

# [Séquence 5]

## *Génération de trames DCC*

### Objectifs :

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

- Elaborer le programme permettant d'envoyer la trame DCC sur les rails du train.
- Valider le cahier des charges par une simulation sous Proteus.
- Vérifier que le train, l'aiguillage et/ou les feux répondent bien aux ordres.

### Pré requis

- Savoir élaborer des programmes en langage C pour le microcontrôleur présent sur la carte élève.
- Avoir compris comment se transmet la trame DCC et quelles sont les contraintes temporelles à respecter.

### Moyens à mettre en œuvre

- Carte élève fonctionnant parfaitement.
- Ordinateur avec logiciel de développement en langage C.
- Le récepteur DCC selon disponibilité.

### Documents de travail :

- Analyse fonctionnelle.
- Document sur la norme DCC.
- Guide de programmation en C et guide d'utilisation du logiciel de développement.

### Documents à inclure au dossier :

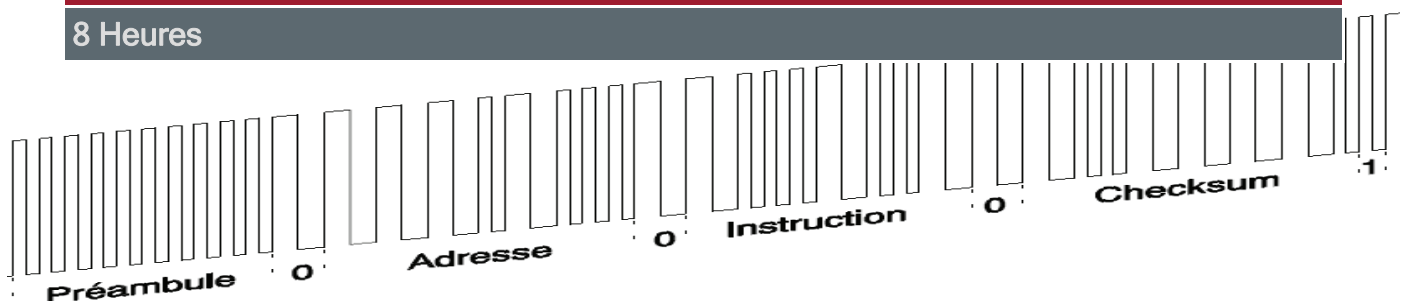
- Réponses aux différentes questions.
- Le fichier du programme en langage C.
- Oscillogramme d'une trame DCC générée parmi les trois possibles.

### Évaluation :

Aptitude à utiliser le logiciel de développement, rapidité pour la mise au point du programme.

### Durée :

8 Heures



## Première partie : Etude préliminaire.

Documents de travail: [Documents sur le protocole DCC](#)

- 1.1. Rappeler les contraintes temporelles d'une trame DCC.
- 1.2. La ligne DCC peut-elle être au repos ?
- 1.3. Que préconise le protocole DCC lorsqu'il n'y a pas d'instruction à envoyer ?
- 1.4. De quoi est composée une trame DCC ?
- 1.5. Quel est le rôle de chaque donnée envoyée dans la trame ?
- 1.6. Retrouver dans la documentation technique NEM 670 la durée minimale pour émettre un « zéro », puis un « un », comparer ceux-ci avec les constantes `alternance_zero`, et `alternance_un` du programme [Seq5-trame-dcc.c](#).

## Deuxième partie : Ouvrir et compléter le fichier « Seq5-trame-dcc.c »

En vous aidant des différentes [Documentations sur le protocole DCC](#) , de la documentation sur la [carte Décodeur DCC](#) et du fichier source [Seq5-trame-dcc.c](#).

2.1. Pour chaque type de trame, compléter le tableau des valeurs à envoyer aux décodeurs  
Voir l'annexe 1 ou le [document en ligne](#).

2.2. Compléter l'algorithme de la procédure de génération d' une trame DCC

/\* Algorithme de la procédure `void generer_trame_dcc(octet_adresse,octet_instruction)`

### DEBUT

Calculer l'octet de checksum ;

Générer la séquence de synchronisation ;

Générer le bit START ;

Sérialiser l'octet d'..... ;

Générer le bit ..... ;

Sérialiser l'octet d'..... qui lui est destiné ;

Générer le bit ..... ;

Sérialiser l'octet de.....;

Générer le bit ..... ;

Programmer une temporisation entre l'envoi de deux trames DCC.

### FIN

### 2.3. Compléter l'algorithme de la procédure de sérialisation d'un octet sur la VOIEA

Note : Sérialiser un octet consiste à écrire les bits de cet octet sur la ligne VOIEA en commençant par le bit de poids faible et en finissant par le bit de poids fort.

/\* Algorithme de la procédure void `serialiser_octet(octet);`

```
DEBUT
déclarer indice et data comme des octets.

POUR indice de 0 jusqu'à 7 par pas de 1 FAIRE
    data ← octet & 128
    SI (data==128)
        ALORS
            { générer un ..... sur VOIEA}
        SINON
            { générer un ..... sur VOIEA}
    Fin de SI
    octet=octet<<1
Fin de POUR.
FIN
```

### 2.4. Ouvrir le fichier [Seq5-trame-dcc.c](#) et compléter les procédures :

-> `serialiser_octet(octet);`  
-> `generer_trame_dcc(octet_adresse,octet_instruction)`

**NB :** Il suffit de préparer les variables «octet\_adresse, octet\_instruction» et appeler la fonction : «generer\_trame\_dcc», de même de mettre le mot que l'on veut sérialiser dans la variable «octet» et appeler la fonction : «serialiser\_octet».

### Troisième partie : Mise en œuvre du programme « Seq5-trame-dcc.c »

La fonction principale (main) du programme «[Seq5-trame-dcc.c](#)», appelle une procédure d'initialisation, envoie une trame de RAZ des décodeurs puis rentre dans une boucle sans fin. Dans cette boucle, on génère par défaut, en absence de toute commande, la trame de remplissage en continue IDLE. Dans cette boucle on scrute aussi les boutons poussoirs qui nous permettent de choisir quatre types d'action, et qui sont :

Si appui sur BP1 : Choix Générer une trame de RAZ des Décodeurs.  
Si appui sur BP2 : Choix Générer Trame Locomotive à l'adresse 3, en marche avant, cran de vitesse 11.  
Si appui sur BP3 : Choix Générer trame aiguillage à l'adresse 1, ce dernier est en position normale.  
Si appui sur le BP4 : Choix Générer trame feux tricolore à leds , feux vert allumé sur la carte décodeur.

NB : Entre deux trames successives on observera un délai de deux millisecondes.

#### 3.1. Pour tester le programme ainsi complété grâce aux précédents algorithmes :

- Implantez le programme dans votre carte élève DCC VOX.
- La brancher à l'oscilloscope numérique (ou simuler sous Proteus).
- Visualisez le chronogramme de la trame DCC générée par le microcontrôleur, et vérifiez que chaque

trame est conforme avec sa séquence propre envoyée sur la ligne VOIEA. (il s'agit des séquences ci-dessus décrites ).

**NB :** la séquence prépondérante dans la trame DCC est celle du remplissage.

3.2 Modifier dans ce programme l'adresse, et l'instruction à envoyer pour chaque décodeur, et visualisez de nouveau la trame DCC.

## Résumé sur les trames DCC

Destinataire	Octet d'adresse								Octet d'instruction							
	A7	A6	A5	A4	A3	A2	A1	A0	I7	I6	I5	I4	I3	I2	I1	I0
<b>RAZ</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<b>Remplissage (Idle)</b>	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
<b>Locomotive</b>	0	1	N5	N4	N3	N2	N1	N0	0	1	D	v	V3	V2	V1	V0
<b>Aiguillage/Feux Roco</b>	1	0	N8	N7	N6	N5	N4	N3	1	N2	N1	N0	D	S1	S1	S0
<b>Feux Leds</b>	1	0	N8	N7	N6	N5	N4	N3	1	N2	N1	N0	D	S1	S1	S0

N désigne le numéro du décodeur adressé.

V désigne le cran de vitesse.

v représente les demi-crans (pour nous v= 0)

D désigne le sens de marche, ou l'état des accessoires.

S désigne le numéro d'accessoires sur le décodeur.

NMAX = 99 pour les locomotives.

NMAX = 512 pour les décodeurs d'accessoires.

SMAX = 8.

Aiguillage et Feux Roco : N2 = J9 , N1 = J8 , N0 = 1.

Feux sur le décodeur d'accessoires : N2 = J9 , N1 = J8 , N0 = 0.

## Annexe 1 : RECAPITULATIF TRAMES DCC

*Note : Valeurs à remplir en Hexadécimal*

### Trame de RAZ des décodeurs de locomotives et d'accessoires (Initialisation)

Octet d'adresse (destinataire)	Octet de donnée (instruction)	Commentaires
		Trame d'initialisation

### Trame de remplissage (IDLE)

Octet d'adresse (destinataire)	Octet de donnée (instruction)	Commentaires
		Trame de remplissage (IDLE)

### Trame à destination d'un décodeur de locomotive (mode 14 crans => Bit 5 à 0)

Octet d'adresse (destinataire)	Octet de donnée (instruction)	Vitesse	Sens
		1	<b>AVANT</b>
		14	<b>AVANT</b>
		Arrêt progressif	
		Arrêt d'urgence	
		1	<b>ARRIERE</b>
		14	<b>ARRIERE</b>

### Trame à destination d'un décodeur d'accessoires

Octet d'adresse (destinataire)	Octet de donnée (instruction)	Destinataire	Action
<b>Feux à Leds</b>			
		Feux à Led (vert)	<b>ALLUMÉ</b>
		Feux à Led (vert)	<b>ÉTEIND</b>
		Feux à Led (orange)	<b>ALLUMÉ</b>
		Feux à Led (orange)	<b>ÉTEIND</b>
		Feux à Led (rouge)	<b>ALLUMÉ</b>
		Feux à Led (rouge)	<b>ÉTEIND</b>
<b>Feux à Leds sur J11 (5V) (Etat niveau actif fixé à 1)</b>			
		Feux sur J11 (F3)	<b>ALLUMÉ</b>
		Feux sur J11 (F3)	<b>ÉTEIND</b>
		Feux sur J11 (F4)	<b>ALLUMÉ</b>
		Feux sur J11 (F4)	<b>ÉTEIND</b>
		Feux sur J11 (F5)	<b>ALLUMÉ</b>
		Feux sur J11 (F5)	<b>ÉTEIND</b>
<b>Instructions spécifiques au décodeur Feux à Leds / J11</b>			
		Instinctions des feux	<b>ÉTEIND</b>
		Fixer Niveau actif de J11	<b>Actif à 1</b>
		Fixer Niveau actif de J11	<b>Actif à 0</b>

**Trame à destination d'un décodeur d'accessoires (suite)**

Octet d'adresse (destinataire)	Octet de donnée (instruction)	Destinataire	Action
<b>Mode ULN2803 (Aiguillage 1 - Output 0,1) - J2</b>			
		Aiguillage 1 (Output 0)	ÉTAT 0
		Aiguillage 1 (Output 0)	ÉTAT 1
		Aiguillage 1 (Output 1)	ÉTAT 0
		Aiguillage 1 (Output 1)	ÉTAT 1
<b>Mode ULN2803 (Aiguillage 2 - Output 2,3) - J3</b>			
		Aiguillage 2 (Output 2)	ÉTAT 0
		Aiguillage 2 (Output 2)	ÉTAT 1
		Aiguillage 2 (Output 3)	ÉTAT 0
		Aiguillage 2 (Output 3)	ÉTAT 1
<b>Mode ULN2803 (Feux ROCO 20V - Output 4,5,6) - J4 et J12</b>			
		FEUX Roco 1 (Output 4)	ÉTAT 0
		FEUX Roco 1 (Output 4)	ÉTAT 1
		FEUX Roco 2 (Output 5)	ÉTAT 0
		FEUX Roco 2 (Output 5)	ÉTAT 1
		FEUX Roco 3 (Output 6)	ÉTAT 0
		FEUX Roco 3 (Output 6)	ÉTAT 1
<b>Instruction spécifique au décodeur ULN2803</b>			
		Instruction globale des sorties	ETEIND