



TP Réseau - La transmission en bande de passe

Réalisé par :

Abbas Adam ABBAS

Encadrant :

Claude Duvallet

TABLES DES MATIERES

1. Rappel du sujet	2
2. Analyse du sujet	3
3. Choix techniques effectués	3
3.1 Structure du programme :	3
3.2 Aspects graphiques :	3
4. Résultats et tests	4
4.1 Résultats	4
4.2 Tests effectués des trames effectuées :	4
4.2.1 Test : 101001101101101 avec NRZ	4
4.2.2 Test : 101001101101101 avec NRZI	4
4.2.3 Test : 101001101101101 avec Manchester.	5
4.2.4 Test : 101001101101101 avec Manchester différentiel	5
4.2.5 Test : 101001101101101 avec Miller	6
5. Difficultés rencontrées et Conclusion	6
5.1 Difficultés rencontrées	6
5.2 Limites du programme :	6
5.3 Conclusion	6

1. Rappel du sujet

Le projet consiste à implémenter un programme Java permettant de tracer graphiquement les signaux électriques correspondant à une chaîne binaire, selon différents types de codage : NRZ, NRZI, Manchester, Manchester différentiel, et Miller.

L'interface doit permettre à l'utilisateur de saisir une chaîne binaire, choisir un type de codage dans un menu déroulant, afficher dynamiquement le signal correspondant dans un panneau graphique.

2. Analyse du sujet

La problématique centrale est basée sur deux points :

Chaque méthode de codage possède des règles spécifiques et stricte à suivre pour traduire la chaîne binaire en transitions électriques ;

Le programme doit adapter les traces en fonction de la longueur de la trame et du typage de codages (NRZ, NRZI, Manchester, Manchester différentiel ou Miller), les signaux doivent être clairement lisibles (les lignes nV , $0V$ et $-nV$) et aussi la ligne vertical qui doit séparer chaque transition.

3. Choix techniques effectués

Langage et outils utilisés :

Java : Langage principal pour la gestion de l'interface utilisateur et du tracé graphique.

Swing : est utilisé pour Créer l'interface graphique (classe principale MainApp.

Graphics2D : Permet de dessiner les signaux avec des fonctionnalités avancées (anticrénelage, motifs pointillés, épaisseur des lignes, etc.).

3.1 Structure du programme :

Une Classe principale MainApp qui :inclut des champs de saisie pour la chaîne binaire et un menu déroulant pour sélectionner le codage puis affiche dynamiquement le signal dans un panneau dédié selon le codage sélectionné ;

Des classes spécialisées pour chaque type de codage (NRZGraphPanel, NRZIGraphPanel, ManchesterGraphPanel, ManchesterDiffGraphPanel, CodeMillerGraphPanel) .

3.2 Aspects graphiques :

- Axes et niveaux de tension : Chaque signal utilise des niveaux définis (nV , $0V$, $-nV$).
- Transitions et étiquettes : Les transitions (rigides ou pointillées) et les bits sont clairement affichés sous le graphe.

- Personnalisation visuelle : Les panneaux incluent des dégradés de fond et des couleurs harmonieuses pour une meilleure lisibilité.

4. Résultats et tests

4.1 Résultats

L'interface utilisateur est fonctionnelle et intuitive. L'utilisateur peut saisir une chaîne binaire (la trame) et choisir un codage, après quoi le signal est affiché dynamiquement dans le panneau. Chaque type de codage respecte les spécifications :

NRZ : le codage du bit 1 par un signal de n volts et le 0 par un signal de $-n$ volts ;

NRZI : le codage du bit 0 par un signal de n volts et le 1 par un signal de $-n$ volts ;

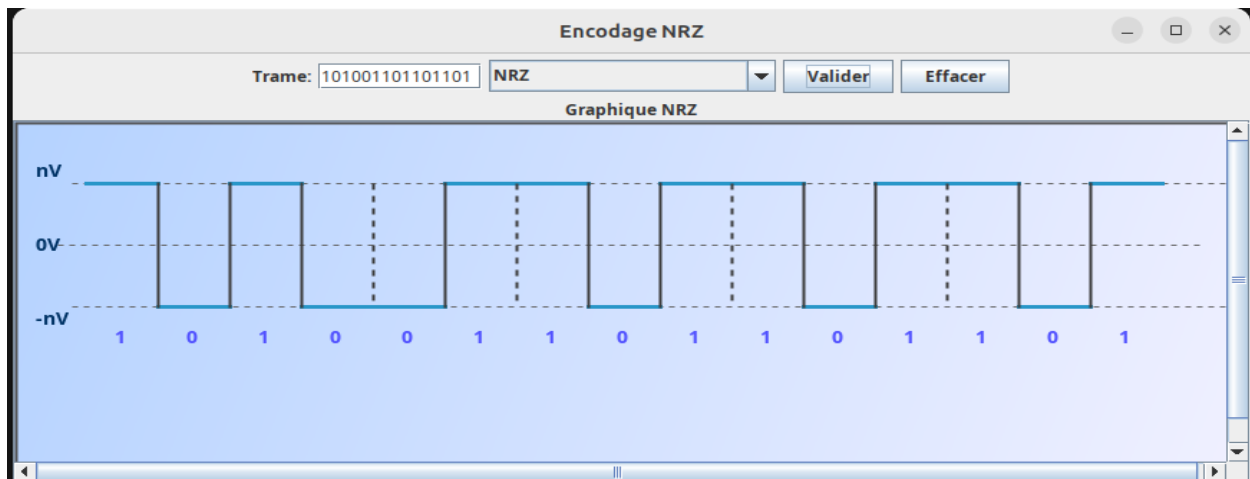
Manchester : Une transition est ajoutée au milieu de chaque bit, respectant le standard ;

Manchester différentiel : Les transitions dépendent des bits précédents et de leur logique ;

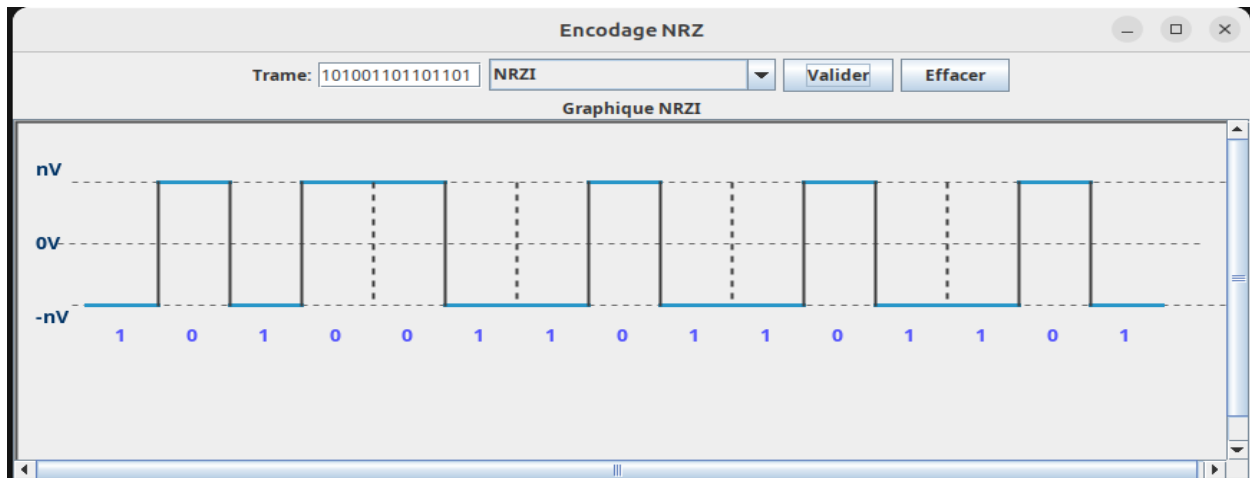
Miller : Les transitions sont conditionnelles, avec une logique adaptée aux longues séquences de 0.

4.2 Tests effectués des trames effectuées :

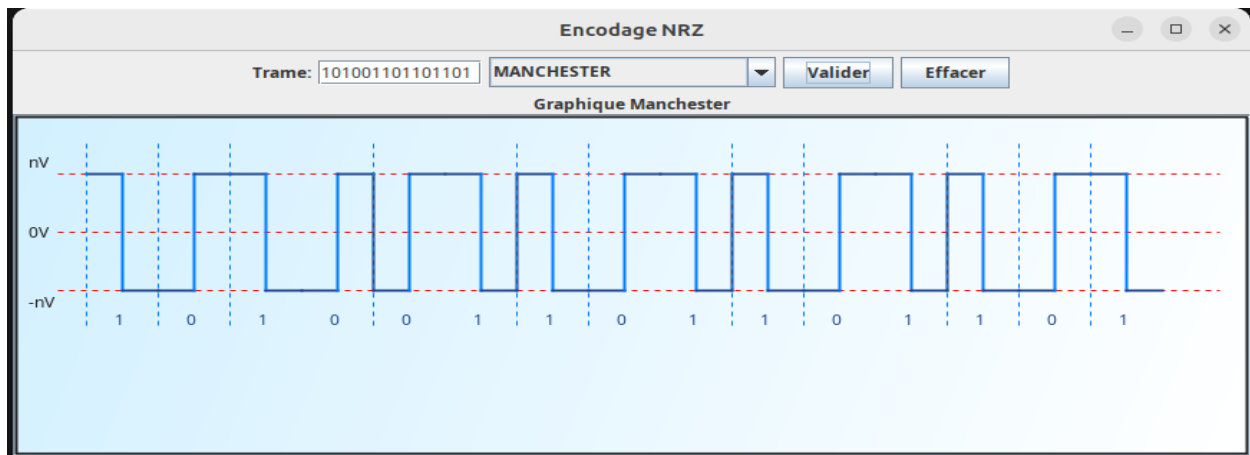
4.2.1 Test : 101001101101101 avec NRZ.



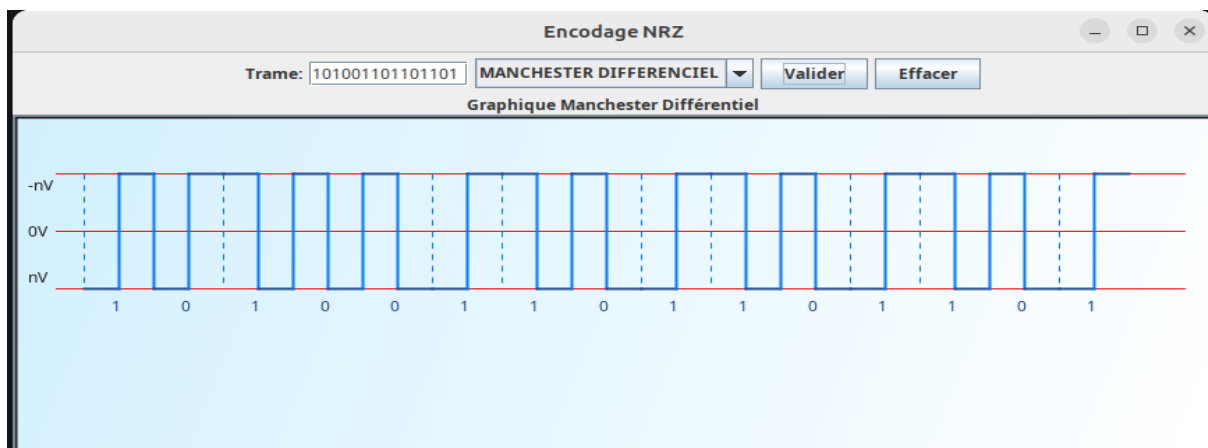
4.2.2 Test : 101001101101101 avec NRZI.



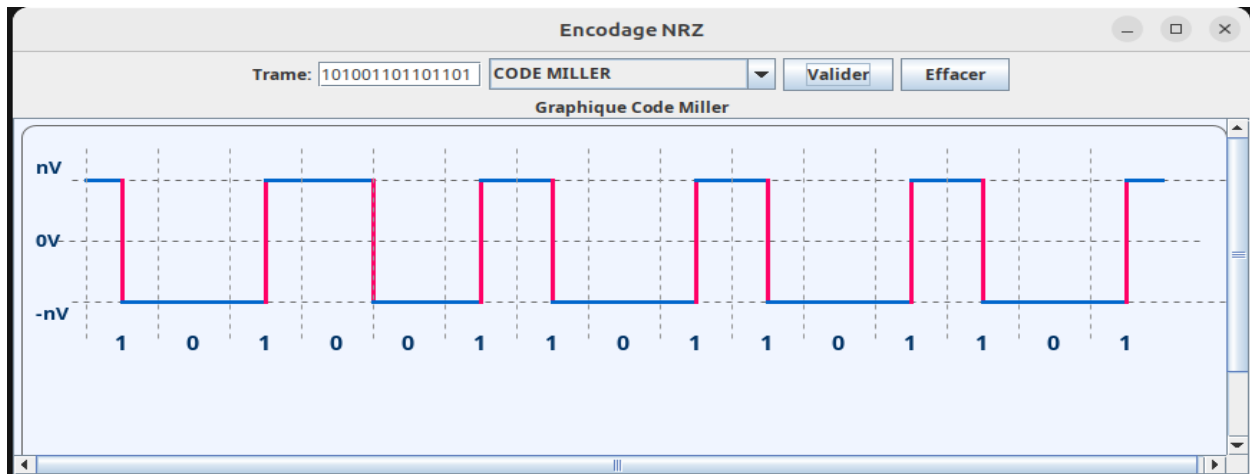
4.2.3 Test : 101001101101101 avec Manchester.



4.2.4 Test : 101001101101101 avec Manchester différentiel.



4.2.5 Test : 101001101101101 avec Miller.



5. Difficultés rencontrées et Conclusion

5.1 Difficultés rencontrées

Pendant l'implémentation, nous avons rencontrées d'énormes difficultés qui sont entre autres :

Gestion des transitions complexes : Les codages comme Manchester différentiel ou Miller nécessitent une logique conditionnelle plus complexe, notamment pour les séquences de 0 et les dépendances au bit précédent.

Alignement graphique : Centrer correctement les étiquettes des bits sous les transitions a demandé des ajustements minutieux.

5.2 Limites du programme :

Bien que le programme vérifie les entrées pour s'assurer qu'elles contiennent uniquement des 0 et 1, aucune gestion avancée des erreurs n'est incluse. Ce dernier est limité aux cinq codages spécifiés.

5.3 Conclusion

Ce programme constitue une solution fonctionnelle et intuitive pour visualiser les signaux électriques associés aux différents codages numériques. Grâce à son architecture modulaire et son interface conviviale, il peut être facilement étendu pour répondre à des besoins futurs.