

Rapport de pré-soutenance

Développement FPGA d'une centrale DCC optimisée pour la commande de trains miniatures

Sarancoumar SOUNDARAMOURTY, Fatma MESSAOUI

Encadrants : Julien DENOULET et Bertrand GRANADO

SOMMAIRE

CONTEXTE & ENCADREMENT	3
OBJECTIFS.....	3
OUTILS.....	4
PRESENTATION DU PROTOCOLE DCC.....	5
<i>exemple d'une trame:</i>	5
<i>Composition de la trame :</i>	5
ORGANISATION DU PROJET	6
<i>Tâches et résultats intermédiaires</i>	6
INTERFACE UTILISATEUR (microcontrôleur) :	7
CENTRALE DCC :	7
AVANCEMENT	7
<i>exemple : send preamble</i>	8
SCHEMA CENTRALE DCC	8
SCHEMA MICROCONTROLEUR	9
VALIDATION	9
PLANIFICATION	10
CONCLUSION.....	10
BIBLIOGRAPHIE	11

CONTEXTE & ENCADREMENT

Avant l'apparition du DCC (Digital Command Control), un modèle réduit de locomotive électrique était commandé en appliquant un courant variable sur les rails : on changeait la direction en inversant la polarité du courant, et la vitesse en faisant varier l'intensité. Ce système de commande a été très utilisé, mais présente plusieurs inconvénients : par exemple, tous les trains circulant sur la même voie ont la même direction, la même vitesse...etc. d'où la nécessité de trouver le moyen de commander chaque train indépendamment des autres. C'est dans ce but que la NMRA (*National Model Railroad Association*) a établi un système de commande de référence, le DCC.

Le DCC est une structure basique de communication pour transmettre des signaux de commande numériques vers les locomotives via les rails. Ce standard définit un protocole de communication entre le décodeur présent dans les locomotives et la centrale de commande, sans spécifier la structure et l'architecture (fonctionnement) de chaque élément (décodeur, centrale). L'un des avantages de ce système est qu'on peut commander n'importe quelle locomotive possédant un décodeur DCC avec n'importe quelle centrale DCC (intérêt économique).

La commande envoyée est sous forme de trames (signal carré) dont nous verrons l'usage et les caractéristiques plus loin.

Ce projet est effectué en binôme et consiste à implémenter une solution qui est la centrale DCC sur une carte de développement FPGA Spartan 6, les domaines de compétences requis sont la connaissance du VDHL, de la plateforme FPGA, la capacité à extraire les informations pertinentes des différentes datasheets et savoir découper le travail de façon optimale.

OBJECTIFS

Le but du projet est de réaliser une centrale qui fonctionne sur le même principe défini plus haut, où nous allons pouvoir envoyer des commandes (une ou plusieurs commandes) aux locomotives. Ces commandes sont composées de plusieurs parties (octets) où l'on retrouve l'adresse de la locomotive qu'on souhaite contrôler, ainsi qu'une ou plusieurs fonctions qu'on souhaite activer sur le train dont la vitesse qu'on souhaite lui appliquer. Nous verrons plus loin en détail le fonctionnement de ces commandes.

L'utilisateur devra alors choisir sur la carte Spartan 6 (via les boutons ou l'UART) quelle locomotive il souhaite actionner et quelles fonctions(instructions) il souhaite lui appliquer, la centrale DCC s'occupera alors d'envoyer aux rails la commande après interprétation de l'information saisie sur l'interface utilisateur. La commande sera récupérée sur les rails et décodée par la locomotive.

Nous verrons plus loin les améliorations que l'on souhaite apporter à la centrale notamment sur l'interface utilisateur.

L'objectif absolu serait de se rapprocher au maximum des centrales de commande qu'on trouve en vente, ces centrales présentent une interface utilisateur très optimisée où on peut commander plusieurs trains (deux trains en même temps) , implémenter plusieurs fonctions à une locomotive presque instantanément. Nous avons eu l'occasion d'en tester une qui a été récemment acquise par l'université au bâtiment Esclançon (voir ci-contre) , nous avons donc pu voir les différentes fonctionnalités qu'offrait ce type de centrales.

Nous ne cherchons pas à copier la centrale déjà existante dans son intégralité mais souhaitons réimplémenter les fonctionnalités majeures de celle-ci comme :



- Un système d'envoi des commandes à de multiples trains
- L'activation de plusieurs fonctions simultanées.

Avec toujours comme but de faciliter la saisie des différentes commandes, via le nombre (restreint) de boutons ou la liaison série (UART) pour plus de commodité et d'interopérabilité (extensions possibles, voir paragraphe « Améliorations »).

OUTILS

Pour mener à bien notre projet nous disposons de ces différents outils/éléments :

- Langage VHDL : pour la description matérielle des différentes IP
- Carte Avnet: équipée d'un FPGA Spartan 6 elle accueillera l'architecture chargée d'envoyer les trames de contrôles DCC, ainsi que d'un microcontrôleur qui servira à implémenter l'interface utilisateur.
- Locomotives miniatures « Jouef » : cc72000 DC/DCC
- Carte Booster : chargé d'adapter la tension sortant du FPGA en (-15V ;+15V) pour avoir une puissance suffisante afin d'être envoyée sur les rails

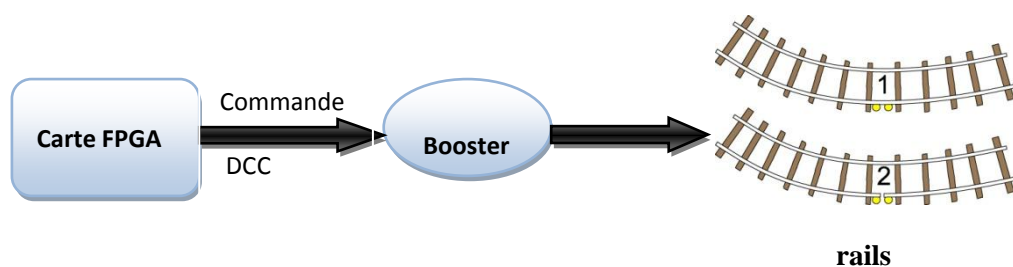
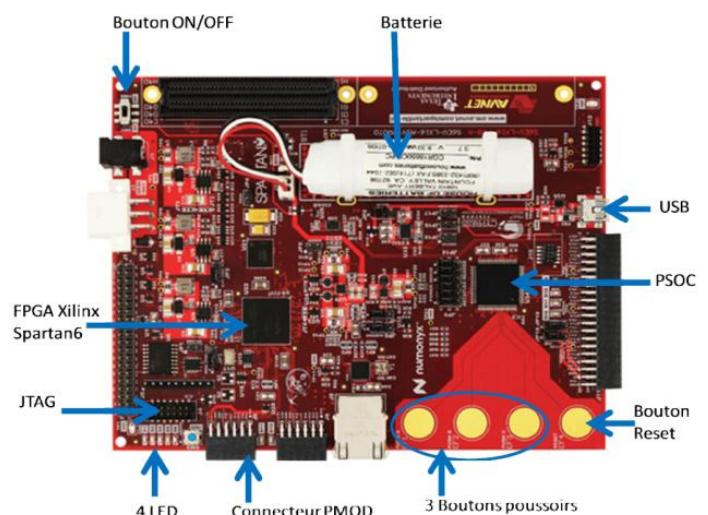
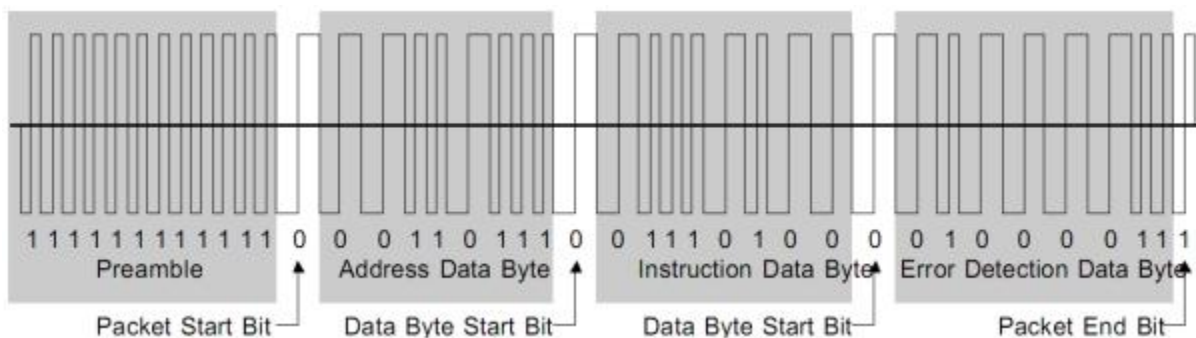


schéma de fonctionnement de l'envoi de la commande depuis la carte FPGA sur les rails

PRESENTATION DU PROTOCOLE DCC

Envoi de commandes sous forme de trames aux locomotives selon un protocole appelé DCC.
La commande est générée par un programme implémenté sur la carte FPGA , et correspond à la commande saisie par l'utilisateur (boutons ou UART), cette commande est sous forme de trames.

EXEMPLE D'UNE TRAME:



sur cet exemple on voit un seul octet d'instruction, mais il est possible de configurer la centrale de sorte à avoir deux octets d'instruction , ce que nous souhaitons faire, afin de pouvoir implémenter toutes les fonctions existantes.

COMPOSITION DE LA TRAME :

- Préambule : suite de 14 bits à 1 généré au début de la trame, il permet au décodeur de se synchroniser sur un début de commande (message).
- Bit de start : bit à 0 qui sert au décodeur à détecter l'arrivée de l'octet d'adresse.
- Adresse : octet qui contient l'adresse de la locomotive.
- Bit de séparation : bit à 0.
- instruction1 : octet d'instruction (fonction ou vitesse) composé de :
 - 3 bits : préfixe qui sert à identifier les groupes de fonctions
 - 5 bits : d'instruction (chaque bit active la fonction correspondante quand il est à 1).
- Deuxième octet d'instruction : instruction 2 (optionnel).
- Octet de vérification : octet qui sert à détecter d'éventuelles erreurs dans la trame envoyée.
- Bit de stop : bit à 1 .

Entre deux octets consécutifs, on retrouve un bit de séparation qui est toujours à 0.

Ce codage est indispensable afin que le décodeur puisse décoder la trame et reconnaître les instructions.

il est indispensable que la trame respecte le protocole DCC pour qu'elle puisse être décodée par le décodeur DCC de la locomotive.

Les différentes instructions possibles(22) qui peuvent être envoyées aux locomotives sont répertoriées dans le tableau suivant avec le codage associé comme expliqué précédemment :

F0	Lumière on/off	F11	Court cor FR #1
F1	Son on/off	F12	Court cor FR#2
F2	Cor FR #1	F13	Annonce station FR #1
F3	Cor FR #2	F14	Annonce station FR #2
F4	Turbo off	F15	Signal d'alerte FR #1
F5	Compresseur	F16	Signal d'alerte FR #2
F6	Accélération/freinage temps, bouchant mode/vitesse de manœuvre	F17	Porte chauffeur ouvrir/fermer
F7	Courbe grincement	F18	Valve
F8	Ferroviaire clank	F19	Attelage
F9	Ventilateur	F20	Sable
F10	Conducteur de signal	F21	Libération des freins

ORGANISATION DU PROJET

TACHES ET RESULTATS INTERMEDIAIRES

Pour faciliter le travail sur le projet, nous avons réparti les tâches de sorte à ce que chaque étudiant puisse correctement travailler sur sa partie indépendamment des résultats de l'autre.

Le travail à réaliser se décompose en plusieurs parties qui sont principalement le codage des différents modules qui composent la centrale DCC .

Le but étant de faire une centrale de contrôle DCC, nous avons réalisé un schéma de l'architecture de l'IP centrale DCC ainsi que du microcontrôleur(pour l'interface utilisateur), afin de pouvoir isoler chaque module.

Comme on peut le voir sur les schémas, le travail à réaliser se décompose en deux principales parties :

- **l'interface utilisateur (microcontrôleur)**
- **la centrale DCC**

L'interface utilisateur comprend principalement les modules qui s'occupent de la saisie des données (adresse de la locomotive, instructions...etc.) et les adaptent afin qu'elles puissent être lues et interprétées par la centrale DCC.

La centrale DCC qui comprend l'IP centrale DCC s'occupe quand à elle de l'interprétation des données envoyées par l'interface utilisateur et génère en fonction de ces données la commande qui sera envoyée à la locomotive.

Ces deux parties sont composées de plusieurs modules qu'on doit coder et implémenter sur la carte .

INTERFACE UTILISATEUR (MICROCONTRÔLEUR) :

- **LEDS** : sortie affichage des instructions, numéro du train, vitesse choisie.
- **Boutons** : entrée saisie des données : numéro du train, vitesse, fonction.
- **UART** : même fonction que les boutons (saisie de la commande).
- **MICROBLAZE** : *cœur de processeur qui sera implémenté sur la carte (microcontrôleur)*
 - récupère les données saisies (bouton, ou UART)
 - sélectionne le codage correspondant à la fonction/adresse/vitesse choisie
 - délivre une consigne qui contient ces informations
 - les écrit dans des registres (fonction write to register) ➡ AXI WRAPPER
- **AXI WRAPPER** : registres de 32 bits où sera enregistrée la consigne

CENTRALE DCC :

- **SEQUENCEUR** : MAE
 - interprétation des données présentes dans les registres de l'AXI WRAPPER , sépare la consigne et met chaque fonction sur 8bits : adresse/vitesse/fonction
 - pilotage des sends ; gère les modules send x de sorte à générer une trame qui correspond aux données saisies (suite d'octets (préambule/adresse/vitesse/fonction/contrôle) séparés par des bits de start , bit de stop à la fin de la trame)
- **SEND X** : modules send one , zero, preamble, byte qui vont générer des 1, 0, octets(consigne) et le préambule qui forment la trame à envoyer

Les bits et octets générés par la centrale dcc sont envoyés sur la carte (spartan 6) qui va envoyer la trame sur une broche de sortie. Le signal sera amplifié en +15 -15v par le booster avant d'être envoyé sur les rails (sous forme de signal carré).

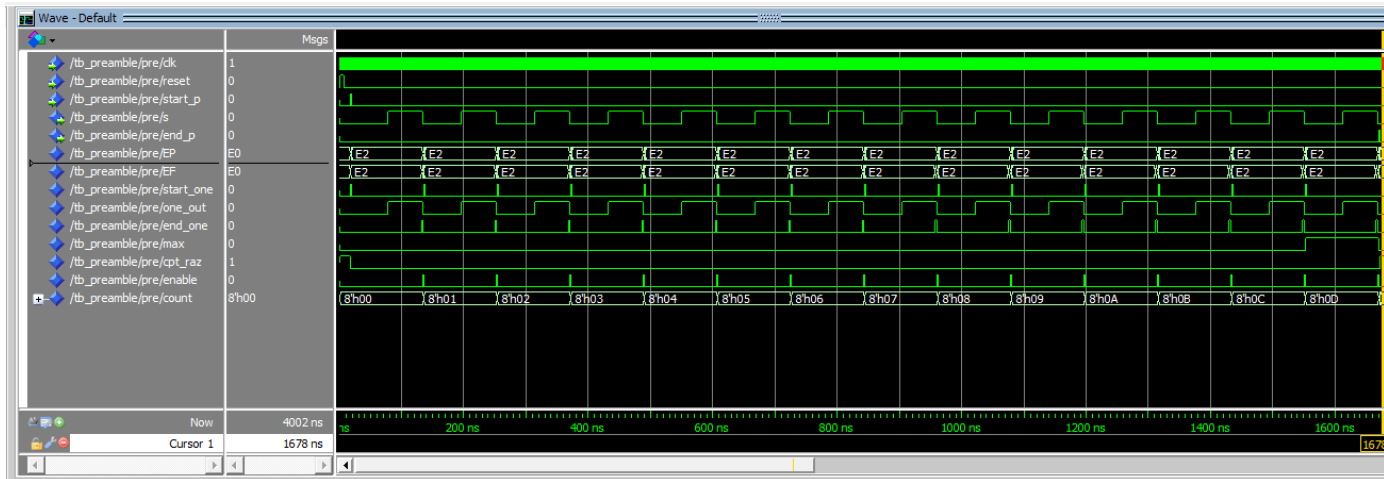
La trame sera ensuite transmise aux locomotives qui pourront la décoder grâce au décodeur présent dans les trains.

Les transferts de données (commandes) entre les modules (centrale DCC , microblaze, GPIO...) se font via un bus AXI qui est un bus d'interconnexion de la carte spartan 6.

AVANCEMENT

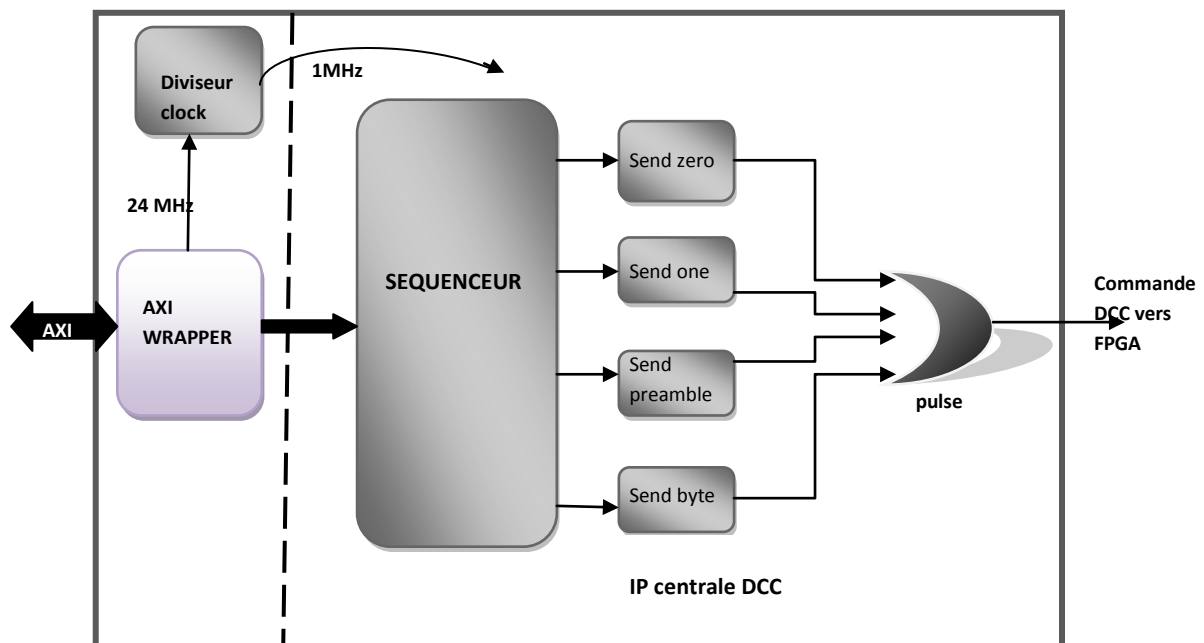
- codage de plusieurs modules : send one, send zero, send preamble , send byte, SEQUENCEUR.
- Test bench pour vérifier le bon fonctionnement de chaque module (simulation)
- prise en main de la carte spartan 6 (séances de TP)
- choix de l'architecture de la centrale, des améliorations que l'on souhaite apporter

EXEMPLE : SEND PREAMBLE



Module qui génère une suite de 14 bits à 1 (synchronisation) qui est déclenché par la mise à 1 de start_p (test bench : horloge, start_p, reset).

SCHEMA CENTRALE DCC



Ce schéma reprend les modules implémentés en VHDL (software) de la centrale DCC qui est chargée de délivrer une commande qui correspond au signal reçu via le bus AXI qui relie le microcontrôleur à la centrale DCC.

SCHEMA MICROCONTROLEUR

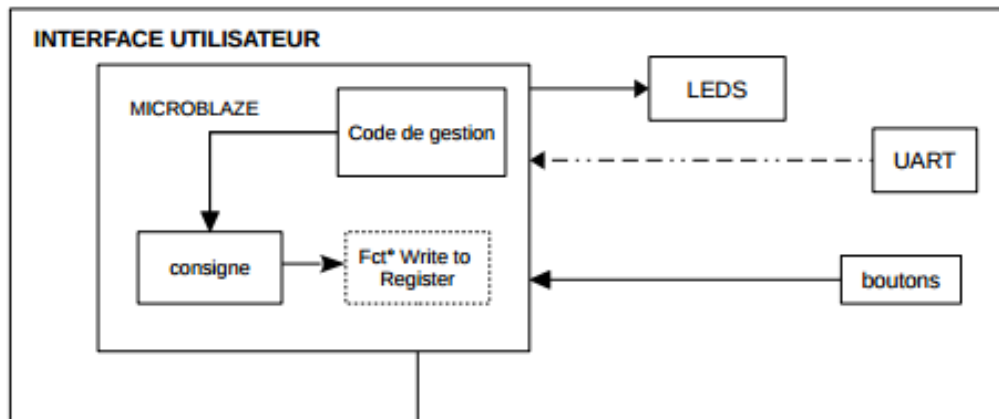
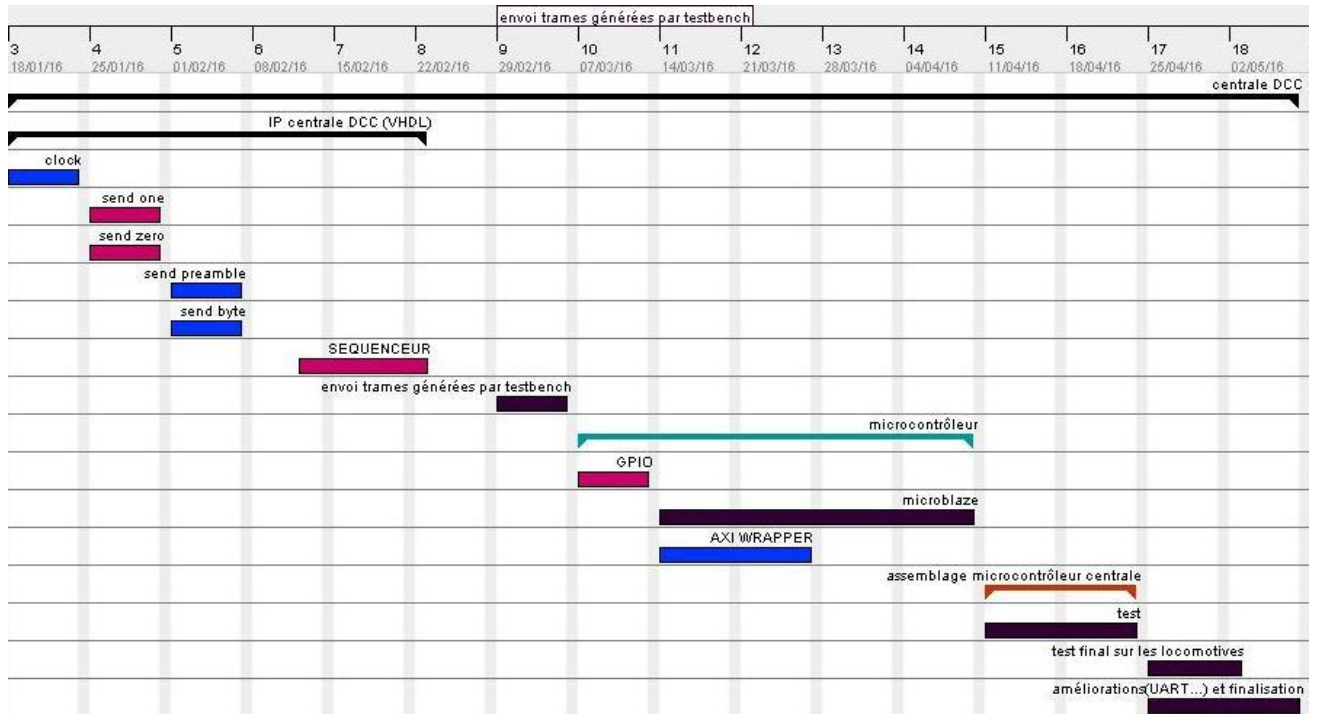


Schéma de l'interface utilisateur qui comprend le microcontrôleur qui gère la saisie de la commande via les boutons et retrouve la fonction/adresse/vitesse associée et la transfère à la centrale DCC (AXI WRAPPER) via un bus AXI.

VALIDATION

Afin de pouvoir valider les différents modules implémentés et par la suite, la centrale DCC complète, nous procéderons à des tests en simulation via des testbench propres à chaque module, ainsi qu'un testbench global (plusieurs modules). Nous effectuerons ensuite une vérification expérimentale via la carte, nous allons d'abord tester la centrale sans le microcontrôleur (sans l'interface utilisateur) en générant une trame par un programme VHDL qui sera envoyée sur la carte puis vers les locomotives afin de vérifier le bon fonctionnement des modules implémentés et de la connexion avec les locomotives (si les fonctions effectuées correspondent aux fonctions demandées par la trame...etc)

PLANIFICATION



- ◆ sarancoumar
- ◆ fatma
- ◆ sarancoumar et fatma

CONCLUSION

Nous avons choisi ce projet car ayant suivi une licence d'électronique, nous avons eu l'occasion de réaliser plusieurs projets au cours de notre cursus, en VHDL, durant lesquels nous avons développé un intérêt certain pour ce langage de programmation et les différentes applications qu'il offre, et avons acquis des compétences dans le domaine de l'électronique numérique, cela nous semblait donc logique de suivre sur cette voie et de postuler au projet qui vise à réaliser une centrale de commande de modèles réduits de locomotives.

Nos encadrants ont mis à notre disposition les moyens nécessaires pour la réalisation du projet, nous avons notamment pu tester la nouvelle centrale de commande récemment acquise par l'université, et nous avons pu constater les différentes fonctions directement sur les trains, ce qui nous a permis de mieux comprendre le fonctionnement des locomotives, et de voir les possibilités qu'offrent le système de commande DCC.

Nous espérons finir le projet dans les délais, et y apporter les améliorations que nous avons prévues. Ce projet nous permettra d'appliquer nos connaissances acquises au cours de notre cursus, de prendre des initiatives, de travailler en équipe et de respecter des délais imposés : des aspects essentiels du métier d'ingénieur.

BIBLIOGRAPHIE

- datasheet Xilinx Spartan -6 LX16 :
http://www.xilinx.com/support/documentation/data_sheets/ds162.pdf
- NMRA Communications Standards For Digital Command Control :
<http://www.nmra.org/sites/default/files/s-92-2004-07.pdf>
- NMRA Extended packet formats for digital command control:
http://www.nmra.org/sites/default/files/s-9.2.1_2012_07.pdf
- DCC - Digital Command Control for Model Train Sets:
<https://www.pc-control.co.uk/dcc.htm>
- Énoncé TP FPGA1 (systèmes programmables) , par J.DENOULET
- Cours FPGA1 , par B.GRANADO
- Datasheet locomotive “jouef” cc 72000