

IUT de Bordeaux 1  
*Département de Génie Electrique et Informatique Industrielle*  
33405 Talence Cedex

## Rapport de stage

# *Etude et mise en œuvre du réseau de terrain CAN*

Vanessa WALAS

**Société :ENSEIRB**

1, Avenue du Docteur Albert Schweitzer  
Domaine Universitaire  
33402 Talence Cedex



**Maître de stage : Patrice KADIONIK(ENSEIRB)**

**Enseignant responsable : Serge BOUTER(IUT GEII)**

*Année scolaire 2001-2002*



**Rapport de stage**

***Etude et mise en œuvre du  
réseau de terrain CAN***

## *Remerciements*

Je tiens à remercier vivement M.Patrice KADIONIK, mon maître de stage, pour m'avoir accueilli au sein de l'école mais également pour ses conseils, la confiance qu'il m'a accordée, et pour avoir mis à ma disposition la bibliographie dont j'avais besoin.

Je voudrais aussi remercier M. Serge BOUTER, enseignant à l'IUT Génie Electrique et Informatique Industrielle, pour m'avoir rendu visite.

Enfin, je remercie tous les techniciens de l'ENSEIRB pour leur disponibilité.

## Résumé

L'ENSEIRB a ouvert le département Télécom pour former des ingénieurs télécoms intégrateurs de systèmes. L'école investit massivement en matériels télécoms afin de permettre la mise en place de travaux pratiques et de projets dans ce domaine. Des travaux pratiques sur les réseaux de terrain de type CAN (Control Area Network) sont prévus à travers lesquels l'étudiant apprendra à utiliser un analyseur de protocoles CAN en vue d'étudier et d'analyser les trames CAN. Il convient donc de monter la manipulation qui se déroulera sur un PC sous Windows avec les logiciels NSI527 et CANalyzer.

## Abstract

ENSEIRB created the Telecom department to train embedded system telecom engineers. The school invest massively in telecom material in order to allow the place of practical exercises and project in this domain. Through practical exercises on the CAN Bus, students will learn to use a CAN protocol analyser in sight of study and analyse frames. Then, the manipulations are executed on a PC under Windows with the NSI527 and CANalyzer softwares.

# SOMMAIRE



<b>INTRODUCTION</b>	<b>1</b>
<b>I. Présentation de l'entreprise</b>	<b>2</b>
<b>II. Matériels et méthodes utilisés pour comprendre leur fonctionnement</b>	<b>3</b>
1) Analyseur de protocole	3
a) Carte CAN PCI	3
b) Carte CAN-AC2-PCI	3
2) Carte application	3
a) Carte E/S	3
3) Logiciels	3
a) NSI527	3
b) CANalyzer	3
<b>III. Présentation du bus CAN</b>	<b>3</b>
1) Historique	3
2) Son protocole	4
a) Couche application	4
b) Couche de communication de données	4
c) Couche physique	4
3) Les différentes trames	5
a) Trames de données	5
b) Trames de requêtes	7
c) Trames d'erreur	8
d) Trames de surcharge	8
e) Intertrame	8
4) Erreur de confinement	9
5) Les différentes erreurs	9
a) Au niveau de la couche physique	10
b) Au niveau du contenu de la trame	10
c) Au niveau de la structure de la trame	10
6) Etude de la couche physique	10
<b>CONCLUSION</b>	<b>12</b>
Glossaire	13
Bibliographie	14
ANNEXES: Gestion temporelle, TP	

# ***INTRODUCTION***

Le CAN (Controller Area Network) fait partie des bus de terrain les plus utilisés tant pour des applications automobiles que des applications industrielles (automatisme, etc.). En effet si nous prenons l'exemple de l'automobile la communication entre les différents modules nécessite un bus d'échange d'information en temps réel, tel que le bus CAN.

L'objectif de mon stage a donc été d'étudier le fonctionnement de 2 logiciels : NSI527 de NSI et CANalyzer de Vector sous Windows. Après avoir acquis les connaissances nécessaires, il m'a donc été possible de comprendre leur déroulement et ainsi de pouvoir mettre au point une série d'exercices pour l'analyse de trames CAN avec ces logiciels. C'est pourquoi après une brève présentation de l'ENSEIRB et du matériel, nous étudierons le bus CAN avant de décrire ces logiciels.

# I.Présentation de l'entreprise

Née en 1920 au sein de la Faculté des Sciences de Bordeaux, l'**ENSEIRB** est l'une des plus anciennes écoles d'ingénieurs dans le domaine de l'électronique. Elle est habilitée à délivrer le titre d'ingénieur diplômé en 1934 et prend le statut d'Ecole Nationale Supérieure d'Ingénieurs en 1975. En 1982, elle est habilitée à délivrer le diplôme d'ingénieur par la voie de la formation continue.

Une seconde filière d'ingénieurs dans le domaine de l'informatique s'est ouverte en 1986.

Actuellement, l'ENSEIRB possède 3 filières de formation :

- La filière Electronique avec les options Automatique - Robotique, Traitement du signal, Informatique Industrielle, Micro - électronique et Télécommunications.
- La filière Informatique avec les options Génie logiciel, Calcul Parallèle et Distribué, Réseaux et Systèmes Répartis, Technologie de l'Image et de la Communication.
- La filière Télécommunication avec les options Systèmes de transmission, Multimédia, Génie logiciel des réseaux et systèmes de télécommunication.

Pour répondre à ses missions : formation initiale et continue, recherche et transferts de technologies, l'**ENSEIRB** s'appuie sur des laboratoires de recherche de niveau international, tel que le laboratoire de Microélectronique **IXL**, le laboratoire Bordelais de Recherche Informatique (**LaBRI**), le laboratoire d'Automatique et de Productique (**LAP**) et l'Equipe Signal et Image (**ESI**), qui garantissent la compétence de ses enseignants, favorisent ses relations avec les entreprises, et facilitent son insertion dans des réseaux d'échanges internationaux.

L'école compte 650 élèves et est implantée au cœur du Campus Universitaire de Talence-Pessac-Gradignan, fréquenté par plus de 50 000 étudiants, à proximité immédiate des laboratoires, d'une bibliothèque universitaire, d'un village universitaire et d'un restaurant universitaire.





## II. Matériels et méthodes utilisés pour comprendre leur fonctionnement

### 1) Analyseur de protocole

#### a) **Cartes CANPCI**

Elles permettent de raccorder au réseau normalisé CAN des systèmes équipés d'un bus PCI. Elles intègrent 2 contrôleurs de protocole CAN 82527 d'Intel. Elles fonctionnent avec le logiciel NSI527 de NSI.

#### b) **Carte CAN-AC2-PCI**

Elles permettent d'interfacer des applications PC et des réseaux de type CAN. Elles fonctionnent avec le logiciel CANalyzer de Vector GMBH.

### 2) Carte application

#### a) **Carte CAN E/S**

Elle permet de se familiariser avec le fonctionnement du bus, de mettre au point des applications dans lesquelles des entrées-sorties sont raccordées sur le bus.

Cette carte intègre le composant 87L84 qui permet de contrôler les entrées et les sorties. Pour programmer ce composant comme on le souhaite, il a fallu comprendre son fonctionnement en étudiant sa data sheet. Par suite nous avons donc compris qu'il contenait plusieurs registres, qu'il faut charger avec les valeurs voulues.

Nous avons mis des interrupteurs dans la zone de câblage pour une utilisation plus simple des connecteurs en entrée.

### 3) Logiciels

#### a) **NSI527**

C'est un logiciel d'analyse et de simulation. C'est un logiciel facile à comprendre lorsque l'utilisateur connaît la composition des différents trames.

#### b) **CANalyzer**

C'est aussi un logiciel d'analyse mais plutôt sous forme graphique et plus performant. Il peut servir d'espion de la carte CAN E/S. Il a été assez facile à comprendre car nous disposions de son mode de fonctionnement qui était fourni avec le logiciel.

## III. Présentation du Bus CAN

### 1) Historique

Depuis le début des années 80, de nombreux systèmes électroniques ont fait leur apparition dans le domaine de l'automobile. Cette apparition peut être divisée en 3 étapes :

- L'époque où chaque système était totalement indépendant des autres :  
 Au début des années 80 quelques grandes sociétés automobiles s'intéressent à des systèmes de communication fonctionnant en temps réel entre différents microcontrôleurs. En 1983, le leader allemand d'équipements automobiles Robert Bosch GmbH commence à développer un protocole de communication.
- Celle où les systèmes commencent à communiquer entre eux :  
 A partir de 1986 des travaux de normalisation à l'ISO(Organisation de Standardisation Internationale) démarrent. En 1987 le 1<sup>er</sup> échantillon de circuit intégré CAN apparaît.
- Et enfin celle où tout le monde doit communiquer avec tout le monde :  
 Puis à partir de 1989, les 1<sup>ères</sup> applications industrielles se développent.

## 2) Son protocole

Le tableau suivant rappelle le découpage de la norme ISO/OSI(Open System Interconnections) et le contenu de ces différentes couches pour le protocole CAN.

Numéro de la couche	Modèle ISO/OSI	Protocole CAN
7	Application	Spécifié par l'utilisateur
6	Présentation	vide
5	Session	vide
4	Transport	vide
3	Réseau	vide
2	Communication de données	Protocole CAN (avec le libre choix du médium)
1	Physique	

### a) Couche Application

Cette couche est totalement vide.

### b) Couches de communication de données

Elle est divisée en 2 sous couches :

- LLC (Logic Link Control) :
  - filtrage des messages
  - notification des surcharges
  - procédure de recouvrement des erreurs
- MAC (Medium Access Control) :
  - mise en trame du message
  - arbitrage
  - acquittement
  - détection des erreurs
  - signalisation des erreurs

### c) Couche physique

Elle définit comment le signal est transmis. Elle définit les niveaux électriques, optiques...des signaux, le support de transmission. Elle est divisée en 3 parties :

- PLS(Physical Signalling) :
  - représentation du bit(codage, timing)

- synchronisation bit
- PMA(Physical Medium Attachment)
  - Caractéristiques Driver/Receiver
- MDI(Medium Dependent Interface)
  - connecteurs

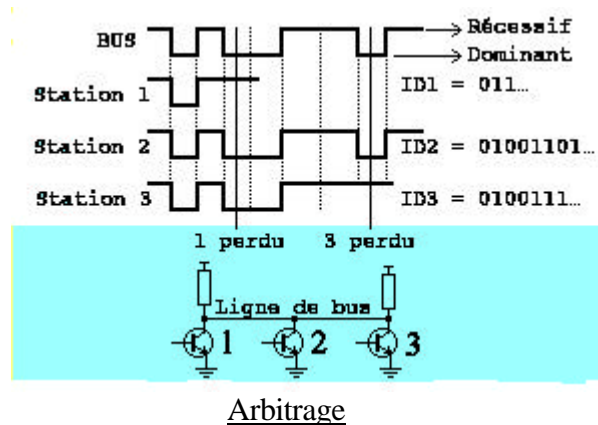
### 3) Les différentes trames

Il existe 2 formats différents :

- format standard : CAN 2.0 A : identificateur 11bits
- format étendu : CAN 2.0 B : identificateur 29 bits

Les bits sont soit dominants : '0' logique, soit récessifs : '1' logique

C'est au niveau des identificateurs que se fait l'arbitrage bit à bit des données. Lorsque plusieurs trames sont envoyées, le niveau du bit reçu est comparé au niveau du bit qu'il est censé transmettre lui-même. Si ces niveaux sont identiques, la station continue d'émettre. Dans le cas contraire, le niveau dominant prend le dessus, l'autre station perd l'arbitrage et cesse d'envoyer.



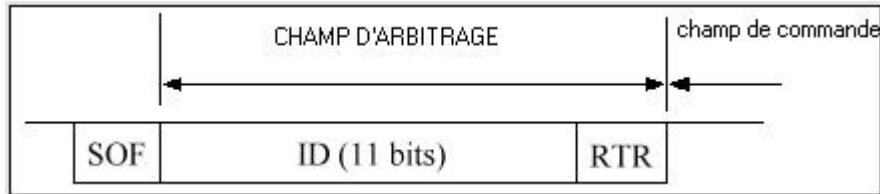
#### a) Trames de données



- Début de trame : SOF

Elle est constituée d'un seul bit dominant (0 logique), signalant à toutes les stations le début d'un échange. Il ne peut démarrer que si le bus était au repos.

- **Champ d'arbitrage**



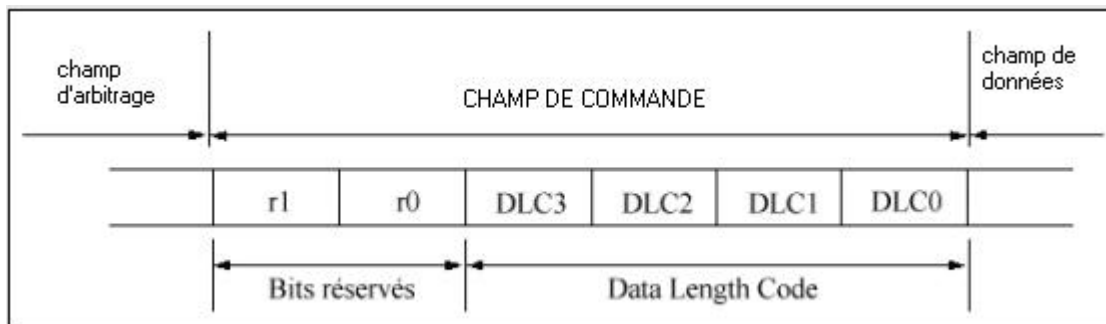
Il est constitué des bits de l'identificateur suivi du bit RTR (Remote Transmission Request).

Bit RTR :

- récessif : trame de requête
- dominant : trame de données

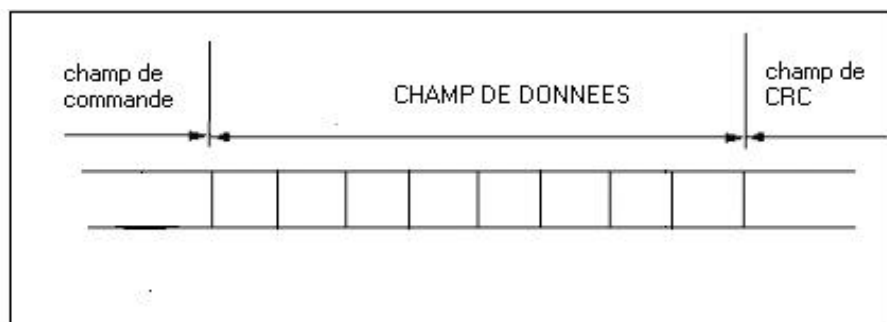
- **Champ de commande**

Il est constitué de 6 bits : les 2 premiers bits sont dominants, les 4 suivants indiquent le nombre d'octets contenus dans le champ de données.



- **Champ de données**

Il peut être composé de 0 à 8 octets transmis avec le MSB en tête. C'est l'endroit où se trouvent les données utiles transmises.



- **Champ de CRC** (Cyclic Redundancy Code : code de redondance cyclique)  
Il est composé de la séquence de CRC suivi d'un bit délimiteur de CRC (bit récessif).



Le polynôme, dont les coefficients sont les bits de début de trame, des champs d'arbitrage, de commande et de données complétés par des 0 pour les 15 coefficients des bits les moins significatifs, est divisé par le polynôme générateur :

$$X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1.$$

Le reste de cette division constitue la séquence CRC transmise sur le bus.

Pour s'assurer de la validité du message transmis, tous les récepteurs doivent procéder à la vérification de la séquence de CRC générée par l'émetteur CAN en relation avec le contenu du message transmis.

- **Champ d'acquiescement (ACK)**

Il est composé de 2 bits : bit ACK slot, bit ACK delimiter.

- ACK slot : bit dominant lorsque la station a validé la séquence CRC
- ACK delimiter : bit récessif

- **Fin de trame (EOF)**

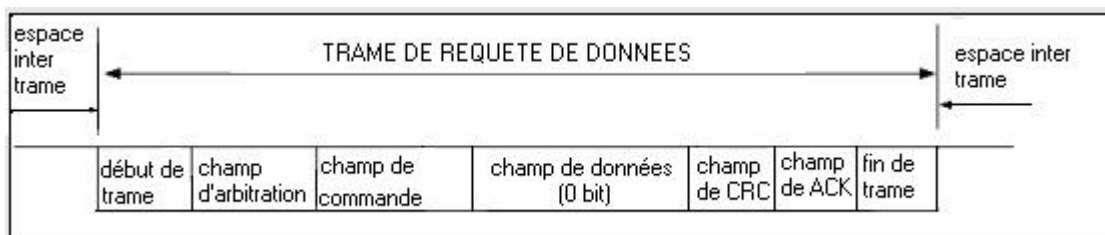
La trame de données se termine par une séquence de 7 bits récessifs.

### b) Trame de requête

Il se peut qu'une station ait besoin d'information, dans ce cas elle peut demander à ce que la station concernée les lui envoie par une trame de données.

Sa constitution est pratiquement identique à celle d'une trame de données.

La différence est qu'une trame de requête n'a pas de champ de données et le bit RTR est récessif.



### c) Trame d'erreur

Elle est composée d'un champ des drapeaux d'erreurs et d'un délimiteur de champ.



Le drapeau d'erreur indique s'il s'agit d'une erreur passive ou active.

Lors d'une erreur active le drapeau d'erreur est constitué de 6 bits dominants consécutifs. Et lors d'une erreur passive le drapeau d'erreur est constitué de 6 bits récessifs successifs. Ils n'obéissent pas à la loi du *bit stuffing* (Après 5 bits de valeur identique, un bit supplémentaire de valeur opposée est introduit).

Le délimiteur d'erreur est constitué de 8 bits récessifs.

### d) Trame de surcharge

Cette trame à pour but d'indiquer qu'une station est surchargée pendant un certain temps.

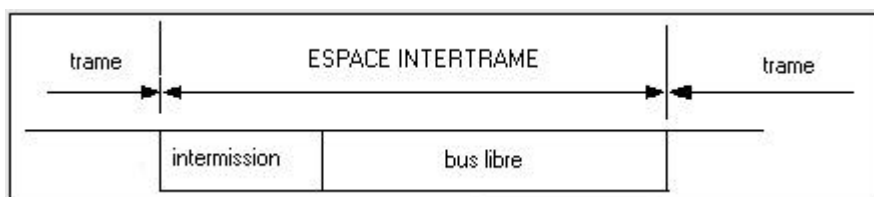


Elle est constituée d'un champ de drapeau de surcharge (6 bits dominants) et d'un délimiteur de champ (8 bits récessifs).

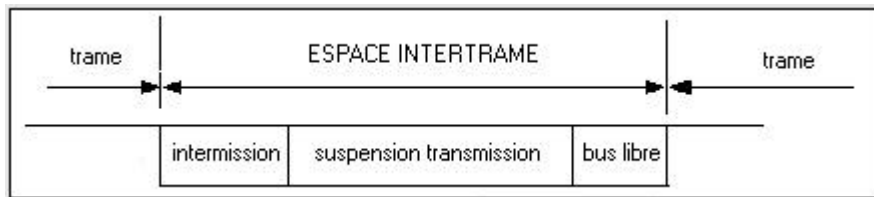
### e) Intertrame

Les trames de données et les trames de requêtes sont séparées des trames précédentes par un champ de bits appelé l'espace intertrame.

- Intertrame « erreur active »



- Intertrame « erreur passive »



- Champ d'intermission : 3 bits récessifs.  
Pendant ce temps aucune station n'est autorisée à démarrer la transmission d'une trame de requête ou de données.
- Champ de bus libre : la durée pendant laquelle le bus est libre est variable.  
Pendant ce temps n'importe quelle station peut émettre.
- Champ de suspension transmission : lorsqu'une station « erreur passive » transmet un message, elle envoie 8 bits récessifs suivant le champ d'intermission avant de démarrer l'émission d'un autre message.

#### 4) Erreur de confinements

C'est un mécanisme qui permet de faire la distinction entre un défaut temporaire et une panne permanente, il permet aussi de localiser les défauts. Une station peut prendre l'un des 3 états suivants :

- error active
- error passive
- bus off

Tous les microcontrôleurs conformes au protocole CAN doivent posséder 2 compteurs, qui doivent enregistrer les erreurs.

Mode de fonctionnement :

- si le message est transmis ou reçu correctement, le contenu du compteur respectif décroît
- si le message contient des erreurs, le contenu du compteur respectif augmente.

De 0 à 127 : état erreur active

La station peut continuer d'émettre, de recevoir et de transmettre un drapeau d'erreur active pendant la trame d'erreur.

De 128 à 255 : état erreur passive

La station peut continuer d'émettre, de recevoir et de transmettre un drapeau d'erreur passive pendant la trame d'erreur.

Au-delà de 255 : état de bus off

La station cesse d'émettre et de recevoir normalement.

#### 5) Les différentes erreurs

### a) Au niveau de la couche physique

- Bit entaché d'erreur :

L'émetteur CAN vérifie que le niveau du bit qu'il veut émettre sur le bus correspond bien à celui qu'il veut y disposer. Si ce n'est pas le cas, il signale un *bit error*.

- Erreur de bit stuffing :

Plus de 5 bits de même valeur consécutifs sont générés. Ceci peut être volontaire pour signaler une erreur.

### b) Au niveau du contenu de la trame

- erreur d'acquittement :

Cette erreur est détectée lorsque l'on n'a pas un bit dominant au ACK slot.

- erreur de CRC :

le récepteur calcule la valeur du CRC de la même façon que l'émetteur. Il y a erreur si les 2 valeurs sont différentes.

### c) Au niveau de la structure de la trame

- erreur de délimiteur de CRC
- erreur de délimiteur de ACK
- erreur de fin de trame
- erreur de délimiteur d'erreur
- erreur de délimiteur de surcharge

Pendant la transmission de ces champs, une condition d'erreur sera reconnue si un niveau dominant est détecté en lieu et place d'un niveau récessif.

Si l'une de ces erreurs est détectées, une trame d'erreur est envoyée sur le bus.

## 6) Etude de la couche physique

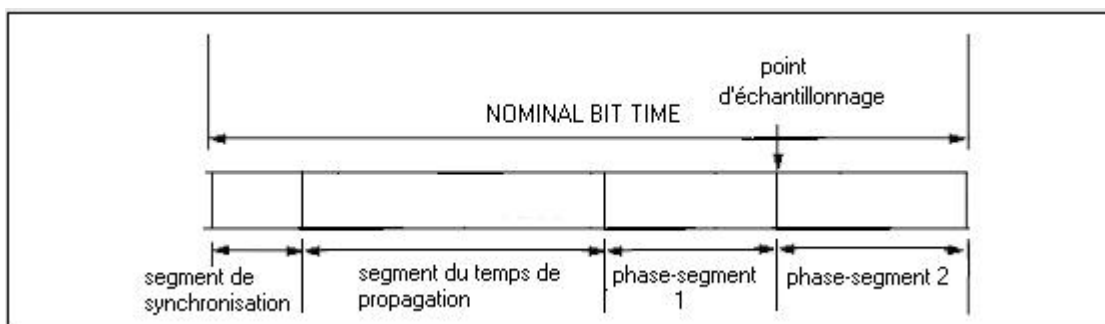
Nous allons maintenant étudier les problèmes concernant le bit (codage, durée, synchronisation...) au niveau de la sous couche de signalisation physique (PLS).

### • Bit time

C'est le temps que dure la présence effective du bit sur le bus.

### • Nominal Bit Time

C'est la durée nominale d'un bit.





Les longueurs des différents segments sont fonction d'une unité appelée ***Time Quantum***.

Il dépend de la période de l'oscillateur.

- *Segment de synchronisation* : 1 Time Quantum

Cette partie est utilisée pour synchroniser les divers nœuds du bus.

- *Segment du temps de propagation* : 1 à 8 fois le Time Quantum

Cette partie est conçue pour prendre en compte et compenser les temps de retard du au phénomène physique de propagation du signal.

- *Phase-segment 1 et 2*: 1 à 8 fois le Time Quantum

Ces segments sont utilisés pour compenser les erreurs de synchronisation. Ils peuvent être allongés ou raccourcis.

- *Point d'échantillonnage* :

Instant où l'état du bus est lu et où la valeur du bit est interprétée.

Il existe 2 modes de synchronisation :

- la synchronisation hardware :

elle est activée lors d'une transition qui se produit dans un bus au repos.

- la resynchronisation bit :

elle est activée lorsqu'il se produit une erreur de phase et il permet de recalculer temporellement le point d'échantillonnage.

## ***CONCLUSION***

Ce stage m'a permis de découvrir le bus CAN, et de me rendre compte de l'importance de ces réseaux dans la vie de tous les jours. En effet de nos jours les équipements dans un véhicule ou dans l'industrie se développent, et une communication entre les différents systèmes est nécessaire.

Suite à l'étude du composant COP 87L84 de la carte CAN E/S, nous avons pu comprendre la programmation du logiciel NSI527. Puis par la lecture du manuel de manipulation du logiciel CANalyzer, nous avons pu l'utiliser correctement. Grâce à la compréhension de ces logiciels, j'ai pu réaliser le manuel de TP qui permettra aux futurs étudiants de comprendre le fonctionnement des logiciels d'analyse NSI527 et du CANalyzer.

## **GLOSSAIRE**

Bus : conducteur ou ensemble de conducteurs communs à plusieurs circuits permettant l'échange de données entre eux.

CAN : Control Area Network

CRC : Cyclic Redundancy Code

DLC : Data Length Code

EOF : fine de trame

LLC : Logic Link Control

MDI : Medium Dependent Interface

MAC : Medium Access Control

PLS : Physical Signalling

PMA : Physical medium Attachment

Protocole : échange de données par des mécanismes standard

RTR : Remote Transmission Request

SOF : début de trame

## **BIBLIOGRAPHIE**

- ❖ *Le bus CAN*. Dominique PARET . Edition Dunod. 1998.
  
- ❖ *Le bus CAN. Application*. Dominique PARET. Edition Dunod.1999.

## *ANNEXES*

❖ Gestion temporelle

❖ TP

# GESTION TEMPORELLE

<b>Semaine</b>										
<b>Travail</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>
<b>Connaissance du bus CAN</b>										
<b>Connaissance et fonctionnement du logiciel NSI527</b>										
<b>Applications</b>										
<b>Connaissance et fonctionnement du logiciel CANalyzer</b>										
<b>Applications</b>										
<b>Rédaction de la manipulation et du rapport</b>										

