

TP 1

Préliminaires

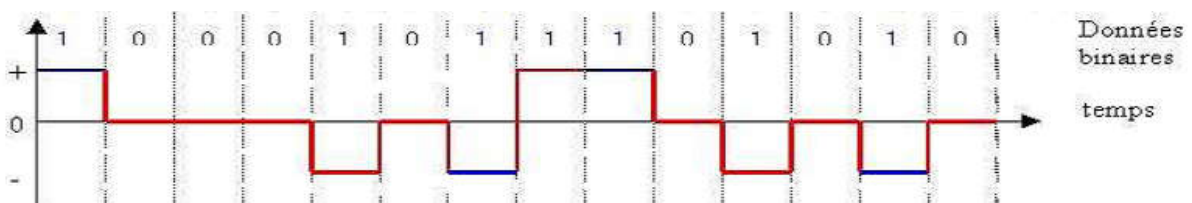
Exécuter le programme Matlab suivant :

```
close all;  
clear all;  
a=[0 1 1 0 1 0 0 0 1 0 0 1];  
b=2*a-1;  
subplot(2,1,1)  
stairs(a)  
grid on  
axis([1 13 -2 2])  
subplot(2,1,2)  
stairs(b)  
grid on  
axis([1 13 -2 2])
```

Que fait ce programme?

Travail à faire

- 1- Ecrire le programme Matlab qui convertie un signal binaire quelconque en code NRZ d'amplitude 20v.
- 2- Ecrire le programme Matlab qui convertie un signal binaire en code bipolaire simple (0 logique est représenté par 0 et 1 logique est représenté par une valeur (+V ou -V) opposée de la valeur 1 transmise au précédent (voir le cours)).
- 3- Ecrire le programme Matlab qui convertie un signal binaire en code bipolaire d'ordre 2 (voir la figure ci-dessous). (NB : dans le code bipolaire d'ordre 2, il y a alternance entre +V et -V une fois sur deux).



TP n°2

L'objectif de ce TP est de simuler une transmission série RS 232 entre un organe de commande et un instrument (capteur ou actionneur). On effectue cette simulation en utilisant le logiciel Matlab et 2 PC de bureau connectés via leurs prises DB9.

Préliminaires

La liaison RS232

La liaison série à la norme RS232 permet la réalisation d'une liaison simple à mettre en œuvre entre 2 équipements. Elle est de type asynchrone, c'est-à-dire qu'elle ne transmet pas le signal horloge, les deux équipements doivent être configurés avec une même vitesse de transmission identique. Ils doivent par ailleurs utiliser le même protocole.

Débit des données

La vitesse de transmission caractérise le nombre de bits transmis par secondes. Elle s'exprime en bauds. Les valeurs courantes sont entre 300 et 9600 bauds.

Format des trames

Au repos, la ligne est à 1. Un passage à 0 indique que la transmission va commencer, c'est le **bit de start**. - Les **bits de données** sont ensuite transmis sur un format de 7 ou 8 bits. - Un **bit de parité** peut être ajouté pour vérifier la validité des données. - Un ou deux **bits de stop** terminent la transmission pour le retour au repos de la ligne.



Exemple Matlab

Le programme suivant permet de créer un objet Matlab, le connecter au port série (en l'occurrence COM1) et d'envoyer et recevoir des données.

```
s = serial('COM1','BaudRate',9600,'DataBits','8','parity','even','StopBits','2') % Crée un objet Matlab port série COM1 avec un débit de 9600 bit/s, des données transmises sur 8 bits, parité paire, 2 bits de stop. Les propriétés de transmission BaudRate, ..., StopBits sont facultatives.
```

```
fopen(s) % Connecte l'objet port série s au périphérique
```

```
get(s) % Affiche les propriétés de l'objet port série
```

`fprintf('Licence Automatique')` % Envoie le text «Licence Automatique » au périphérique.

`fscanf(s)` % Lecture des données ASCII à partir du périphérique et leur formatage en tant que texte.

`fwrite(s,32)` %Envoie au périphérique 32 en code binaire.

`fread(s)` % Lecture des données binaires à partir du périphérique.

`fclose(s)` % déconnecte l'objet port série s du périphérique.

Manipulation

- 1- Écrire un programme Matlab qui transmet et reçoit sur le port série des données en code ASCII et en code binaire.
- 2- On désire écrire un programme Matlab qui simule le comportement d'un moteur à courant continu dont le modèle mathématique est décrit ci-dessous. La simulation commence et s'arrête après la réception des codes « go » et « stop » respectivement envoyés par le deuxième PC (qui joue le rôle de l'organe de commande) sur le port série. À la fin de la simulation, le programme envoie au deuxième PC la dernière valeur du vecteur d'état.

Indication

- Modèle à simuler : $\dot{x} = \begin{pmatrix} 0 & 1 \\ 0 & -1/5 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u$ avec $u=1$.

- Boucle Matlab infinie :

<code>m=true</code>	
<code>while m</code>	<code>while true</code>
<code>étape1 ;étape2 ; étape3 ;</code>	<code>étape1 ;étape2 ; étape3 ;</code>
<code>if étape3==condition_d'arret</code>	<code>if étape3==condition_d'arret</code>
<code> m=false ;</code>	<code> break ;</code>
<code>end</code>	<code>end</code>
<code>end</code>	<code>end</code>

- Comparaison des chaînes de caractères en Matlab:

`Test=strcmp(ch1,ch2)` % retourne Test=1 si ch1=ch2 , Test=0 si non.

- On suggère d'utiliser les commandes Matlab `fwrite` et `fscanf`.