

Travaux Pratiques de Physique

Lycée
Charlemagne
Paris

MP

4 heures

Calculatrices autorisées

Numérisation des signaux

Objectif

De nombreux dispositifs traitent numériquement des informations analogiques qui leur sont fournies. Une transcription est alors nécessaire, à l'issue de laquelle une image binaire de dimension finie doit approcher au mieux le signal d'origine ; c'est la conversion analogique/numérique qui fait l'objet de ce TP.

I- Repliement de spectre

1. Visualisation à l'écran de l'ordinateur

Le logiciel LAVIS PRO, associé à l'interface EUROSART, constitue un convertisseur analogique/numérique dont les caractéristiques de l'échantillonnage peuvent être imposées :

- **Te** : désigne la période d'échantillonnage, à laquelle est associée une fréquence $f_e = \frac{1}{T_e}$;
- **Points** : correspond au nombre de valeurs numériques acquises ;
- **Total** : donne le temps d'échantillonnage : **Total** = **Te** × **Points**.

Soit $u(t) = U_0 \cos(\omega t)$ une tension de fréquence $f = \frac{\omega}{2\pi}$. Rappeler la condition sur f et f_e pour qu'il n'y ait pas de repliement de spectre lors de l'acquisition numérique de $u(t)$.

Connecter le GBF à l'interface EUROSART afin de visualiser $u(t)$ avec le logiciel LAVIS PRO (on pourra choisir le mode d'acquisition **permanent** afin de rendre l'observation plus confortable). Choisir alors f et f_e de manière à visualiser un repliement de spectre.

Appeler l'examineur

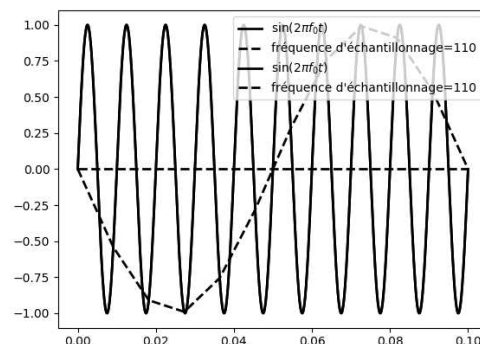
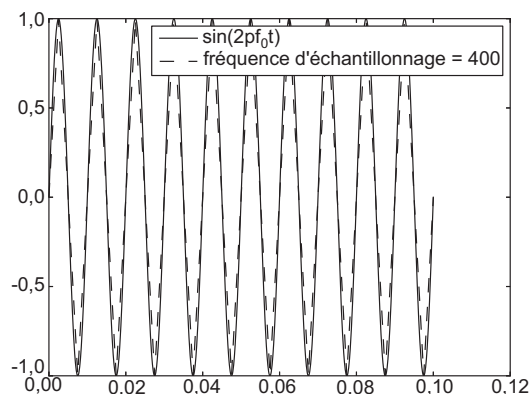
Présenter à l'examineur les courbes obtenues et expliquer clairement en quoi elles illustrent le repliement de spectre.

2. Simulation numérique

Écrire, sous Python, un script permettant de visualiser simultanément :

- un signal $s(t) = \sin(2\pi f_0 t)$ de fréquence $f_0 = 100$ Hz, pour $t \in [0; 10]$ s ;
- le signal échantillonné $s_e(t)$ à partir de $s(t)$, avec une fréquence d'échantillonnage f_e que l'on pourra modifier

En modifiant la fréquence d'échantillonnage, visualiser le phénomène de repliement de spectre.



Afin de parvenir à ces courbes, on pourra s'inspirer du script suivant, dans lequel certaines lignes ont été volontairement effacées, ou écrire un script original :

```

1 from pylab import*
2 def s(t):
3     f0=100
4     return sin(2*pi*f0*t)
5 fe=400
6
7
8
9 plot(t,s(t),label="\sin(2\pi f_0 t)",color="black",linewidth=2)
10 plot(T,S(T),color="black",linewidth=2,linestyle='--',label="fréquence d'échantillonnage=110")
11 legend(loc=1)
12 show()

```

Question

Imprimer les courbes obtenues et les interpréter en termes de repliement de spectre. Comparer notamment $f_e - f_0$ à la fréquence du signal échantillonné.

II- Conversion numérique/analogique (CNA)

Notre environnement est régi par des informations analogiques pouvant prendre une multitude d'états. Un ordinateur (ou tout autre système informatique) est géré par des informations binaires, ou encore logiques, ne pouvant prendre que deux états : 0 ou 1. Afin d'émettre des informations vers le monde extérieur, l'ordinateur doit générer des informations analogiques depuis ses propres informations binaires. C'est le rôle du convertisseur numérique-analogique (CNA). Un CNA a donc pour fonction de convertir des valeurs binaires en informations analogiques. Tous les systèmes analogiques devant communiquer avec le monde extérieur en sont automatiquement équipés.

1. Approche théorique

Cette partie a pour objectif de comprendre la transformation qui fait passer d'un nombre binaire N codé sur n bits à un nombre s décimal. Cette étude sera limitée au type de convertisseur à réseau $R/2R$ et, dans un premier temps, on choisira $n = 4$ bits. Ainsi, N est représenté par une séquence $[\varepsilon_3, \varepsilon_2, \varepsilon_1, \varepsilon_0]$ où $\varepsilon_i = 0$ ou 1 (par exemple : $[1, 0, 0, 1]$) à laquelle sera associée une tension s .

Le réseau utilisé sera celui d'une platine (Plana3) alimentée sous une tension $U_0 = 5,12$ V ; les entrées de ce réseau seront alors alimentées sous des tensions $a_i = \varepsilon_i \times U_0 = 5,12$ V ou 0 V, selon la disposition des interrupteurs, c'est-à-dire selon la séquence imposée par N :

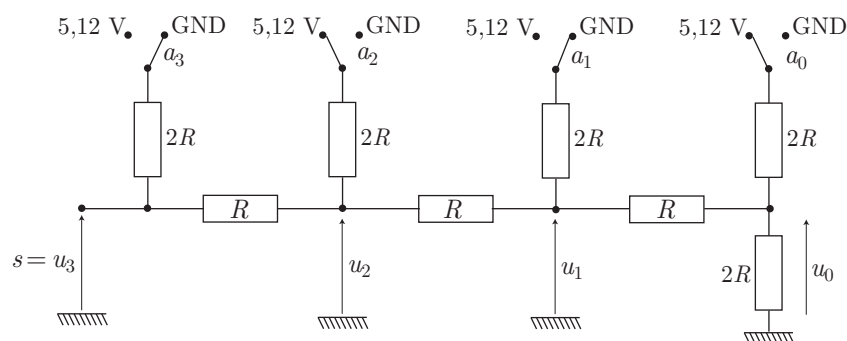


Figure 1

En utilisant la méthode de votre choix, établir le système de 4 équations :

$$\begin{cases} 4u_0 - 2u_1 = a_0 \\ -2u_0 + 5u_1 - 2u_2 = a_1 \\ -2u_1 + 5u_2 - 2u_3 = a_2 \\ -2u_2 + 3u_3 = a_3 \end{cases}$$

Une combinaison linéaire de ces équations permet de se débarrasser des termes des termes u_0, u_1, u_2 afin de ne conserver que :

$$s = u_3 = K(8a_3 + 4a_2 + 2a_1 + a_0)$$

Question

Chercher cette combinaison linéaire et donner la valeur de la constante K' qui permet de convertir le nombre binaire N :

$$s = K' (8\varepsilon_3 + 4\varepsilon_2 + 2\varepsilon_1 + \varepsilon_0) \text{ pour } N = [\varepsilon_3, \varepsilon_2, \varepsilon_1, \varepsilon_0] \quad (1)$$

La table de vérité ci-dessous comporte 8 colonnes :

- les colonnes 1, 2, 3, 4 comportent les valeurs respectives $\varepsilon_3, \varepsilon_2, \varepsilon_1, \varepsilon_0$ des bits qui composent le nombre N ;
- la colonne 5 comportera les valeurs théoriques s_{th} de s , issues de la formule (1);
- les colonnes 6 et 7 comporteront les valeurs expérimentales s_{exp} de s qui seront obtenues ultérieurement (lors de l'approche expérimentale de la partie II-2.);
- la colonne 8 donne la conversion décimale du nombre N .

1	2	3	4	5	6	7	8
ε_3	ε_2	ε_1	ε_0	s_{th}	s_{exp}	s_{exp}	décimal
0	0	0	1				1
0	0	1	0				2
0	0	1	1				3
0	1	1	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				

Remplir la colonne 5 (il pourrait être utile de composer un petit script Python pour y parvenir) et compléter la colonne 8.

La **résolution numérique** r d'un CNA désigne le plus petit écart qui sépare deux valeurs de s , pour un nombre binaire composé de n bits.

Question

Exprimer r en fonction de U_0 et de n .

Le tableau précédent confirme-t-il cette formule ?

La **limite maximale en tension** désigne la valeur maximale s_{max} que peut prendre la tension s associée à un nombre binaire N de n bits.

Question

Exprimer s_{max} en fonction de U_0 et de n .

Le tableau précédent confirme-t-il cette formule ?

On admettra que pour un nombre N de 8 bits, la formule (1) peut être généralisée (avec la même valeur U_0) :

$$s = K' \sum_{k=0}^7 2^k \varepsilon_k \text{ pour } N = [\varepsilon_7, \varepsilon_6, \varepsilon_5, \varepsilon_4, \varepsilon_3, \varepsilon_2, \varepsilon_1, \varepsilon_0] \quad (2)$$

Question

Quelle est l'expression de K' et sa valeur numérique ?

La table de vérité doit alors être agrandie en proportions :

1	2	3	4	5	6	7	8	9	10	11
ε_7	ε_6	ε_5	ε_4	ε_3	ε_2	ε_1	ε_0	s_{th}	s_{exp}	décimal
0	0	0	0	0	0	0	0			0
0	0	0	0	0	0	0	1			1
0	0	0	0	0	0	1	0			2
0	0	0	0	1	0	1	0			
0	0	0	0	1	0	1	1			
0	0	0	1	0	0	1	1			
0	0	0	1	1	0	0	1			
0	0	1	1	0	0	0	1			
1	1	1	1	1	1	1	0			
1	1	1	1	1	1	1	1			

Remplir les colonnes 9 et 11 de ce tableau (peut être en s'aidant d'un petit script Python).

Question

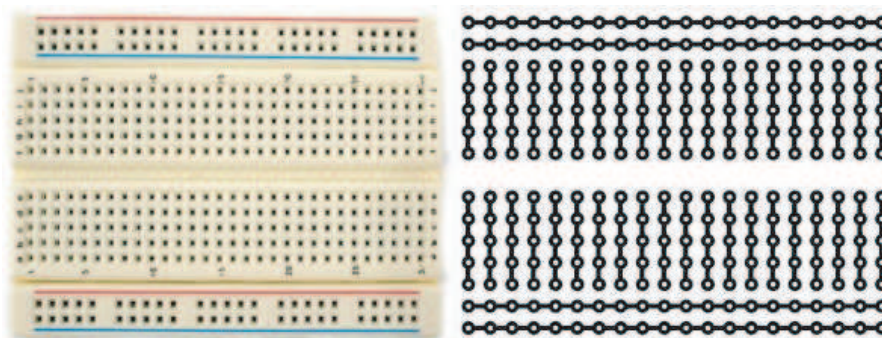
À l'aide de ce tableau, évaluer la résolution en tension r et la limite maximale en tension. Ces valeurs sont-elles compatibles avec les expressions théoriques ?

2. Approche expérimentale

L'objectif de cette partie est d'illustrer le fonctionnement réel du montage schématisé sur la figure 1. Dans un premier temps, ce montage sera réalisé sur une plaque d'expérimentation (appelée *breadboard*) et sera limité à 4 bits, puis un CNA (platine de conversion Plana) sera mise en œuvre, avec possibilité d'étudier la conversion analogique de nombres binaires de 4 ou 8 bits.

a- Réalisation du convertisseur

Au laboratoire, avant d'être soudés sur des circuits imprimés, les circuits électroniques sont fréquemment testés sur une *platine d'expérimentation* ; il s'agit d'un dispositif équipé de trous accueillant les composants électroniques et dont les connexions sont représentées ci-dessous (les traits pleins figurent les connexions électriques entre les divers trous), lesquelles connexions ne sont pas visibles par le haut de la platine.



Plateau de connexions

Schéma des connexions

Des bornes équipées de fils conducteurs assurent également le branchement de la platine à des éléments extérieurs (générateurs, oscilloscope, boîtes à décades, ...).

Réaliser, sur la *breadboard*, le montage schématisé sur la figure 1, dans lequel :

- les résistances R et $2R$ prennent respectivement les valeurs $1\text{ k}\Omega$ et $2\text{ k}\Omega$;
- les interrupteurs des bits ε_0 , ε_1 , ε_2 , ε_3 seront simplement matérialisés par des fils électriques, qui seront reliés soit à l'alimentation de tension U_0 , soit à la masse ;
- la tension U_0 est délivrée par un générateur de tension stabilisée, réglée à la valeur $U_0 = 5,12\text{ V}$;
- un voltmètre permet d'ajuster la valeur de U_0 et de mesurer la valeur de s .

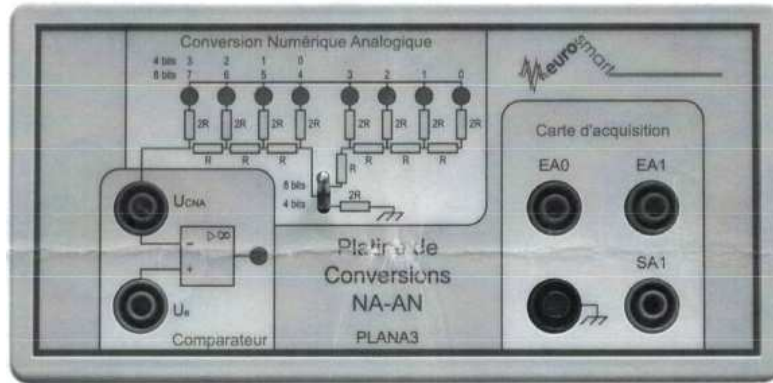
Remplir alors la colonne 6 du tableau de la page 3.

Question

Les valeurs obtenues sont-elles compatibles avec celles attendues ?

b- Utilisation de la platine de CNA Plana

La conversion analogique/numérique peut être effectuée directement à l'aide d'un convertisseur, disponible dans la salle de TP sous la forme d'une platine (Plana) :



Cette platine permet de travailler sur 4 ou 8 bits de résolution. Cette possibilité permet d'étudier et de comparer le pas de conversion, précision et autres valeurs typiques de ces deux modes de travail. Connecter la platine Plana à l'interface Sysam :

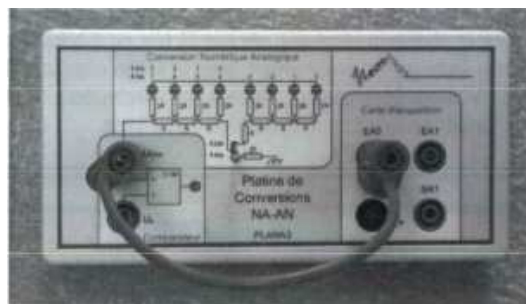


La platine est alors automatiquement alimentée. Les entrées EA0 et EA1 sont directement reliées aux entrées correspondantes de l'interface Sysam. De même, un signal délivré par cette interface peut être récupéré sur la sortie SA1 de la platine Plana.

Exécuter Latis Plana, puis sélectionner la plage CNA en cliquant sur le bouton 

Sélectionner un mode de travail sur 4 bits, en positionnant correctement l'interrupteur prévu à cet effet sur Plana.

Relier, sur la platine, la borne repérée par U_{cna} à la borne EA0.

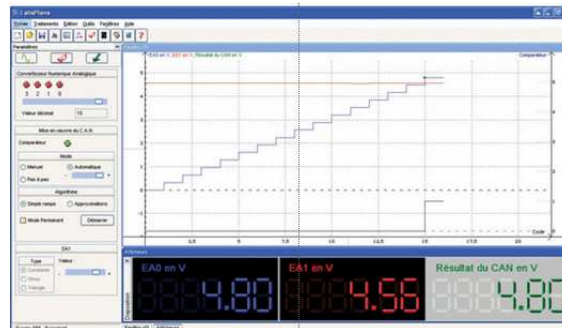


Ainsi, la tension s générée par le convertisseur N/A va être envoyée sur l'entrée EA0 de la carte Sysam, puis à l'ordinateur. Dans cette configuration, le «comparateur» de la platine n'est pas utilisé (il sera étudié plus loin dans le TP).

Pour activer une porte logique, cliquer à l'aide de la souris sur le voyant lumineux correspondant dans la fenêtre :

Convertisseur Numérique Analogique

Pour chaque combinaison réalisée, la tension s est visualisée dans l'afficheur EA0 de Latis Plana (en bas de l'écran de l'ordinateur).



Il est également possible d'utiliser le *barregraphe*, ou les flèches Haut et Bas du clavier, après avoir positionné le curseur dans le champ **Valeur Décimale**.

En utilisant le convertisseur N/A de Latis Plana, générer les nombres binaires proposés dans le tableau de la page (3) puis remplir la colonne 7. On fournira ce tableau avec le compte-rendu (en recomplant le tableau ou en joignant à la copie celui fourni en annexe 4)

Question

1. Évaluer à nouveau la résolution en tension r ainsi que la limite maximale en tension. Conclure.
2. Comparer et commenter les valeurs consignées dans les colonnes (5), (6), (7).

Sélectionner enfin un mode de travail sur 8 bits, puis recommencer l'expérience précédente, afin de remplir la colonne (10) du tableau de la page 4 (le tableau ainsi rempli sera rendu avec le compte-rendu ; on trouvera un exemplaire détachable de ce tableau en annexe 4). Répondre alors aux mêmes questions :

Questions

1. Évaluer la résolution en tension r pour $n = 8$ bits, ainsi que la limite maximale en tension. Conclure.
2. Comparer et commenter les valeurs consignées dans les colonnes (9), (10).

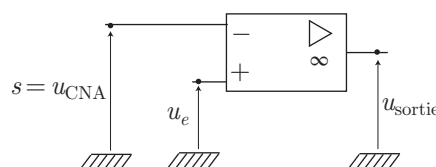
Question

En conclusion, sur quels paramètres faut-il jouer pour améliorer la résolution en tension ?

III- Conversion Analogique/Numérique

Il s'agit ici d'effectuer la conversion inverse de la précédente : une tension s (en volt) doit être convertie en nombre binaire N à n bits ($n = 4$ ou 8 dans ce TP).

Trois méthodes vont être présentées dans ce TP : la *conversion simple rampe*, la *conversion par approximations* et la *conversion parallèle* (à l'aide d'Amplificateurs Opérationnels¹ en mode comparateurs). Quel que soit le mode de conversion adopté, il mettra en œuvre un module comparateur (son étude sera abordée dans la partie IV) :



1. Encore appelés ALI : Amplificateur Linéaire Intégré.

À cette étape du TP, il suffit de retenir que la tension de sortie u_{sortie} de ce module présente un signe qui dépend des entrées s et u_e :

- si $u_e > s$, u_{sortie} est positive ($u_{\text{sortie}} = +V_{\text{sat}}$ où $V_{\text{sat}} \simeq 15 \text{ V}$) ;
- si $u_e < s$, u_{sortie} est négative ($u_{\text{sortie}} = -V_{\text{sat}}$).

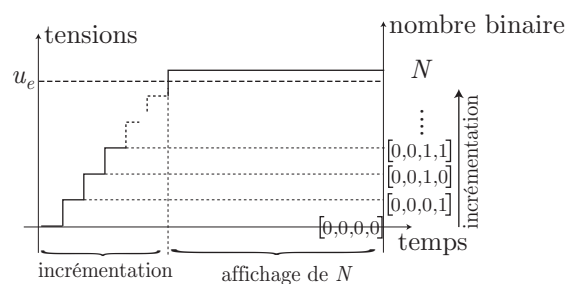
Ainsi, lorsque u_{sortie} change de signe, cela signifie que s est devenu plus grand ou plus petit que u_e .

1. Conversion simple rampe

a- Présentation

La tension à convertir (u_e) alimente l'entrée (+) du comparateur. Un dispositif numérique crée un nombre binaire $N = [0, 0, 0, 0]$ (pour $n = 4$ bits) et la convertit en tension $s = u_{CNA}$ qui est envoyée sur l'entrée (-) du comparateur. Si $u_{CNA} < u_e$ (ce qui est probable en début de conversion), un dispositif *incrémenteur* ajoute une unité à N , qui devient $[0, 0, 0, 1]$. Si $u_{CNA} < u_e$, le processus se reproduit ($[0, 0, 1, 0] \rightarrow [0, 0, 1, 1] \rightarrow \dots$) jusqu'à ce que $s = u_{CNA}$ atteigne la valeur u_e : le signe négatif de u_{sortie} indique :

- qu'il faut cesser l'incrémentation de N ;
- qu'il faut afficher la valeur de N . C'est le résultat de la conversion.



b- Approche numérique

Nous cherchons ici à mettre en œuvre la méthode présentée précédemment à l'aide d'un script Python : en précisant préalablement le mode de conversion (4 ou 8 bits), le script devra afficher le nombre binaire N associé à toute tension (appelée "décimale") u_e comprise entre 0 V et $U_0 = 5,12 \text{ V}$.

Pour cela, il va d'abord être nécessaire de concevoir une fonction qui incrémente tout nombre binaire N qui lui est fourni en entrée.

Écrire un tel script et vérifier son fonctionnement.

Par exemple, dans le script ci-dessous (privé de quelques instructions), la fonction `addition(n-1,N)` fournit le nombre N incrémenté d'une unité binaire, dans un mode de conversion à n bits :

```

1 #Fonction qui incrémente le nombre binaire N
2 #Lancer addition(Nbit-1,N) pour incrémenter N
3 def addition(i,N):
4     if N[i]==0:
5
6     else :
7
8
9     return(N)

```

En choisissant comme valeur initiale $N = [0, 0, 0, 0]$, l'appel successif de cette fonction sur N a fourni les résultats suivants :

```

> > > addition(3,N)
{0,0,0,1}
> > > addition(3,N)
{0,0,1,0}
> > > addition(3,N)
{0,0,1,1}
> > > addition(3,N)
[0,1,0,0]

```

Écrire ensuite un script (ou compléter celui fourni ci-dessous) qui convertit une tension analogique $u_e = u_{ana}$ en nombre binaire à n bits. Par exemple, le script ci-dessous (plutôt les fragments encore existants) définit la fonction `conversion(Uana,Nbit)` qui reçoit la tension analogique $u_e = u_{ana}$ à $n = N$ bits, fait sa conversion en s'inspirant de la méthode simple rampe, puis renvoie comme résultat le nombre binaire N correspondant à

la valeur $u_{CNA} = s$ à partir de laquelle a cessé l'incrémentation :

```

1 Nbit=4#Nombre de bits du nombre binaire (4 ou 8)
2 N=[0]*Nbit#Création du nombre binaire
3
4 #Fonction qui incrémente le nombre binaire N
5 #Lancer addition(Nbit-1,N) pour incrémenter N
6 def addition(i,N):
7     if N[i]==0:
8
9     else :
10
11
12     return(N)
13
14 #Conversion du décimal Uana en binaire N à Nbits
15 def conversion(Uana,Nbit):
16     L=[]
17     N=[0]*Nbit
18     r=5.12/2**Nbit
19     Unum=0
20     if Uana<r:
21         return(N,Unum)
22     else :
23         while Unum<Uana-r:
24             Unum=Unum+r
25             addition(Nbit-1,N)
26         return(N,Unum)

```

Le lancement de ce script sur le nombre décimal 2,3 a fourni les résultat suivants, qui donnent non seulement le nombre N , mais également sa correspondance décimale :

```
> > > conversion(2.3,4)
```

```
([0, 1, 1, 1], 2.24)
```

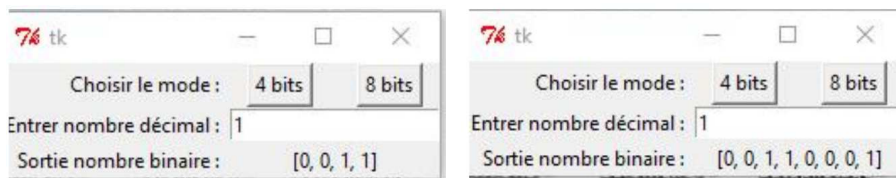
Fournir, avec le compte-rendu du TP, le script ainsi rendu opérationnel, puis l'utiliser pour remplir les colonnes (2) et (4) du tableau suivant, qui donnent pour chaque valeur de U_e (colonne 1) la conversion N (de 4 bits) et le nombre décimal (en base 10) correspondant, à l'aide de votre script.

(1)	(2)	(3)	(4)	(5)
U_e	N_{Python}	N_{Plana}	u_{cna}^{Python} (V)	u_{cna}^{Plana} (V)
0,50 V				
1,00 V	0011		0,96	
1,10 V				
2,00 V				
2,30 V	0111		2,24	
2,50 V				
3,00 V				
4,00 V				
5,00 V				

Utiliser maintenant votre script pour faire des conversion A/N sur 8 bits et remplir les colonnes (7) et (9) du tableau ci-dessous :

(6)	(7)	(8)	(9)	(10)
U_e	N_{Python}	N_{Plana}	u_{cna}^{Python} (V)	u_{cna}^{Plana} (V)
0,50 V				
1,00 V	00110001		0,98	
1,10 V				
2,00 V				
2,30 V				
2,50 V				
3,00 V				
4,00 V				
5,00 V				

Pour les étudiants intéressés par Python, le script peut être rendu plus convivial grâce à l'interface graphique **Tkinter** de Python. Par exemple, la conversion d'un nombre décimal (à entrer par l'utilisateur dans la fenêtre correspondante) dans un mode 4 ou 8 bits (choisi en cliquant sur les boutons correspondants) peut apparaître dans une fenêtre :



On trouvera le script correspondant en annexe 2 (auquel manquent évidemment les instructions relatives à l'incrémention!).

c- Approche expérimentale

Le logiciel Latisplana associé à la platine Plana permet de convertir, par la méthode *simple rampe*, une tension délivrée par l'ordinateur. Le mode opératoire est fourni en annexe 3.

Lire attentivement ce mode d'emploi. Compléter alors les colonnes (3), (5), (8) et (10) des tableaux précédents.

Question

Mettre en œuvre cette mesure pour les tensions $u_{\text{ext}} = 3,00 \text{ V}$ et $u_{\text{ext}} = 6,00 \text{ V}$ délivrées par un générateur de tension stabilisée.

On décrira avec précision les branchements réalisés sur la platine Plana.

Conclure.

2. Conversion par approximations

Cet algorithme de conversion est basé sur une méthode dichotomique pour déterminer la valeur binaire de conversion. Comme dans la méthode précédente, l'algorithme compare la tension la tension U_e à convertir avec la sortie $s = u_{cna}$ du réseau $R - 2R$. La différence entre les deux méthodes réside dans la logique des valeurs proposées par le réseau $R - 2R$. Celle-ci n'est plus basée sur une incrémentation linéaire, mais sur une méthode dichotomique.

La méthode : Pour une valeur de U_e à convertir, l'algorithme propose en comparaison une valeur de u_{cna} égale à la moitié de la valeur maximale qu'elle peut prendre (et qui correspondrait à un niveau haut sur tous les bits logiques du réseau $R - 2R$ disponibles). Sur 8 bits, la valeur proposée est alors (en décimal) : $\frac{256}{2} = 128$.

Le système compare alors les deux valeurs et indique si u_{cna} est supérieure ou inférieure à U_e . En fonction de cette réponse, le réseau $R - 2R$ fournit une nouvelle tension u_{cna} incrémentée ou décrétementée du quart de la valeur maximale que peut prendre u_{cna} .

Le système compare à nouveau les deux tensions et indique si u_{cna} est supérieur ou inférieur à U_e . En fonction de cette réponse, le réseau $R - 2R$ fournit une nouvelle tension u_{cna} incrémentée ou décrétementée du huitième de la valeur maximale que peut prendre u_{cna} .

Et ainsi de suite jusqu'à un nombre de comparaisons égal au nombre de bits utilisés dans cette conversion.

Exemple sur 4 bits

1. $U_e = 4$ V
2. u_{cna} prend la valeur décimale 8 ($= 16/2$), soit une valeur de tension 2,56 V
3. Le système répond que $U_e > u_{cna}$
4. u_{cna} est incrémenté en décimal de la valeur 4 ($= 16/4$) et prend alors une valeur décimale 12 (3,84 V en tension)
5. Le système répond que $U_e < u_{cna}$
6. u_{cna} est incrémenté en décimal de la valeur 2 ($= 16/8$) et prend alors la valeur décimale 14 (4,48 V en tension)
7. Le système répond que $U_e < u_{cna}$
8. u_{cna} est décrémenté en décimal de la valeur 1 ($= 16/16$) et prend alors la valeur décimale 13 (4,16 V en tension)

La conversion est terminée.

On constate que la donnée est bien convertie sur un nombre de coups égal au nombre de bits. Si le travail est réalisé sur 8 bits, il faudra 8 coups pour convertir la donnée analogique. Cette procédure a la particularité d'être très rapide en comparaison avec la méthode *simple rampe*.

b- Approche expérimentale

Sélectionner sur Plana3 une résolution de 8 bits.

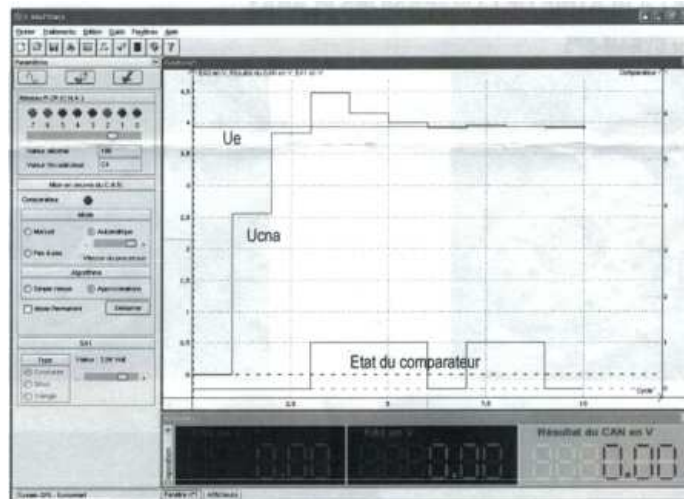
Effectuer sur la platine les câblages tels que spécifiés en annexe 3.

Dans le logiciel Latis Plana :

- sélectionner le mode **Automatique** et l'algorithme **Approximations**.
- À l'aide du potentiomètre disponible au champ **Sorties A1**, régler une tension quelconque à convertir.

Démarrer la conversion en validant le bouton **Depart**.

L'algorithme contrôle alors les valeurs obtenues par dichotomie et les teste afin d'effectuer l'encadrement de la valeur. Sur une résolution de 8 bits, l'algorithme propose 8 valeurs à tester pour déterminer la valeur finale de conversion.



La valeur binaire de conversion est indiquée par des DEL sur Latis Plana (et Plana3) et sa valeur décimale est affichée dans le champ **Valeur Décimale**.

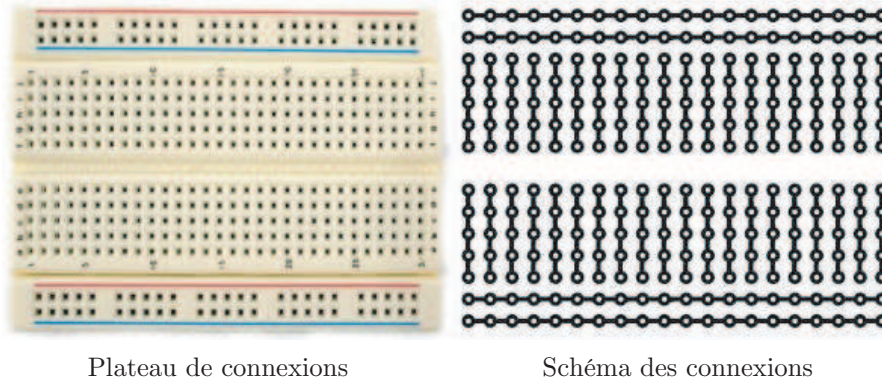
Question

Réaliser la manipulation sur 8 bits et sur 4 bits puis comparer les résultats en termes de précision et de limites.

IV- Bloc comparateur

1. Présentation

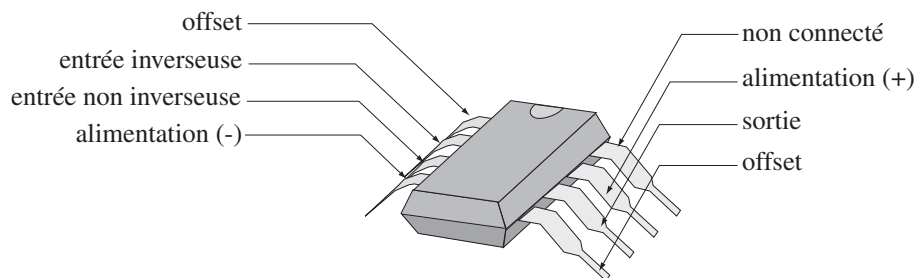
La comparaison des tensions se fera à l'aide d'un Amplificateur Linéaire Intégré (ALI)². Ce composant peut être connecté à la platine d'expérimentation (aussi appelée *breadboard*) lorsque le nombre de fils de connexion est important.



Plateau de connexions

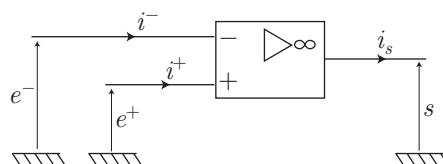
Schéma des connexions

à condition d'identifier les différentes bornes (toujours à partir d'une encoche faite en haut du boîtier) :



Attention : l'ALI doit être alimenté à l'aide d'une source de tension stabilisée symétrique +15 V ; 0 V ; -15 V. Une erreur de connexion à cette étape endommagera instantanément le composant. C'est pourquoi cette opération est la première à être réalisée, avec attention.

Conventionnellement, l'ALI est schématisé de la manière suivante (on omet de représenter les alimentations +15 V et -15 V) :



Dans cette configuration, ce composant est caractérisé par :

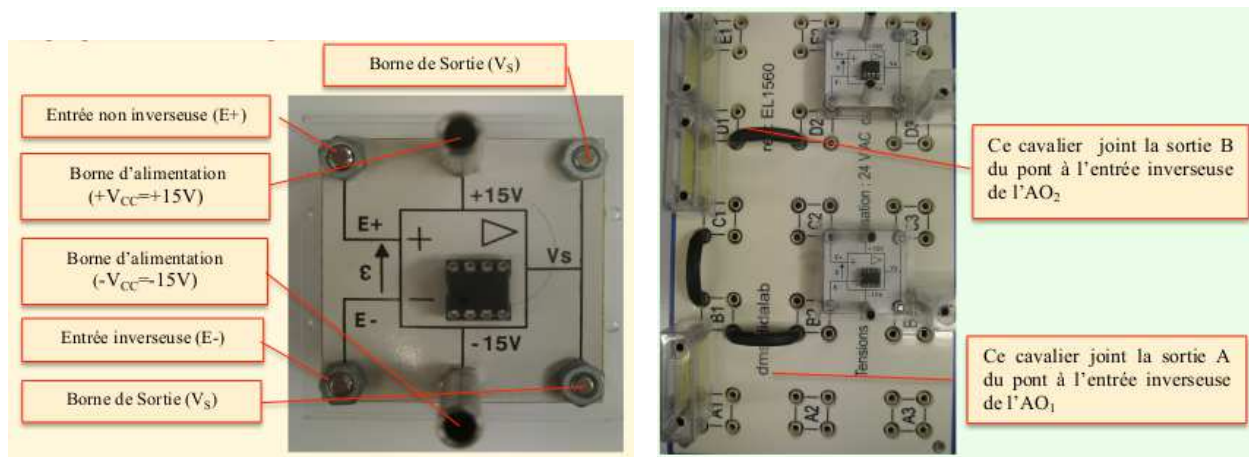
- des courants d'entrée nuls : $i^- = i^+ = 0$;
- une tension de sortie s dont le signe dépend des valeurs relatives des tensions d'entrée e^+ et e^- :

$$\begin{cases} s = V_{\text{sat}} & \text{si } e^+ > e^- \\ s = -V_{\text{sat}} & \text{si } e^+ < e^- \end{cases} \quad \text{où } V_{\text{sat}} \text{ est une tension constante de saturation.}$$

C'est dans cette configuration qu'il joue son rôle de comparateur des tensions e^+ et e^- .

Dans les montage qui exigent peu de fils électriques, on pourra utiliser des ALI montés sur plaquette, elles-même connectées au reste du circuit par une plaquette (DMS Didalab EL 1500) :

². Autrefois appelé Amplificateur Opérationnel (A.O.)



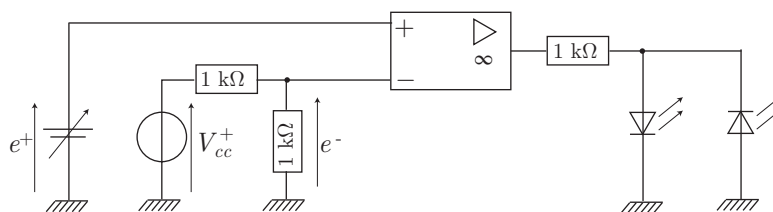
Une telle option semble simplifier les montages, mais :

- elle exige beaucoup de place ;
- elle favorise l'apparition de faux-contacts que l'étudiant devra apprendre à repérer (et réparer) par lui-même ;
- elle ne sera pas toujours disponible le jour du concours.

C'est pourquoi on apprendra, dans cette partie, à réaliser les deux modes de connexion.

2. Le comparateur simple

En utilisant les ALI montés sur plaquette, ainsi que les plaquette DMS Didalab, réaliser le montage schématisé ci-dessous :



dans lequel :

- $V_{cc}^+ = 15 \text{ V}$ est la tension délivrée par l'alimentation stabilisée symétrique de l'ALI ;
- e^+ est une tension réglable de 0 à 15 V délivrée par un générateur de tension stabilisée ;
- les DEL rouges permettent d'identifier rapidement l'état de sortie de l'ALI.

De nombreux faux-contacts peuvent se produire sur les plaquettes, ainsi que dans les fils gainés. Les étudiants devront mettre en œuvre des stratégies permettant de les repérer rapidement : par exemple, on ne réalisera pas de circuit complexe sans contrôler chaque sous-partie du circuit.

Questions

Mesurer au voltmètre la tension e^- et interpréter théoriquement cette valeur.

Faire varier la tension e^+ de 0 à 15 V et interpréter les observations.

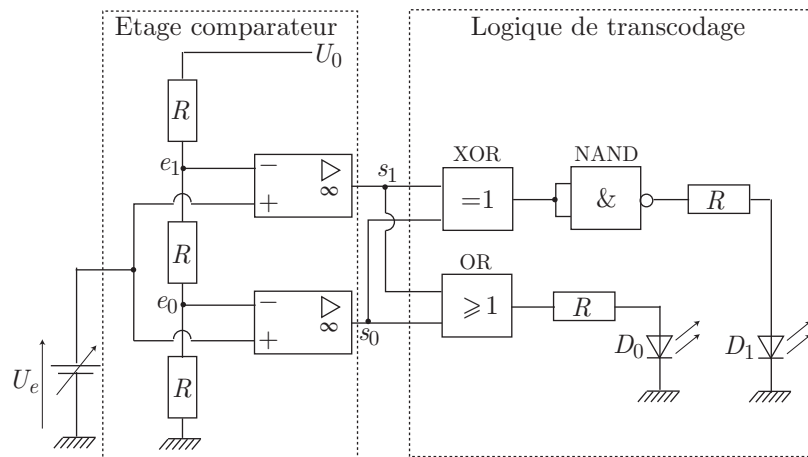
3. Conversion Analogique/Numérique

a- Aspects théoriques

Ce type de convertisseur³ utilise de nombreux ALI, ainsi que des portes logiques. Afin de simplifier l'étude, nous limiterons la conversion à 2 bits : Dans ce dispositif, on distingue :

- un étage «comparateur», composé d'ALI et de résistances R identiques, qui transforment le signal analogique (tension u_e) en signaux binaires s_1 et s_2 (les états **haut** et **bas**, soit 0 et 1, correspondent aux tensions de saturation respectives $+V_{\text{sat}}$ et $-V_{\text{sat}}$ des ALI) ;
- un bloc de «logique de transcodage» associant des composants de logique booléenne (présentés plus bas).

3. Aussi appelé *convertisseur parallèle* (ou *flash converter*), il réalise le type de conversion le plus rapide puisque tous les bits sont déterminés en même temps. Cette rapidité s'obtient cependant au détriment du nombre de composants nécessaires, lequel progresse en 2^n .



On s'intéresse, dans un premier temps, au bloc comparateur.

Question

Exprimer les potentiels e_0 et e_1 en fonction de U_0 .

On rappelle que le potentiel d'un point représente la tension entre ce point et la masse du circuit.

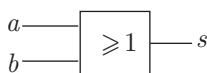
Dans la pratique, U_0 prendra la valeur $U_0 = 15 \text{ V}$ et U_e variera entre 0 et 15 V.

Remplir dans le tableau ci-dessous, les colonnes 2 et 3 (qui donnent les potentiels e_0 et e_1 en volt) ainsi que les colonnes 4 et 5, qui donnent les états des sorties (0 ou 1 pour les états **bas** ou **haut**).

1	2	3	4	5	6	7
U_e	e_1 (V)	e_0 (V)	s_1	s_0	D_0	D_1
2 V						
7 V						
12 V						

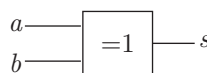
Quant aux portes logiques qui composent le *bloc logique de transcodage*, elles sont caractérisées par leur *tables de vérités* :

Porte OU (OR)



a	b	$s = a + b$
0	0	0
0	1	1
1	0	1
1	1	1

Porte OU EXCLUSIF (XOR)



a	b	$s = a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

Porte NON ET (NAND)



a	b	$s = \overline{a \cdot b}$
0	0	1
0	1	1
1	0	1
1	1	0

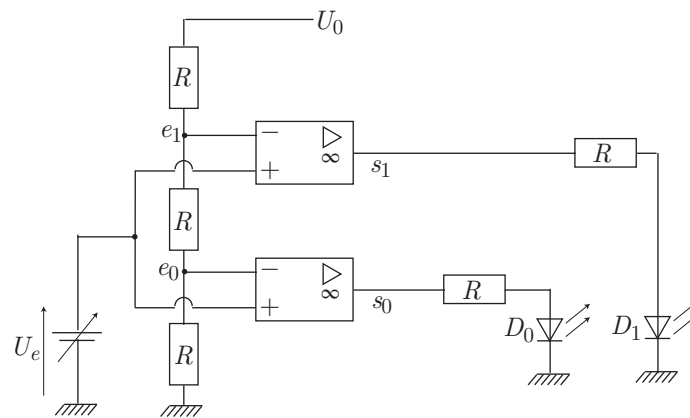
Compléter les colonnes 6 et 7 du tableau précédent, donnant l'état des diodes D_0 et D_1 (0 ou 1). Afin de ne pas recopier ce tableau, on pourra remplir et rendre le tableau fourni en annexe 4.

b- Approche expérimentale

Afin de simplifier les montages :

- l'étude portera uniquement sur l'étage comparateur ;
- les branchements se feront sur une plaquette *breadboard*.

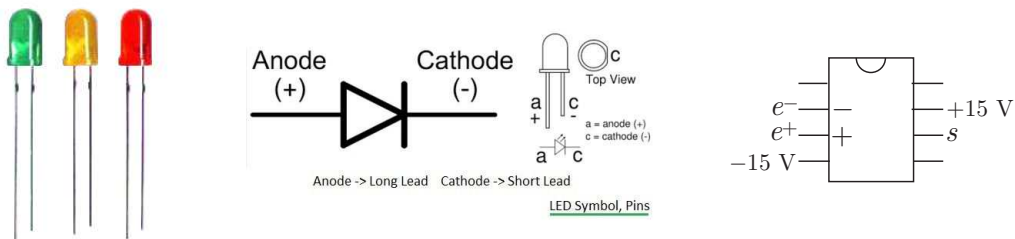
Réaliser l'étage comparateur schématisé ci-dessous :



dans lequel :

- les résistances prennent toutes la valeur $R = 1 \text{ k}\Omega$;
- la tension U_e est délivrée par un générateur de tension stabilisée, réglable de 0 à 15 V ;
- la tension $U_0 = 15 \text{ V}$ sera délivrée par l'alimentation symétrique (+15V ; 0V ; -15V) des ALI ;
- l'état des sorties s_1 et s_2 sera analysé grâce aux DEL D_0 et D_1 ;
- les tensions U_e , s_1 , s_2 seront mesurées à l'aide de voltmètres ;
- les deux ALI sont connectés sur la plaque *breaboard*.

On rappelle les connexions des DEL et des ALI :



Le montage exige de nombreux fils et connexions qui seront autant de sources de faux contacts et d'erreurs. Pour les identifier à temps, il est recommandé de vérifier, isolément, chaque portion de circuit (comparateur, diviseur de tensions, LED...) avant des les raccorder les unes aux autres.

Faire varier la tension U_e de 0 à 15 V et remplir les colonnes 2,3 (en volt) et 4,5 (avec 0 ou 1 selon l'état des diodes) du tableau ci-dessous :

1	2	3	4	5
U_e	s_0 (en V)	s_1 (en V)	D_0	D_1
2 V				
7 V				
12 V				

Questions

Commenter les résultats obtenus, notamment :

- en les comparant à ceux de la page 13 (dernier tableau) ;
- en évaluant les tensions U_e qui permettent le changement d'état des diodes ;
- en évaluant la précision de la conversion.

Matériel disponible

PAILLASSES ÉLÈVES GROUPE A

- 1 ordinateur équipé de Latis Pro, Latis Plana et Python ;
- 1 imprimante reliée à l'ordinateur ;
- 1 interface Sysam SP5 ;
- 1 GBF ;
- 1 oscilloscope numérique ;
- 1 platine Plana 3 ;
- 1 générateur de tension stabilisée réglable de 0 à 15 V ;
- 1 voltmètre numérique.

PAILLASSES ÉLÈVES GROUPE B

- 1 ordinateur équipé de Python et d'une imprimante ;
- 5 résistances de 1 k Ω ;
- 5 résistances de 2 k Ω ;
- 2 DEL rouges ;
- 2 ALI
- 1 platine *breadboard* + nombreux petits fils électriques ;
- 1 générateur de tension stabilisée réglable de 0 à 15 V ;
- 2 multimètres numériques ;
- 1 alimentation stabilisée symétrique +15V ; 0V ; -15V ;
- 1 plaquette DMS Didalab EL 1560 + dipôles enfichables :
 - 1 ALI sur plaquette (+3 cavaliers) ;
 - 2 DEL rouges ;
 - 3 résistances de 1 k Ω

PAILLASSES PROFESSEUR

- Plusieurs ALI de réserve.

Répartition des groupes A et B

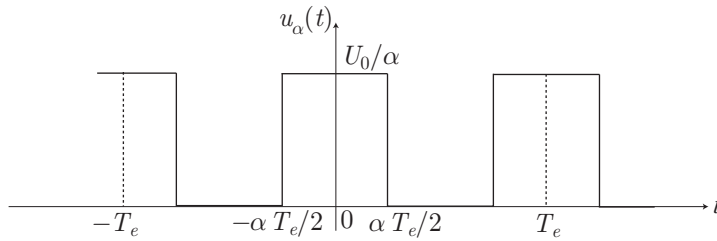
I- Repliement de spectre	
1. Visualisation à l'écran de l'ordinateur	Groupe A
2. Simulation numérique	
II- Conversion Numérique/Analogique	
1. Approche théorique	
2. Approche expérimentale	
a- Réalisation du convertisseur	Groupe B
b- Utilisation de la platine	Groupe A
III- Conversion Analogique/Numérique	
1. Conversion simple rampe	
a- Présentation	
b- Approche numérique	Groupes A & B
c- Approche expérimentale	Groupe A
2. Conversion par approximations	
a- Approche théorique	Groupe A
b- Approche expérimentale	
IV- Bloc comparateur	
1. Présentation	
2. Comparateur simple	
3. Convertisseur Analogique/Numérique	Groupe B
a- Aspects théoriques	
b- Approche expérimentale	

ANNEXE 1

Repliement du spectre

1. Peigne de Dirac

Soit $u_\alpha(t)$ un échelon de période T_e , de largeur $\frac{\alpha T_e}{2}$ (où α est un paramètre réel sans dimension) et de hauteur $\frac{U_0}{\alpha} / U_0 = 1$ V :



Puisque cette fonction est symétrique par rapport à l'axe des ordonnées, son développement en série de Fourier ne comporte que des fonctions paires :

$$u_\alpha(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos(n\omega_e t) \text{ où } \omega_e = \frac{2\pi}{T_e}$$

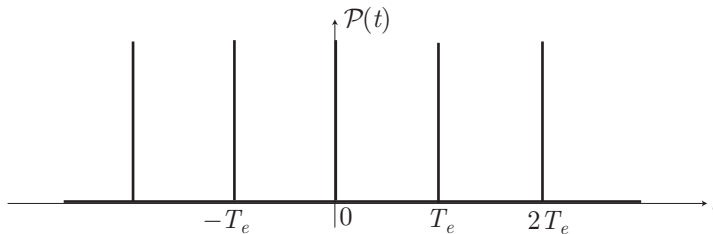
Les coefficients a_0 et a_n valent alors :

$$a_0 = \frac{1}{T_e} \int_{-T_e/2}^{T_e/2} u_\alpha(t) dt = \frac{1}{T_e} \times \frac{U_0}{\alpha} \times \alpha T_e = U_0$$

et :

$$\begin{aligned} a_n &= \frac{2}{T_e} \int_{-T_e/2}^{T_e/2} u_\alpha(t) \cos(n\omega_e t) dt = 2 \times \frac{2}{T_e} \int_0^{T_e/2} u_\alpha(t) \cos(n\omega_e t) dt \\ &= \frac{4}{T_e} \times \frac{U_0}{\alpha} \int_0^{\alpha T_e/2} \cos(n\omega_e t) dt = \frac{4U_0}{\alpha T_e} \times \frac{1}{n\omega_e} \left[\sin \left(n\omega_e \times \alpha \frac{T_e}{2} \right) \right] \\ &= \frac{4U_0}{2\pi n\alpha} \sin(n \times \pi\alpha) \text{ car } \omega_e T_e = 2\pi \\ &= 2U_0 \times \frac{\sin(n\pi\alpha)}{n\pi\alpha} = 2U_0 \operatorname{sinc}(n\pi\alpha) \end{aligned}$$

Le *peigne de Dirac* est une limite de $u_\alpha(t)$ lorsque α tend vers zéro : $\mathcal{P}(t) = \lim_{\alpha \rightarrow 0} u_\alpha(t)$:



En remarquant que :

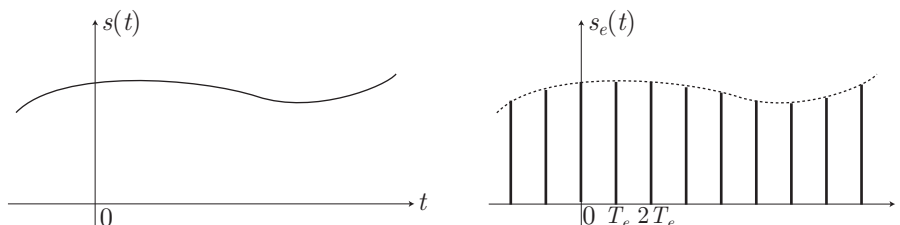
$$\lim_{\alpha \rightarrow 0} a_n = 2U_0 \lim_{\alpha \rightarrow 0} [\operatorname{sinc}(n\pi\alpha)] = 2U_0$$

le peigne de Dirac présente le développement en série de Fourier :

$$\mathcal{P}(t) = U_0 + \sum_{n=1}^{\infty} 2U_0 \cos(n\omega_e t)$$

2. Signal échantillonné

Soit $s(t)$ un signal analogique, qui doit être échantillonné avec une fréquence $f_e = \frac{1}{T_e}$. Cet échantillonnage peut être réalisé à l'aide du peigne de Dirac :



Le signal échantillonné s'écrit alors :

$$s_e(t) = s(t) \times \mathcal{P}(t) \quad (3)$$

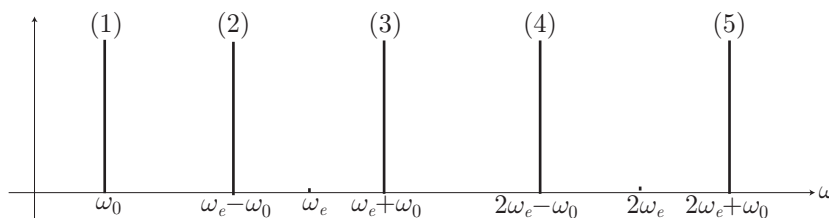
Pour simplifier les calculs, supposons $s(t)$ purement sinusoïdal : $s(t) = S_0 \cos(\omega_0 t)$. Le signal échantillonné correspondant s'écrit :

$$s_e(t) = U_0 S_0 \cos(\omega_0 t) + \sum_{n=1}^{\infty} 2S_0 U_0 \cos(n\omega_e t) \cos(\omega_0 t)$$

c'est-à-dire, puisque $\cos a \times \cos b = \frac{1}{2} [\cos(a+b) + \cos(a-b)]$:

$$s_e(t) = U_0 S_0 \cos(\omega_0 t) + \sum_{n=1}^{\infty} S_0 U_0 \cos[(n\omega_e + \omega_0)t] + \sum_{n=1}^{\infty} S_0 U_0 \cos[(n\omega_e - \omega_0)t]$$

Le spectre de ce signal comporte alors plusieurs composantes :

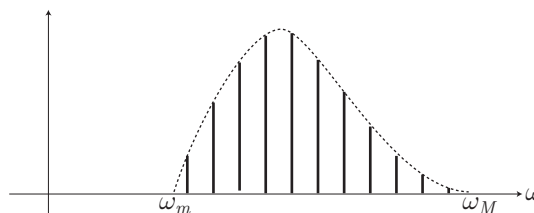


Lorsque le pic (2) est situé à gauche du pic (1), il y a recouvrement du spectre : le pic (2) apparaît dans le spectre du signal comme s'il composait le signal $s(t)$ (il y a confusion entre les pics (1) et (2)). Dans ce cas, une conversion numérique-analogique produirait un signal possédant dans son spectre un signal de fréquence $\omega_e - \omega_0$. Pour éviter qu'une telle situation se produise, il faut que $\omega_e - \omega_0$ demeure supérieur à ω_0 :

$$\omega_e - \omega_0 > \omega_0 \Rightarrow \boxed{f_e > 2f_0} \text{ où } \omega_e = 2\pi f_e \text{ et } \omega_0 = 2\pi f_0$$

Cette inégalité démontre le *critère de Shannon* qui fixe la valeur minimale à la fréquence d'échantillonnage permettant d'éviter le repliement du spectre.

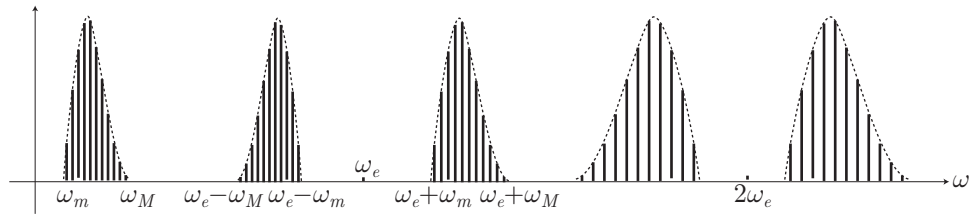
Plus généralement, le spectre du signal $s(t)$ peut présenter des harmoniques dont les pulsations sont comprises entre ω_m et ω_M :



Le produit (3) génère alors un signal :

$$s_e(t) = U_0 \times s(t) + \sum_{n=1}^{\infty} 2U_0 \cos(n\omega_e t) \times s(t)$$

dont le spectre comporte des composantes symétriques par rapport aux pulsations $\omega_e, 2\omega_e, \dots$



Pour éviter le repliement de spectre, il faudra alors assurer :

$$\omega_M < \omega_e - \omega_M \Rightarrow \omega_e > 2\omega_M \Rightarrow \boxed{f_e > 2f_M}$$

ANNEXE 2

Script Python

Conversion sur 4 bits ou 8 bits

```

1  from tkinter import*
2  Nbit=4
3  #Fonctions associées aux boutons bou1 et bou2
4  def quatre():
5      global Nbit
6      Nbit=4
7  def huit():
8      global Nbit
9      Nbit=8
10 N=[0]*Nbit#Création du nombre binaire
11 #Fonction qui incrémente le nombre binaire N
12 def addition(i,N):
13     IL MANQUE DES LIGNES ICI !
14     return(N)
15 #Conversion du décimal Uana en binaire N à Nbits
16 def conversion(Uana,Nbit):
17     L=[]
18     N=[0]*Nbit
19     r=5.12/2**Nbit
20     Unum=0
21     Uana=float(Uana)
22     if Uana<r:
23         return(N,Unum)
24     else :
25         while Unum<Uana-r:
26             Unum=Unum+r
27             addition(Nbit-1,N)
28         return(N,Unum)
29 #instructions d'affichage
30 def afficher(event):
31     chaine.configure(text=str(conversion(entree1.get(),Nbit)[0]))
32 fen=Tk()
33 bou1=Button(fen,text='4 bits',command=quatre)
34 bou2=Button(fen,text='8 bits',command=huit)
35 entree1=Entry(fen)
36 entree1.bind("<Return>",afficher)
37 chaine=Label(fen)
38 txt1=Label(fen,text='Choisir le mode:')
39 txt2=Label(fen,text='Entrer nombre décimal:')
40 txt3=Label(fen,text='Sortie nombre binaire:')
41 txt1.grid(row=0,column=0,sticky=E)
42 txt2.grid(row=1,column=0,sticky=E)
43 txt3.grid(row=2,column=0,sticky=E)
44 bou1.grid(row=0,column=1,sticky=W,padx=10)
45 bou2.grid(row=0,column=1,sticky=E,padx=10)
46 entree1.grid(row=1,column=1)
47 chaine.grid(row=2,column=1)
48 fen.mainloop()

```

ANNEXE 3

Mode d'emploi constructeur

Généralités sur PLANA3

Platine connectée sur SYSAM-SP5, exécuter LatisPlana.

Opter pour la page CAN en cliquant sur le bouton 

Sélectionner sur la platine le nombre de bits à utiliser pour l'étude à réaliser (4 ou 8 bits).

Sélectionner l'un des trois modes de travail disponibles :

- **Automatique** : recherche automatiquement (selon l'algorithme choisi) la valeur binaire de conversion correspondante à la donnée analogique à transformer. Toutes les étapes de la conversion sont enchaînées automatiquement jusqu'à obtention de la valeur binaire recherchée.
- **Pas à Pas** : recherche automatiquement (selon l'algorithme choisi) la valeur binaire de conversion correspondante à la donnée analogique à transformer. Cependant contrairement au mode automatique, chaque étape de la conversion doit être validée par l'utilisateur.
- **Manuel** : recherche manuelle de la valeur binaire de conversion correspondante à la donnée analogique à transformer. L'utilisateur n'est pas lié à un algorithme de recherche particulier, mais applique sa propre méthode pour déterminer la bonne valeur binaire de conversion. Il peut ainsi l'obtenir en appliquant manuellement les méthodes des deux algorithmes proposés en modes Automatique et pas à pas ou simplement déterminer cette valeur par tâtonnement.

Sélectionner enfin l'algorithme de travail (pour l'utilisation d'un mode Automatique ou Pas à Pas seulement).

Afin de profiter des possibilités de tracé graphique de LatisPlana, il est fortement conseillé d'utiliser l'interface SYSAM-SP5 pour effectuer les mesures (entrées de mesures EA0 et EA1) et l'émission de la tension à convertir (par la sortie analogique SA1).

Pour cela, relier à l'aide de cordons bananes $\varnothing 4\text{mm}$ sur PLANA3 les bornes :

- Ucna et EA0 (mesure de la tension Ucna par le canal 0 de SYSAM-SP5)
- SA1 et Ue (émission de la tension à convertir par la sortie analogique SA1)
- SA1 et EA1 (mesure de la tension à convertir par le canal 1 de SYSAM-SP5)
- Afin de régler le niveau de la tension que l'on souhaite convertir, agir sur le curseur de réglage prévu à cet effet dans la fenêtre SA1.

**Remarque :**

Il est également possible d'utiliser une source externe en tension pour fournir au système la tension analogique Ue à convertir. De même il est possible d'utiliser un oscilloscope et deux voltmètres pour mesurer les tensions Ucna et Ue. Néanmoins, aucun tracé témoignant de l'évolution de la conversion, ne pourra alors être effectué par LatisPlana.

ANNEXE 4

Tableaux de mesure

Conversion sur 4 bits

1	2	3	4	5	6	7	8
ε_0	ε_1	ε_2	ε_3	s_{th}	s_{exp}	s_{exp}	décimal
0	0	0	1				1
0	0	1	0				2
0	0	1	1				3
0	1	1	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				

Conversion sur 8 bits

1	2	3	4	5	6	7	8	9	10	11
ε_7	ε_6	ε_5	ε_4	ε_3	ε_2	ε_1	ε_0	s_{th}	s_{exp}	décimal
0	0	0	0	0	0	0	0			0
0	0	0	0	0	0	0	1			1
0	0	0	0	0	0	1	0			2
0	0	0	0	1	0	1	0			
0	0	0	0	1	0	1	1			
0	0	0	1	0	0	1	1			
0	0	0	1	1	0	0	1			
0	0	1	1	0	0	0	1			
1	1	1	1	1	1	1	0			
1	1	1	1	1	1	1	1			

Conversion parallèle

1	2	3	4	5	6	7
U_e	e_1 (V)	e_0 (V)	s_1	s_0	D_0	D_1
2 V						
7 V						
12 V						