

[Séquence 7]

Mise en œuvre finale

Objectifs :

A l'issue de cette étude l'élève sera capable de :

- Faire fonctionner sa maquette dans différentes configurations.

Pré requis :

- Programmation de séquence DCC (séquence 5)
- Programmation du système de menus (séquence 6)
- Carte élève réalisée et fonctionnelle
- Carte DCC_VOX réalisée et fonctionnelle

Nature du travail à effectuer :

- Mise en œuvre complète de la maquette élève
- Programmation en langage C

Documents de travail :

- [Langage C](#) adapté au microcontrôleur.
- Documentation constructeur du [PIC 16F876A](#)
- Documentation sur la carte [Décodeur DCC VOX](#)
- Documentation sur [le standard DCC](#).

Documents à inclure au dossier :

- Programmes réalisés et documentés

Evaluation :

- Les différents programmes seront évalués pendant la mise en œuvre du thème lors de l'examen final.

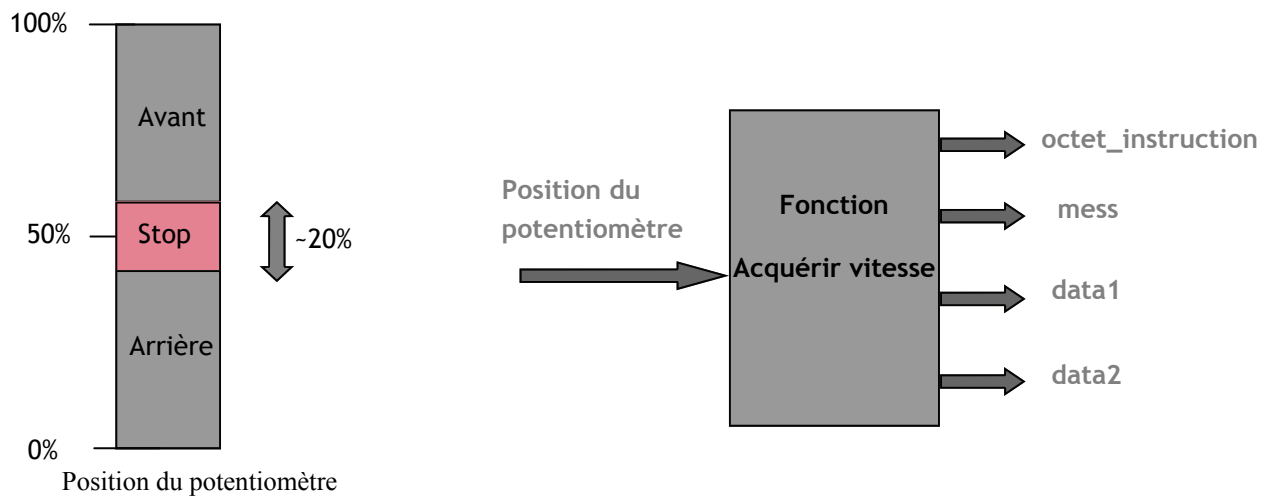
Durée :

4 semaines

Première partie : Programme 1 - Réglage de la vitesse de la locomotive

1.1. Réaliser un réglage de la vitesse de la locomotive en utilisant le potentiomètre.

Ce programme doit permettre à partir de la valeur de tension « émise » par le potentiomètre de calculer le contenu de l'octet d'instruction à envoyer au décodeur de locomotive.



Valeurs délivrées par la fonction « Acquérir_Vitesse_Loco » en fonction de la position du potentiomètre :

- **octet_instruction** : octet2 (2^o octet de trame DCC) de la trame de commande de la locomotive
- **mess** : valeur 1, 2 ou 3 permettant l'affichage sur le LCD
- **data1** : adresse de la locomotive
- **data2** : valeur du pas de vitesse à afficher sur le LCD (de 1 à 14)

Note : La zone de la vitesse « zéro » doit correspondre à environ 20% de la variation totale du potentiomètre.

Activité préparatoire : Etude du Convertisseur.

A partir de la documentation du [PIC 16F876A](#) (et des notions vues sur [les CAN](#)), sur une demi-page environ présenter la configuration du Convertisseur Analogique Numérique utilisée dans notre application.

Etape 1 : Acquisition de la position du curseur et calcul des bits de vitesse SSSS.

Réaliser un programme qui permet de :

- Acquérir la valeur « émise » par le potentiomètre.
- Tester cette valeur de façon à effectuer un traitement correspondant aux trois zones « Avant », « Stop » et « Arrière ».

Selon le cas :

▶ Cas des zones « avant » et « arrière » :

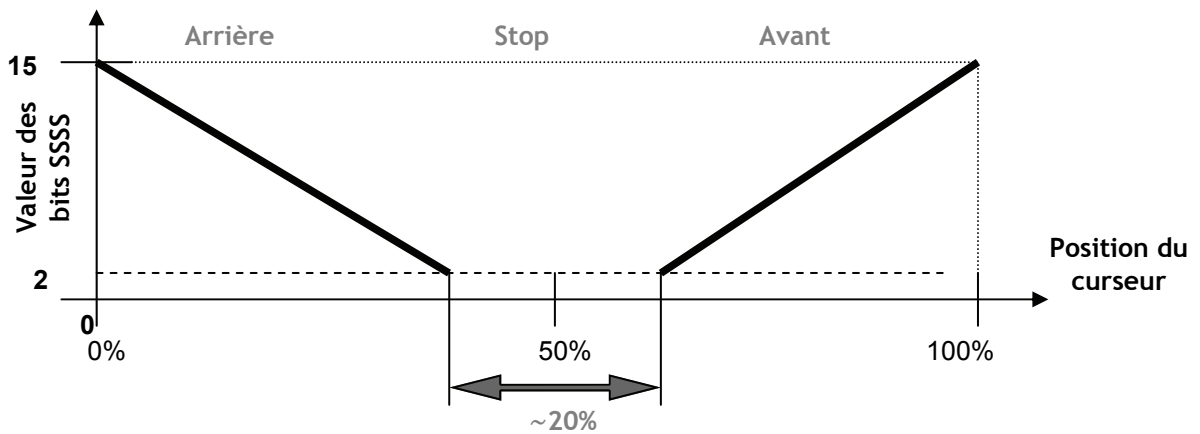
→ A l'aide de la caractéristique donnant la relation liant les bits de vitesse SSSS et la position du potentiomètre (page suivante), établissez les équations dans ces deux zones, permettant une transformation de l'image de la position du curseur du potentiomètre en une valeur de bits vitesse SSSS de la locomotive comprise entre 0010 et 1111 (2 et 15).

▶ Cas de la zone « Stop » :

→ Il va de soit de mettre dans ce cas la valeurs des bits vitesses SSSS à la valeur adéquate... (sans commentaires ...)

Validation : Valider ce premier programme en affichant la valeur en décimal de la position du curseur du potentiomètre et la valeur du pas comprise entre 2 et 15 (bits vitesse SSSS).

Nota : Il sera peut être nécessaire d'ajuster une ou plusieurs constantes des équations pour coller au cahier des charges. En effet, nous ne calculons que sur un octet (arrondi) d'où ce problème d'erreur.



Etape 2 : Calcul des valeurs à placer dans les variables `octet_instruction`, `mess`, `data1` et `data2`.

Réaliser un programme qui permet de :

- ☑ Initialiser votre variable de sortie (ici `octet_instruction`) pour la vitesse stop (`pas = 0`)
- ☑ Acquérir la valeur « émise » par le potentiomètre
- ☑ Tester cette valeur de façon à effectuer un traitement correspondant aux trois zones « Avant », « Stop » et « Arrière ».

Selon le cas :

▶ Cas des zones « avant » et « arrière » :

- déterminez la valeur des bits SSSS (`pas`) en fonction de l'image de la position du curseur.
- placez la valeur des bits SSSS (`pas`) sur les 4 bits correspondants dans l'octet d'instruction.
- forcez le bit D de l'octet d'instruction à la valeur voulue
- mise à jour des variables d'affichage `mess`, `data1` et `data2`

▶ Cas de la zone « Stop » :

- Il va de soit de mettre dans ce cas la valeurs des bits vitesse SSSS à la valeur adéquate... (sans commentaires ...)
- mise à jour des variables d'affichage `mess`, `data1` et `data2`

Etape 3 : Programme final (Programme 1)

Réaliser un programme qui permet de remplir les fonctions suivantes :

- ☑ La mise en marche et l'arrêt de la locomotive est commandée par l'`inter1`,
- ☑ L'arrêt d'urgence est commandé par l'`inter2`,
- ☑ Le programme 1 appelle la fonction `Acquisition_vitesse_loco` pour déterminer le contenu de l'octet instruction et les messages à afficher,
- ☑ Le programme 1 appelle le sous programme d'envoi d'une trame de commande de la vitesse de la locomotive,
- ☑ Si le BP4 est appuyé Le programme 1 appelle le sous programme d'affichage pour afficher les bonnes valeurs en fonction du contexte.

page 3

1.2. Test et mise au point du programme

S'assurer que votre programme permet de réaliser les fonctions suivantes :

- ☑ Affichage de la valeur du cran de vitesse sur l'afficheur LCD (**Vérification par le Jury à ORAL**)
- ☑ Test de la commande de la locomotive à la bonne vitesse sur la maquette du modèle réduit de train. (**Validé par l'enseignant en classe**)

Deuxième partie : Programme 2 - Commande des feux

1.3. Réaliser un programme de commande des feux avec la carte DCC_VOX

- ☑ Réaliser le programme générant la trame DCC pour commander l'allumage du feu tricolore à Leds sur la carte DCC_VOX.
- ☑ Chaque appui sur **BP4** provoque le changement de l'état du feu Vert=>Orange=>Rouge

1.4. Test et mise au point du programme

S'assurer que votre programme permet de réaliser les fonctions suivantes :

- ☑ Commande des feux à Leds (**Vérification par le Jury à ORAL**)
- ☑ Affichage de contrôle sur l'afficheur LCD

Troisième partie : Programme 3 - Commande des aiguillages

1.5. Réaliser un programme de commande d'un aiguillage avec DCC_VOX

- ☑ Réaliser un programme générant la trame pour commander le mouvement des aiguillages.
- ☑ L'**inter1** commande l'aiguillage 1, l'**inter2** commande l'aiguillage 2
- ☑ Prévoir une commande du relais en mode « monostable » : le relais bistable de manœuvre de l'aiguillage doit être laissé au repos. La commande de type impulsionnel doit durer 50 ms.

Principe : Dans ce mode, la commande consiste à provoquer le changement de position de l'aiguillage à chaque manœuvre de l'interrupteur correspondant. (Impulsion de durée 50ms).

1.4. Test et mise au point du programme

- ☑ Essais de manœuvre des aiguillages (**Vérification par le Jury à ORAL**)
- ☑ Affichage de contrôle sur l'afficheur LCD

Quatrième partie : Programme 4 - Mode décodeur de locomotive - Carte Physique - Commande PWM

Ici nous allons utiliser la carte DCC_VOX dans le mode de fonctionnement 'locomotive'. Le comportement de la carte est semblable à celui d'un décodeur embarqué à bord des modèles réduit de locomotive. Dans ce mode elle transmet sous la forme d'une modulation PWM de 0 à 100% la valeur de l'octet PWM reçu dans la trame DCC à la carte de physique appliquée.

- ☑ Acquisition de la valeur du potentiomètre sur 8 bits
- ☑ Envoi de cette valeur pour générer le signal PWM vers la carte DCC_VOX, configurée en mode Décodeur de locomotive.
- ☑ Envoi de cette modulation sur la carte physique appliquée (**Vérification par le Jury à ORAL**)

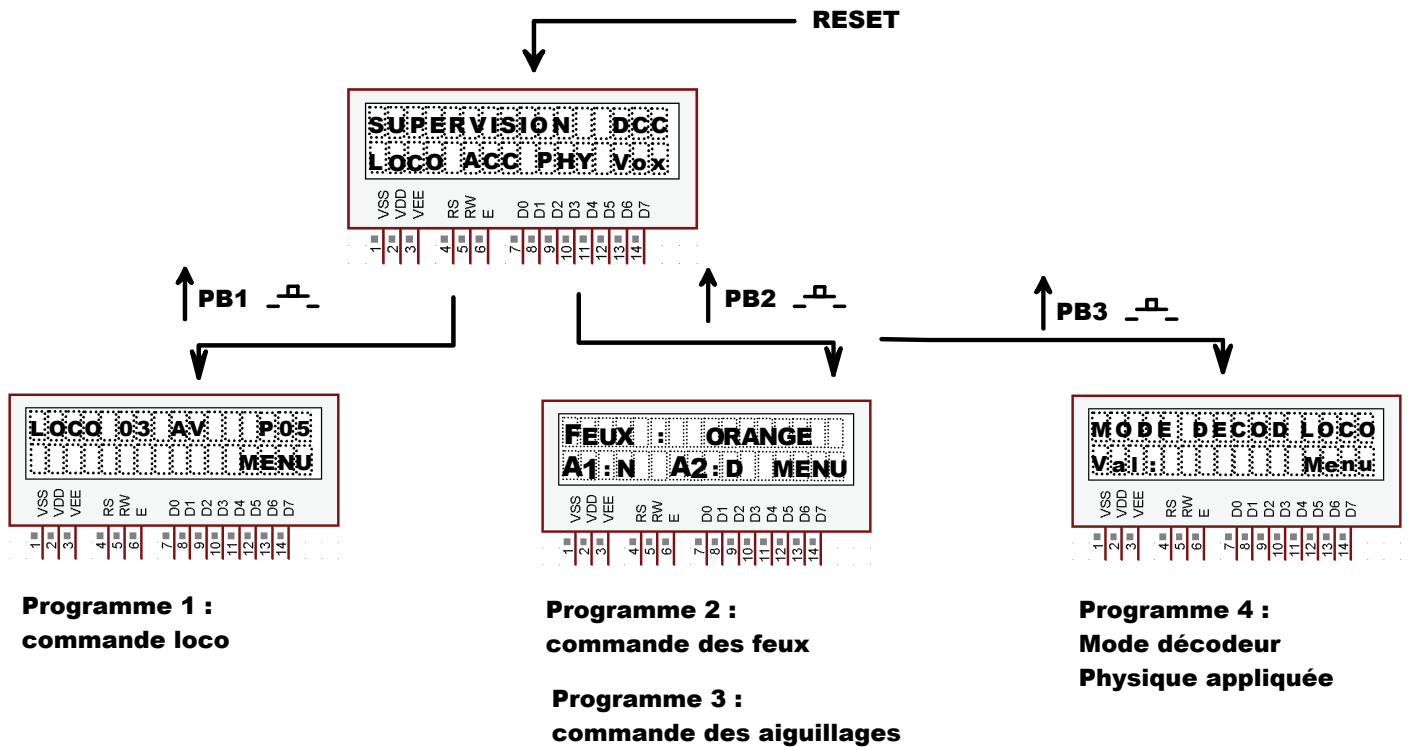
Cinquième partie : Mise en œuvre finale [Configuration présentée le jour de l'examen]

Intégrer tous les programmes selon le schéma ci-dessous, après un reset le menu général est affiché sur l'écran de la console. Le choix se fait par l'appui sur l'un des trois boutons poussoirs BP1, BP2, BP3. Ce choix est définitif, il n'y a pas de retour au menu général.

Le choix sélectionne l'un des programmes mis au point précédemment :

- Programme 1 : commande de la loco (BP1)
- Programme 2 : commande des feux (BP2)
- Programme 3 : commande d'un aiguillage (BP2)
- Programme 4 : mode décodeur (Physique appliquée) (BP3)

Commande DCC Présentation finale



Situation de la mise en œuvre finale devant jury (avec afficheur LCD)

page 5

