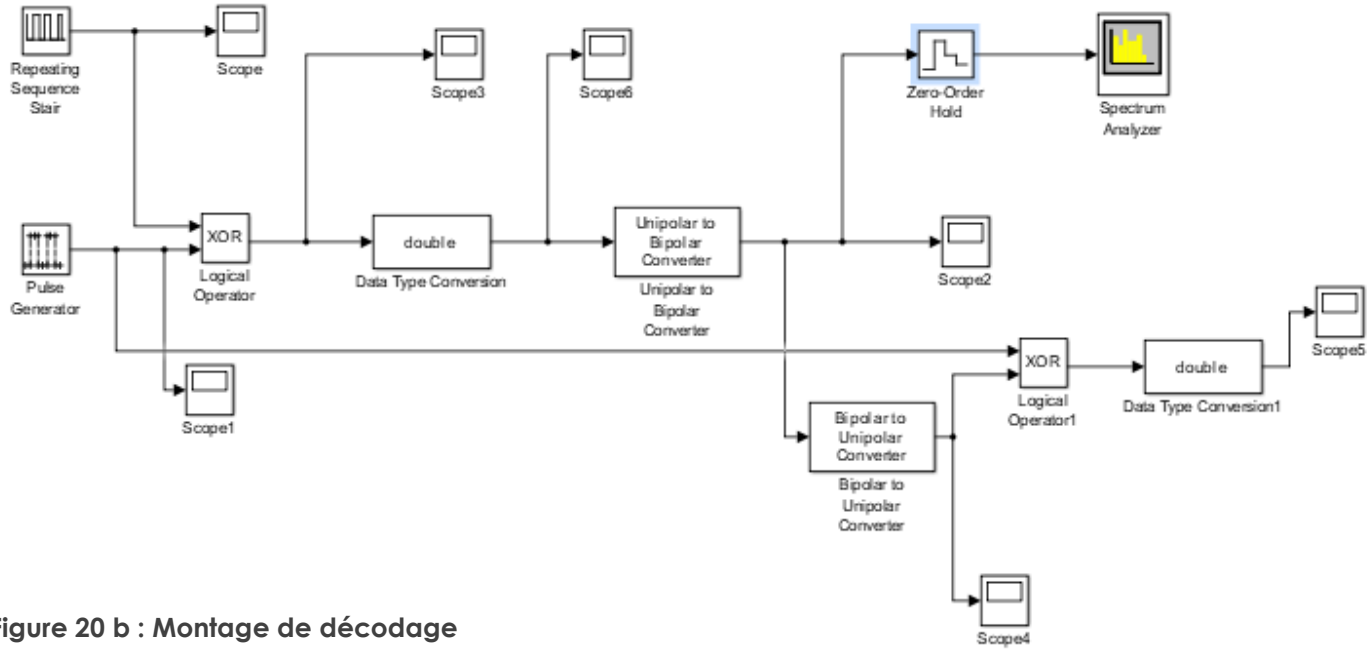


Figure 20 b : Montage de décodage

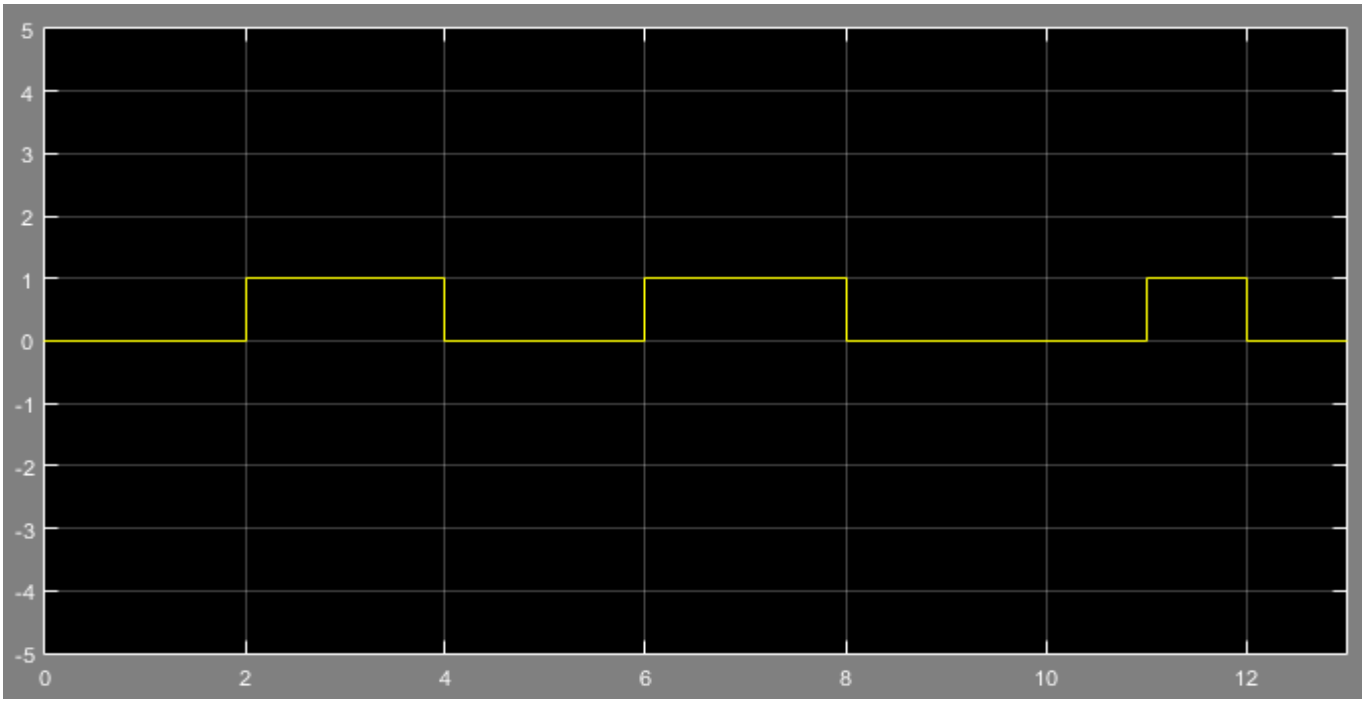


Figure 25 signal décodé

♦ Spectre du signal codé Manchester :

Pour visualiser le spectre on ajoute « zero order holader » puis « spectrum analyzer » à la sortie du montage de codage Manchester comme le montre la figure 20 b

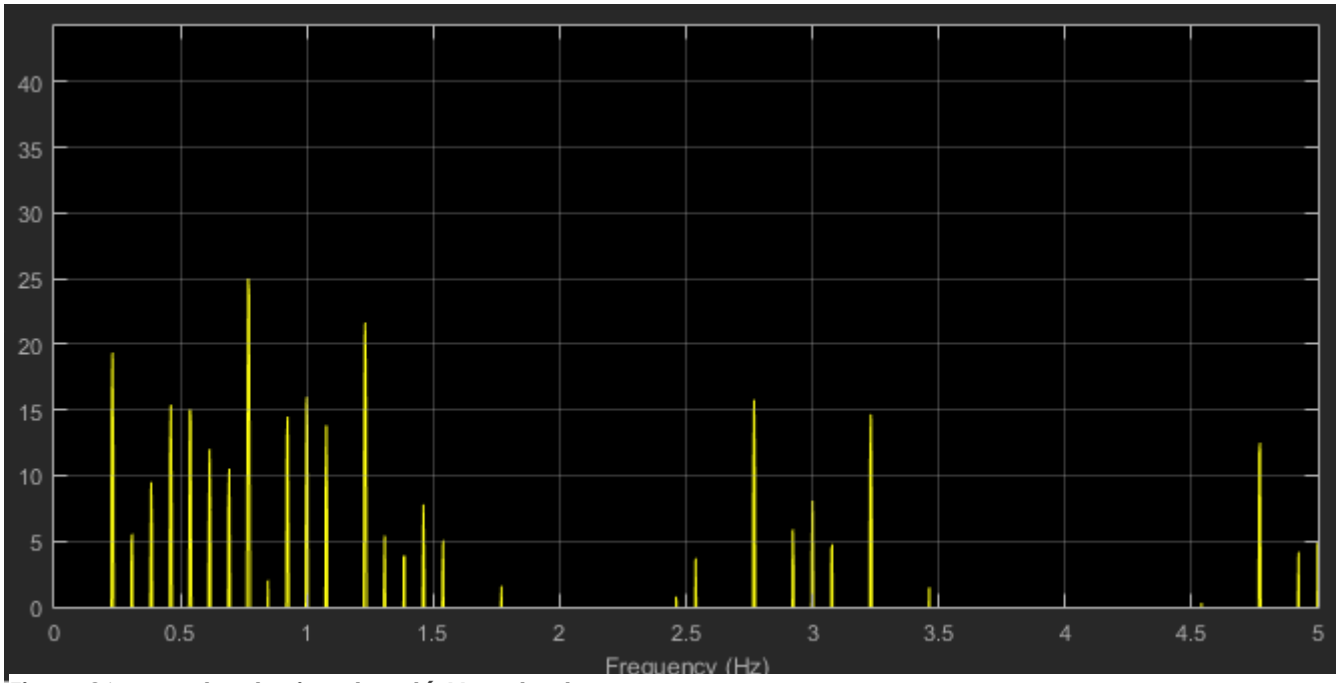


Figure26 : spectre du signal codé Manchester

Le spectre du codage Manchester montre une bande passante doublée [0,2/T[et qui ne présente pas une composante continue (sur f=0), le spectre est nul sur 2k/T.

Le codage Manchester : Caractéristiques de ce codage :

- Bonne résistance au bruit (2 niveaux)
- Bonne adaptation aux supports à bande passante large
- Beaucoup de transitions, donc facilité de synchronisation d'horloge

Le principal inconvénient de ce code réside dans la grande largeur de son spectre, ce qui le confine aux supports à large bande comme les câbles coaxiaux

TP2 Codage : Cod. Hauffman
Sahar BEN MABROUK
MP MATIS2

Faculté des sciences de Sfax
Dép. Des Mathématiques
17/12/2020

Objectif :

L'objectif est de réaliser le code source de codage en ligne Hauffman

« Le principe du codage de Hauffman repose sur la création d'une structure d'arbre composée de nœuds.

L'arbre est créé de la manière suivante, on associe chaque fois les deux nœuds de plus faibles poids, pour donner un nouveau nœud dont le poids équivaut à la somme des poids de ses fils. On réitère ce processus jusqu'à n'en avoir plus qu'un seul nœud : la racine. On associe ensuite par exemple le code 0 à chaque embranchement partant vers la gauche et le code 1 vers la droite. » Wikepedia

Manipulation :

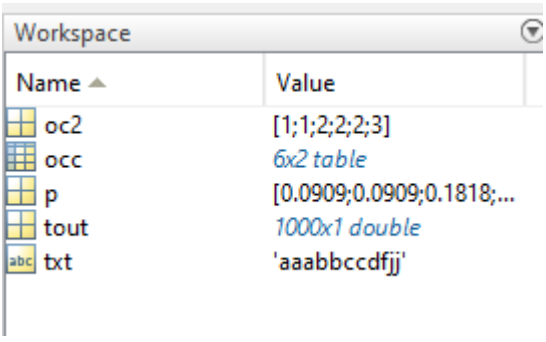
On construit une fonction qui calcule L'occurrence de chaque alphabet dans le Message .
Inputs : message à coder
Outputs : p= vecteur de probabilité de chaque Alphabet
Occ : un tableau de type cell qui affiche l'alphabet et son occurrence
oc2 : la liste d'occurrence

principe :

on commence par ordonner le message et on convertit le msg en liste L
puis on efface la répétition et on ne laisse que les caractére sans répétition
puis à l'aide de la commande strfind
on peut détermine le nombre d'occurrence à partir de la liste d'indice d'apparition qu'elle retourne.

```
function [occ ,oc2,p] = occurence(txt);
%txt = input('Tapez votre texte :    ','s')
M=strfind(txt,txt(1));
p=length(txt);
L=sort(txt);

i=2;
  while i <=length(L)
     if L(i)== L(i-1)
       L(i)= '';
       i=i-1;
    end
    i=i+1;
end
oc=[];
for j=1:length(L)
    k=length(strfind(txt,L(j)));
    %num2str(k);
    oc=[oc;{L(j),k}];
end
occ= cell2table(oc);
oc2=sort(table2array(occ(:,2)))
p=oc2./sum(oc2);
%A= table2array(T);
end
```

```
>> txt='aaabbccdfjj'

txt =

aaabbccdfjj

>> [occ ,oc2,p] = occurence(txt)
```




```
occ =

    oc1     oc2
    ___     ___

    'a'     3
    'b'     2
    'c'     2
    'd'     1
    'f'     1
    'j'     2


oc2 =

     1
     1
     2
     2
     2
     3
```

```
p =

    0.0909
    0.0909
    0.1818
    0.1818
    0.1818
    0.2727
```

Code source de codage Hauffman: licence Sean Danaher University of Northumbria at Newcastle UK 98/6/4:

Cette fonction est la fonction principale elle
Prend un vecteur de probabilité et retourne
Le code de hauffman et le taux de compression
Cette fonction va Faire appel à d'autre fonctions :

- Huff5 : donne l'arbre de hauffman
- Getcodes : retourne le code binaire de hauffman

```
function [code,compression]=huffman(p);
% Sean Danaher University of Northumbria at
Newcastle UK 98/6/4
p=p(:)/sum(p);    %normalises probabilities
c=huff5(p);
code=char(getcodes(c,length(p)));
compression=ceil(log(length(p))/log(2))/
(sum(code' ~= ' ')*p);
```

Sous fonction Huff5 :

Inputs : p vecteur de proba

Outputs : c arbre de hauffman

Principe : génére une celle ligne
De Longueur celui de vecteur p
Une boucle while dans laquelle on ordonne p, reordonnement de l'arbre de hauffman, sommer les probas et effacer chaque probabilité classée dans l'arbre.

```
function c=huff5(p);
Note Matlab 5 version
% Sean Danaher 98/6/4        University of Northumbria,
Newcastle UK
c=cell(length(p),1);            % Generate cell structure
while size(c)-2                 % Repeat till only two
branches
    [p,i]=sort(p);                  % Sort to ascending
probabilities
    c=c(i);                         % Reorder tree.
    c{2}={c{1},c{2}};c(1)=[];   % join branch 1 to 2 and
prune 1
    p(2)=p(1)+p(2);p(1)=[];     % Merge Probabilities
end
```

Fonction getcodes donne le code de
Chaue branche en partant du racine
Ceci à l'aide de la fonction récursive
Getcodes2 qui donne le code binaire

```
function y= getcodes(a,n)
% Y=GETCODES(A,N)
% Sean Danaher 98/6/4   University of
Northumbria, Newcastle UK
global y
y=cell(n,1);
getcodes2(a,[])
```

Objectif :

Le but de ce TP est de réaliser les montages aboutissant à la modulation numérique selon trois types : modulation d'amplitude ASK , modulation de fréquence FSK et modulation de phase PSK.

Pour cela on va utiliser Matlab Simulink.

Manipulation :

❖ La modulation d'amplitude (ASK « *Amplitude Shift Keying* ») :

Le principe de cette modulation est de transmettre :

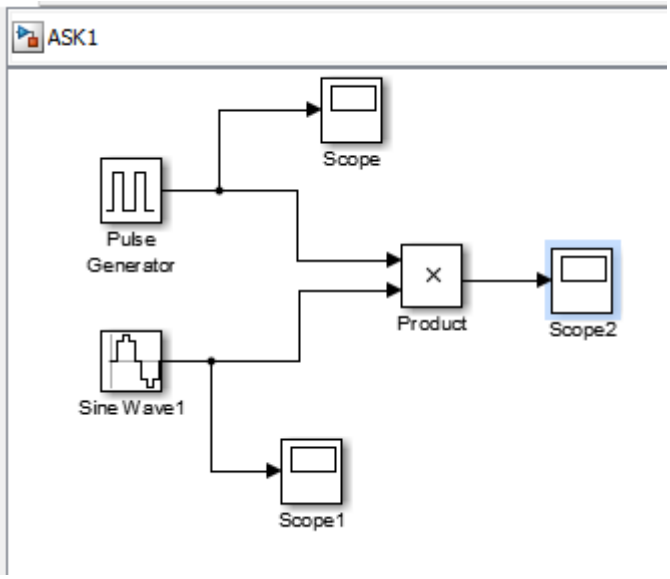
- Une fréquence f pour l'information de niveau haut.
- Une fréquence nulle pour l'information de niveau bas.

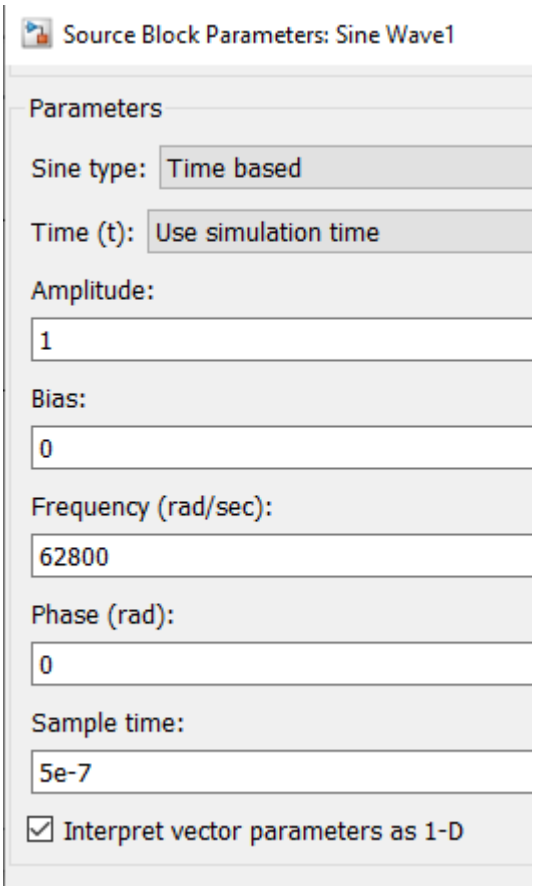
♦ Montage :

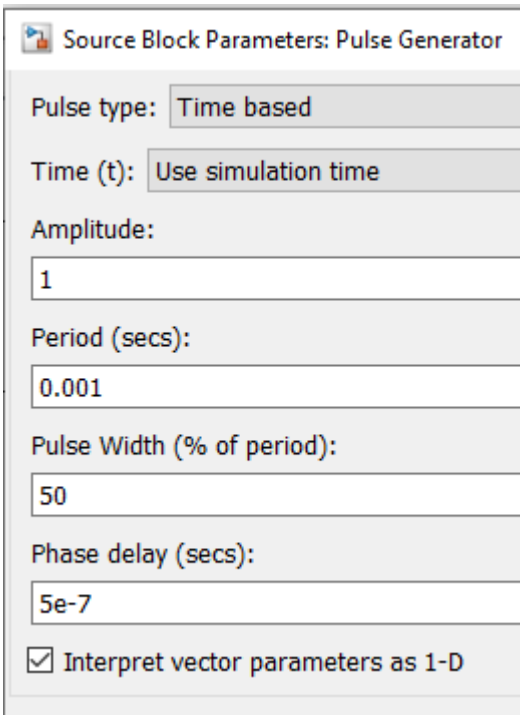
-la porteuse est un signal sinusoïdale généré par le bloc « Sine Wave », cette porteuse est d'amplitude1 et de fréquence 10 KHz soit 62800 rad/s

-le signal modulant qui est binaire est généré par le bloc « pulse generator » , il est d'amplitude 1 et de période 0.001s (chaque période contient 0 et 1 comme on a choisi 50% pulse width)

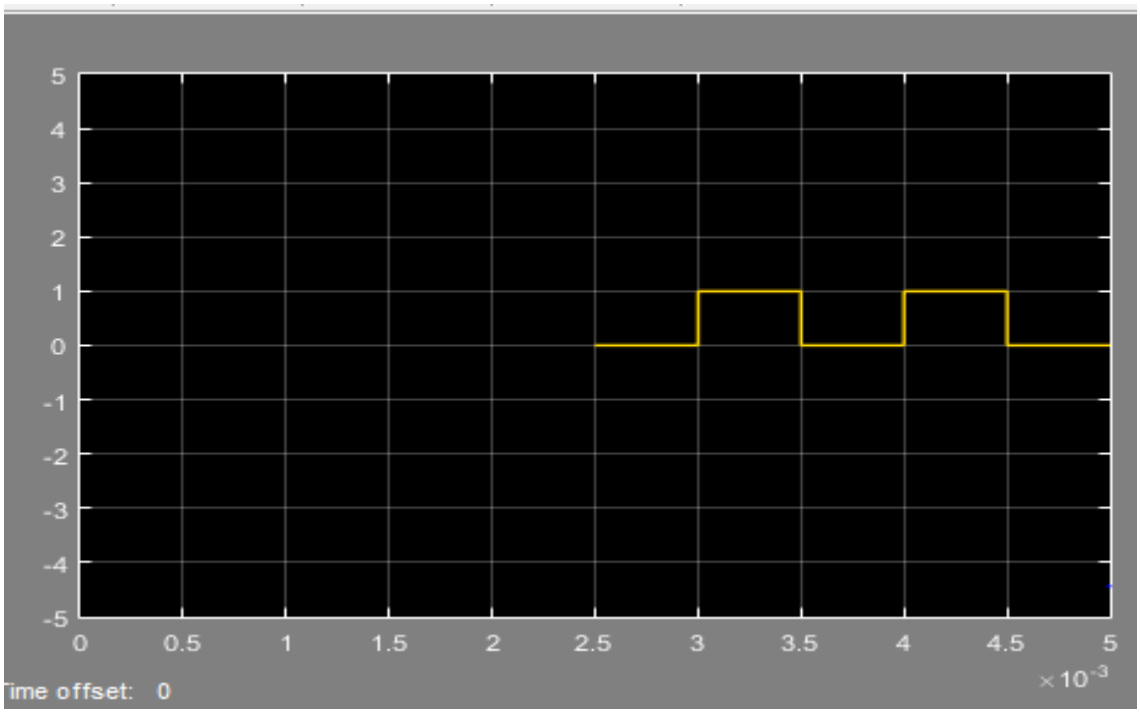
-le signal modulé est obtenu donc on faisant le produit entre la porteuse et le signal modulé.



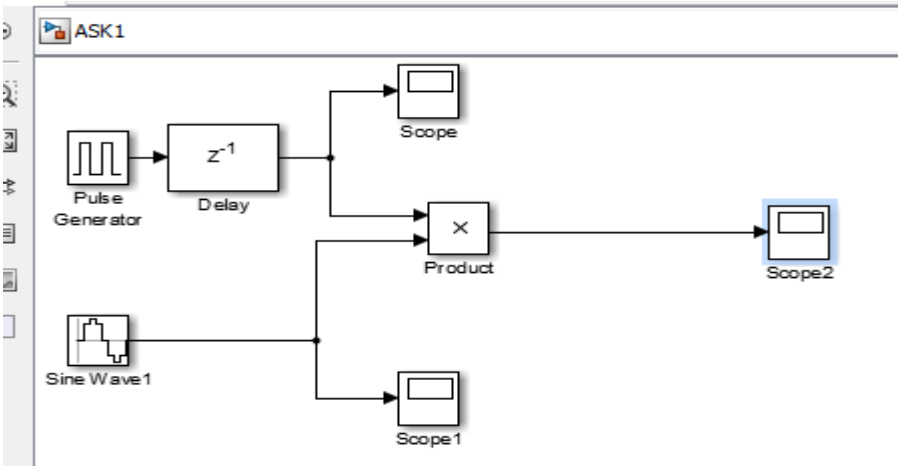




- **Visualisation :**



En visualisant le résultat de pulse generator et wave sine et meme le signal modulé, un retard de l'ordre 2.5 e-3 s. On peut l'enlever en appliquant un retard (t-2.5e-3) à tous ce signal. (time delay).



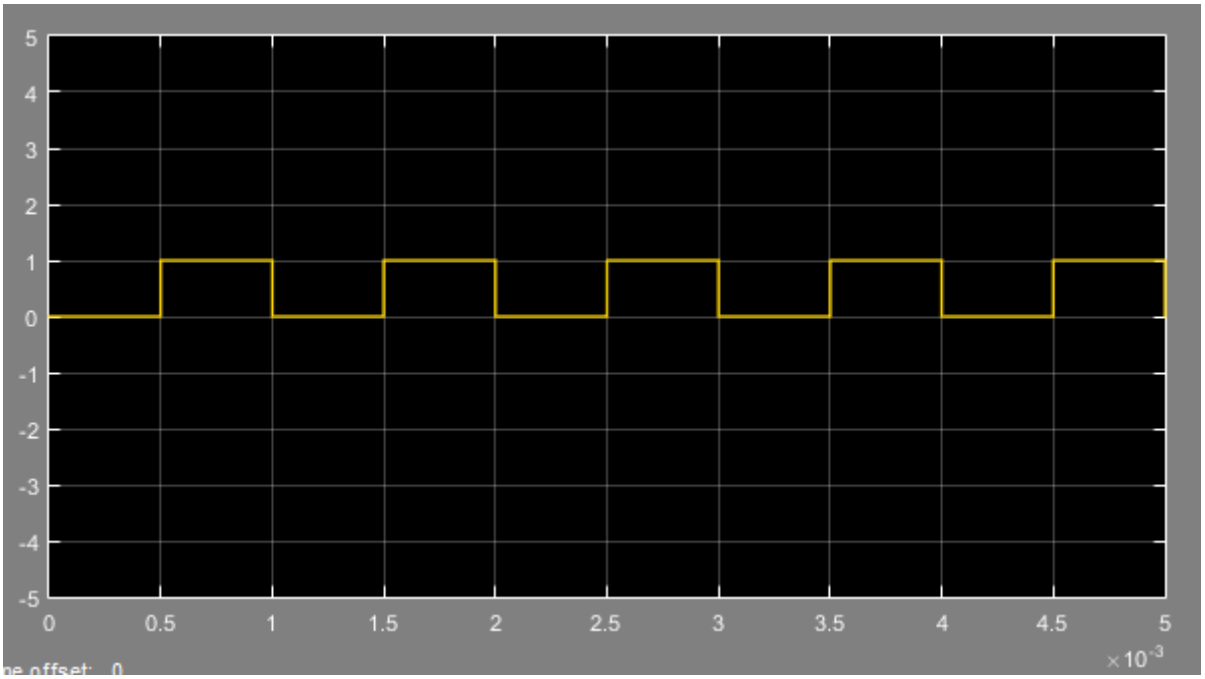


Figure 1 : Scope: trame binaire

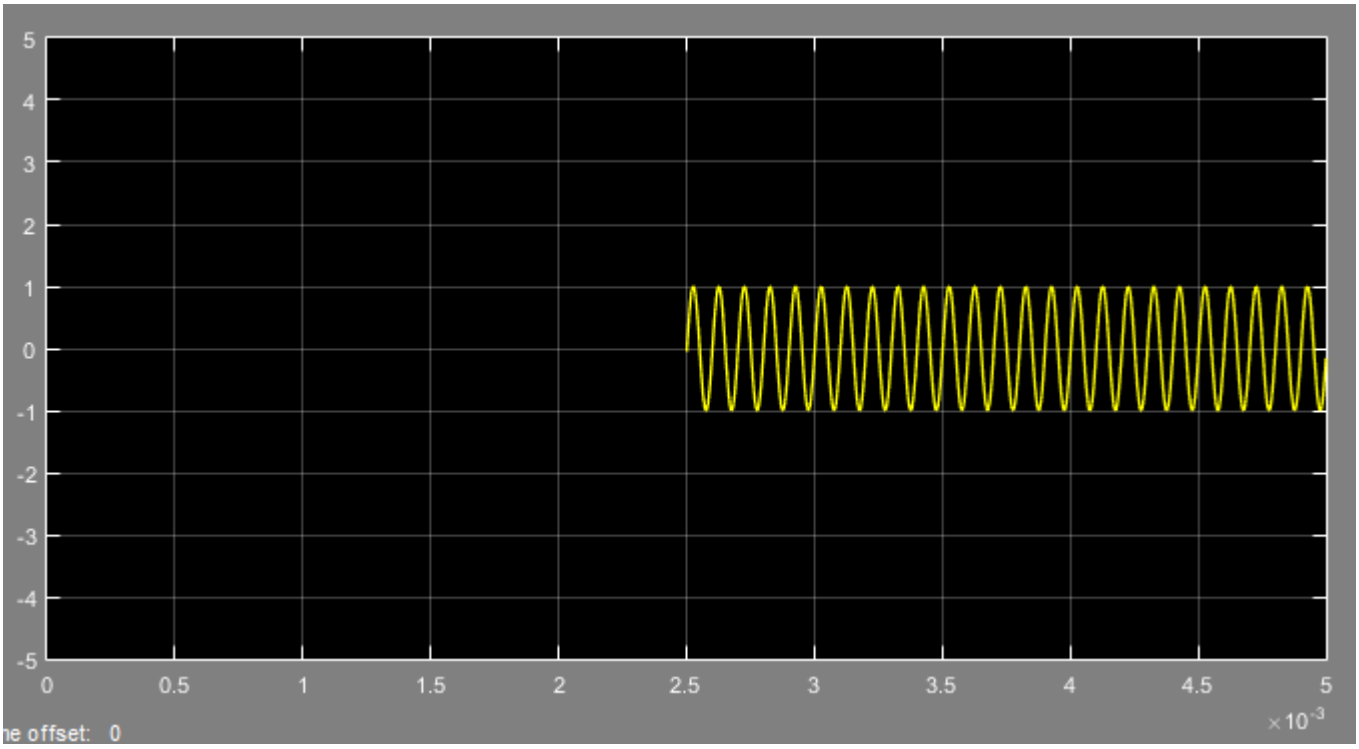


Figure 3: scope1 : la porteuse

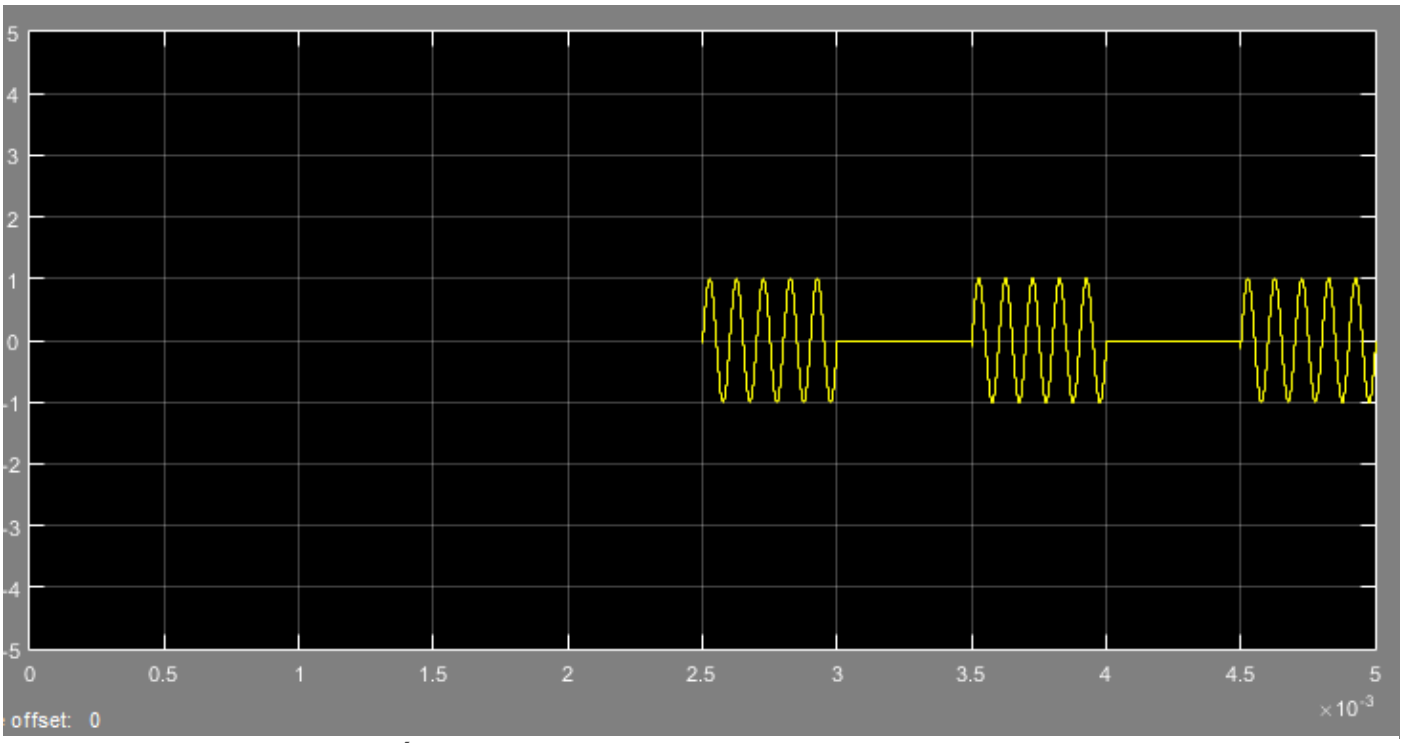


Figure 2: Scope2: signal modulé ASK

- **Vitesse de transmission =nbre de bits/ΔT =1/0.001=10e3bit/s**
- **Nombre d'oscillations= Tmodulant/Tporteuse=0.001/0.0001=10 oscillations pour T donc pour T/2 on a 5 oscillations ce qui est le cas dans le graphe.**
- **Fmin=2Fmodulant=2/Tmodulant=2×10e3 ➔ Fmin≥ 2×10e3 HZ**

- **Signal modulé avec Fmin=2×10e3 :**

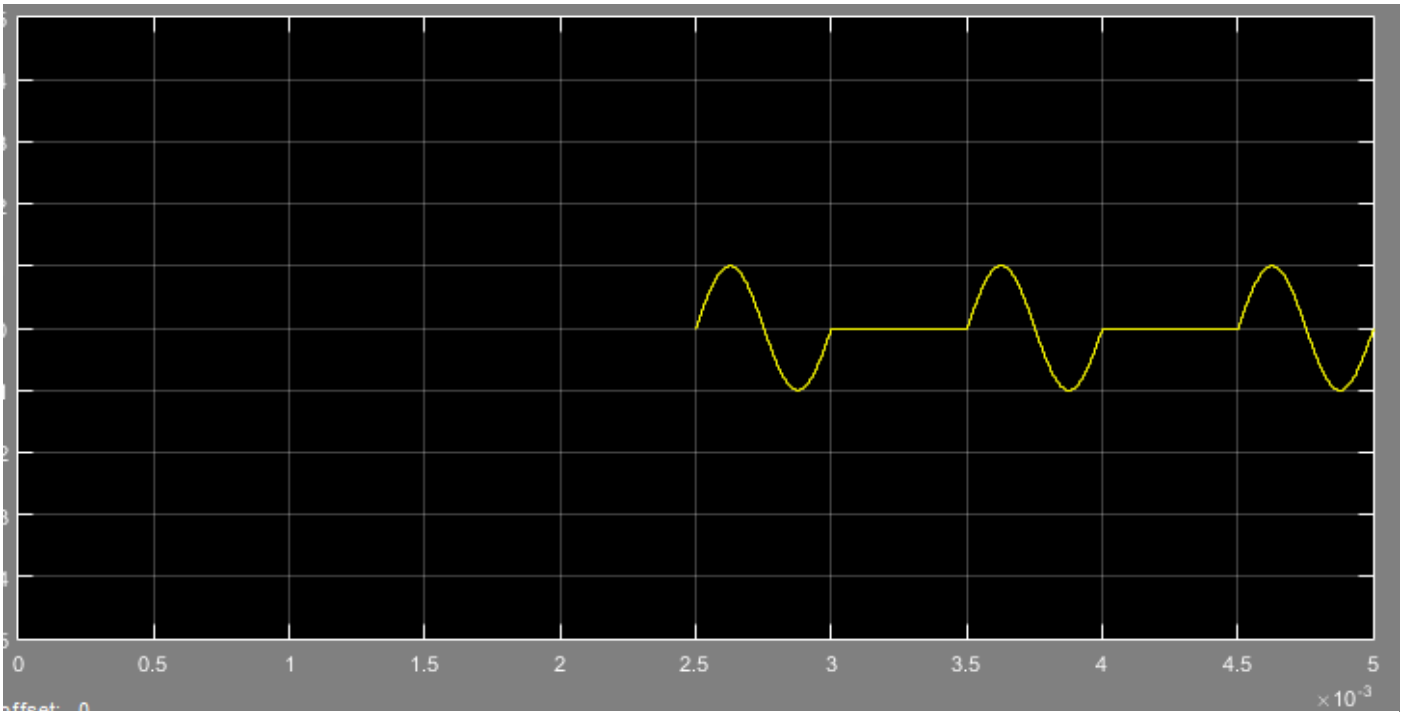


Figure 4: signal modulé avec Fmin

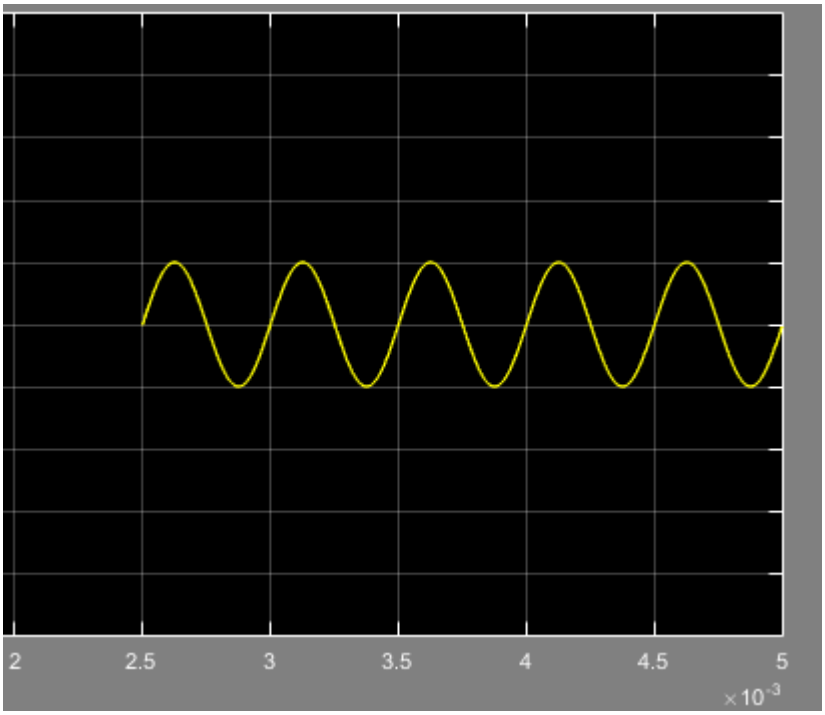
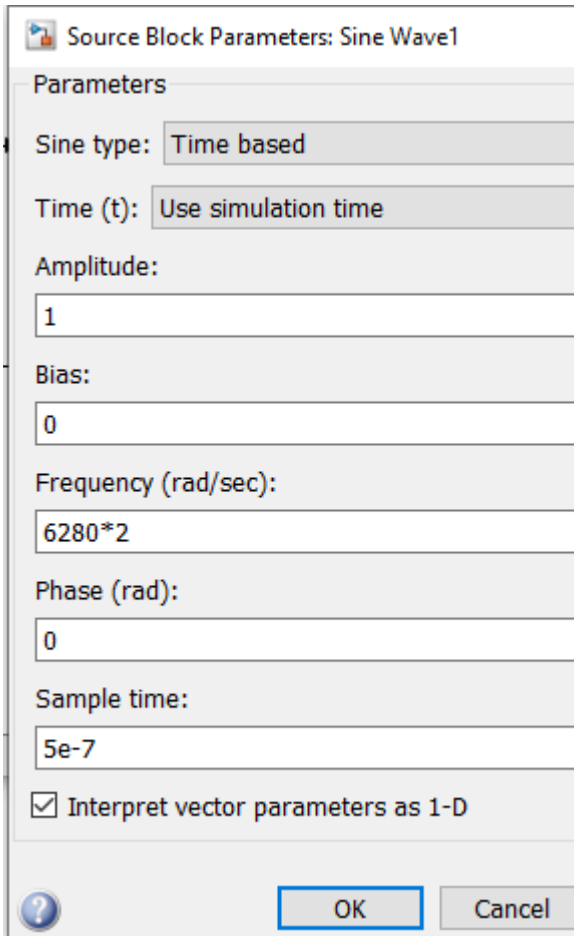


Figure 5:porteuse de fréquence Fmin



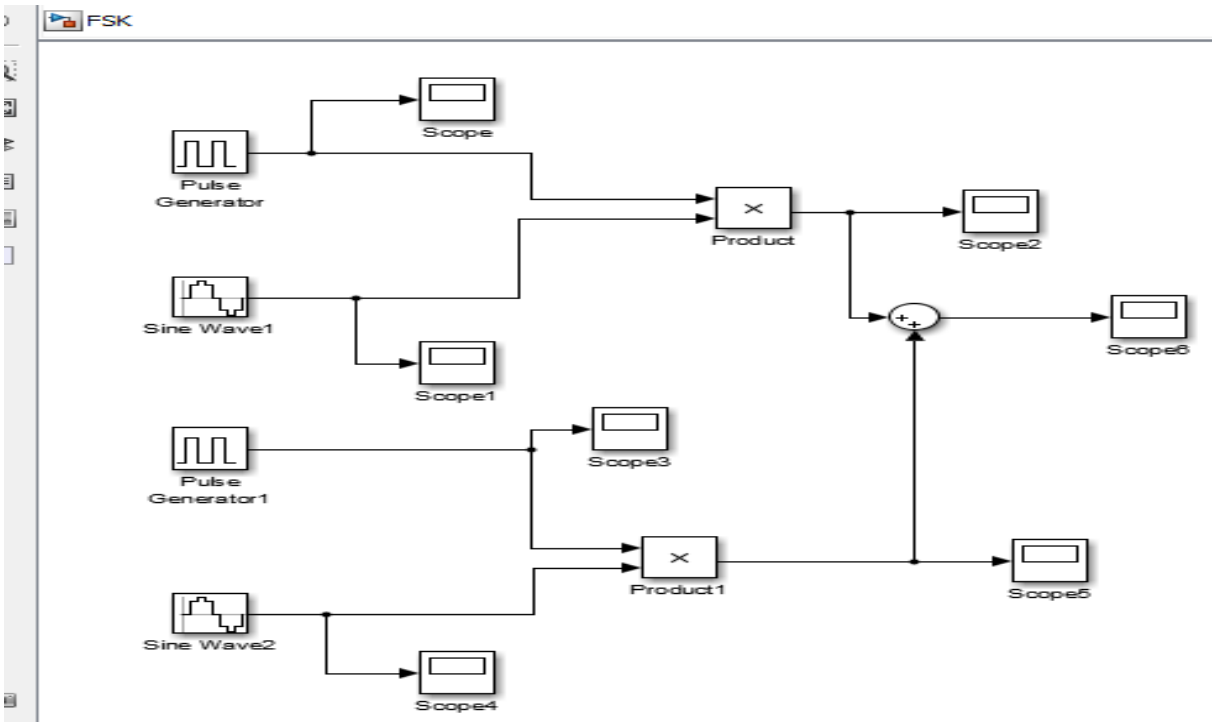
❖ **La modulation de fréquence (FSQ « *Frequency Shift Keying* ») :**

Cette modulation est de transmettre:

- Une fréquence f1 pour l'information de niveau haut.
- Une fréquence f2 pour l'information de niveau bas.

Elle peut être considérée comme un cas particulier d'une modulation d'amplitude : une somme de deux modulations ASK.

♦ **Montage :**



-on refait le meme montage de modulation ASK mais en donnant deux fréquences différentes :

20KHz pour P1 et 10KHz pour P2 et pour le signal modulant2 on lui applique un retard de T/2 pour pouvoir obtenir les oscillations juxtaposées

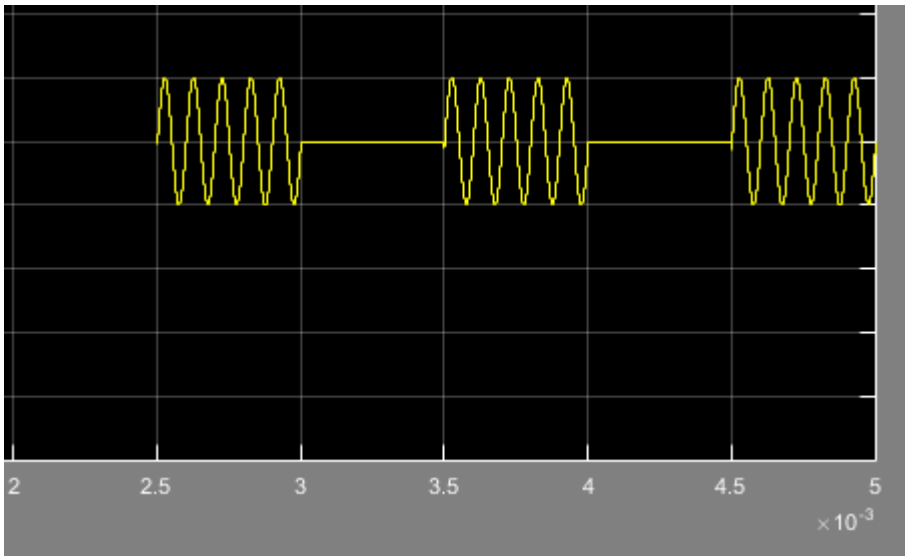


Figure 6:Scope5: ASK2

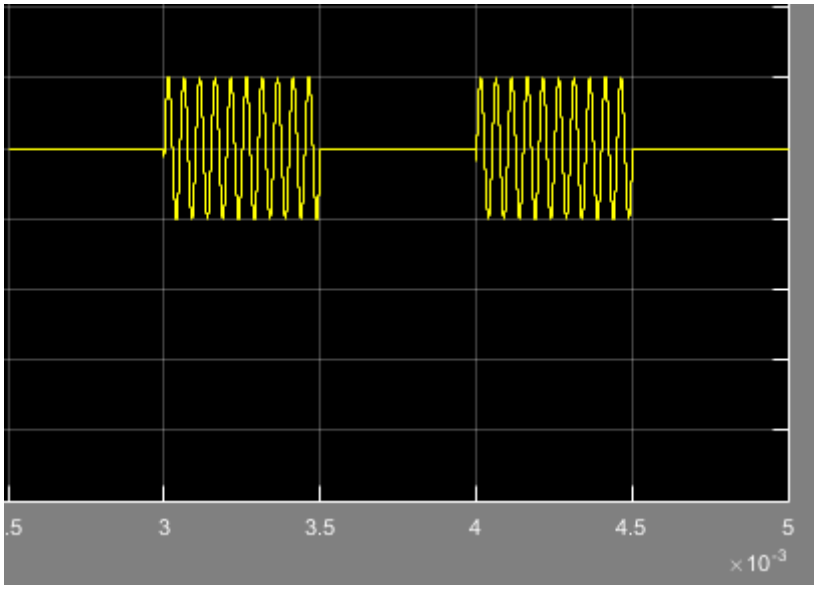


Figure 7:Scope2: ASK1

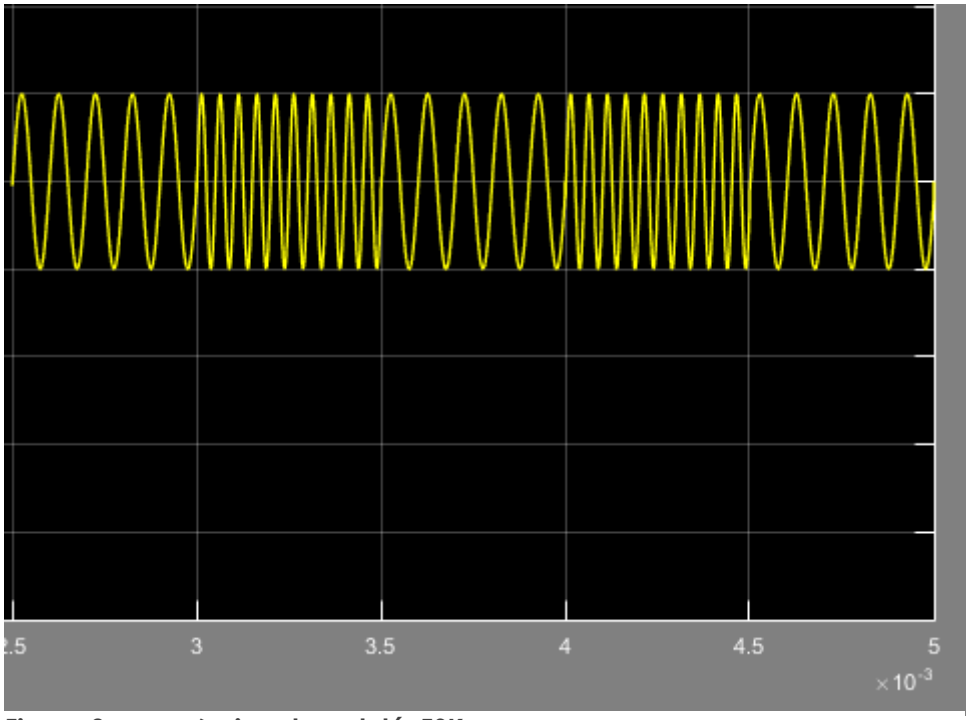


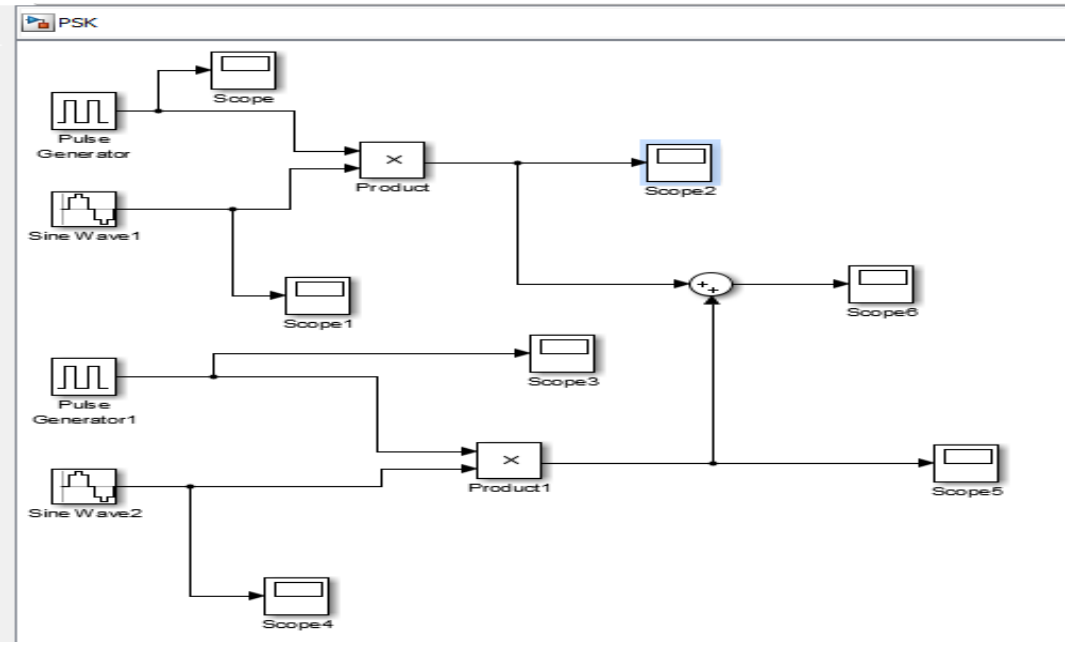
Figure 8:scope6: signal modulé FSK

❖ La modulation de phase (PSK « Phase Shift Keying »)

Cette modulation provoque un changement de phase pour chaque changement de niveau du signal
modulant. On parle de la modulation de phase binaire (BPSK) lorsqu'il les phases des porteuses
modulées sont en opposition de phase. C'est un cas particulier de modulation ASK.

- **Montage :**

on refait le meme montage de modulation ASK mais en donnant deux phases opposées : 0 pour P1 et π pour P2 et pour le signal modulant2 on lui applique un retard de T/2 pour pouvoir obtenir les oscillations juxtaposées.



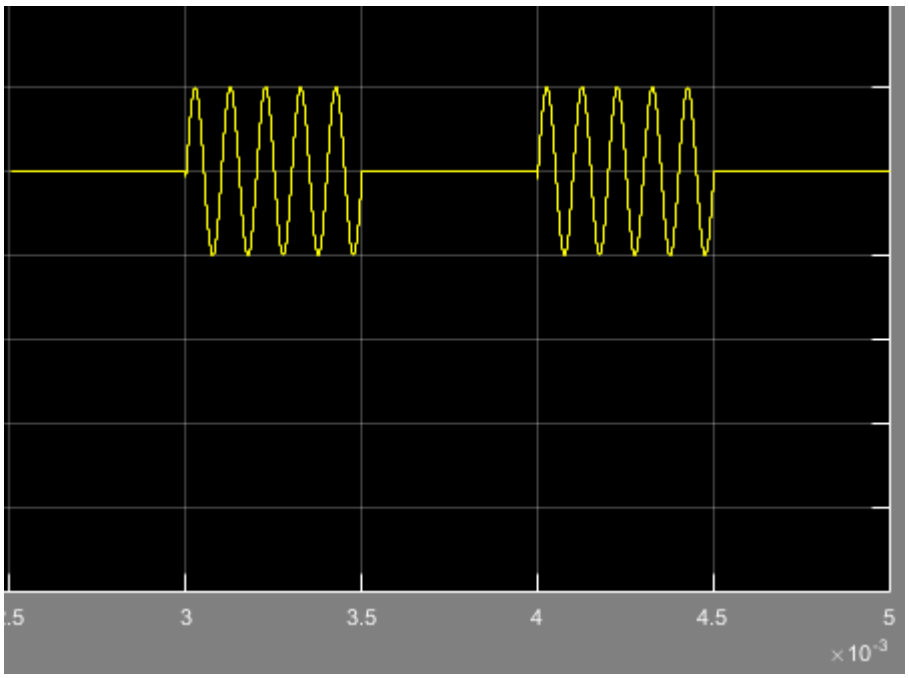


Figure 9:Scope2: ASK1 phi=0

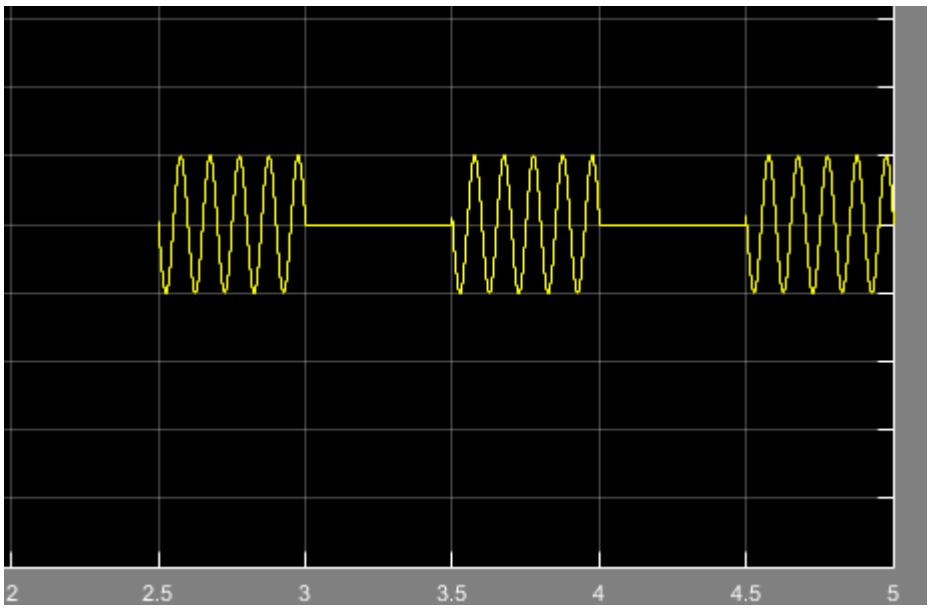


Figure 10:scope5: ASK2 phi=pi

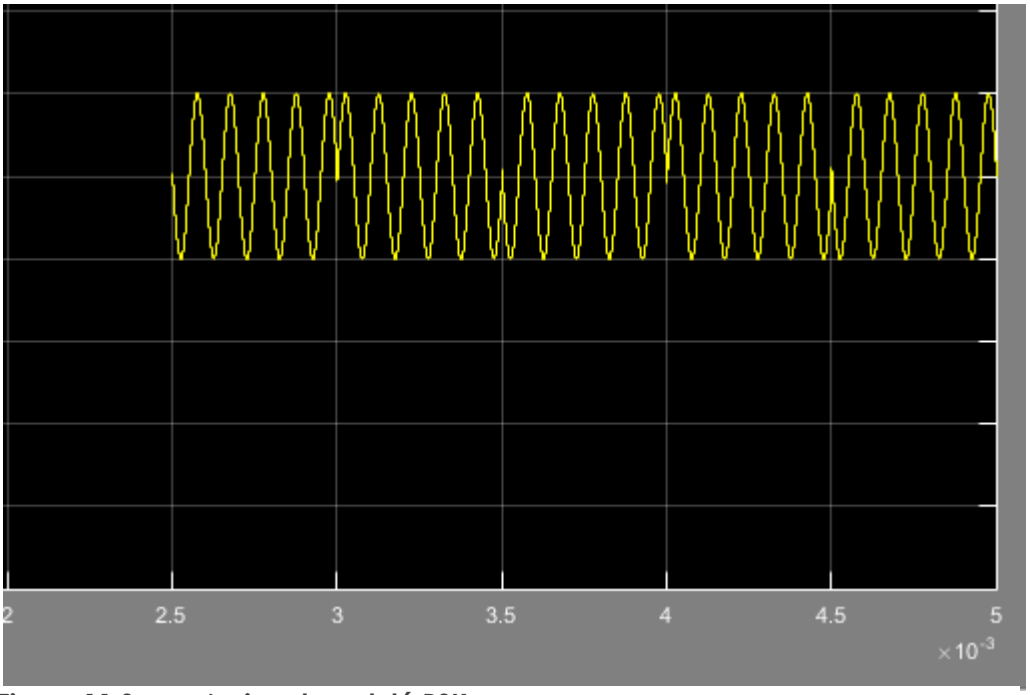


Figure 11:Scope6: signal modulé PSK