RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université de Mohamed El-Bachir El-Ibrahimi - Bordj Bou Arreridj
Facultédes Sciences et de la Technologie
Département d'Électronique

Mémoire

Présenté pour obtenir
LE DIPLÔME DE MASTER
FILIÈRE : **ÉLECTRONIQUE**
Spécialité : Électronique des Systèmes Embarqués

Par
- KHAMADJ Walid
- BAKOUR Bidjad

Intitulé

Etude et mise en œuvre du bus CAN : application à un tableau de bord de voiture

Soutenu le : 27/06/2022

Devant le jury composé de :

M. Med Elhossine DAACHI	*Président*	*MCA*	*- U. de Bordj Bou Arreridj*
M. Abdelhakim LATOUI	*Encadreur*	*MCA*	*- U. de Bordj Bou Arreridj*
Mlle Nacira DIFFELLAH	*Examinateur*	*MCA*	*- U. de Bordj Bou Arreridj*

Année Universitaire 2021/2022

Remerciement

Je remercie avant tout, DIEU le tout puissant qui m'a donnée la santé, la volonté, la patience et l'énergie pour réaliser ce travail.

Au seuil de ce travail je tiens à exprimer mes plus vifs remerciements et ma profonde gratitude à Mr LATAOUI Abdelhakim, pour ses encouragements, ses conseils avisés, orientations et critiques sui m'ont permis de mener à bien cette étude. Et je suis très reconnaissant pour la confiance qu'il nous a donnés.

Enfin, très nombreuses de gens qui ont participé à la réalisation de notre travail; mes amis, mes proches, qui m'ont toujours soutenu sans cesse et beaucoup d'autres personnes qu'ils m'excusent de ne plus pouvoir les citer.

Dédicace de BAKOUR BIDJAD

Je dédie ce travail

À ma mère belkorchia merzaka qui n'a jamais eu à exprimer son amour

À ma grand-mère c'est la personne la plus idéale dans ce monde, que je le dédie.

C'est vrai qu'elle n'est pas avec nous pour récolter le fruit de ses sacrifices, mais elle reste toujours la plus présente.

À mon frère bakour mouloud et ma sœur bakour noussaiba qu'ils m'ont toujours encouragé

À toute ma famille (Bekorchia Ibrahim, Djamar Mesouda, Fatima, Belkacem, Baziz)

A tous les personnes que j'ai autant aimées.

Dédicace de KHAMADJ WALID

Je dédie ce mémoire

À mes chers parents qui en été toujours à mes cotés et m'ont toujours soutenu tout au long de ces longues année d'études. En signe de reconnaissance, qu'ils trouvent ici, l'expression de ma profonde gratitude pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études,

À mes chères sœurs,……, pour leurs encouragements permanents

À mes chèrs frères,…….., pour leurs appuis continu

À mes chèrs amis,……, pour leurs soutien moral

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fuit de votre soutien infaillible.

Résumé

Dans ce travail nous avons mis en en œuvre une communication série par le fameux bus CAN pour la gestion et génération des trames CAN de contrôle du tableau de bord d'une voiture. Pour ce faire, nous avons utilisé deux cartes Arduino et deux modules interface Bus CAN MCP2515. La première carte Arduino permet d'acquérir les données des différents capteurs de la voiture et de les envoyer sur le bus CAN. La deuxième carte Arduino, quant elle, permet la gestion du tableau de bord de la voiture : affichage de la température, la vitesse, l'état des portes (ouvertes ou fermées), etc. De ce fait, un afficheur LCD I2C 4X20 caractères a été utilisé dans ce projet comme tableau de bord. Le système ainsi conçu a été réalisé et testé pour les différent capteurs utilisés dans ce projet à savoir : le capteur de température, de vitesse, de distance (dans le cas d'une marche arrière) et de capteur d'ouverture de porte. Le fonctionnement de notre système réalisé est tel que prévu dans le cahier les charges.

في هذا العمل قمنا بتنفيذ اتصال تسلسلي بواسطة ناقل CAN الشهير لإدارة و توليد إطارات CAN للتحكم في لوحة قيادة السيارة. للقيام لذلك إستخدمنا لوحين من Arduino و وحدتين لواجهةCAN MCP2515تستخدم بطاقة Arduino الأولى للحصول على البيانات من أجهزة الإستشعار المختلفة للسيارة وإرسالها على ناقل CAN .تسمح بطاقة Arduino الثانية, من جانبها, بإدارة لوحة قيادة السيارة : عرض درجة الحرارة, و السرعة, وحالة الأبواب (مفتوحة أو مغلقة), إلخ. لذلك, تم استخدام شاشة20*4 LCD I2C حرف في هذا المشروع كلوحة قيادة السيارة. تم تصميم النظام واختباره لمختلف المستشعرات المستخدمة في هذا المشروع, وهي : مستشعر درجة الحرارة, ومستشعر السرعة, ومستشعر المسافة (في حالة السير الخلفي), ومستشعر فتح الباب. يعمل نظامنا المحقق كما هو مخطط في المواصفات.

In this research, we have implemented a serial communication using the famous CAN bus for the sake of managing as well as generating CAN frames in order to control a dashboard of a car. In order to do so, we used two Arduino boards and two MCP2515 CAN Bus interface modules. The first board allows us to collect data from different sensors of the car and send them right on the bus CAN. For the second Arduino board, it manages the dashboard of the car as it displays the car's temperature, speed, the

status of doors (open or closed), etc. Therefore, a a 4X20 character I2C LCD display has been used in this project as a dashboard. Thus, the designed system was produced and tested for the various sensors applied in this work, namely: the temperature sensor, the speed sensor, distance sensor (in case of reverse gear) as well as the door opening sensor. The process of functioning of our realized system is as planned in the specifications.

Sommaire

Chapitre II: Concepts de base du Bus CAN

Chapitre III: Mise en oeuvre du bus CAN à base d'Arduino

Liste des Figures

Chapitre I: Généralités sur les Systèmes Embarqués pour l'automobile

Chapitre II: Concepts de base du Bus CAN

Chapitre III: Mise en oeuvre du bus CAN à base d'Arduino

Introduction générale

Introduction générale

Le bus CAN (Controller Area Network) a été initialement créé, en 1986, par Bosch et INTEL pour les applications automobiles. Les principaux objectifs étaient, de ce fait, la fiabilité et un faible coût. Depuis, les applications CAN gagnent du terrain et s'étend à l'automatisation industrielle, y compris l'électronique pour la marine et l'aéronautique [1]. En fait, le CAN est un bus série conçu pour fournir un réseau fiable, efficace, et une liaison très économique entre capteurs et actionneurs. INTEL et Philips furent les premiers à implémenter le protocole CAN dans des microcontrôleurs. En 1993, le CAN a été normalisé par l'ISO dans les normes 11898 pour les applications à haute vitesse (jusqu'à 1 Mb/s) et ISO 11519 pour les applications à basse vitesse (jusqu'à 125 kb/s).

Par ailleurs, en coopération avec d'autres experts CAN, Bosch a prédéveloppé la spécification CAN FD (Controller Area Network Flexible Data-Rate), officiellement introduite en 2012. Ceci a permet au bus CAN de s'offrir alors une deuxième jeunesse qui va certainement lui prolonger sa durée de vie non seulement dans le secteur de l'automobile, mais peut-être aussi dans l'industrie. Certains experts annoncent une prolongation de 10 à 20 ans [2]. En effet, l'automobile d'aujourd'hui dispose de nombreux équipements électroniques communicants (systèmes embarqués) et le bus CAN reste cependant incontournable dans ce genre communications.

Notre travail rentre dans ce cadre et nous nous proposons de mettre en œuvre une communication série par le fameux bus CAN pour la gestion et génération des trames CAN de contrôle du tableau de bord d'une voiture. Pour ce faire, nous avons utilisé deux cartes Arduino et deux modules interface Bus CAN MCP2515. La première carte Arduino permet d'acquérir les données des différents capteurs de la voiture et de les envoyer sur le bus CAN. La deuxième carte Arduino, quant elle, permet la gestion du tableau de bord de la voiture : affichage de la température, la vitesse, l'état des portes (ouvertes ou fermées), etc. De ce fait, un afficheur LCD I2C 4X20 caractères a été utilisé dans ce projet comme tableau de bord.

Outre cette introduction, le présent manuscrit est organisé en trois chapitres. Le premier chapitre introduit les concepts de base des systèmes embraqués pour l'automobile.
Le deuxième chapitre, quant à lui, est consacré à une présentation détaillée du bus CAN. Dans le troisième chapitre, nous avons présenté notre système réalisé au terme de notre projet de fin d'étude qui consiste à mettre en œuvreun bus CAN à base d'Arduino modélisant un tableau de bord d'une voiture.
Enfin, une conclusion situe la valeur de ce travail et présente perspectives.

Chapitre I

Généralités sur les Systèmes Embarqués pour l'automobile

I- Introduction

les progrès de l'électronique, de l' informatique embarquée et de l' automatisation moderne jouent tous un rôle important. Il permet le remplacement des composants mécaniques et hydrauliques (tels que la direction, les freins et la suspension). Tous les composants de la voiture sont devenus plus intelligents, permettant le couplage électronique des fonctions, rendant le véhicule plus communicant et permettant le développement de services associés. Dans l' automobile d'aujourd'hui, l' électronique est devenue un besoin.Impossible de passer à côté sans se faire remarquer. Elle est, après tout, le principal générateur de nouvelles idées. L'électronique est responsable de 80 à 90 % des progrès des véhicules [15].

À mesure que le nombre de calculatrices augmente, la complexité de leurs fonctionnalités et de leurs interactions augmente également. De même au fil du temps, les constructeurs ont acquis plus de compétences techniques et d' outils de test.

I-1- Présentations des Systèmes Embarqués

I-1-1- Définition

Un système embarqué est un système électronique et informatique qui est intégré dans un produit plus grand. Son premier objectif est de traiter les informations reçues de son entourage afin de maximiser les bénéfices du produit d' hébergement.

De nombreux chercheurs s'accordent à dire qu'un système embarqué peut être un système électronique et informatique autonome dédié à une tâche spécifique [16].

Les automobiles modernes, par exemple; peuvent être équipées d' un système de climatisation entièrement contrôlé par un système télématique. Ce dernier collecte des données sur la température et l'humidité de l' air à l' intérieur de la voiture et, sur cette base, détermine si le climatiseur, l'humidificateur ou le déshumidificateur doit être allumé ou éteint.

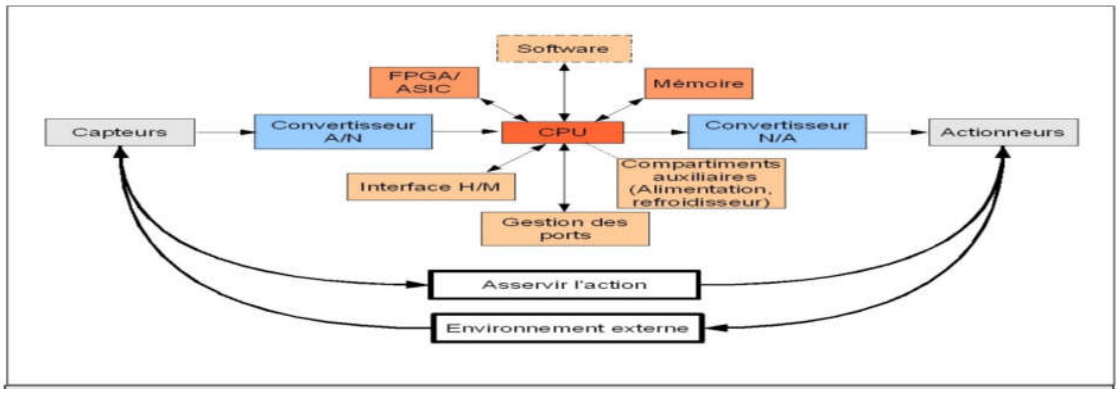


Figure I-1- Système embarqué typique.

I-1-2- La Conception d'un Système Embarqué pour l'automobile

L'application des systèmes embarqués dans les voitures est comme au message nerveux d'un humain. Il a besoin d'une main pour recevoir le sens du toucher (froid ou chaud) et ensuite envoyer l'information au cerveau pour analyse et décider ensuite si c'est dangereux. Il donne l'ordre au muscle de se contracter. La même chose dans les voitures Les sens résident dans les capteurs qui captent L'information provient du support externe et l'envoie au Calculateur qui est intelligent grâce à mon programme, puis prend la décision et instruit l'Actionneur pour faire une réaction et quel que soit le contrôle du Calculateur on l'appelle actionneur.

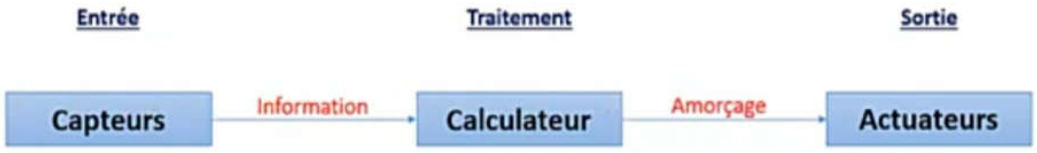


Figure I-2- Mode de fonctionnement d'un systèmes embarqués dans l'automobile

I-1-3- Composition d'un Système Embarqué

Un système embarqué doit pouvoir effectuer des tâches en temps réel. Il dispose de capteurs, d' actionneurs et d' une interface utilisateur pour y parvenir. Nous pouvons dire

que le système embarqué divisés en deux catégories principales(software et Hardware) Peut être considéré le Calculateur comme une partie software et pour les capteurs et les Actionneur Comme une partie Hardware.

De ce point de vue, nous aborderons les composants de base des systèmes embarqués:

I-1-3-1- Calculateur

La partie importante dans un système embarqué, se présente sous la forme de code machine utilisé les opérations du jeu d'instruction du processeur placement en mémoire en fonction de ses caractéristiques.généralement écrits dans les langages de programmation C++ ou Java; Dans une voiture, les calculateurs gèrent chaque système individuellement, mais ils peuvent aussi échanger des informations et interagir [16].

Pour ce faire, les calculateurs communiquent entre eux en utilisant différents langages selon le type de fonction (gestion moteur, confort, sécurité active et passive…etc) appelés réseaux multiplexés. Les réseaux diffèrent les uns des autres en termes de mode ou de protocole de communication (CAN, LIN, MOST) et de vitesse d' exécution, ou débit, en kbits/s (le réseau "confort" demandera moins de débit que le réseau "sécurité active ").

I-1-3-2- Les Actionneurs

Lorsque le calculateur termine le traitement, un signal électrique est envoyé aux actionneurs, leur permettant d' agir physiquement sur le véhicule. Ces actionneurs(ou actuateurs) convertissent des signaux électriques en énergie mécanique [15].

Tout ce qui est contrôlé par le calculateur on peut l'appeler actionneur (Relais, injecteur, moteur électrique et électrovanne…etc).

I-1-3-3- Les Capteurs

Le capteur est un dispositif de détection qui transforme une grandeur physique (température, Pression, position, accélération, luminosité...etc) à une grandeur

électrique (tension, intensité, capacité...etc) exploitable par un calculateur. Il présentée les grandeurs physiques.

Capteur inductif

Pour la détection de mouvements (vitesses de rotation, rotations de vilebrequin...ext) et de positions (position de vilebrequin), on utilise par exemple des capteurs à fonctionnement inductif (également appelés capteurs inductifs) [15].

Le principe physique régissant la génération de la tension inductive est basé sur la fluctuation du temps du champ magnétique. Le régime capteur, par exemple: équilibre les bosses de l'aile de l' avion; moteur, ainsi qu'un système de propulsion entraîné par les bosses.

Capteur à effet Hall

À l' aide d' un capteur à effet Hall, il est également possible de déterminer des vitesses de rotation (capteur de vitesse de rotation, capteur de vitesse du véhicule) et des emplacements (point d'allumage). Dans une sonde à effet Hall, l' UH (tension de Hall) est proportionnelle à la densité. Un champ magnétique B a été créé. Un écran rotatif permet de changer le champ de grossissement .phase avec la vitesse de rotation de la lumière, permettant la création d' un signal Avec le Champ Magnétique B, il y a une tension variable [15].

Capteur de température

La température du moteur et la quantité d' air inhalé fournissent des informations essentielles sur les étapes de charge du moteur au dispositif de contrôle électronique. Les capteurs de température mesurent électroniquement la température en fonction des changements dans l' environnement.Résistances mesurées comme une combinaison de résistances NTC et PTC. La plupart du temps, Les résistances NTC sont utilisées

L' abréviation NTC signifie Coefficient de température négatif, ce qui signifie que lorsque la température augmente, la résistance diminue.

L' abréviation PTC signifie Coefficient de température positif, ce qui signifie que lorsque la température augmente, la résistance augmente également [15].

Capteur de pression

Ces capteurs piézoélectriques sont utilisés comme capteurs de clic et de pression dans les collecteurs d' admission, tels que les systèmes d' injection, et ils signalent l' état de charge du moteur au dispositif de contrôle. Des capteurs piézoélectriques ou capacitifs sont utilisés dans la mesure de pressions absolues ou relatives. Lorsque ceux- ci sont soumis à une force, ils produisent une tension électrique.

I-1-4- Multiplexage

Le multiplexage est un système de communication qui permet la transmission simultanée de nombreux flux de données sur un seul support de transmission; Le multiplexage fait référence à la transmission de nombreux flux de données via un seul câble à l' ordinateur du véhicul. Réduction du nombre de capteurs et de la communication inter-boîtier puisque chacun fournit à l' autre, via le bus, les informations qu'il reçoit en filaire, c'est -à - dire le partage d'informations Son importance réside dans la facilité avec laquelle il peut être réparé [15].

- Son Multiplexage

Si vous voulez connaître la vitesse du véhicule, vous pouvez l'obtenir à partir d' un capteur de vitesse sur le boîtier, qui est censé fournir les données au calculateur d' injection , au calculateur d' ABS, au compteur de blocs, au calculateur de climatisation.

Le capteur attribue une valeur à chaque calculateur via des liaisons filaires utilisant de longs tronçons de câble.

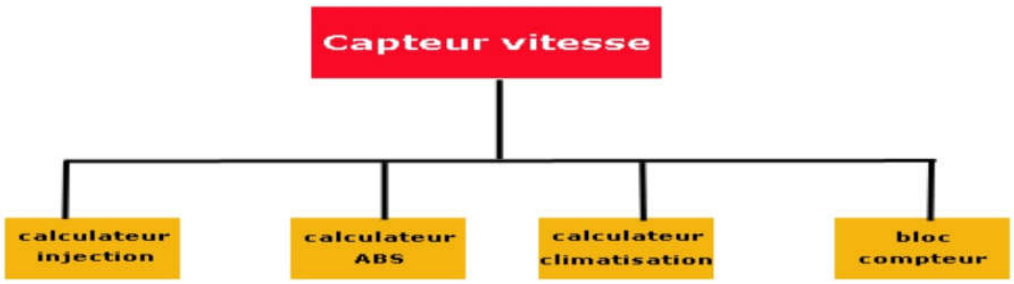


Figure I-4- Transfert des données son Multiplexage

- Avec Multiplexage

Le calculateur ABS fournit des informations sur le réseau multiplexé aux nombreux calculateurs qui en ont besoin.Toutes les données des différents capteurs sont fournies sur un réseau multiplexé sur une seule paire de fils croisés appelée Bus(CAN,VAN,LIN); et tous les calculateurs ont accès aux données dont ils ont besoin.

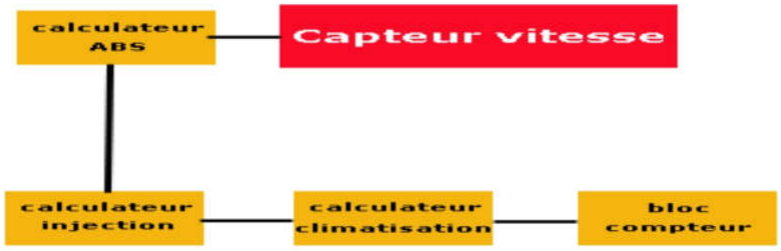


Figure I-5- Transfert des données avec Multiplexage

I-2- Principales caractéristiques d'un Système Embarqué

Les différences entre un système embarqué qui est un système dédié, et un PC, qui est un système à usage général, révèlent les caractéristiques d' un système embarqué.

Sur la liste suivante, il y a quelques différences à noter:
-Les PC sont des plates-formes informatiques à usage général, tandis que les systèmes embarqués sont dédiés à des tâches spécifiques.

-Une large gamme de processeurs et d' architectures de processeurs est disponible pour prendre en charge les systèmes embarqués.

- Dans certains cas, les systèmes embarqués doivent fonctionner dans des conditions environnementales extrêmes.

-La conception d'un système embarqué diffère de la conception logicielle ou matérielle traditionnelle en ce qu'elle prend en compte un certain nombre de facteurs liés à une variété de domains [16].

I-2-1- Les Systèmes Embarqués et les Contraintes temps réel

Le concept de temps réel est un peu flou; on peut le définir comme un système, mais il reste pertinent en matière de collecte et de traitement des données. Cela signifie que dans le cas d' informations asynchrones (souvent sous forme d' interruption); les temps d' acquisition et de traitement doivent être inférieurs au temps de rafraîchissement de l' information. Pour cela, le noyau ou le Real-Time Temps System doit être prédéterminé et proactif afin de toujours avoir le dessus lors de la prochaine interruption de la tâche la plus prioritaire [5].

Le fait d' être constamment en alignement temporel avec la réalité est ce qui définit le temps réel. Un système en temps réel doit non seulement produire un résultat correct mais doit le faire dans un laps de temps spécifique, sinon le résultat sera inexact.

Par conséquent, le système temps réel doit produire un résultat sous une contrainte de temps. L' environnement dans lequel se trouve le système détermine le temps; cet environnement doit avoir l' image la plus réaliste de son environnement extérieur, qui évolue avec le temps. Les systèmes embarqués sont soumis à diverses contraintes, en fonction de l'usage auquel ils sont destinés. En effet, l'une des contraintes les plus importantes de l' industrie automobile, tant en termes de performances que, plus important encore, en termes de sécurité, est le chronométrage en temps réel. L' échelle de temps varie en fonction de l' environnement dans lequel le système est utilisé et peut

aller de quelques millisecondes à plusieurs heures. Lorsque nous sommes dans notre voiture, il est préférable de voir la vitesse à laquelle nous roulons en millisecondes; afficher la vitesse à plusieurs minutes d' intervalle rendrait les données totalement fictives [6].

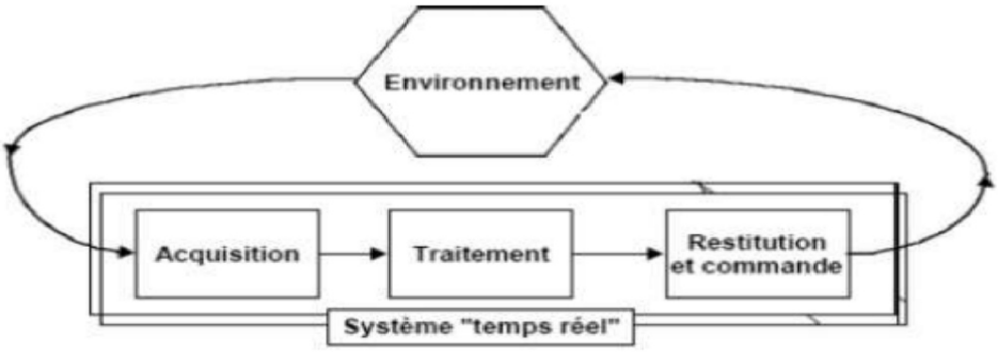


Figure I-6- Représentation en temps réel d'un système avec son environnement

I-2-2- Système d'exploitation temps réel (RTOS)

Un système d'exploitation temps réel (RTOS : Real-Time Operating System) est un système d' exploitation multi-tâches conçu pour les applications temps réel. Il fournit un support fondamental pour la planification des tâches (ordonnancement), la gestion des ressources, la synchronisation et la communication inter-processus, et la gestion des entrées/ sorties, entre autres. La différence entre un RTOS et un système d'exploitation non temps réel est qu'il fournit des services supplémentaires similaires à ce dernier, tels que des services adaptés au développement d' applications temps réel et un support pour le portage de l' ensemble du système/des applications vers le matériel desystèmes embarqués. sont fréquemment soumis à des contraintes de temps temps réel, et un RTOS est rarement conçu pour un usage général, car l'installation de nouveaux logiciels sur un RTOS est difficile.

Bien qu'un RTOS facilite la construction d' un système en temps réel, il ne garantit pas que le résultat final sera en temps réel. Une grande partie de cette assurance réside dans le logiciel qui s'exécute sur le RTOS, ce qui nécessite un développement

minutieux et approfondi sur lepartie des développeurs. Il est également important de prendre en compte ce que l'on appelle les paramètres non fonctionnels, qui sont généralement associés aux caractéristiques techniques du système telles que la fiabilité, la consommation d'énergie ou la latence de la prise en charge des communications [7].

I-2-3- Les Systèmes Embarqués et Contraintes d'énergie

Dans les systèmes électroniques, et particulièrement dans les systèmes embarqués, la question de la consommation est devenue critique. Comme on peut le voir, la gestion de la consommation d' un système est une question difficile et de grande envergure qui nécessite des compromis réguliers et dynamiques entre performance et consommation d'énergie.

Il est vrai que les PC ont peu ou pas de restrictions alimentaires car ils sont alimentés par des prises électriques domestiques courantes de 110 à 220 volts. Pour les systèmes embarqués; On ne peut pas dire autant, Ils sont généralement alimentés par des batteries dont la capacité et la durée sont limitées. Ceci est clairement visible sur les téléphones portables et les PDA lorsque les utilisateurs doivent les recharger régulièrement. Pour les sondes d' exploration autonomes, telles que les sondes spatiales, ce problème est évident. Ils n'ont que des panneaux solaires comme source d' énergie parce qu'ils sont dans un environnement si difficile.

En général si la conception du système est contrainte par des contraintes de puissance le logiciel doit être conçu de manière à ce que le processeur reste en mode veille la majorité du temps, ne se révélant qu'en cas de perturbations; En d'autres termes le système est entièrement axé sur le réveil et la surveillance du processeur [8].

Dans les lignes précédentes, on remarque que le processeur est fréquemment placé en premier lors de la conception d' un système soumis à des contraintes énergétiques. C'est tout à fait raisonnable, étant donné que c'est le composant des systèmes embarqués qui consomme le plus d'énergie en pratique. Contrairement au marché des PC, la fréquence d' exécution des processeurs embarqués n'est pas le seul critère de

qualité; il en existe plusieurs autres, comme la consommation d'énergie, la résistance aux éléments physiques tels que la chaleur, les interférences radio, le rayonnement solaire…etc.

I-2-4- La Connectivité de Système Embarqué

Les systèmes embarqués sont désormais hautement connectés. C'est désormais possible grâce à la puissance de calcul accrue offerte par les processeurs embarqués récents (notamment 32 bits), ainsi qu'à l' essor de la connectivité Internet ou IP. La connectivité IP est essentielle pour contrôler à distance un système connecté à Internet. Il s'agit en fait de la mise en œuvre d' un contrôle à distance d' un système électronique utilisant une variété d' interfaces : RS.232, RS.485, bus de terrain…etc.

Cela permet à l' utilisateur d' utiliser les technologies Web modernes pour le contrôle à distance; il suffit d' installer un serveur Web sur son équipement électronique, ce qui lui permet de le contrôler de n'importe où à l' aide d' un simple navigateur Web. La construction d'un IHM devient plus facile grâce au rôle joué en grande partie par le navigateur Web.

Il est également intéressant de noter que la puissance des communications sans fil a augmenté dans l' embarqué, au détriment des communications filaires, afin de réduire le câblage et de faciliter l' installation du système embarqué. Les réseaux Wifi et toutes les normes de réseaux sans fil IEEE 802.15 sont généralement utilisés dans les applications embarquées, notamment en domotique (réseaux de capteurs sans fil par exemple). Mais, malgré ces commodités et avantages il y a un inconvénient : la sécurité du système embarqué, sachant qu'il est connecté à Internet ou accessible via un moyen ne nécessitant pas de connexion physique, comme le sans fil .De ce fait, la sécurité des systèmes embarqués est critique, et elle doit être prise en compte dès le début de leur développement [5].

I-3- Système Embarqué et Médias externes

I-3-1- Classification et Domaine d'application des Systèmes Embarqués

Pour les systèmes embarqués, il existe plusieurs critères de classification tels que le gabarit, le type de calcul, et la catégorie de marché économique dans laquelle l' appareil est placé. C'est le critère le plus utilisé dans la littérature sur les systèmes embarqués; il s'agit d'une catégorisation dans laquelle les appareils embarqués sont grossièrement divisés en différents segments de marché.

Nous pouvons le résumer comme suit :

- Transport (véhicule en générale) :

Mis-en-marche du véhicule -Contrôle du moteur -Système de freinage.

- Consommables électroniques :

Téléviseur numérique -Domotiqu -Jeux vidéo -PDA -Téléphone -GPS -Caméra

- Contrôle industriel :

Robotiques -Systèmes de contrôle industriel -Automate programmable industriel

- Médicale :

Moniteur cardiaque -IRM -Endoscope -Dialyseur

- Réseaux et telecommunication :

Routeur -Hub -Passerelle

- Équipements de bureau :

Fax -Photocopieur -Imprimante -Scanner -Moniteur

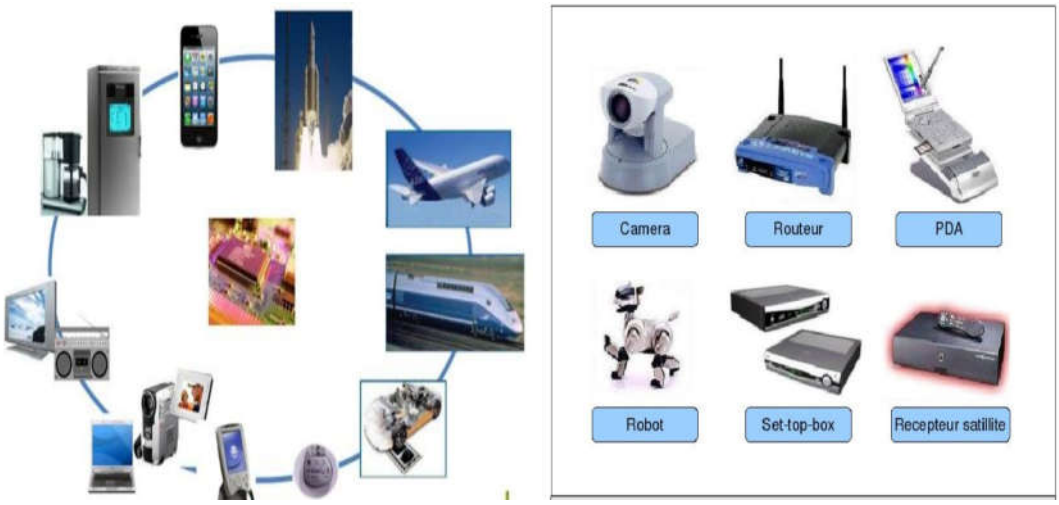


Figure I-7- Quelques applications du système embarqué

I-3-2- La touche des systèmes embarqués dans la voiture moderne

Récemment, la révolution électronique dans la conception automobile a été maîtrisée grâce au rôle essentiel joué par les systèmes embarqués multi-usages.

Examinons quelques des différents rôles des systèmes embarqués dans les voitures :

I-3-2-1- Système de freinage (ABS)

Sur les routes lisses ou glissantes, ce système consiste à empêcher les voitures de glisser.Ce système de freinage aide les roues à maintenir une meilleure connexion avec la route. Des capteurs, un contrôleur, une vitesse, une pompe et des vannes de suivi peuvent tous être utilisés pour le construire.

L'unité de commande surveille le mouvement des roues du véhicule et distribue la force de freinage de manière égale aux deux roues du même essieu, en augmentant ou en diminuant la pression de freinage provoquée par les soupapes, et ainsi un freinage précis en un temps très court et avec peu de conducteur [9].

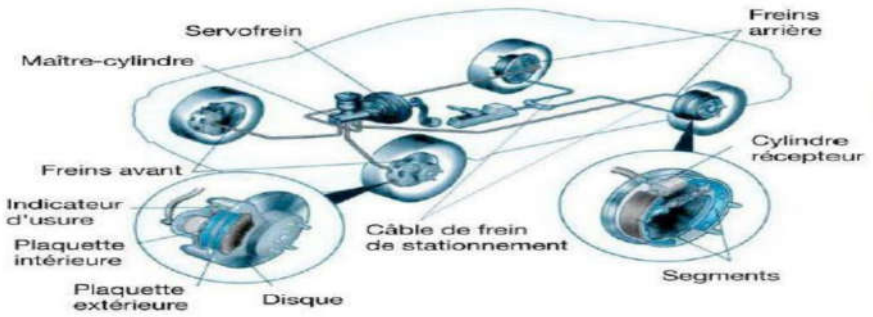


Figure I-8- Système de freinage dans l'automobile.

I-3-2-2- Système d'airbag (SRS)

Un courant sera envoyé au système d' éclairage chaque fois que la méthode de collision est utilisée.Le filament absorbera la chaleur du courant électrique et éclairera la tablette, permettant la production de gaz. Chaque fois que la pression du gaz augmente, le coussin gonflable se dilate pendant une fraction de seconde. ils sont conçus pour augmenter en cas de collision frontale dans les automobiles [10].

I-3-2-3- Système de climatisation

Le système de climatisation peut être considéré comme un dispositif de sécurité active car il fait perdre certains de ses réflexes au conducteur lorsqu'il roule longtemps à des températures extrêmes. Pendant l' hiver, le chauffage augmente la température de l' air qui circule dans la voiture, ce qui crée un environnement plus confortable, La climatisation rafraîchit la voiture en été [9].

I-3-2-4- Régulateur de vitesse

Chaque voiture est équipée d' un régulateur de vitesse adaptatif et d' un radar qui agit comme un émetteur-récepteur et est monté sur le tableau de bord pour déterminer la vitesse et la distance du véhicule par rapport aux autres véhicules sur la route; L' ordinateur connecté à l' unité ACC aide à contrôler les freins et l' allumage du véhicule.

Le meilleur exemple est un véhicule autonome appelé «véhicule sans conducteur ». Ceci n'est réalisable qu'avec l' utilisation de systèmes embarqués dans les automobiles.

Actuellement, il est largement utilisé dans les automobiles pour créer l' espacement le plus court entre les véhicules en circulation dense sur les autoroutes. Au fur et à mesure que la quantité d' embouteillage diminue, ce système aidera à modifier la vitesse du véhicule en utilisant un système de freinage [10].

I-3-2-5- Système de stationnement automatique

Ce système est un système autonome de manipulation de véhicule qui permet de déplacer un véhicule d' une voie de circulation vers une place de stationnement afin d' obtenir un stationnement perpendiculaire, parallèle et en angle. Les capteurs de ce système sont montés sur la face avant du véhicule, et un parachute orienté vers l' arrière sert de Tx et Rx. Lorsque ces capteurs détectent un obstacle autour du véhicule, ils envoient un signal à l' ordinateur, qui reçoit le signal horaire, et les pare-chocs utilisent le radar pour localiser l' objet. Après être entré dans l' aire de stationnement, le véhicule remarquera la place de stationnement ainsi que l' espace le long du côté de la route [10].

Pour détecter des objets à proximité d' un véhicule, le système proposé utilise une variété de procédés.

En plus des systèmes que nous avons mentionnés précédemment, il y en a beaucoup que nous ne pouvons pas mentionner dans notre mini-mémoire, qui sont les suivants:

-Système hydraulique -Conduire à travers le fil -Vision nocturne -Affichage tête haute

-Sauvegarde des capteurs de collision -Radio satellite -Contrôle des émissions

-Contrôle de traction -Vision nocturne -Moniteur de pression des pneus -Boîte noire

-Systèmes de divertissement embarqués -Systèmes de navigation -Télématique

I-3-3- Le Coût des Systèmes Embarqués

La plupart du temps, le coût de l' architecture globale du système (microprocesseur, mémoire et bus) est associé au coût d' un système embarqué. Étant donné que le

processeur est généralement le composant le plus cher d' un système embarqué, certains concepteurs pensent qu'il s'agit du facteur le plus important pour déterminer le coût d' un système embarqué,

De plus, la légitimité de cet agrément s'érode du fait de l' émergence des puces SoC (System On Chip). Un SoC typique contient les composants suivants : un processeur, un DSP, de la mémoire (RAM, ROM, flash…etc) des convertisseurs analogique/numérique et plusieurs unités E/S. En d'autres termes, l' ensemble du système est hébergé sur un seul ordinateur, ce qui réduit considérablement l' encombrement, la consommation d' énergie et le coût du système (d' un point de vue matériel) [8].

Comme ces petits appareils électroniques sont fréquemment vendus à grande échelle, le coût d' un système est crucial dans le domaine des systèmes embarqués. Cela signifie qu'un petit changement dans le prix d' un produit peut entraîner un gros profit ou une grosse perte lorsqu'il est multiplié par le nombre d' unités vendues. Il est souvent affirmé que les systèmes embarqués sont rentables.

I- Conclusion

Ce que nous pouvons retenir de ce chapitre, c'est que le domaine des systèmes embarqués est un domaine vaste et multidisciplinaire qui a réussi à rassembler des domaines auparavant disparates pour devenir l' une des technologies les plus utilisées aujourd'hui.

Les systèmes embarqués sont donc des outils extrêmement puissants capables de fournir un large éventail de fonctions et d'effectuer un nombre insondable de tâches. Cependant, en raison de leur complexité de conception et de leur environnement intrinsèquement instable, ces systèmes ne sont pas en mesure de fournir une fiabilité et une sécurité opérationnelles maximales à leurs utilisateurs. Ces travaux démontrent la présence et l' intégration des systèmes embarqués dans le domaine automobile, ainsi qu'une augmentation significative de leurs applications dans plusieurs domaines, tels que la sécurité des véhicules, le confort et la conduite directe des véhicules.

Chapitre II

Concepts de base du Bus CAN

II- Introduction

Le Controller Area Network (CAN) est un protocole de communication série qui permet un contrôle distribué en temps réel tout en maintenant un haut niveau de sécurité.

Bosch GmbH l'a développé dans les années 1980; Il a été conçu pour remplacer le faisceau de câbles standard par un système simplifié à deux bus dans le secteur automobile. Les exigences du système prévoient des tolérances EMI élevées ainsi que la capacité d'auto-diagnostiquer et de résoudre les problèmes. Il est connu sous le nom de bus multi-maître car il connecte de nombreuses unités de contrôle appelées nœuds.

CAN est un système dans lequel tous les nœuds reliés au bus CAN peuvent entendre simultanément chaque message reçu. Des débits de données CAN jusqu'à 1 Mbit/s sont utilisés pour connecter les appareils. Les microcontrôleurs et autres processeurs peuvent communiquer entre eux de cette manière.

II-1- Présentation générale du Bus CAN

II-1-1- Définitions

Pour bien appréhender le Bus CAN, il est nécessaire de définir les nombreux termes techniques utilisés tout au long du document.

Bus de communication : Un bus de communication est un dispositif non bouclé qui relie plusieurs composants, sous-ensembles ou équipements pour permettre le transfert d'énergie et la circulation d'informations entre eux.

Bus CAN : Le protocole CAN est le protocole de communication le plus utilisé pour les systèmes de contrôle et de commande des véhicules [6].

Il possède les qualités suivantes : La capacité de travailler avec plusieurs maîtres avec une communication diffuse et fonctions de détection d'erreur sophistiquées.

De nombreuses télécommunications entre équipements informatiques composent le réseau.Le réseau peut être (Etoile, Maillé, Bus, Anneau).

Une trame: est un regroupement logique d'éléments numériques qui est défini par un protocole de communication.

L'architecture de la communication: fait référence à la structure des éléments qui composent une communication (Entités qui communiquent et Les règles qui régissent la communication entre les entités communicantes).

Un protocol: est un ensemble de règles et de procédures qui doivent être suivies pour envoyer et recevoir des données sur un réseau. Il s'agit d'une méthode standard qui permet la communication entre processus (éventuellement exécutés sur différentes machines) [6].

Un nœud: est un objet connecté à un réseau.

Le multiplexage: est le processus de combinaison de plusieurs signaux en un seul signal composite qui peut être reproduit. Le multiplexage peut être effectué de différentes manières, notamment en termes de fréquence, de temps, de code, de longueur d'onde…etc.

Couche OSEK :

OSEK signifie «Offene Système Under en Schnittstellenfür die Elektronikim Kraftfahrzeug», qui se traduit par «Systèmes ouverts et interfaces correspondantes pour l'électronique automobile» en français. Il comprend des entreprises telles que Daimler, Chrysler, Opel, Siemens et Volkswagen, ainsi qu'un département universitaire. Leur objectif était de créer une norme pour une architecture ouverte qui connecterait les nombreuses commandes électriques d'un véhicule. Renault et PSA, qui travaillaient sur un projet similaire appelé VDX (VehicleDistribute by Xecutive), ont rejoint le consortium en 1994.L'architecture ouverte OSEK/VDX est divisée en trois parties :

la communication échange de données entre les unités de contrôle, l'administration du réseau et le système d'exploitation temps réel [11].

II-1-2- Historique

CAN (Controller Area Network) a été proposé pour la première fois par Bosch en février 1986 lors du congrès SAE (Society of Automotive Engineers). Le premier véhicule à être équipé du nouveau protocole est sorti en 1992.

Les réseaux CAN sont désormais installés dans pratiquement toutes les nouvelles voitures particulières fabriquées en Europe. CAN s'est imposé comme le premier protocole de réseau, utilisé dans différentes formes de véhicules de transport (trains, bateaux, avions, etc.) ainsi que dans le secteur industriel.

1983: Lancement du projet interne Bosch en partenariat avec l'université Karlsruhe pour développer un réseau dans le véhicule.
1986: Bosch présente officiellement et pour la première fois le protocole CAN pendant le congrès de la SAE (Society of Automotive Engineers).
1987: Apparition des premiers circuits imprimés liés au CAN par Intel et Philips Semi-conducteurs.
1991: Publication par Bosch des spécifications de la version 2.0.
Kvaser introduit le CAN Kingdom (protocole de haut niveau basé sur CAN) utilisé par les fabricants de machines textiles.
1992: Création du groupe CiA (CAN in Automation) établi par des fabricants et utilisateurs du CAN. Premiers véhicules équipés du réseau CAN; Mercedes Classe S.
1993: Publication de la norme ISO 11898.
Création du groupe OSEK (Systèmes ouverts et interfaces correspondantes pour l'électronique des véhicules automobiles).
1994: CiA organise la première conférence internationale sur le CAN (iCC).
Allen-Bradley introduit le protocole DeviceNet utilisant la technologie du CAN.
PSA (Peugeot – Citroen) et Renault rejoignent le groupe OSEK.
1995: Publication d'un amendement pour la norme ISO 11898 (format de trame étendu).
CiA publie le protocole CAN open.

2000: Développement du protocole TTCAN (time-triggered communication protocol for CAN).

2012: Bosch présente officiellement l'évolution du Bus CAN : le CAN-FD lors de la 13e conférence internationale (iCC).

II-1-3- Principe de fonctionnement

Le bus CAN est un bus de communication de données qui relie plusieurs nœuds ou ECU (unité de contrôle électronique). Chaque nœud ou ECU a la capacité de communiquer avec les autres. Les données sont transmises sur une paire connectée en utilisant la transmission différentielle, qui consiste à mesurer la différence de tension entre les deux lignes (CAN H et CAN L). À chaque extrémité de la ligne de bus, des résistances de 120 ohms doivent être installées. Les paquets de messages sont utilisés pour la communication et les vitesses de transmission peuvent approcher 1 mégabit par seconde (pour Can High Speed) [7].

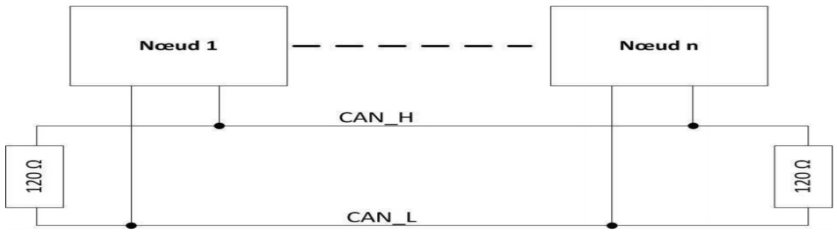


Figure II-1- Fonctionnement du Bus CAN.

II-1-4- Normes et spécifications

Les organismes de normalisation nationaux et internationaux sont les deux types d'organismes de normalisation.

• International : Toutes les utilisations non électriques sont régies par l'Organisation internationale de normalisation (ISO), tandis que tous les équipements électriques et électroniques sont régis par la Commission électrotechnique internationale (CEI).

• Nationaux : le Cenelec est l'homologue européen de la IEC, tandis que le Comité européen de normalisation est l'équivalent européen de l'ISO.

Une spécification publiée par Bosch a été la première à décrire le protocole CAN.

II-1-4-1- Les normes ISO

L'Organisation internationale de normalisation (ISO) a établi la norme ISO 11898 sur le protocole CAN dans les automobiles en 1993, qui a remplacé toutes les normes précédentes, y compris la définition Bosch. La norme ISO 11898 est divisée en plusieurs sections, comme suit :

- ISO 11898-1:2003
- la norme ISO 11898-2:2003
- L'ISO 11898-3
- ISO 11898-4:2004
- ISO 11898-5:2007
- ISO 11898-6:2013 [7]

II-1-4-2- Les normes IEC

L'organisation IEC a également développé des normes sur le CAN (CEI 61375-3-3 et CEI 61800-7-201/301)

II-1-4-3- Divers organismes

Parmi les autres organismes qui publient des normes CAN figurent :

- les deux organismes européens , qui collaborent aujourd'hui étroitement avec la CEI et l' ISO pour éviter la duplication des norms.

- le Airline Electronic Engineering Committee, qui a initié le développement de l' ARINC (Aeronautical Radio, Incorporated) spécification 825, qui est une normalisation générale pour l' utilisation du CAN dans les aéronefs [7].

II-1-5- Situation par rapport au modèle OSI

Étant donné que le bus CAN est un protocole réseau , il doit respecter le modèle OSI, qui divise les fonctionnalités requises pour la communication et l' organisation du réseau en sept couches. Les couches liées au CAN du modèle OSI sont :

II-1-5-1- Couche applicative

Ce canapé est vide, mais il sera rempli par l' utilisateur en fonction de l' application qui sera concernée par le Bus CAN.

II-1-5-2- Couche liaison de données

Cette couche fournira les ressources nécessaires à l' établissement, à la libération et à la maintenance des connexions entre les différentes entités du bus .Elle est également chargée de corriger les erreurs de premier niveau [7].

II-1-5-3- Couche physique

La couche physique déterminera comment le signal est transmis .Elle assure le transfert physique des informations entre chaque nœud, en fonction des propriétés du système (électrique, électronique, etc). Il convient de noter que la couche réseau de chaque nud doit être la même.

II-2- Analyse technique

II-2-1- Caractéristiques électriques

Nous commencerons par les caractéristiques électriques avant de passer à l' aspect logiciel du Bus CAN.

II-2-1-1- Support de transmission

Le câblage se présente par une paire filaire de type différentielle:

-CAN H (CAN High). -CAN L (CAN Low).

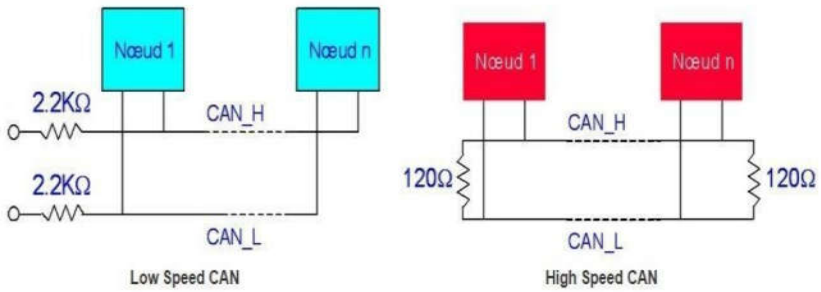


Figure II-2- Schéma de câblage Bus CAN.

La possibilité de transmettre par paires différentes permet d' éliminer les parasites pouvant être présents sur les lignes de communication.

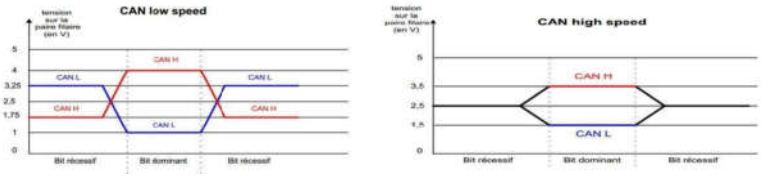


Figure II-3- Niveaux de tension en Low Speed & High speed

Il y a une nécessité d'utiliser des résistances de terminaison aux extrémités du bus afin de minimiser les réflexions sur le câble en adaptant l'impédance caractéristique de la ligne [7].

II-2-1-2- Effet des longueurs de câble

La longueur du bus est déterminée par les facteurs suivants**:**

- Le temps de propagation des lignes physiques du bus.

- La différence quantique résultant du temps de propagation défini précédemment. Cela est dû aux variations de la cadence des oscillations des nœuds.

- L' amplitude du signal varie en fonction de la résistance du câble et de l' impédance d' entrée des nœuds.

Il est nécessaire d' utiliser un optocoupleur pour faire fonctionner le bus CAN avec une longueur de câble supérieure à 200 mètres. Si la distance est supérieure à un kilomètre, des systèmes d' interconnexion doivent être utilisés.Il est possible d' utiliser un répéteur pour régénérer un signal ou des ponts pour connecter des réseaux locaux du même type. Les modules connectés au bus CAN doivent pouvoir supporter un débit de données d' au moins 20 kilobits par seconde [7].

II-2-1-3- Le Non-Return to Zero

C'est le code NRZ qui est utilisé dans le cas du Bus CAN (Non-Retour à Zéro). En effet, que le mors soit récessif ou dominant, le niveau reste constant tout au long.

Ce type de codage présente un inconvénient important. Il est possible que des problèmes de synchronisation surviennent lors de la transmission d' une longue série de bits au même niveau. Pour ce type de séquence, le codage Manchester, qui entraîne un décalage de niveau pour chaque bit transmis, est préféré.Dans le cas du Bus CAN, la procédure de "bit stuffing" est utilisée [7].

II-2-1-4- Le « bit-stuffing »

Un code NRZ a été appliqué à un trame.Alternativement, une trame peut avoir un grand nombre de bits de même niveau, faisant croire à différents nœuds qu'il y a une erreur de réseau. De ce fait, lors de la transmission, un bit opposé est ajouté à une suite de 5 bits de même valeur afin de « casser » le rythme et confirmer que tout fonctionne correctement. Les morceaux de « remplissage » ou de « bourrage », comme on les appelle en anglais, sont appelés « stuff ». Logiquement, cette technique allonge quelque peu le temps de transmission d'un message, mais elle en assure le bon transport de son contenu.

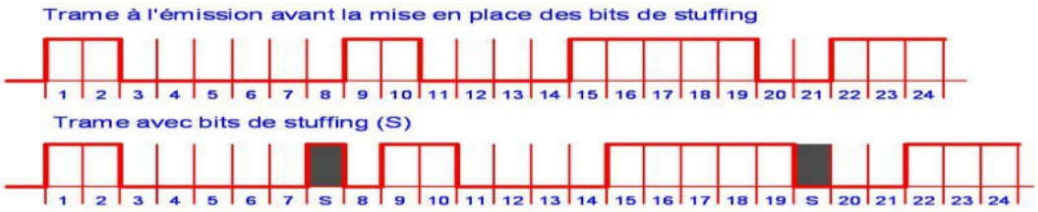


Figure II-4- Démonstration du "bit stuffing".

Les récepteurs CAN doivent comprendre la technique du bourrage pour que cette méthode fonctionne. Lorsque la trame est reçue, le récepteur effectuera le contraire du "bourrage" en enlevant les pièces de remplissage et en réassemblant le message d' origine. Cette technique de "surcodage" est totalement transparente pour l' utilisateur, mais les éventuelles erreurs seront retournées si elles se produisent.Il est également intéressant de noter que l' insertion de ces bits de bourrage permet un plus grand nombre de transitions, ce qui améliore la synchronisation des communications malgré le codage du tram NRZ [7].

II-2-2- Trames CAN

Dans le bus CAN, il existe quatre types de trame différents et un intervalle de temps:

- La trame de données (Data Frame) est créée par un noeud appelé "producteur" qui veut transférer des données, ou en réponse à la demande d' un autre noeud.

- La trace de la requête (Remote Frame) est générée par un noeud appelé " consumer" ou "data requestor ".

- La trame d' erreur (Error Frame) est créée par n'importe quelle entité du bus lorsqu'une erreur est détectée.

-La trame de surcharge (Overload Frame) permet de demander un délai supplémentaire entre Data Frames ou Remote Frames lorsqu'elles sont séquentielles [13].

II-2-2-1- Décomposition d'une trame

II-2-2-1-1- Trames de données

Un trame de données peut être décomposé en sept champs différents:

- le début de trame SOF (Start Of Frame), 1 bit dominant. - le champ d'arbitrage, 12 bits.

-le champ de données, 0 à 64 bits. - le champ d'acquittement, 2 bits.

- le champ de CRC (Cyclic Redundancy Code), 16 bits. - le champ de contrôle, 6 bits.

- le champ de fin de trame EOF (End Of Frame), 7 bits récessifs.

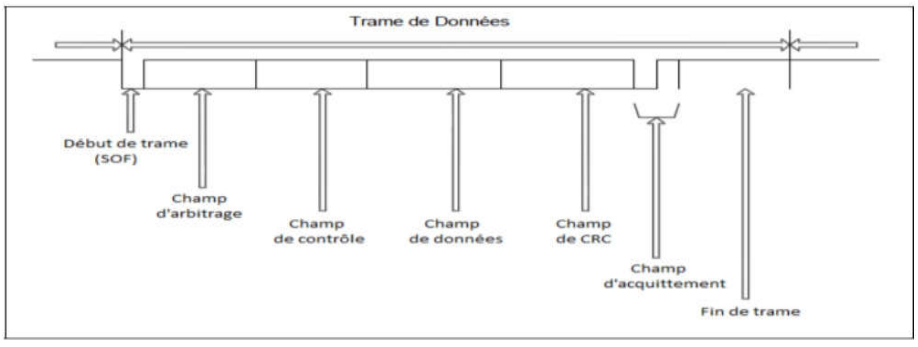


Figure II-5- Décomposition d'une trame de données.

II-2-2-1-2- Trames de requête

Une trame de requête est constituée de la même manière qu'une trame de données sauf que le champ de données est vide [13].

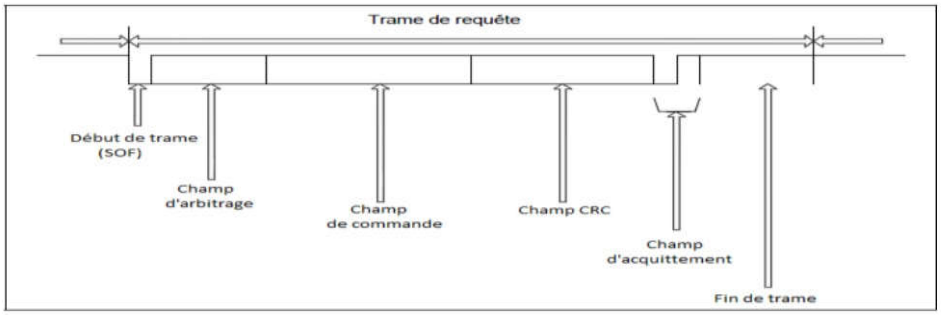


Figure II-6- Décomposition d'une trame de requête

II-2-2-1-3- Trames d'erreur

Une trame d'erreur permet d' alerter les autres nœuds CAN de la présence d' une erreur (Passive ou Active).

La trame d'erreur est composée de deux sections principales : le drapeau d'erreur et le délimiteur de champ [13].

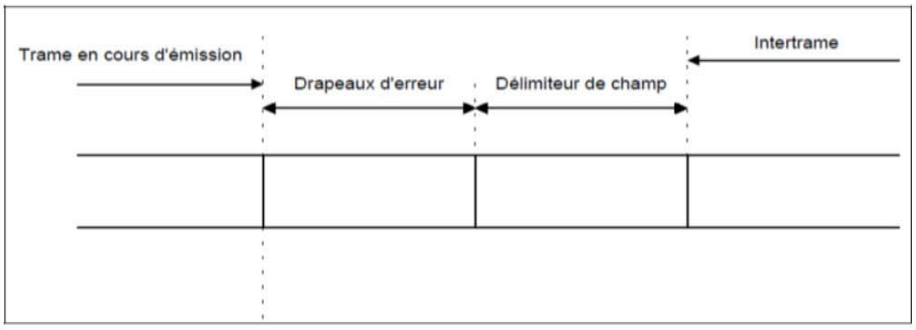


Figure II-7- Constitution de la trame d'erreur.

II-2-2-1-4- Trames de surcharge

Le but de cette trame est d'indiquer qu'une entité est surchargée pendant un certain laps de temps. Elle se déclenche sous deux conditions :

- Lorsqu'un nœud demande un délai spécifié avant d' accepter le prochain ensemble de données ou de demandes.

- Pendant l' inter trame (3 bits récessifs entre les trams); un nœud détecte un bit dominant.

Seuls deux trams de surtaxe peuvent être générés dans un ordre séquentiel afin d' éviter d' emmêler le bus indéfiniment. Elle est divisée en deux sections : la section des drapeaux de surcharge et la section des délimiteurs de surcharge [13].

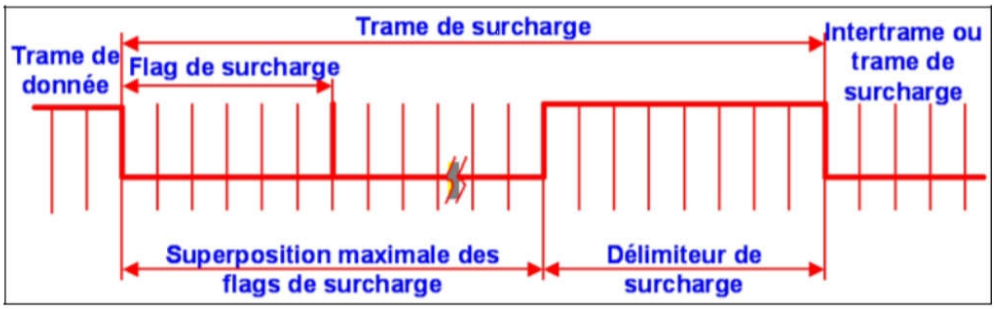


Figure II-8- Décomposition de la trame de surcharge.

Le champ des indicateurs de surcharge est composé de 6 bits dominants à la suite. Il efface la zone d' intervalle de l'inter trame. Délimiteur de surcharge : 8 bits consécutifs.

Suite à la transmission d' un drapeau de surcharge, l' entité examine le bus jusqu'à ce qu'elle détecte une transition indiquant le passage d' un bit dominant à un bit récessif.

À ce stade, chaque entité sur le bus a fini d' envoyer son indicateur de surcharge, et elles envoient maintenant 7 bits, récessifs consécutifs pour former le délimiteur de surcharge [13].

II-2-2-2- Analyse des différents champs

Chaque trame commence par un bit SOF (Start of Frame) qui indique le début d' une transaction. Cette transaction ne peut commencer que si le bus a été préalablement arrêté. Le bit SOF désigne le début d' une trame (Data frame ou Remote frame). C'est seulement une partie simple qui doit être au centre. Sur le bit de SOF, tous les nœuds du réseau doivent se resynchroniser. Il exister 6 champs comme suit :

- Le champ d'arbitrage (Arbitration field). - Le champ de contrôle (Control field).
- Le champ de données (Data field). - Le champ de CRC (Cyclic Redundancy Code field).
-Le champ d'acquittement (ACK field). - Le champ de fin de trame (End of frame field).

II-2-2-3- Période d'inter trame

Les trames de données et de requête sont séparées des autres trames par une inter trame, mais les trames d' erreur et de surcharge ne sont pas séparées par cette inter trame.Celui - ci est composé de deux ou trois champs selon la situation : un champ d' intermission, un champ sans bus et un champ de transmission de suspension pour les unités qui ont été émises à partir du message précédent.

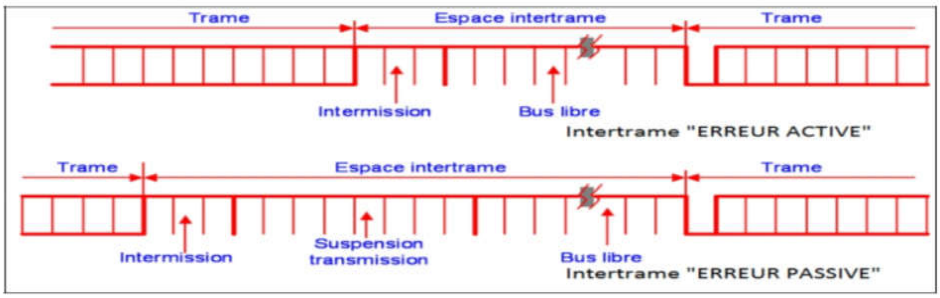


Figure II-9- Composition période d'inter trame.

Trois bits récessifs constituent le champ d' entracte. Pendant l' entracte, aucun trame de données ou de requête n'est autorisé à circuler; seul le trame de surcharge est autorisé à circuler. La durée pendant laquelle le bus est libre peut être définie à volonté .Pendant cette période, la ligne est "ouverte", ce qui signifie que n'importe qui peut monter à bord du bus.Si un message était en attente d' être envoyé lors de la transmission précédente, il sera envoyé dès le premier bit après la pause. Ce bit sera traduit par le SOF (Start Of Frame) de la nouvelle transmission [13].

Après l' envoi d' un message par une entité avec le statut "Erreur passive", elle envoie le champ "Suspension Transmission", qui est composé de 8 bits qui sont récessifs, avant de commencer une nouvelle transmission ou de reconnaître que le bus est "libre". "Si une transmission (par une autre entité) débute dans cet intervalle de temps, la première entité deviendra le destinataire du message [13].

II-2-3- Gestion des priorités

Une trame donnée est toujours prioritaire sur une trame de requête. Lorsqu'un noeud veut un nombre spécifique d' octets, il envoie une requête avec le nombre d' octets dont il a besoin.Un bit appelé RTR (Remote Transmission Request) peut être utilisé pour ouvrir la priorité qui sera donnée aux trams. En effet, dans le cas d' une trame donnée, le bit RTR sera dominant, mais dans le cas d' une trame de requête, il sera récessif. Du fait de la vérification du bit RTR, une trame donnée est toujours prioritaire. En comparant deux trames avec les mêmes identifiants, il est facile de voir que la trame donnée prime sur la trame de requête car le bit RTR est dominant.

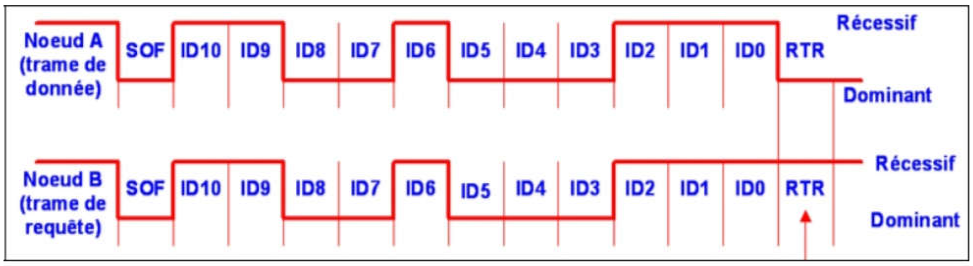


Figure II-10- Comparaison de deux niveaux de priorité.

II-2-4- Gestion des erreurs

Pour mieux comprendre comment les erreurs sont gérées, nous allons passer en revue toutes les erreurs possibles, expliquer comment le Bus CAN les gère, puis passer en revue comment le Bus CAN les gère.

Les erreurs de transmission peuvent causer des problèmes avec le fonctionnement de divers noeuds sur le bus CAN. Un nud pourrait être la source d' une communication BUS saccadée ou impossible. Il existe des méthodes pour détecter les erreurs de transmission dans le protocole CAN dans certaines situations [13].

II-2-4-1- Les différents types d'erreurs

-Le Bit Error

-L'erreur de Stuffing

-L'erreur de Cyclic Redundancy Code (CRC)

- L'erreur d'Acknowledge Delimiter

-L'erreur de Slot Acknowledge (Acknowledgment Error)

II-2-4-2- Les modes d'erreur

S'il y a des erreurs de transmission sur une trame, le nœud voyeur renvoie la trame jusqu'à ce que la transmission se termine sans erreur. Lorsqu'il y a des erreurs dans la transmission des trams, la valeur de ces compteurs augmente. Si les trams sont correctement transmis, les compteurs augmentent au fur et à mesure de la transmission. Si la valeur de ces comptes est trop élevée, le noeud incriminé se déconnectera du bus (Mode Bus Off) et ne pourra ni envoyer ni recevoir de trames.

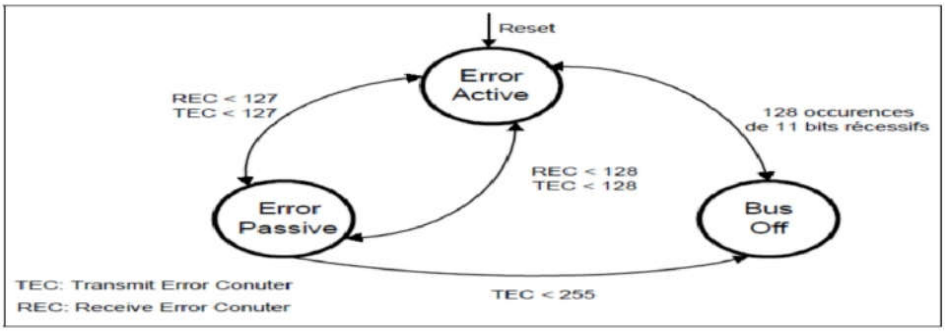


Figure II-11- Compteur d'erreur et état d'un nœud.

II-3- caractéristiques du bus CAN

II-3-1- Modes de fonctionnement

II-3-1-1- Le mode sommeil

CAN peut être mis en " mode veille " pour réduire la consommation d' énergie; Le nœud CAN en question est inactif et les pilotes ne sont pas connectés au bus .

II-3-1-2- Le mode de réveil

Après avoir utilisé le " mode veille ", le bus peut utiliser l' option "réveil" pour se réveiller. Cela se produit lors d' une reprise d' activité du bus ou à la suite d' une décision interne prise au niveau de l' élément CAN. On observe une période d' attente due à la resynchronisation de l' oscillateur local. Autrement dit, il vérifie la présence de 11 bits consécutifs sur le bus. Les pilotes se connectent alors au bus, mais du fait de la resynchronisation, les premiers messages sont perdus.Ils ont été redirigés afin

qu'aucune donnée ne soit perdue. Pour obtenir de bonnes performances de débit du réseau , des oscillateurs à quartz sont utilisés.

II-3-2- Domaines d'application

Développé à l' origine par Bosch pour l' industrie automobile, il s'est depuis démocratisé au cours des 20 dernières années; Son application va des réseaux à haut débit aux réseaux de multiplexage à faible coût. Le protocole CAN a de nombreuses applications dans une variété de domaines.

Le système de navigation assure la sécurité des machines-outils et des machines à commande numérique. Photocopieurs, jeux tactiles et encodeurs sont tous intégrés directement dans un même nombre de capteurs [16].

-Applications pour automobiles

Pour les voitures il existe généralement deux réseaux CAN

1er réseaux : injection électronique, alimage électronique, boîte à vitesse Pour les voitures automatiques, système de freinage ABS (Anti Blocage Système).

2eme réseaux: chauffage, climatisation, vitre et rétroviseur électrique, commande centralisée.

La différence entre deux applications de protocole CAN dans une voiture réside dans l' ordre de priorité; par exemple: si l' ABS est le premier domaine d' application, il a une priorité beaucoup plus élevée que la commande de rétraction électrique.

- applications médicales

En plus des applications automobiles et médicales , on peut trouver le système PMS, une nouvelle technologie Philips Medical System qui utilise le protocole CAN comme moyen de communication entre les unités de passage et les unités de production de rayons X, dans la foutaie de dontiste et, bien sûr,irm (résonance magnétique réceptrice d'image).

II-3-3- Intérêts du Bus CAN

L' objectif principal du Bus CAN est de réduire les coûts .Ces passions peuvent être divisées en trois catégories :

Réduction des coûts initiaux:

- Plutôt que d' utiliser un seul câble pour chaque équipement, un seul câble est nécessaire pour tous.

- Dans certains cas, le câblage analogique peut être réutilisé.

- Le temps d'installation est réduit.

- Une réduction de la quantité d' équipement nécessaire à l' installation du système .

Réduction des coûts de maintenance:

- Maintenance simplifiée : temps de dépannage réduit, emplacement de panoramique potentiel avec diagnostic à distance.

- Moins d'entretien requis en raison de la fiabilité croissante du système.

- La possibilité d' étendre à un nouveau bus de terrain avec flexibilité.

- Simplification du raccordement.

Performances globales accrues :

- Précision accrue : Il n'y a pas d' erreurs de distorsion ni de réflexions dues à la précision du signal, ce qui n'est pas le cas des signaux analogiques.

- La communication entre deux équipements est possible sans passer par le système de surveillance.

- Les données et les mesures sont souvent disponibles pour tous les types d'équipements de terrain [8].

Dans l' exemple suivant, nous pouvons voir comment le nombre de connexions est réduit pour permettre à plusieurs unités dans une voiture de communiquer entre elles. A l' aide d'un bus unique , les unités communiquent entre elles.

II-3-4- Les Aventages et les inconvénients

Voici les avantages du bus CAN par rapport aux autres types de bus :

-Il prend en charge un taux de transfert de données de 1 Mbps.La version CAN FD (débit flexible de données) utilise plus que cette vitesse, c'est-à -dire qu'elle utilise 2 + Mbps; Le CAN FD traitera huit fois plus de trafic que le bus CAN normal.

-Il est utilisé dans une variété d' applications automobiles pour réduire la quantité de frottement.Il est largement utilisé dans une variété d' industries en raison de son interface moins compliquée.

-Il vous permet de gagner du temps et de l'argent en utilisant un système de câblage moins nombreux et plus facile à utiliser, ainsi qu'une programmation flash.

-Dans la partie données, le protocole standard CAN utilise 8 octets, tandis que le protocole CAN FD utilise 64 octets.

-Le protocole gère une variété de capacités de détection d' erreur, y compris l'erreur de bit, l'erreur de réception, l'erreur de formulaire, l'erreur CRC et l' erreur de bourrage.

-Il gère la retransmission automatique des messages perdus et fonctionne parfaitement dans une variété d' environnements électriques.

Voici les inconvénients du bus CAN par rapport aux autres types de bus :

-Cependant, le nombre maximum de nœuds pour le réseau n'est pas spécifié. En raison de la charge électrique , il peut contenir jusqu'à 64 nuds.

-Pour réduire les problèmes d' intégrité du signal tels que les réflexions, le bus CAN doit être correctement terminé aux deux extrémités avec des résistances.

-Le retrait du nœud nécessite l' utilisation de résistances de terminaison de 120 ohms aux endroits appropriés sur le bus CAN.

-Cela entraîne une augmentation des coûts de développement et de maintenance des logiciels.

-Le réseau doit être configuré de manière à réduire au minimum les stubs.

-Il est possible qu'il y ait des interactions défavorables entre les nœuds.

-Le pilote CAN doit générer au moins 1,5 V dans une charge typique de 60 ohms.

-Il a une longueur maximale de 40 mètres qu'il peut gérer.

II- Conclusion

Les perspectives du bus Le CAN est étroitement lié à ses applications automobiles car c'est ce secteur qui lui assure une grande partie de sa crédibilité technologique ainsi qu'un marché à grande échelle qui permet de baisser les coûts .Alternativement, dans les années à venir, des méthodes pour aider les véhicules à devenir plus autonomes seront développées.La conduite autonome nécessite en effet.

Cette analyse explique pourquoi le Bus CAN est obligatoire dans l' industrie automobile et pourquoi il devient de plus en plus populaire dans l' industrie aéronautique : son faible coût de mise en œuvre dû à l' utilisation d' un seul fil, sa robustesse de fonctionnement, et sa gestion des prioritésen font un réseau de choix dans les environnements automobiles et aéronautiques les plus exigeants.

Ces évolutions ouvrent de nombreuses opportunités de développement d'applications du bus CAN qui continuera à jouer un rôle majeur dans les prochaines années aussi bien dans les véhicules que dans les machines, les procédés et dans diverses applications industrielle.

Chapitre III

Mise en oeuvre du bus CAN à base d'Arduino

III- Introduction

Lorsque deux périphériques neoud1 et neuod2, partagent le même bus, il est important de préciser comment se rendre à ce bus. Le protocole CAN spécifié la connexion précise et détaillée de plusieurs dispositifs à un bus, qui est un bus largement utilisé dans l'industrie. Leprotocole spécifie les conditions d'accès au bus et d'obtention des données des capteurs via neoud2, ainsi que l'envoi des valeurs à afficher sur un LCD.

III-1- Présentation des composants utilisé

III-1-1- Carte Arduino uno

La carte Arduino Uno est une carte électronique équipée d'un microcontrôleur construite autour de l'ATmega 328, le microcontrôleur permet à partir d'événements détectés par des capteurs, de programmeur et commandant des actionneurs.

La carte Uno possède 14 broches d'entrée/sortie numériques (dont 6 peuvent servir de sorties PWM), 6 entrées analogiques Le convertisseur analogique numérique possède 10 bits et sa tension de pleine échelle est par défaut de 5 V mais peut être réglée entre 2 et 5V, des entrées et des sorties numériques un oscillateur à quartz de 16 MHz, un port USB il suffit de la relier à un ordinateur, un jack d'alimentation, une embase ICSP, et un bouton d'initialisation (reset).

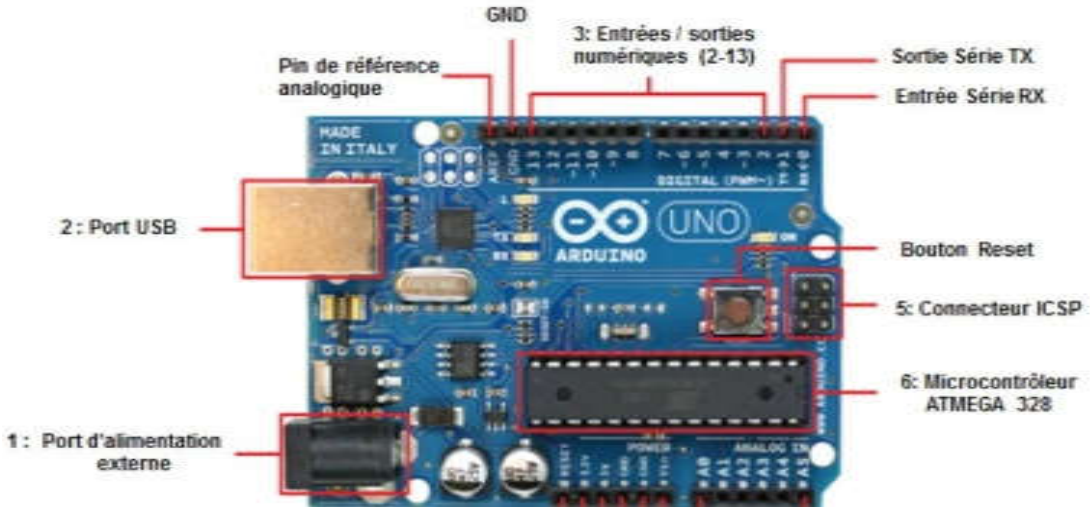


Figure III-1- Broches de carte Arduino Uno

La carte Arduino Uno est donc clair et simple de programmation, il contiant nombreuses bibliothèques disponibles avec diverses fonctions implémentées avec simplifie grandement les schémas électroniques et la charge de travail à la conception d'une carte électronique tout cela diminué le coût de la réalisation.

III-1-2- Carte Bus CAN MCP2515

Le CAN est un protocole de communication série qui supporte efficacement le contrôle en temps réel de systèmes distribués tels qu'on peut en trouver dans les automobiles, et ceci avec un très haut niveau d'intégrité au niveau des données.

Le controleur de bus CAN MCP2515 est un module de base qui gère le protocol CAN 2.0B et permet une connexion à 1Mbps. Vous aurez besoin de deux modules de bus CAN pour mettre en place un système de communication entièrement fonctionnel.

Ce module spécifique est basé sur le controleur CAN MCP2515 et le CI émetteur-récepteur CAN TJA1050. Le MCP2515 IC est un controleur CAN autonome avec un port SPI intégré pour communication du microcontroleur. En ce qui concerne le CI TJA1050, il sert de lien entre le controleur CAN MCP2515 et le bus CAN physique.

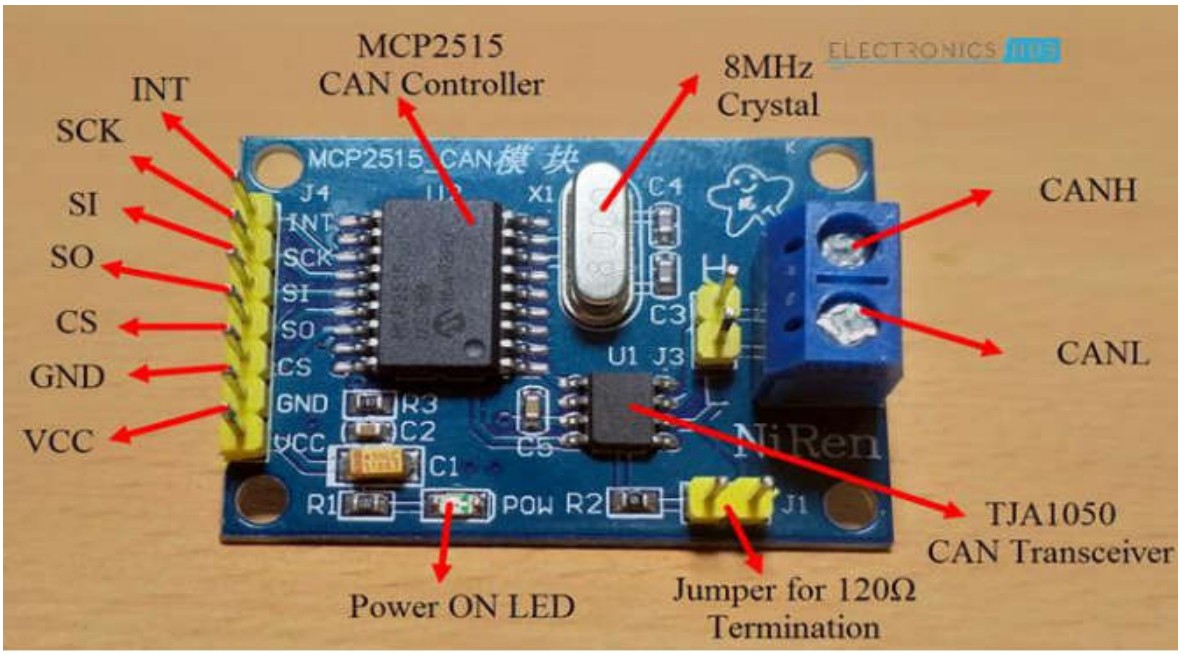


Figure III-2- Broches du module MCP2515

III-1-3- Capteur de distance Ultrasons

Un capteur à Ultrasons enregistre de courtes impulsions sonore à haute fréquence à intervalles réguliers. Ces impulsion se propagent dans l'air à la vitesse du son. Lorsqu'ils rencontrent un objet, ils s'arretent et reviennent sous la forme d'un écho au capteur.
A partir du temps écoulé entre l'émission du signal et la réception de l'écho, il calcule la distance entre la cible et l'émetteur.

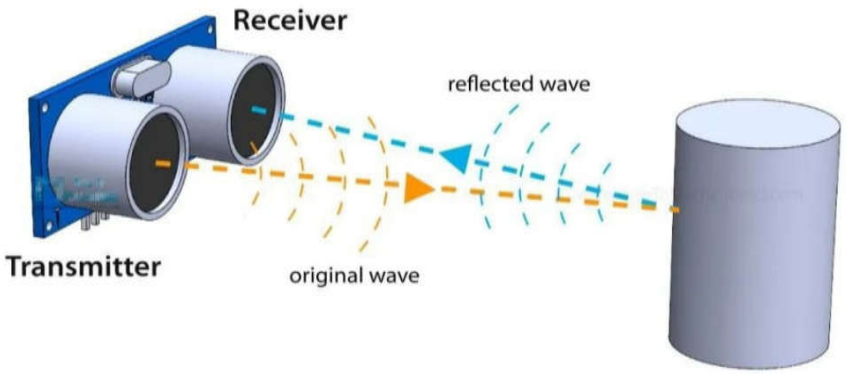


Figure III-3- Foncionnement du capteur Ultrasons

Pratiquement, tous les matériaux qui réfléchit le son peut etre détecté, quelle que soit sa couleur, ne posent aucun problème pour un capteur à ultrasons.

III-1-4- Capteur de température DH11

Ce capteur utilise une thermistance avec un coefficient de température négatif pour mesurer la température, ce qui entraine une diminution de la résistance à mesure que la température augmente. Ce capteur est généralement fait de céramique ou de polymètres semi-conducteurs pour obtenir une valeur de résistance plus élevée meme pour de petits changements de température.
Chaque seconde Le DH11 donne une lecture avec une fréquence de 1Hz et la plage de température entre 0 et 50 degrés celsuis avec une tension de fonctionnement de 3 à 5 volts. Le DH11 a trois broches GND, VCC et une broches de données avec une résistance pull-up de 5k à 10k.

Figure III-4- Broches d'un capteur de température DH11

III-1-5- Encoder (Driver moteur L298N)

Le controleur de moteur L298N est basé sur l'architecture de pont H est vraiment simple, en activant deux transistors à la suite (détectant des directions opposées), on peut controler le sens de circulation du courant dans charge connectée (au milieu du H), la direction du flux de courant dans la charge qui est connectée, peut etre controlée. Et c'est ce changement de sens actuel qui permet aux moteur en marche contine de tourner vers la droite ou vers la gauche.

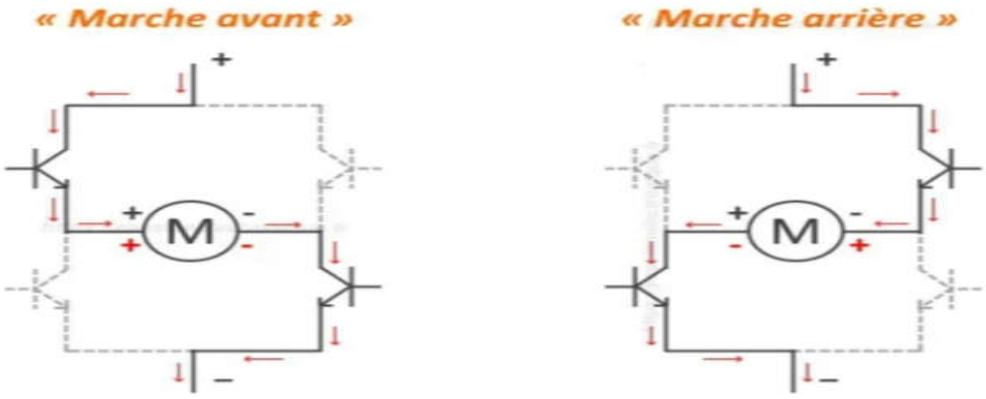


Figure III-5- Les configurations de pont en H

Les interrupteurs IN1 et IN2 sont utilisés pour controler la direction du moteur1. Lorsque IN1 est HIGH et IN2 est LOW, le moteur1 tournera dans une direction spécifique. Faites IN1 LOW et IN2 HIGH pour changer la direction.

Les interrupteurs IN3 et IN4 sont utilisés pour controler la direction du moteur2. Lorsque IN3 est HIGH et IN4 est LOW, le moteur2 tournera dans une direction spécifique. Faites IN3 LOW et IN4 HIGH pour changer la direction.

Le moteur1 ou moteur2 s'arretera si les deux entrées sont HIGH ou LOW.

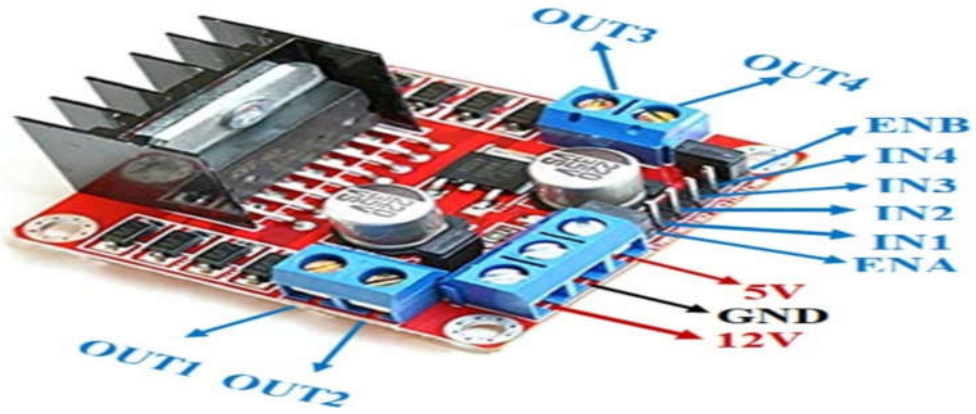


Figure III-6- Broches de L298N

III-1-6- LCD 4*20 +I2C

Il se distingue par l'utilisation du système I2C pour le fonctionnement et le controle via seulement deux fils, ce qui est un avantage signification pour les entrées du panneau de commande utilisé dans notre projet et simplifie la procédure de connexion et de fonctionnement.

Figure III-7- Ecran LCD avec I2C

III-2- La technique de contrôle de bus CAN MCP2515 avec Arduino uno

III-2-1- Présentation de logiciel Arduino

Le logiciel est un environnement de développement intégré (IDE) basé sur Java qui peut etre téléchargé gratuitement à partir du site Web Arduino, www.arduino.cc. Après avoir correctement installé le logiciel et ses pilot, nous pouvons programmer nos cartes Arduino.

La fonction setup() n'est exécutée qu'une seule fois, immédiatement après le lancement de programme; il contient souvent les instructions d'instalation de certaines ressources de la carte.

La fonction loop() est la boucle principale, et elle se répète indéfiniment tant que l'Arduino est maintenu en tension.

Le langage de programmation Arduino comprend des fonctions arithmétiques et mathématiques, des opération de comparaison et logique, des structures de contrôle et des fonctions de gestion du temps.

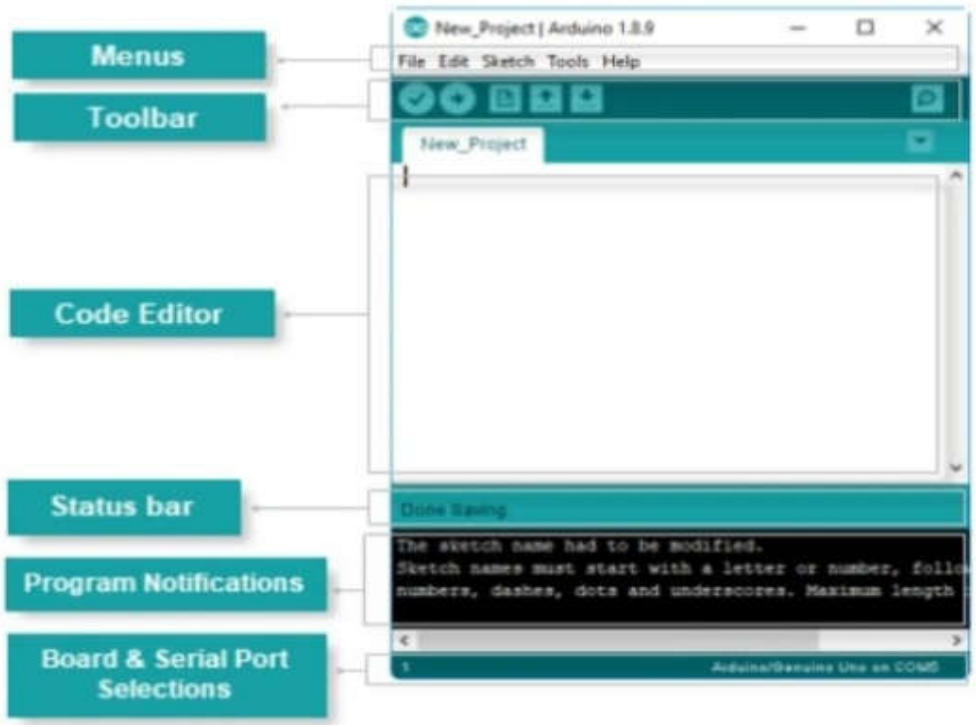


Figure III-8- Interface utilisateur du langage Arduino

III-2-2- Le choix de bus can mcp2515 en Arduino uno

Nous ajoutons d'abord la bibliothèque de bus CAN MCP2515 dans IDE Arduino pour pour commencer la programmation. Aprée taper "mcp2515" dans la recharche puis installer la bibliotheque autowp-mcp2515.

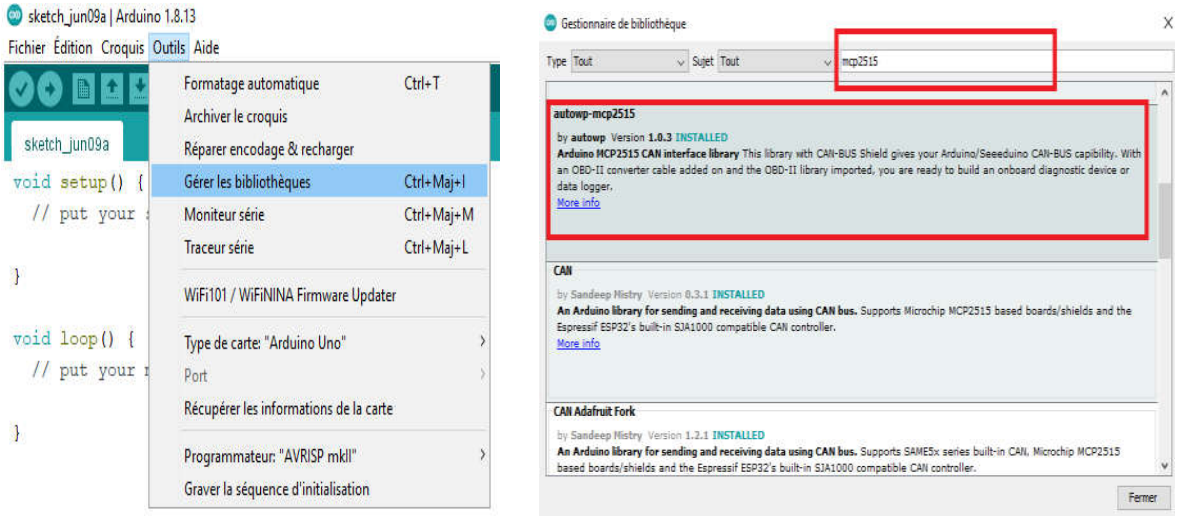


Figure III-9- Gestionnaire de bibiothèque de l'Arduino

III-2-3- Le câblage de bus CAN MCP2515 avec Arduino uno

L'image suivante montre le schéma de circuit du module CAN MCP2515 avec Arduino et la communication possible.

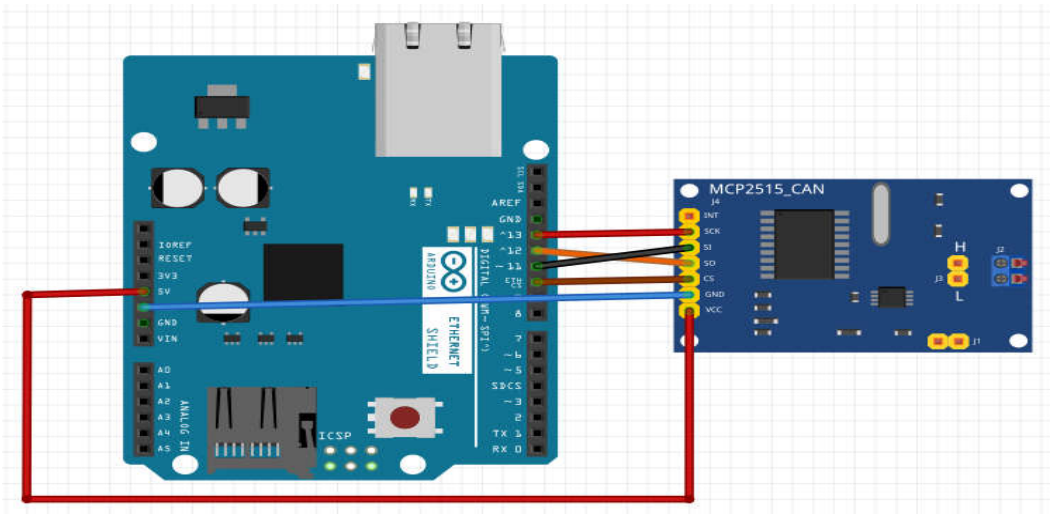


Figure III-10- Schéma du circuit du bus CAN MCP2515 et Arduino

III-2-4- Allumage une led par un bouton poussoir

Dans cette exemple on à vu commet controle une led avec un bouton poussoire, lorsque les deux pins de bouton connecté à Arduino1 (pin7, GND) puis envoie l'information vers Arduino2 (allumer la led) par deux CAN MCP2515.

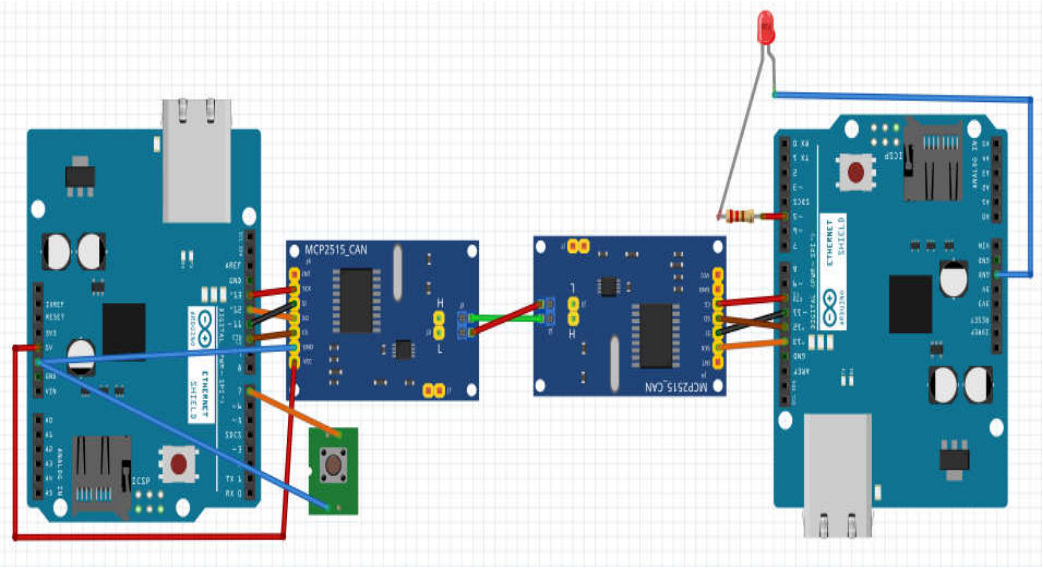


Figure III-11- Montage d'un bouton poussoir et led avec deux Arduino et deux CAN.

Le code source pour Arduino1 (émetteur) :

```
#include <SPI.h>
#include <mcp2515.h>      //la bibliotheque de mcp2515
MCP2515 mcp2515(10);      //Déclaration de Cs que brancher ver pin 10 de arduino
struct can_frame canMsg1; //La taille de message par octet

void setup()
{
  pinMode(7,INPUT);
  mcp2515.reset();
  mcp2515.setBitrate(CAN_1000KBPS,MCP_8MHZ);//Configuration de vitesse de message  et le quartez
  mcp2515.setNormalMode();
  canMsg1.can_id  = 0x011; //Adresse de message
  canMsg1.can_dlc = 1; //dlc de message
}
void loop()
{
int p = digitalRead(7);
canMsg1.data[0] = p;
mcp2515.sendMessage(&canMsg1);//La transmission de message
}
```

Le code source pour Arduino2 (récepteur) :

```
#include <SPI.h>
#include <mcp2515.h>      //la bibliotheque de mcp2515
MCP2515 mcp2515(10);      //Déclaration de Cs que brancher ver pin 10 de arduino
struct can_frame canMsg1; //La taille de message par octet
void setup()
{
  delay(1000);
  pinMode(5,OUTPUT);
  mcp2515.reset();
  mcp2515.setBitrate(CAN_1000KBPS,MCP_8MHZ);//Configuration de vitesse de message  et le quartez
  mcp2515.setNormalMode();
}
void loop()
{
     if (mcp2515.readMessage(&canMsg1) == MCP2515::ERROR_OK)//test pas error dans le mwssage
     {
        if (canMsg1.can_id==0x011) //test L'adresse de message
        {
         int z = canMsg1.data[0];
           if(z==1)
            {
             digitalWrite(5,HIGH);
            }
           else
            {
             digitalWrite(5,LOW);
            }
        }
     }
}
```

III-3- Présentation de la réalisation

Nous avons décidé d'effecteur notre réalisation sur le bus CAN avec une application qui se rapproche de celle de l'automobile.

III-3-1- Aspects technique

Le projet que nous avons développé est compose de deux noeuds. Le premier noeud se compose de different capteurs, il est realisé avec une carte Arduino. La deuxième noeud affiche les données des different capteurs dans un lcd.

en présenté par le schema suivant :

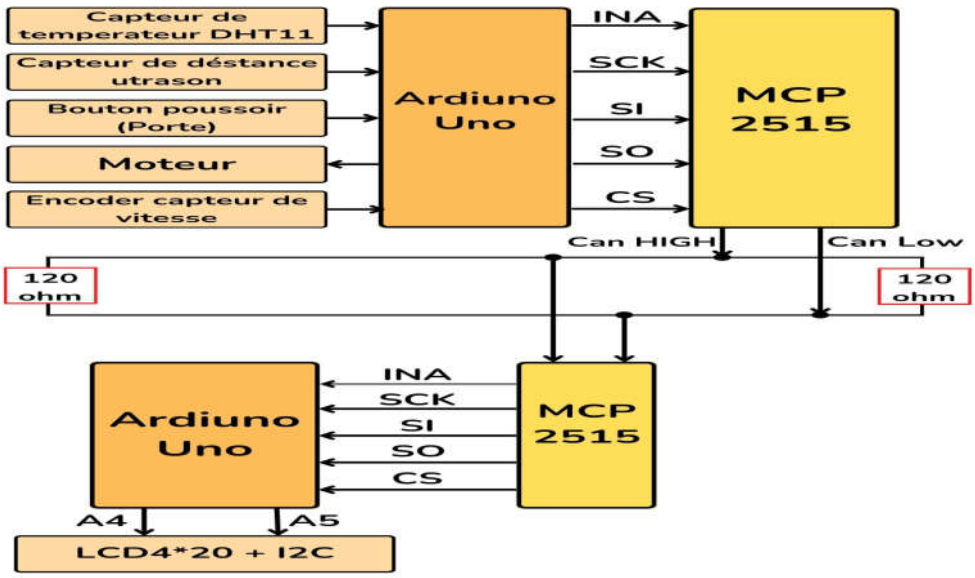


Figure III-12- Schématique de communication de la réalisation

III-3-2- Circuit de la réalisation

Pour commencer, et fin de prendre en main la communication CAN via Arduino UNO, les deux Arduino fonctionnnée émetteur et l'autre récepteur.

III-3-2-1 Circuit émetteur

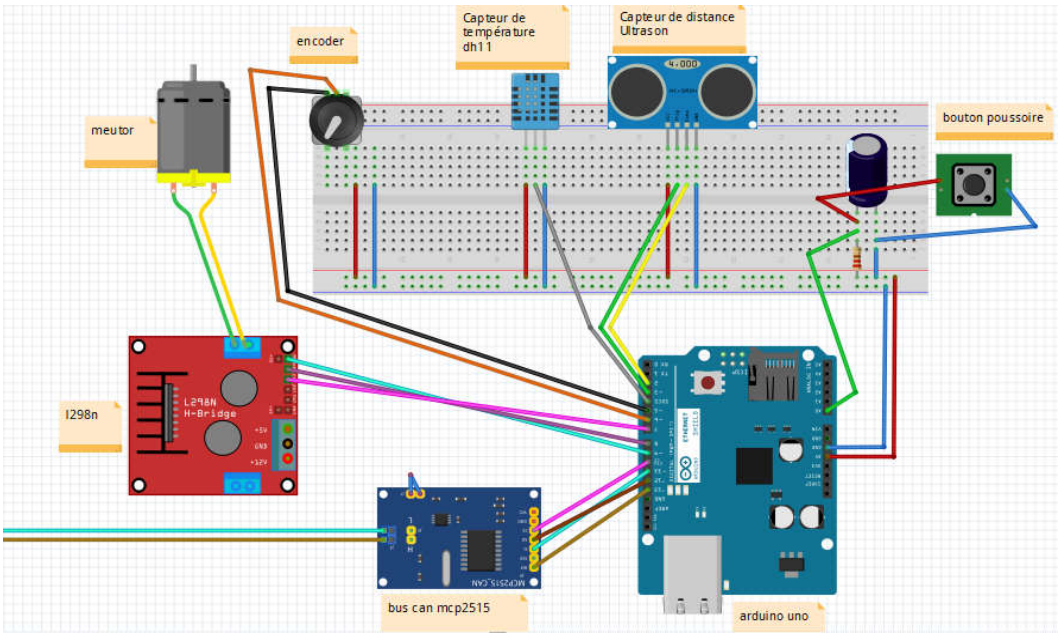


Figure III-13- Circuit de la réalisation émetteur

III-3-2-2- Circuit récepteur

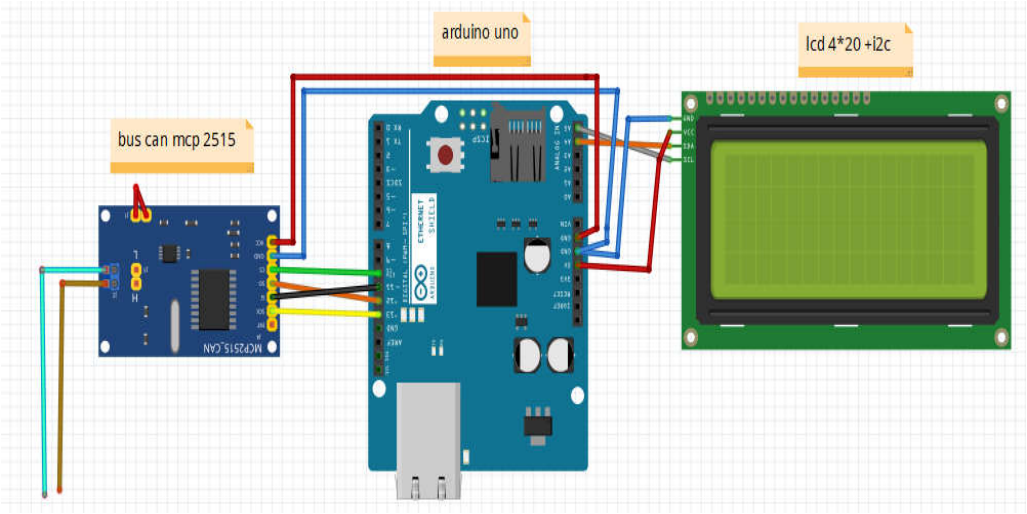


Figure III-14- Circuit de la réalisation récepteur

III-3-2-3- Montage globale

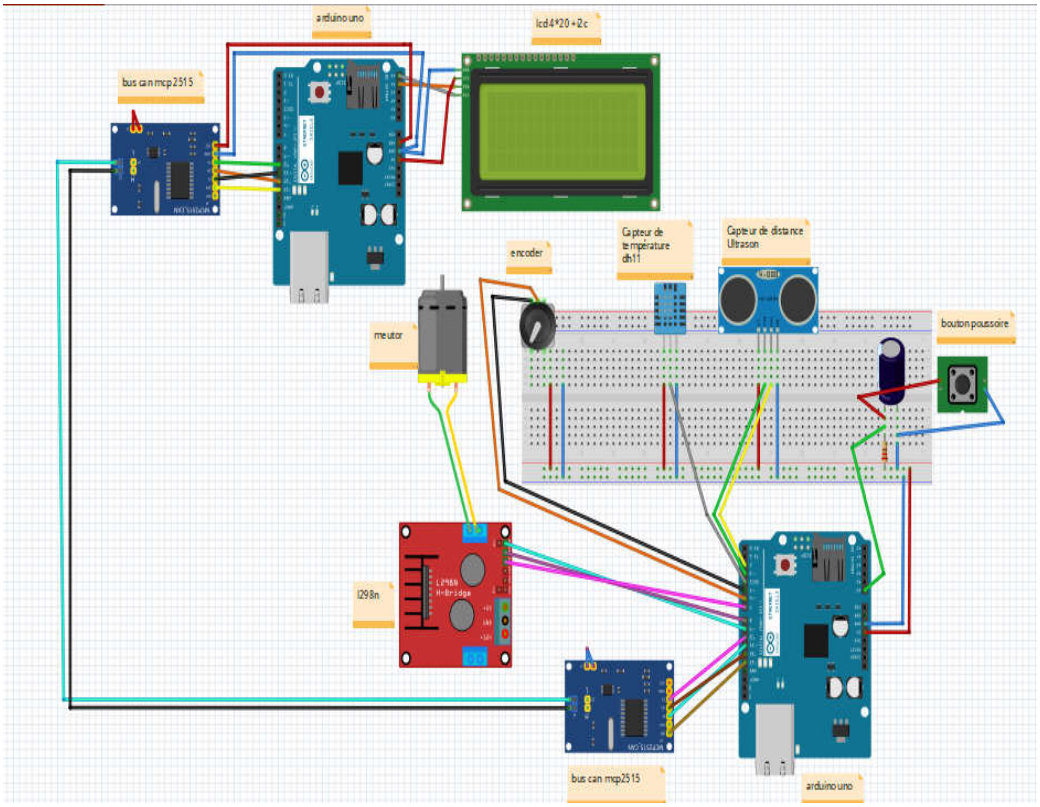
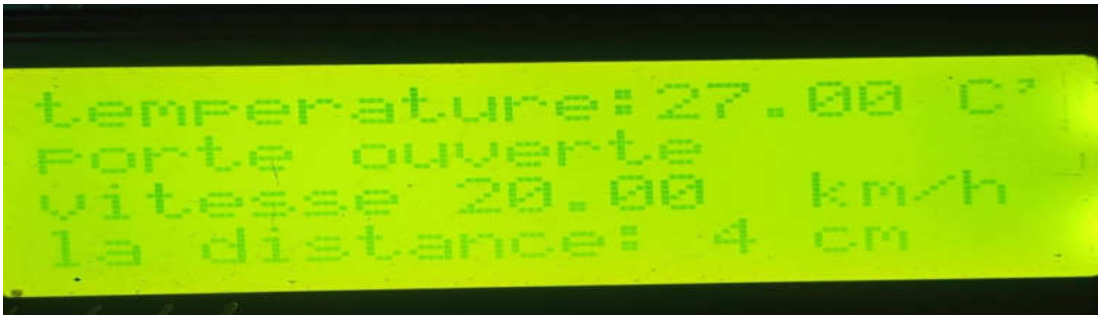
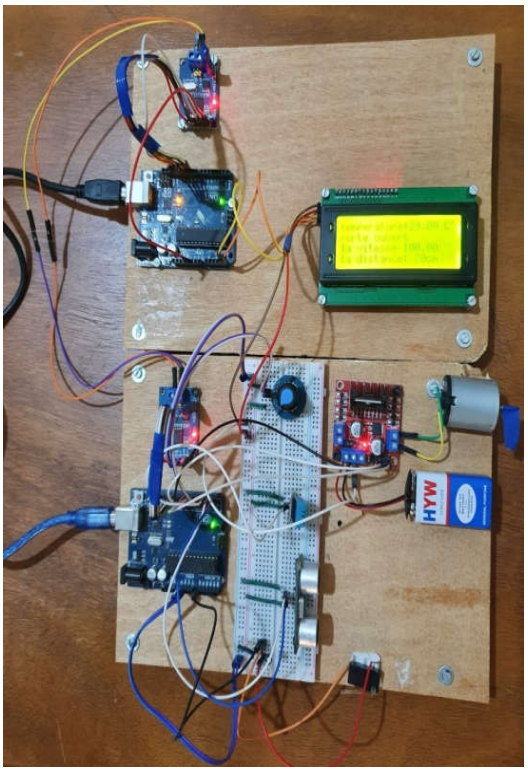
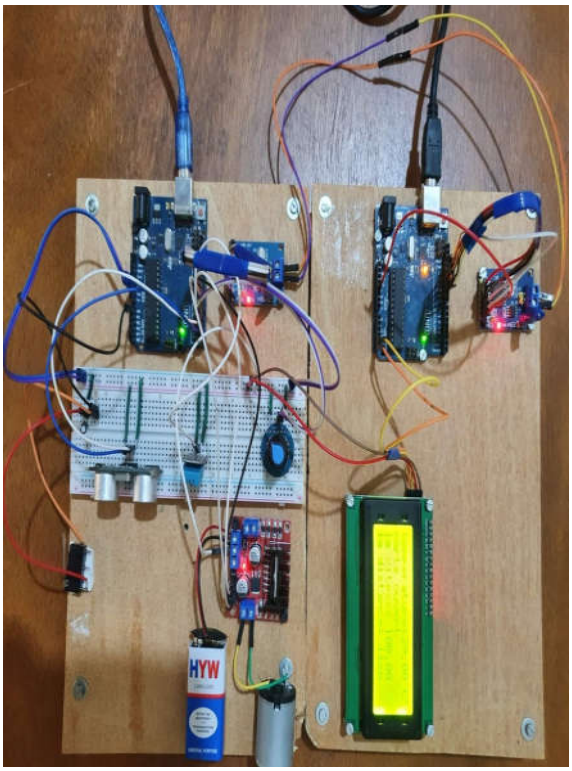


Figure III-15- Circuit de la réalisation (émetteur et récepteur)

Quelques photos prises du travail appliqué







III- Conclusion

Nous avons concevoir et implémenter un système de communication entre deux ARDUINO sur bus CAN. nous avons mesuré la température avec un DH11, la vitesse du véhicule avec un moteur L298n et la distance avec un ultrason branchés sur un nœud1, puis avons envoyé les donné à un écran LCD pour l'affichage sur un deuxième nœud2. Notre travail est basé essentiellement sur la réduction des nombres de câbles utilisé.

Le manque des cartes de simulation et de développement MCP sur proteus a rendu difficile le débogage des application. Malgré les défis, nous avons put mettre en place une communication CAN et tester sa fonctionnalité.

Conclusion générale

Conclusion générale

Dans ce travail nous avons réalisé un système représentant un tableau de bord d'une voiture. Nous nous sommes principalement intéressésà la mise en œuvre du bus CAN à base d'un Arduino. Pour ce faire, nous avons utilisé deux cartes Arduino Uno et deux modules interface Bus CAN MCP2515. La première carte Arduino permettait d'acquérir les données des différents capteurs (température, vitesse, …) de la voiture et de les envoyer sur le bus CAN. La deuxième carte Arduino, quant elle, permettait la gestion du tableau de bord de la voiture : affichage de la température, la vitesse, l'état des portes (ouvertes ou fermées), etc. De ce fait, un afficheur LCD I2C 4X20 caractères a été utilisé dans ce projet comme tableau de bord.

Toutefois, il est à noter que notre système ainsi conçu a été réalisé et testé pour les différent capteursutilisés dans ce projet à savoir : le capteur de température, de vitesse, de distance (dans le cas d'une marche arrière)et de capteur d'ouverture de porte. Le fonctionnement de notre système réalisé est tel que prévu dans le cahier les charges. Néanmoins, il nous reste à tester notre système sur une petite voiture ce qui est perspective pour notre présent travail.

Références

[1] Othman, H. F., et al. "Controller area networks: Evolution and applications." 2006 2nd International Conference on Information & Communication Technologies. Vol. 2. IEEE, 2006.

[2] Ahmed RACHID&Frédéric COLLET, “Bus CAN”, *Edition Technique de l’Ingénieur*, 2020.

[3] https://www.fichier-pdf.fr/2012/12/20/technique-de-diagnostic-dans-le-domaine-automobile.

[4] A. S. Berger. Embedded Systems Design: An Introduction to Processes, Tools, and Techniques. Éditions CMP Books, 2002, 237 pages

[5] J. Bortolazzi. T. Hirth. T. Raith. Specification and Design of Electronic Control Units. EURO-DAC '96/EURO-VHDL '96, Éditions IEEE Computer Society Press, Septembre 1996.

[6] Real-time systems. The Encyclopædia Britannica 15th edition, Éditeur Encyclopædia Britannica Inc, 1985.

[7] J. Stankovic. Misconceptions About Real-Time Computing: A Serious Problem for Next-Generation Systems, IEEE Computer, Vol. 21, No. 10, Octobre 1988.

[8] T. Noergaard. Embedded systems Architecture: A Comprehensive Guide ForEngineers And Programmers Embedded Technology. Éditions Newnes, Avril 2005, 638 pages.

[9] Mr DJELLAL DJAMAL Système embarqué pour l'automobile, Université Bachir El Ibrahimi Bordj Bou Arreridj, Algérie, 2019/2020.

[10] https://fr.jf-parede.pt/embedded-systems-role-automobiles#:

[11] Dominique Paret, Hassina Rebaine Réseaux de communication pour systèmes embarqués - 2e éd. – 2014 –

[12] Le Bus CAN – Patrice Kadionik – [En Ligne]

[13] Tavernier, Christian. Arduino Maîtrisez sa programmation et ses cartes d'interface (shields). Dunod, Paris, 2011.

[14] Seminaire Industriel CAN – CFAI Languedoc Roussillon – [En Ligne] : http://www.cfai languedocroussillon.com/telechargement/Seminaire_Bus_Industriel_CAN.pdf

[15] Benkhelifa, A. Les systèmes embarqués dans l'automobile (Doctoral dissertation, Haute école de gestion de Genève), 2018.

[16] Barr, M. Embedded Systems Glossary. Netrino Technical Library, www.netrino.com. Vu en janvier 2009.

Annexe

code arduino Partie émetteur :

```
////////////////// motor/////////////////

#define pinA 6

#define pinB 5

int in3 = 7;

int in4 = 8;

intConb = 9;

float compteur = 0 ;

float v;

booletatA ;

booldernierEtatA ;

long unsigned tempsA ;

////////////////mcp2515/////////////////

#include <SPI.h>

#include <mcp2515.h>

#define porte A0

#define trig 2

#define echo 3

structcan_frame canMsg1;
```

```
structcan_frame canMsg2;
structcan_frame canMsg3;
structcan_frame canMsg4;
MCP2515 mcp2515(10);
//////////////////////////////ultrasons////////////////////////////////
int distance=0,t=0;
//////////////////////////////dht11 ///////////////////////////////////
#include "DHT.h"
DHT dht(4,DHT11);
float temp;
floathumi;
void setup()
{
Serial.begin(9600);
   ////////////////////////////motor////////////////////////////
pinMode(8, OUTPUT);
pinMode(9, OUTPUT);
pinMode(10, OUTPUT);
pinMode(pinA,INPUT);
pinMode(pinB,INPUT);
```

```
dernierEtatA = digitalRead(pinA);
tempsA = millis();
  ////////////////////////////ultrasons////////////////////////////
Serial.begin(9600);
pinMode(trig,OUTPUT);
pinMode(echo,INPUT);
  ////////////////////
dht.begin();
pinMode(porte,INPUT);
SPI.begin();
  //////////////bus can mcp2515//////////////////////
mcp2515.reset();
mcp2515.setBitrate(CAN_1000KBPS,MCP_8MHZ);
mcp2515.setNormalMode();
  canMsg1.can_id  = 0x011;          //dht11
  canMsg1.can_dlc = 1;
  canMsg3.can_id  = 0x014;          //ultrasons
  canMsg3.can_dlc = 1;
  canMsg2.can_id  = 0x036;          //port
  canMsg2.can_dlc = 1;
```

```
  canMsg4.can_id  = 0x26;          //port
  canMsg4.can_dlc = 1;
}
void loop()
{
  ////////////////////////motor//////////////////////////////////
etatA = digitalRead(pinA);
if(etatA != dernierEtatA )
  {
if( abs(millis() - tempsA) > 50 )
    {
if(digitalRead(pinB) != dernierEtatA)
      {
compteur++;
if (compteur>25)
        {compteur=25;}
      }
else
      {
        compteur--;
```

```
if (compteur<0)
       {compteur=0;}
     }   }
dernierEtatA = etatA ;
v=compteur*10;
digitalWrite(in3,HIGH );
digitalWrite(in4, LOW);
analogWrite(Conb, v);
Serial.println(v);
}
floatvt = v; ;
canMsg4.data[0] = vt;
mcp2515.sendMessage(&canMsg4);
/////////////// port//////////
int p = digitalRead(porte);
canMsg1.data[0] = p;
mcp2515.sendMessage(&canMsg1);
   ////////////////////////dht/////////////////
float temp=dht.readTemperature();
canMsg3.data[0] = temp;
```

```
mcp2515.sendMessage(&canMsg3);
   ////////////////////ultrason//////////
digitalWrite(trig,LOW);
delayMicroseconds(5);
digitalWrite(trig,HIGH);
delayMicroseconds(10);
digitalWrite(trig,LOW);
   t=pulseIn(echo,HIGH);
distance=t/57;
canMsg2.data[0] = distance;
mcp2515.sendMessage(&canMsg2);
delay(5);
}
////////////////////code arduino Partie  récepteur :////////////////////////////
#include <SPI.h>
#include <mcp2515.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,20,4);
structcan_frame canMsg1;
```

```
structcan_frame canMsg2;
structcan_frame canMsg3;
structcan_frame canMsg4;
MCP2515 mcp2515(10);
long time1;
void setup()
{
lcd.init();
lcd.backlight();
delay(1000);
SPI.begin();
Serial.begin(9600);
  time1=millis();
mcp2515.reset();
mcp2515.setBitrate(CAN_1000KBPS,MCP_8MHZ);
mcp2515.setNormalMode();
pinMode(4,OUTPUT);
}
void loop()
{
```

```
if (millis()-time1>=4000)
  {
lcd.init();
    time1=millis();
   }
  ////////////////////////ultrason/////////////////////
if (mcp2515.readMessage(&canMsg2) == MCP2515::ERROR_OK)
    {
if (canMsg2.can_id==0x036)
      {
int d=canMsg2.data[0];
lcd.setCursor(0,3);
lcd.print("la distance: ");
lcd.print(d);
lcd.setCursor(15,3);
lcd.print("cm");
}          }
  ////////////////////port//////////////
if (mcp2515.readMessage(&canMsg1) == MCP2515::ERROR_OK)
  {
```

```
if (canMsg1.can_id==0x011)
   {
int z = canMsg1.data[0];
if(z==0)
{
digitalWrite(4,HIGH);
lcd.setCursor(0,1);
lcd.print("porte ouverte  ");
}
else
     {
digitalWrite(4,LOW);
lcd.setCursor(0,1);
lcd.print("                ");
      }                  }              }
    ////////////////////////////////////dht1//////////////////
if (mcp2515.readMessage(&canMsg3) == MCP2515::ERROR_OK)
    {
if (canMsg3.can_id==0x014)
    {
```

```
float t=canMsg3.data[0];
lcd.setCursor(0,0);
lcd.print("temperature:");
lcd.print(t);
lcd.setCursor(18,0);
lcd.print("C'");
}                }
    ////////////////motor////////////////
if (mcp2515.readMessage(&canMsg4) == MCP2515::ERROR_OK)
    {
if (canMsg4.can_id==0x26)
     {
floatvs=canMsg4.data[0];
lcd.setCursor(0,2);
lcd.print("vitesse ");
lcd.print(vs);
lcd.setCursor(15,2);
lcd.print("km/h");
Serial.println(vs );
}          }          }
```