

REPUBLIQUE DU CAMEROUN

Paix – Travail – Patrie

UNIVERSITE DE YAOUNDE I
ECOLE NORMALE SUPERIEUR
D'ENSEIGNEMENT TECHNIQUE
D'EBOLOWA
DEPARTEMENT DE GENIE
ELECTRIQUE



REPUBLIC OF CAMEROUN

Peace – Work – Fatherland

UNIVERSITY OF YAOUNDE I
HIGHER TECHNICAL TEACHER
TRAINING COLLEGE OF
EBOLOWA
DEPARTMENT OF ELECTRICAL
ENGINEERING

**Filière
Electrotechnique (ET)**

**MISE EN ŒUVRE D'UNE PLATEFORME DIDACTIQUE
POUR L'APPRENTISSAGE DES RESEAUX LOCAUX
INDUSTRIELS ET SUPERVISION**

Mémoire de fin d'étude
En vue de l'obtention du Diplôme de Professeur d'Enseignement
Technique et
Professionnel de 2e grade (DIPET II)

Par : NGUEMETA Selambi Bienvenue

Sous la direction de
NNEME NNEME Léandre
Maître de Conférences, Université de Douala
AKOUE Henry Joël
Assistant, Université de Douala

Année Académique : 2019 - 2020



TABLE DES MATIERES

DEDICACES	vi
REMERCIEMENTS	viii
LISTE DES TABLEAUX.....	ix
LISTES DES FIGURES.....	x
LISTE DES ACRONYMES ET ABREVIATIONS.....	xii
RESUME.....	xiv
ABSTRACT	xv
INTRODUCTION GENERALE.....	1
CHAPITRE 1: REVUE DE LA LITTERATURE	3
INTRODUCTION.....	4
1.1 Les réseaux locaux industriels et la notion de supervision	4
1.1.1 Les types de réseaux informatiques	4
1.1.2 Les réseaux locaux industriels dans le modèle OSI.....	4
1.1.3 Réseaux locaux industriels et pyramide CIM	2
1.1.4 Typologie de canaux de transmission	3
1.1.5 Modèles de coopération	4
a) Modèle client/serveur	4
b) Modèle producteur/consommateurs	4
c) Modèle producteurs-distributeur-consommateurs	5
1.1.6 La notion de supervision	6
a) Les organes de supervision	6
b) Fonctions existantes dans un système automatisé de production.....	7
c) Architecture matérielle du système de supervision	7
1.1.7 Les bus de terrain	8
a) Le bus ASI.....	8
b) Le bus CAN.....	9

c) DeviceBus.....	9
b) ModBus	10
c) profibus.....	10
d) Bus FipIo	11
e) Interbus.....	11
f) Ethernet TCP/IP.....	11
1.1.8 Quelques maquettes didactiques d'étude des réseaux locaux industriels	14
a) Maquette didactique d'étude des réseaux locaux industriels ZE4112 de SSEDU	14
b) Maquette didactique d'étude des réseaux locaux industriels CAN01 de DIDALAB	16
CONCLUSION.....	17
CHAPITRE 2: MATERIELS ET METHODES	19
INTRODUCTION.....	20
2.1 Présentation des bus de terrain CAN et MODBUS	20
2.1.1 Le Bus CAN.....	20
a) Principe de fonctionnement	20
b) Intérêts du bus CAN	20
c) Situation par rapport au modèle OSI	21
d) Support de transmission.....	23
e) Effet des longueurs de câble	25
f) Le Non-Return to Zéro.....	26
g) Trames CAN.....	26
h) Gestion des erreurs	29
2.1.2 Le bus de terrain MODBUS.....	31
a) Principe general	31
b) Liaison série RS485	32
c) Câblage liaison ModBus RS485	33
d) Liaison symétrique (ou différentielle).....	33
2.1.3 Les types de transmissions	34

a) Transmission simple : monodirectionnel	34
b) Transmission half-duplex : bidirectionnel alterné.....	34
c) Transmission full-duplex : bidirectionnel simultané.....	35
2.1.4 Les supports physiques.....	36
a) Les principaux supports utilisés	36
b) Trame ModBus.....	36
2.2 Application des bus de terrain CAN et ModBus: maquette didactique des feux tricolores et supervision du trafic routier	38
2.2.1 Cahier de charge.....	38
2.2.2 Matériels et logiciels utilisés	39
a. Logiciel Arduino (IDE)	39
b. Le logiciel LabVIEW	40
c. La carte Arduino Uno	41
d. Les diodes	42
e. Les résistances	43
f. Réalisation du bus CAN	44
g. Réalisation du bus ModBus.....	45
h. Méthode de mise en œuvre de la maquette.....	46
CHAPITRE 3: SIMULATIONS, REALISATION ET DISCUSSIONS	47
INTRODUCTION.....	48
3.1) descriptions des différents blocs	48
3.2) Principe de réalisation des différents modules et schémas	49
a) Le module bus de terrain	49
b) Le module de commande	50
c) Module partie opérative.....	51
d) Module de supervision	51
3.3) les étapes de réalisation pratique de la maquette finale	54
a) La soudure	54

b) La coque	56
3.4) présentation et caractéristiques de la maquette finale.....	57
a) Présentation	57
b) Caractéristiques générales et performances techniques	58
CONCLUSION	59
CONCLUSION GENERALE	60
PERSPECTIVES.....	61
REFERENCES BIBLIOGRAPHIQUES	62
ANNEXES	64

FICHE DE CERTIFICATION DE L'ORIGINALITE DU TRAVAIL

Je soussigné NGUEMETA SELAMBI Bienvenue, atteste que se présent Mémoire de fin d'étude en vue de l'obtention du Diplôme de Professeur des Lycées d'Enseignement Technique et Professionnel de deuxième grade (DIPETP II) en Génie Electrique option Electrotechnique est le fruit de nos propres travaux effectués à l'Ecole Normale Supérieur de l'Enseignement Technique (ENSET) d'Ebolowa de l'université de Yaoundé I, sous la direction de NNEME NNEME Léandre, *Maître de Conférences, Université de Douala* et l'encadrement de AKOUE Henry Joël, *Assistant, Université de Douala*.

Ce travail est authentique et n'a pas encore fait l'objet d'une présentation pour l'obtention d'un diplôme universitaire.

Visa de l'hauteur

NGUEMETA SELAMBI Bienvenue

Visa directeur de mémoire

visa encadreur

Pr NNEME NNEME Léandre

M. AKOUE Henry Joël

Visa Chef de Département

Pr. NDJAKOMO ESSIANE Salomé

DEDICACE



MA FAMILLE

REMERCIEMENTS

Au terme de ce cycle de formation, nous souhaitons adresser nos sincères remerciements à toutes les personnes qui de près ou de loin ont contribué à l'élaboration de ce travail ; plus particulièrement :

- Au professeur **NDJAKOMO ESSIANE Salomé** Directeur de l'ENSET de d'Ebolowa pour nous avoir prodigué une formation de qualité.
- Au **professeur NNEME NNEME Léandre**, directeur de l'ENSET de Douala pour la supervision de ce travail.
- A **Mme FOUA BELLA Régine** responsable du Département du Génie Electrique (**GEL**) pour son accompagnement dans la programmation et le suivi des enseignements tout au long de notre formation.
- A mon encadreur **M. AKOUE HENRY Joël** pour le coté professionnelle donné à ce travail à travers son encadrement permanent.
- A Nos enseignants ainsi qu'à l'administration générale de l'ENSET d'Ebolowa qui nous ont suivies durant notre parcours, pour leurs conseils et encouragement multiforme.
- A mes mamans, **Mme PENAYE Jacqueline** et **Mme BIJIHO Marie claire** pour leur amour et leur soutien sans faille. Que ce travail soit pour eux un sujet de fierté et de réjouissance.
- A mes frères et sœurs notamment **M. Désiré TOUOYEM** et son épouse **Mme LAMBO Nadège**, **Mr Francis TAFBOU** et son épouse **Mme TCHOFFO Rosine**, **Mr LEPAFFO Arnaud** et son épouse, **Mr KAMTA Rodrigue** et son épouse, **Mme FEPI Josseline** et bien d'autres pour leur soutien moral et financier.
- A **Mlle KENGNE DZEUGUI Divine Flore** pour le soutien, l'amour et l'accompagnement qu'elle m'a accordé tout au long de cette formation.
- A mes oncles et tantes, cousins et cousines pour leurs soutiens et réconfort.
- A mes camarades et collègues de promotion pour l'ambiance chaleureuse qui nous a toujours animés durant ce cycle de formation.
- A tous mes amis notamment **Valdes TIETSAP**, **Nelson MBOUGNA**, **Anicet TCHOFFO**, Et tous ceux qui ont participé de près ou de loin à l'élaboration de ce travail.
- Aux membres du jury qui examineront ce travail

LISTE DES TABLEAUX

Tableau 1 : Synthèse du modèle OSI pour les RLI (Les réseaux locaux industriels synthèse informatique, Jean Claude CABIANCA, P2) 1

Tableau 2: étude comparative des différents bus de terrain au niveau liaison et application **Erreur ! Signet non défini.**

Tableau 3 : étude comparative des différents bus de terrain au niveau physique **Erreur ! Signet non défini.**

Tableau 4: type de transmission en CAN..... 24

Tableau 5: Caractéristiques de la maquette par transmission CAN..... 58

Tableau 6: Caractéristiques de la maquette par transmission ModBus..... 58

LISTES DES FIGURES

Figure 1: la pyramide SIM	2
Figure 2: les niveaux de la pyramide CIM et types de données.....	2
Figure 3: schéma requête/réponse dans le modèle client/serveur	4
Figure 4: le modèle producteur/consommateur	5
Figure 5: Modèle Producteurs/Distributeur/Consommateur	6
Figure 6: hiérarchie de la supervision en entreprise.....	8
Figure 7: présentation et règles d'arbitrage du bus CAN.....	9
Figure 8: maquette ZE4112 de SSEDU	15
Figure 9: maquette CAN 01A de DIDALAB.....	17
Figure 10: maquette CAN 01B de DIDALAB.....	17
Figure 11: principe de fonctionnement du bus CAN	20
Figure 12: application du CAN au modèle OSI	23
Figure 13: schéma de câblage du bus CAN	24
Figure 14: Niveaux de tension en Low Speed & High speed	25
Figure 15: décomposition d'une trame de donnée	27
Figure 16: décomposition d'une trame de requête	28
Figure 17: constitution d'une trame d'erreur	28
Figure 18: décomposition de la trame de surcharge.....	29
Figure 19: Les sources d'erreur dans la trame CAN.....	31
Figure 20: principe de fonctionnement du bus modBus	32
Figure 21: Cablage liaison ModBus RS485.....	33
Figure 22 : liaison symétrique.....	33
Figure 23: transmission simplex monodirectionnel	34
Figure 24: transmission half-duplex bidirectionnelle alternée.....	35
Figure 25: transmission half-duplex série asynchrone.....	35
Figure 26: carrefour à deux voies avec croisement	38
Figure 27: IDE Arduino	39
Figure 28: carte Arduino Uno	41
Figure 29: schémas de la LED	43
Figure 30: synoptique bus CAN.....	44

Figure 31: la carte MCP2515-CAN	45
Figure 32: connexion MCP2515_CAN et Arduino.....	45
Figure 33: carte MAX485	46
Figure 34: connexion MAX485 Arduino	46
Figure 35: synoptique de la maquette	49
Figure 36: schéma électronique du bloc bus de terrain	50
Figure 37: schéma de commande	50
Figure 38: schéma des feux de carrefour	51
Figure 39: face avant Labview	52
Figure 40: face arrière Labview	53
Figure 41: panneau de signalisation	54
Figure 42: circuit à microcontrôleur ATmega328P	55
Figure 43: module CAN/ModBus	56
Figure 44: la coque.....	56
Figure 45: structure globale de la maquette finale	57

LISTE DES ACRONYMES ET ABREVIATIONS

- **RLI** : Réseau Local Industriel
- **WAN** : Wide Area Network
- **MAN** : Metropolitan Area Network
- **LAN** (Local Area Network)
- **OSI** : Open Systems Interconnection
- **CIM** : Computer Integrated Manufacturing
- **CSMA/CD** : Carrier Sense Multiple Access with Collision Detection
- **CSMA /CA** : Carrier Sense Multiple Access with Collision Avoidance
- **MMS** : Manufacturing Message Service
- **FIP** : Fieldbus Instrumentation Protocol
- **IHM** : Interface Homme Machine
- **PC** : Personal Computer
- **API** : Automates Programmables Industriels
- **ASI** : Actuators Sensors Interface
- **MAC** : Medium Access Control
- **LLC** : Logical Link Control
- **PLS** : Physical Signalling
- **PMA** : Physical Medium Attachment
- **MDI** : Medium Dependant Interface
- **CAN** : Control Area Network
- **ProFiBus** : Process Field Bus
- **DP** : Decentralized Periphery
- **FMS** : Fielbus Message Specification
- **PA** : Process Automation
- **BA** : Building Automation
- **WiFi** : Wireless Fidelity
- **PVC** : Polychlorure de Vinyle
- **TOR** : Tout Ou Rien
- **CRC** : Contrôle de Redondance Cyclique

- **ECU** : Electronic Control Unit
- **NRZ** : No Return to Zéro
- **SOF** : Start Of Trame
- **RTU** : Remote Terminal Unit
- **ASCII** : American Standard Code For Information Interchange
- **BP** : Bouton Poussoir
- **VI** : Instruments Visuels
- **LabVIEW** : Laboratory Virtual Instrument Engineering Workbench

RESUME

Le but de notre étude porte sur la mise au point d'une maquette didactique pour l'apprentissage des réseaux locaux industriels et la supervision. Au-delà des références bibliographiques, nous avons fait recours au matériel électronique et aux logiciels Arduino, Proteus et LabVIEW. Pour aboutir aux principales fins de notre projet de mémoire nous avons premièrement présenté de façon brève les réseaux locaux industriels, la notion de supervision, les bus de terrain et quelques maquettes d'étude de ces réseaux. Ensuite nous avons particulièrement choisis deux bus de terrain qui intègrent notre plateforme et enfin nous avons réalisé une maquette intégrant quatre blocs à savoir : le bloc partie opérative constitué des feux de carrefour, le bloc de commande constitué de deux boutons poussoirs pour l'augmentation et la diminution automatique du temps de commutation entre ces feux, le bloc bus de terrain constitué des deux bus de terrain (CAN et ModBus) au choix pour la communication entre différents blocs et le bloc supervision constitué d'une interface homme-machine sur LabVIEW.

Mots et expressions clés : **plateforme didactique, supervision, apprentissage, réseaux locaux industriels**

ABSTRACT

The aim of our study is to develop a didactic model for learning local industrial networks and supervision. Beyond the bibliographical references, we have made use of electronic hardware and software Arduino, proteus and Labview. To reach the main ends of our dissertation project, we first briefly presented industrial local networks, the concept of supervision, field buses and some models of study of these networks. Then we particularly chose two field buses which integrate our platform and finally we made a model integrating four blocks namely: the operational part block consisting of crossroad lights, the control block consisting of two push buttons for the increase and automatic reduction of the switching time between these lights, the field bus block consisting of the two field buses (CAN and ModBus) of your choice for communication between different blocks and the supervision block consisting of a man-machine interface on Labview.

Key words and phrases : **didactic platform, supervision, learning, local industrial network**

INTRODUCTION GENERALE

Les réseaux locaux industriels ont été introduits petit à petit dans les systèmes automatisés. Ils sont nés avec le développement de l'électronique et des matériels numériques programmables et sont enseignés dans les universités camerounaises.

La massification qu'a connue le système éducatif camerounais ces dernières années, concomitante à une quasi-stagnation du nombre d'enseignants pose un énorme défi d'encadrement pédagogique, particulièrement au niveau des établissements à accès ouvert. Ce défi semble insurmontable quand il s'agit d'enseignement nécessitant un encadrement de proximité et différencié comme c'est le cas des travaux pratiques dans les filières scientifiques et techniques. Au niveau de ces activités, la situation est aggravée par la limitation du nombre de plateforme didactiques et d'espaces appropriés qui permettraient aux étudiants les possibilités d'observation et de manipulation requises. Ces contraintes ont réduit bon nombre d'activités sensées être pratiques à de simples séances de démonstrations au cours desquelles les étudiants assistent à titre de spectateurs passifs. La question qu'on se pose est celle de savoir si dans le cadre de l'apprentissage des réseaux locaux, de la supervision et de la commutation industrielle la mise en exergue d'une maquette didactique encouragerait la performance, la réussite et la qualification des étudiants ? Une contribution en vue de remédier à cette situation ne peut que faire l'objet d'un encouragement et d'une appréciation à sa juste valeur. C'est ce qui met en relief l'importance de notre travail dont la thématique est la « **mise en œuvre d'une plateforme didactique pour l'apprentissage des réseaux locaux industriels et la supervision** ».

Ce travail vise à exploiter les immenses possibilités des outils didactiques d'apprentissages disponibles actuellement pour mettre en place une maquette de travaux pratiques permettant aux étudiants de mieux appréhender les notions de réseaux locaux industriels et de supervision.

La démarche de travail retenu pour cette mise au point est une revue de la littérature au premier chapitre portant sur une présentation brève des réseaux locaux industriels, de la supervision, des bus de terrain et quelques maquettes d'étude de ces réseaux. Au deuxième chapitre abordant le matériel et la méthode nous avons fait un état des lieux sur les bus de terrain CAN et ModBus que nous avons utilisé pour mettre en œuvre la maquette

des feux de carrefour tout en présentant leurs intérêts, leurs situations par rapport au modèle OSI, leurs caractéristiques électriques et leurs modes de fonctionnement. Au troisième chapitre portant sur la simulation, la réalisation et les commentaires nous avons simulé et réalisé les différents blocs et systèmes qui entre dans la mise en œuvre de notre maquette et discuté l'atteinte des objectifs précédemment énoncés.

Chapitre

1

CHAPITRE 1: REVUE DE LA LITTERATURE

Aperçu

1.1 Les réseaux locaux industriels et la notion de supervision

1.2 Les bus de terrain

1.3 Quelques maquettes didactiques d'étude des réseaux locaux industriels

INTRODUCTION

L'apparition des régulateurs numériques et des automates programmables a conduit les offreurs à mettre sur le marché des réseaux pour les interconnecter et rapatrier à moindre coût de câblage les informations nécessaires à la conduite par les opérateurs dans les salles de commande. De même, le souci de commander et contrôler à distance des procédés industriels a favorisé le développement de la supervision à travers les interfaces graphique homme-machine. Ce chapitre sera organisé en trois parties majeures à savoir les réseaux locaux industriels et la notion de supervision, les bus de terrain et quelques maquettes d'étude de ces réseaux.

1.1 Les réseaux locaux industriels et la notion de supervision

1) Les types de réseaux informatiques

Les réseaux sont généralement répartis en fonction de leur étendue géographique et on rencontre en général les :

- **WAN** (Wide Area Network)

Réseaux de grandes distances internationaux et nationaux (Ex : réservation aérienne, échanges bancaires, Internet...) : distance > 100km

- **MAN** (Metropolitan Area Network)

Réseaux pour les besoins d'une agglomération (Ex : gestion des bâtiments communaux, trafic urbain...) : distance > 1km

- **LAN** (Local Area Network)

Réseaux limités à 1 ou plusieurs bâtiments (Ex : réseau d'entreprise) : distance < 1km

2) Les réseaux locaux industriels dans le modèle OSI

Un réseau local industriel est basé le plus souvent sur le modèle OSI à 3 couches :

- La couche Application (qui peut être vide dans de nombreux réseaux)
- La couche Liaison qui doit assurer un transport de données
- La couche Physique qui doit respecter des contraintes de l'environnement

Mémoire de fin d'étude en vue de l'obtention du DIPETP II

Tableau 1 : Synthèse du modèle OSI pour les RLI (Jean Claude CABIANCA, '' Les réseaux locaux industriels synthèse informatique'', P2)

Couche OSI	Nom de la couche	Rôle de la couche		Format des données		
7	Application	Interface entre le réseau et l'utilisateur	Elle assure tous les services qui peuvent simplifier la vie de l'utilisateur et qui ne sont pas assurés par les autres couches		Messages	
2	Liaison de données	Méthodes d'accès au support	Aléatoires	CSMA/CD: détection de collision	Trame	
				CSMA/CA : évitement de collision		
			Déterministes	Maitre/esclave		
				Arbitre de bus		
		Anneau à jeton				
Sécurisation des échanges	En-tête de trame (question, réponse,...)		Bits de redondance (parité, checksum, CRC,...)			
1	Physique	Codage de l'information	Tension		Bits	
		Topologie	Bus			
			Etoile			
			Anneau			
		Support de l'information (média)	Le cuivre	Câble coaxial		
				Paire torsadée		
			Fibre optique	Monomode		
				Multimode		
			Autres	Courant porteur		
		Radio				

3) Réseaux locaux industriels et pyramide CIM

La pyramide CIM désormais incontournable a été créée dans les années 80 afin de mieux faire comprendre l'apport de l'informatique au monde des automatismes. Les réseaux locaux industriels répondent aux besoins du système d'automatisme (niveaux 2, 1 et 0), les réseaux informatiques et les systèmes de télécommunication apparaissent dans les niveaux 3 et 4.

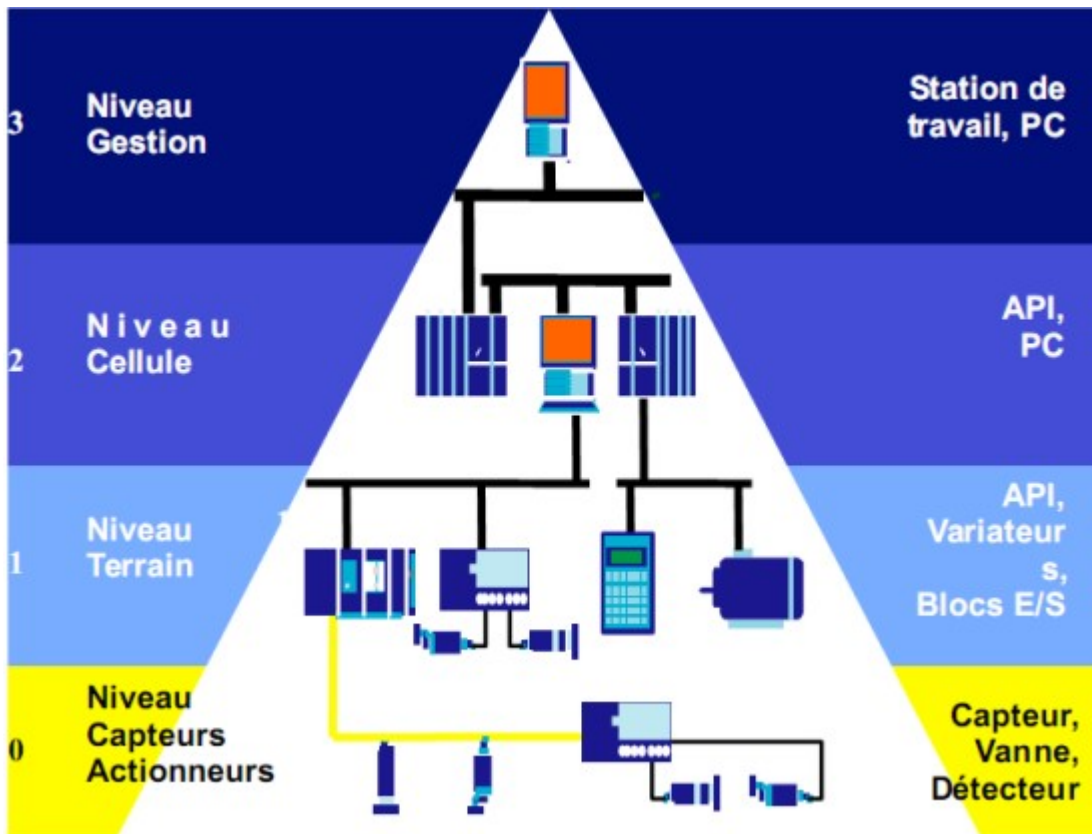


Figure 1: la pyramide CIM (Jean Claude CABIANCA, "Les réseaux locaux industriels synthèse informatique", P9)

Niveau	Temps de réponse	Distance maximale	Données échangées
3	$n \times s$	$n \times 10 \text{ km}$	fichiers
2	$n \times 100 \text{ ms}$	$n \times 1 \text{ km}$	messages
1	$n \times 10 \text{ ms}$	$n \times 100 \text{ m}$	mots
0	$n \times 1 \text{ ms}$	$n \times 10 \text{ m}$	bits

Figure 2: les niveaux de la pyramide CIM et types de données

4) Typologie de canaux de transmission

L'échange de données dans les réseaux se fait via des supports de transmissions dont la nature peut fortement varier. Dans les médias de transport, nous trouvons entre autres :

- Les supports métalliques

Qui peuvent en fait être construits avec n'importe quel métal bon conducteur. C'est en général le média le moins coûteux, mais également le plus limité, surtout en termes de distance.

Exemple : câbles en "cuivre"

- Les supports optiques : fibres optiques

Qui ont l'avantage d'être insensibles aux perturbations électromagnétiques dans lesquelles nous baignons. De plus, la vitesse de propagation de la lumière dans ces fibres autorise de longues distances et de nombreuses solutions permettent une très grande bande passante, donc un gros débit de données. Malheureusement, la fibre optique souffre de quelques défauts :

- Sa relative fragilité.
- La difficulté d'y adapter de la connectique.
- Le prix de cette connectique.

Les liaisons "hertziennes" qui assurent la communication par ondes électromagnétiques et qui couvrent elles-mêmes plusieurs technologies:

- La liaison "classique" c'est à dire en émission omnidirectionnelle qui encombre beaucoup l'espace mais apporte une grande souplesse dans la mobilité des équipements connectés. Un exemple typique en est le téléphone mobile dit "cellulaire".
- La liaison par faisceau hertzien plus intéressante car l'émission est extrêmement directive. L'inconvénient est que les émetteurs et les récepteurs doivent être rigoureusement alignés et ne peuvent donc pas être mobiles.
- Les liaisons par satellite. Ces dernières peuvent utiliser : Des satellites géostationnaires et des satellites à basse altitude qui ne sont pas géostationnaires. Il est courant, pour aller d'un point à un autre, d'emprunter plusieurs de ces technologies...

5) Modèles de coopération

a) Modèle client/serveur

Deux processus sont en relation : le client et le serveur. Le client émet une demande de service vers le serveur. Le serveur, selon ses possibilités et ses ressources, traite la requête et renvoie la réponse au client. Ce modèle est très général. Une grande diversité de services peut fonctionner selon ce modèle. La plupart des services/protocoles de couche application respectent ce modèle (MMS en est le prototype). La durée totale de l'opération est imprévisible, sauf à faire des hypothèses sur la disponibilité du serveur et du réseau chargé d'acheminer la demande et la réponse.

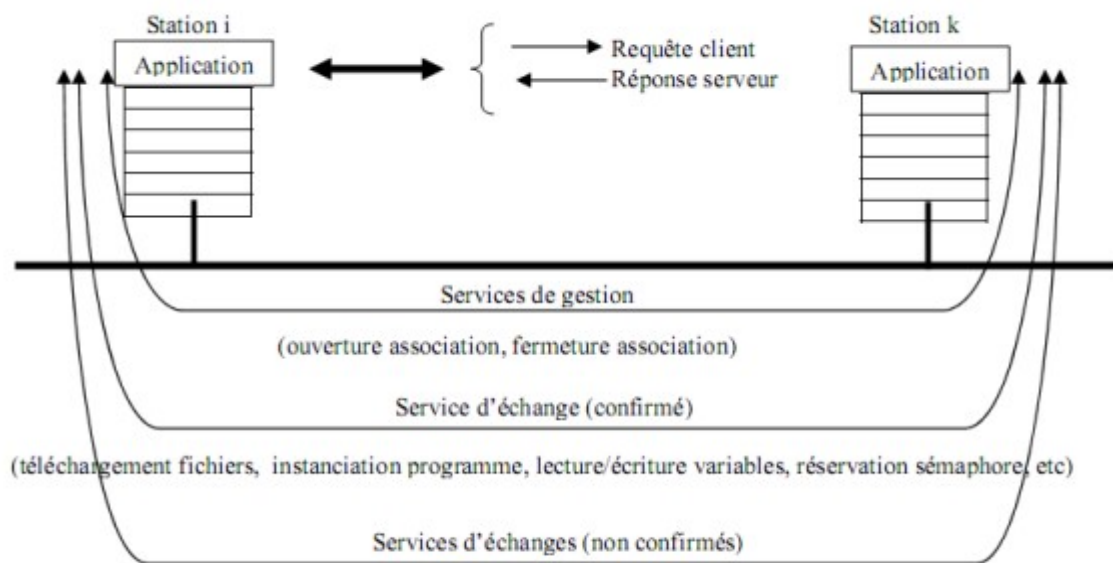


Figure 3: schéma requête/réponse dans le modèle client/serveur (Jean-Pierre Thomesse, *"réseaux locaux industriels"*, P18)

b) Modèle producteur/consommateurs

L'initiative de l'émission est laissée au producteur, mais celle de la production peut être confiée à un client parmi les consommateurs qui déclenche l'opération.

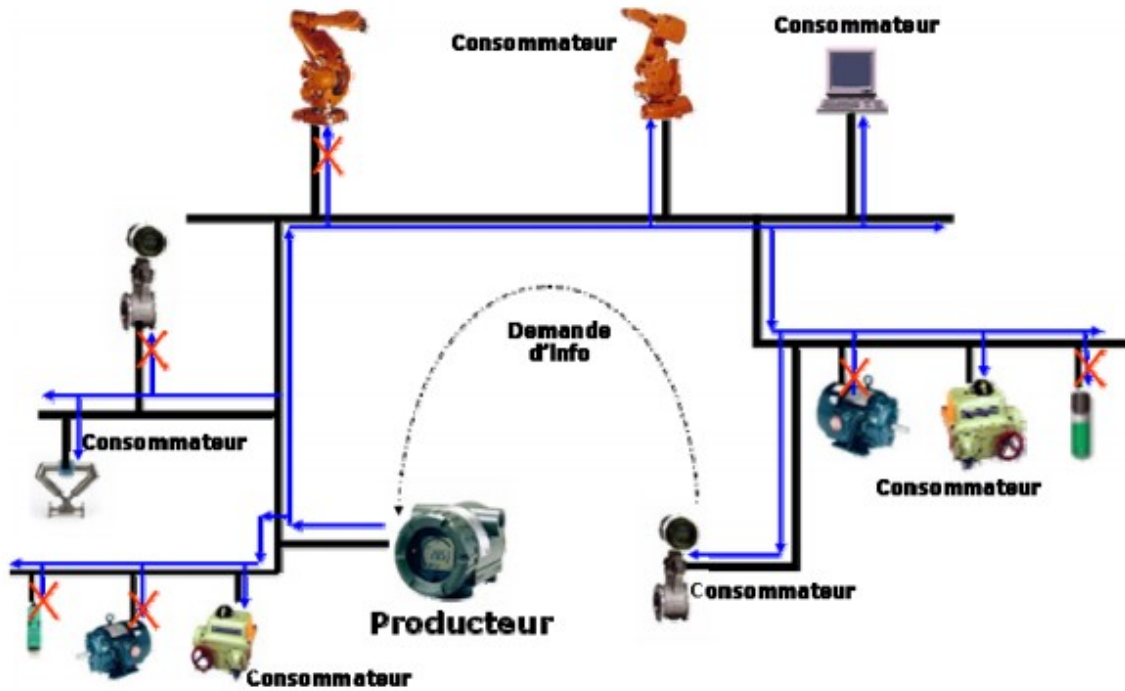


Figure 4: le modèle producteur/consommateur ((Jean-Pierre Thomesse, 'réseaux locaux industriels', P20)

c) Modèle producteurs-distributeur-consommateurs

Trois types de processus cohabitent, les producteurs, le ou les distributeurs et les consommateurs. Un producteur produit localement la valeur d'un objet ; le distributeur déclenche le transfert et la réception, l'opération de recopie, chez les consommateurs, de la valeur originale prise chez le producteur ; les consommateurs utilisent la copie locale de l'objet. Ces processus sont plus ou moins dépendants les uns des autres. Ils peuvent être complètement indépendants, ou être coordonnés de façon étroite, par exemple, produire de façon périodique, transmettre à la même fréquence, et consommer de manière identique, mais toute autre situation peut être définie. Les comportements temporels des divers processus sont définis selon les besoins de l'application. Dans le cas de protocole FIP (Fieldbus Instrumentation Protocol), des mécanismes de validation temporelle des valeurs d'objets ont été introduits pour permettre aux consommateurs de savoir si les délais impartis au producteur et au distributeur sont ou non respectés. C'est cette notion d'original et de copies qualifiées temporellement de la valeur d'un objet qui fait parfois dire que FIP est un système réparti de gestion de base de données de terrain.

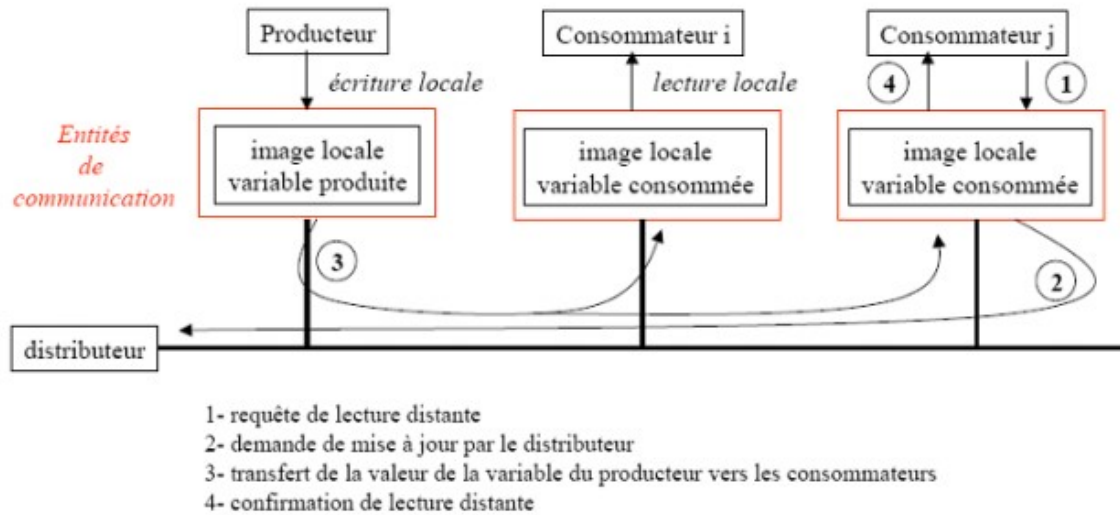


Figure 5: Modèle Producteurs/Distributeur/Consommateur ((Jean-Pierre Thomesse, 'réseaux locaux industriels', p22)

6) La notion de supervision

Ayant constaté l'importance des réseaux dans la bonne marche et le développement des industries, il est important pour chaque entreprise de pouvoir contrôler et retracer les activités des leurs équipements. La supervision réseau est donc cette branche de l'informatique qui a pour but de surveiller les équipements et ressources du réseau.

a) Les organes de supervision

Il s'agit de l'ensemble des composants utilisé pour la transmission des consignes à la machine par l'opérateur, et le retour d'information vers ce dernier. Il s'agit en particulier des outils permettant de confectionner des pupitres utilisateur physiques, ainsi que des interfaces Homme-Machine. Nous citons quelques-uns :

Organes de visualisation :

- Colonnes
- Voyants
- Compteurs
- Afficheur texte
- Afficheur graphique (sur PC par exemple)

- Colonne lumineuse

Organes d'émission de consignes

- Les boutons poussoirs (ordinaire, lumineux, coup de poing)
- Les sélecteurs (à bouton rotatif, lumineux, à clef)
- Les manipulateurs (à levier)
- Les claviers
- Les terminaux d'exploitation (texte, graphique, exemple XBT Magellis (Schneider), Siemens ...)

b) Fonctions existantes dans un système automatisé de production

Les fonctions de base dans une entreprise industrielle peuvent être décomposées en trois niveaux :

- Niveau planification,
- Niveau exécution,
- Niveau contrôle.

Ces trois niveaux sont généralement présents dans les systèmes industriels.

c) Architecture matérielle du système de supervision

La supervision dans la hiérarchie d'une entreprise manufacturière est représentée par la figure suivante :



Figure 6: hiérarchie de la supervision en entreprise (P. Bonnet, "introduction à la supervision" P 3)

On peut distinguer ainsi deux mondes de supervision :

- Monde industrielle
- Monde bureautique

7) Les bus de terrain

a) Le bus ASI

Le bus ASI est un bus série permettant de diminuer le nombre de câbles nécessaire à l'utilisation d'un grand nombre de capteurs ou de pré-actionneurs : ces derniers étant reliés directement ou par l'intermédiaire d'une embase (base de donnée) à l'automate central via un seul câble. Il permet d'utiliser jusqu'à 31 embases soit 124 capteurs tout en gardant un temps de réaction faible de 5ms. Avec la dernière version, il est possible de raccorder jusqu'à 62 embases soit 248 capteurs mais le temps de réaction est alors doublé. Il est possible de remplacer le câble AS-I par un câble rond standard à deux conducteurs. La longueur maximale d'un segment AS-I est de 100 mètres ; toutefois il est possible d'augmenter cette distance en utilisant des récepteurs (deux au maximum) en atteignant la limite de distance entre l'automate et la dernière embase qui est de 300 mètres.

b) Le bus CAN

- Présentation

Le bus CAN est utilisé dans de nombreux domaines : automobile, agricole, industriel et médical. Ce bus de terrain est connu pour être une solution de communication économique et efficace, sa vitesse de transmission peut atteindre 1 Mbit/s. Chaque équipement connecté, ou « nœud », peut communiquer avec tous les autres. L'accès au bus CAN suit la technique des collisions par priorité).

- Règles d'arbitrage du bus CAN :

En cas d'émission simultanée de plusieurs stations, l'attribution du bus suit le principe d'arbitrage suivant. Les stations émettant simultanément sur le bus, comparent bit à bit l'identificateur de leur message (ID) avec celui des messages concurrents. Les ID de priorités moins élevée perdent la compétition face à celle qui a la priorité la plus élevée, pour cela les stations sont câblées sur le bus par le principe du "ET câblé". En cas de conflit, c'est à dire émission simultanée, la valeur 0 écrase la valeur 1. On appelle donc "état dominant" l'état logique 0, et "état récessif" l'état logique 1.

Lors de l'arbitrage bit à bit, dès qu'une station émettrice se trouve en état récessif et détecte un état dominant, « elle perd » et arrête d'émettre.

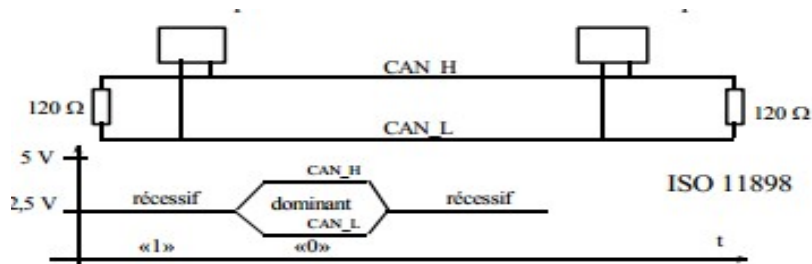


Figure 7: présentation et règles d'arbitrage du bus CAN

c) DeviceBus

DeviceNet est un bus de terrain normalisé basé sur CAN, qui a été développé par Rockwell Automatisation et qui est utilisé dans l'automatisation industrielle. Il permet la communication au plus bas niveau d'une installation automatisée entre appareils de différents fabricants sans nécessiter d'adaptateur. Le bus de capteurs actionneurs deviceNet utilise un modèle producteur- consommateur et un modèle client-serveur pour gérer les flux de donnée

entre les différents équipements installés. Selon sa configuration, DeviceNet peut opérer soit au sein d'une architecture maître-esclave, soit au sein d'une architecture distribuée en mettant en œuvre des échanges de données point à point.

b) ModBus

Le bus ModBus répond aux architectures Maître/Esclave. Il a été créé par la société Modicon en 1978 pour interconnecter les automates programmables. Le bus est composé d'une station Maître et de stations Esclaves. Seule la station Maître peut être à l'initiative de l'échange (la communication directe entre stations Esclaves n'est pas réalisable). Le réseau ModBus Plus (ModBus+) est un réseau local industriel, déterministe et performant, répondant à des architectures étendues de type Client/serveur, combinant haut débit (1 Mbit/s), supports de transmission simples et économiques ainsi que de nombreux services de messagerie. On peut lui connecter 64 nœuds sur 1 800 m (32 nœuds sur 450 m sans répéteur) sur paire torsadée.

c) Profibus

Profibus est le résultat d'un projet lancé en 1989 par le ministère fédéral allemand de la recherche et de la technologie, il a été développé et financé par des entreprises d'automatismes comme Siemens. (**Conservatoire National des Arts et Métiers, II A10**, bus profibus 2004-2005).

Il est composé de deux protocoles de transmission appelés profils de communication DP et FMS et de profils applicatif tels que PA, ou Profisafe. Ces profils de communication définissent la façon dont les données sont transmises sur le support physique. Le profil de communication FMS est destiné à la communication entre équipements dits intelligents tels que des ordinateurs ou des automates, tandis que le profil DP est plutôt destiné à la communication entre équipements intelligents et équipements basiques tels que des capteurs et actionneurs. Le protocole PROFIBUS est composé de 3 couches de communications, une couche physique, une couche liaison et une couche application. Il peut être utilisé avec trois types de média, la paire torsadée (RS485), le bifilaire (CEI 1158-2) et la fibre optique, cette dernière a pour avantage de fournir des débits élevés, d'avoir une forte immunité aux parasites et peut être utilisée pour des grandes distances, par contre elle a pour inconvénient d'être plus fragile et plus coûteuse. Il est à noter cependant que PROFIBUS est utilisé sur des infrastructures déjà existantes telles que le réseau Ethernet.

d) Bus FipIo

Le bus de terrain FipIo est un standard de communication entre différents constituants d'automatisme (niveau 1). Il permet la connexion de 127 équipements à partir du point de connexion intégré au processeur. Ce bus de terrain est basé sur les mécanismes producteurs/consommateurs. Il est destiné au déport d'Entrées/Sorties(E/S) jusqu'à 1 5 km et accueille des équipements.

e) Interbus.

Interbus est un standard de fait (1987), développé par Phoenix Contact, spécialiste des capteurs/actionneurs et qui respecte le modèle OSI (1, 2,7). Le bus de terrain Interbus est un produit reconnu : Norme CENELEC EN50254 (1 997) et Norme DIN 1 9258 (1 993). Interbus est à ranger dans la catégorie des bus capteur/actionneur. Il offre néanmoins la possibilité d'émettre des messages de 1 0 à 1 00 octets Interbus est beaucoup utilisé dans l'industrie : 700 fournisseurs de modules Interbus et 1 20000 réseaux mis en place reliant 1 500000 modules.

f) Ethernet TCP/IP.

Ethernet est une technologie de plus en plus utilisée comme solution de communication dans l'industrie :

- Utilisation inéluctable dans les ateliers : technologie banalisée, performante, fiable, peu onéreuse.
- Son point faible : son indéterminisme dû à la méthode d'accès CSMA/CD.

Mais Ethernet sera incontournable pour mettre en œuvre des programmes d'automatisation répartis d'autant plus que les flux de données ne cessent de croître.

Tableau 2 : étude comparative des différents bus de terrain au niveau liaison et application

	ASI	CAN	DeviceNet	Ethernet TCP/IP ModBus	Profibus-DP	FIPIO	Interbus	ModBus
Méthode d'accès au médium	Maître Esclaves	CSMA/CA	CSMA/CA	CSMA/CD	Token ring et maître/esclave	Gestionnaire de bus	Maître Esclaves Trame unique	Maître Esclaves
Type et taille des données échangées	ASI V1 : Cycliques: 4 bits E 4 bits S Acycliques: 4 bits P ASI V2 : Cycliques: 4 bits E 3 bits S Acycliques: 3 bits P	E/S cycliques : PDO 8 octets E 8 octets S Acycliques : SDO Param./réglage >8 octets par fractionnement des informations	E/S cycliques : I/O messages 8 octets E 8 octets S ou >8 si fragmentation Acycliques : Explicit messages Param./réglage >8 octets par fractionnement des informations	E/S cycliques: I/O scanning 125 mots E 125 mots S Acycliques: Param. /réglage par messagerie asynchrone 507mots	E/S cycliques: PZD 244 mots E 244 mots S PKW = 1 mot à la fois	E/S cycliques: 32 mots E 32 mots S Acycliques: Param.= 30 mots Réglage. = 30 mots	E/S cycliques: 256 mots E/S Acycliques: 256 mots par fragmentation	Variables acycliques 1920 bits 120 mots

Mémoire de fin d'étude en vue de l'obtention du DIPETP II

Tableau 2 : étude comparative des différents bus de terrain au niveau physique

	ASI	CAN	DeviceNet	Ethernet TCP/IP ModBus	Profibus-DP	FIPIO	Interbus	ModBus
Médium	Câble plat jaune Câble rond non blindé Câble rond blindé	Paire torsadée blindée	Double paires torsadées blindées	Câble coaxial: 10 base 2 - 10 base 5 Paire torsadée blindée: 10 base T - 10 base TX Fibre optique 10 base F - 10 base FX	Paire torsadée blindée Fibre optique	Paire torsadée blindée Fibre optique	Double paires torsadées blindées	Paire torsadée blindée
Distance maxi sans répéteur	100 m	Suivant débit: 25 m à 1 Mbits/s 1 km à 10 Kbits/s	Suivant débit: 100m à 500 Kbits/s 500m à 125 Kbits/s	Paire torsadée 100m Fibre optique 2000m	Suivant débit : 100m à 12 Mbits/s 1,2km à 10 Kbits/s	1000 m en paire torsadée 3000 m en fibre optique	400 m	1300 m
Distance maxi avec répéteurs	300 m	Fonction du type de répéteur	Fonction du type de répéteur	10km fibre optique	400 à 4800 m suivant débit	15 km	12,8 km	Fonction du type de répéteur
Débit	166 Kbits/s	9 débits possibles de 10 Kbits/s à 1 Mbits/s	125, 250 ou 500 Kbits/s	10/100Mbits/s	9,6 Kbits/s à 1 Mbits/s	1 Mbits/s	500 Kbits/s	jusqu'à 19200 bits/s
Nombre maxi d'équipements	ASIV1 : 1 maître + 31 esclaves. ASI V2 : 1 maître + 62 esclaves.	1281 clients et 127 serveurs	641 maîtres et 63 esclaves	64 I/O scanning et ModBus	Mono ou Multi-maîtres 126 équipements maxi	1 gestionnaire+ 126 équipements	512	321 maîtres et 31 esclaves

8) Quelques maquettes didactiques d'étude des réseaux locaux industriels

La formation pratique du réseau d'automatisation industrielle est basée sur le système de contrôle basé sur l'automate d'automatisation intégré. La plateforme de formation à l'automatisation industrielle vise l'enseignement supérieur professionnel et technique et le développement d'équipements de formation complet, applicable à tout type d'intégration mécanique et électrique d'école professionnelle, à l'automatisation et à d'autres types d'enseignement professionnel via la formation du personnel.

1) Maquette didactique d'étude des réseaux locaux industriels ZE4112 de SSEDU

C'est une maquette pouvant être non seulement utilisée comme système de formation pédagogique pour l'automatisation et le réseau, mais aussi comme machine à enseigner dans une salle de classe en réseau avec des sous-systèmes. Dans cette maquette, la fusion organique du bus de terrain profibus, le réseau WIFI industriel, les automates S7-300, siemens S7-200 et WINCC, .net, transmission standard, entrées/sorties à distance, écran tactile et électricien, la puissance électrique trainée et divers contenus de formation, réalisent le partage des ressources éducatives, optimisent la gestion de l'enseignement expérimental.

a) Caractéristiques de la maquette ZE4112

L'équipement est fonction du niveau de réseau subordonné dans la phase expérimentale pour décorer les couches « de contrôle de champ », « de surveillance de processus » et « de gestion d'entreprise » ceci dans le but de faciliter la compréhension des couches du modèle OSI utilisées par les réseaux locaux industriels.

L'équipement du contenu de la formation est la technologie principale pour l'application de l'industrie, pour s'adapter au développement de la technologie de contrôle moderne, peut répondre aux étudiants pour faire des expériences, la conception du curriculum, les besoins de conception des diplômes et les besoins de leur école à entreprendre l'ingénierie.

Le système utilise la plaque de maille mobile et la combinaison de la boîte suspendue, pouvant être pratique pour toute sorte de petite conception de projet d'ingénierie.

L'entrée et la sortie des terminaux, l'entrée et la sortie des automates et la carte d'interface de test externe peuvent être utilisés pour la formation pratique.

b) Performances techniques

- Puissance d'entrée : triphasé à cinq fils ~ 380V plus ou moins 10% /50Hz
- Puissance de sortie du groupe 24V/4.5A
- Sortie du signal de tension réglable de 0-10V
- Sortie du signal de courant réglable de 4-20mA
- Bouton d'arrêt d'urgence, indicateur d'alimentation, instructions d'alimentation.
- Module WIFI industriel, module émetteur-récepteur à fibre optique
- Mesure de voltmètre à courant continu et d'ampèremètre a courant continu
- Environnement de travail : température -10 ~ + 40°C, humidité relative<85%, altitude<4000m
- Capacité de l'appareil :<1.5kVA
- Taille d'aspect : 1600*800*1800mm

c) Composition du produit

La table d'entraînement proprement dite est supportée par la colonne de profil en aluminium et le frein de roue inférieur wanxiang peut être déplacé et positionné d'une manière flexible, l'ordinateur de bureau est avec substrat de 25mm d'épaisseur à haute densité, haute température et haute pression sur la surface du panneau de prévention des incendies pour coller un visage, avec un rail de guidage de type deux, une armoire inferieure à deux, une table de formation pouvant être placée entre les accessoires informatiques, industriels et les outils. La structure est ferme et la beauté généreuse.



Figure 8: maquette ZE4112 de SSEDU (www.ssedu.com)

2) Maquette didactique d'étude des réseaux locaux industriels CAN01 de DIDALAB

C'est une maquette d'étude du réseau local industriel CAN ayant pour support les commandes au volant au volant d'une automobile (commande des feux et de l'essuie-glace).

Il est décliné en 2 sous systèmes :

- CAN 01 A : maquette équipée, signalisation éclairage,
- CAN 01 B : maquette commande et régulation de vitesse de moteur d'essuie-glace

➤ COMPOSITION :

Le pack CAN 01 A est une option constituée d'un support en PVC sur lequel sont implantées les cartes CAN d'entrées sorties et leur bus. C'est une option du pack de base EID210B (voir la documentation correspondante pour tous les détails techniques).

Il inclut :

- Une carte industrielle (commercialisée par ATON SYSTEME) qui réalise l'interface CAN, elle permet le dialogue avec :
- Une carte d'acquisition 8 entrées TOR (des microswitchs implantés sur les cartes permettent de simuler le commodo),
- Quatre cartes de 4 sorties TOR (des LEDs implantées sur les cartes de sortie permettent de simuler et visualiser l'état des filaments des blocs optiques),

L'ensemble est livré avec pack logiciel éditeur assembleur débogueur, compilateur C et C++ et un lot de TP complets et très progressifs. Les fichiers sources SRC, C et C++ sont fournis sur le CDROM d'installation.

Le CAN 01 B est aussi une option du pack EID210B, il comprend :

- Une carte industrielle (commercialisée par ATON SYSTEME) qui réalise l'interface CAN, elle permet le dialogue avec :
- Une carte commande et régulation de vitesse moteur au travers du CAN, le moto-réducteur codeur, l'ensemble pare-brise,

- Une carte d'entrée (Lecture d'une grandeur analogique, un potentiomètre et des microswitchs permettent de simuler le commodo essuie-glace).

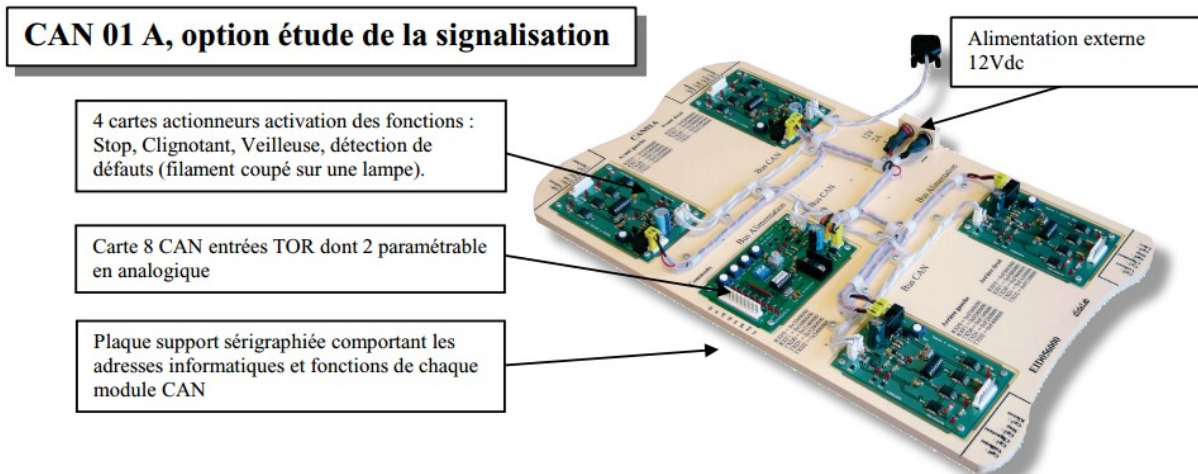


Figure 9: maquette CAN 01A de DIDALAB (www.didalab.fr)

CAN 01 B, option étude de la régulation de vitesse d'un moteur CC

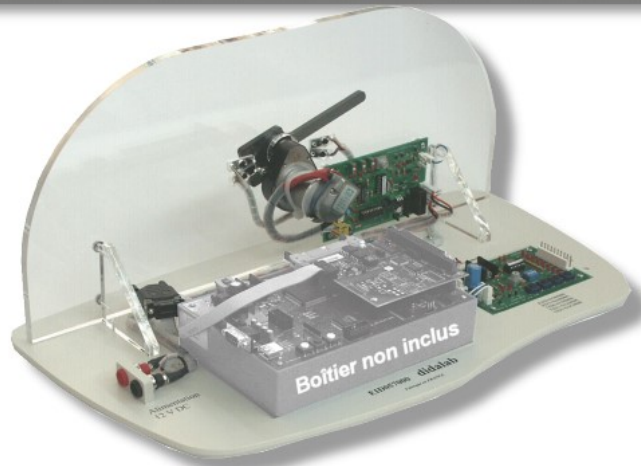


Figure 10: maquette CAN 01B de DIDALAB (www.didalab.fr)

CONCLUSION

Au-delà d'une brève présentation des réseaux locaux industriels, de la notion de supervision et des bus de terrain, nous avons fait l'état des lieux sur quelques maquettes didactiques existantes dans le cadre de l'étude des réseaux locaux industriels. Après une comparaison faite entre les différents bus de terrain présentés, nous avons choisi deux bus de terrain qui seront utilisés dans le cadre de la mise en œuvre de notre maquette. Le chapitre suivant portera

sur la description générale de ces bus ainsi que la description du matériel qui sera utilisé pour une application de ceux-ci à travers un cahier de charge bien défini.

Chapitre

2

CHAPITRE 2: MATERIELS ET METHODES

Aperçu

2.1 Présentation des bus de terrain CAN et MODBUS

2.2 Application des bus de terrain CAN et modBus: maquette didactique des feux tricolores et supervision du trafic routier

INTRODUCTION

Ce chapitre met en exergue les différents matériels, méthodes et logiciels utilisés dans le cadre de la conception de notre maquette. Il est structuré en deux principales parties : la présentation générale des bus de terrain CAN et ModBus choisis pour la communication et une application de ces deux bus à travers un cahier de charge.

I. Présentation des bus de terrain CAN et MODBUS

1) Le Bus CAN

1.1. Principe de fonctionnement

Le bus CAN est un bus de communication qui permet d'effectuer des échanges de données entre plusieurs nœuds ou ECU. Chaque nœud ou ECU peut communiquer avec les autres. La transmission des données s'effectue sur une paire filaire par émission différentielle : une mesure de la différence de tension entre les deux lignes (CAN H et CAN L) est réalisée. La ligne du bus doit se terminer par des résistances de 120 ohms à chacun des bouts. Les communications sont réalisées avec des paquets de messages et la vitesse de transmission peut atteindre jusqu'à 1Mb/s (Pour le Can High Speed).

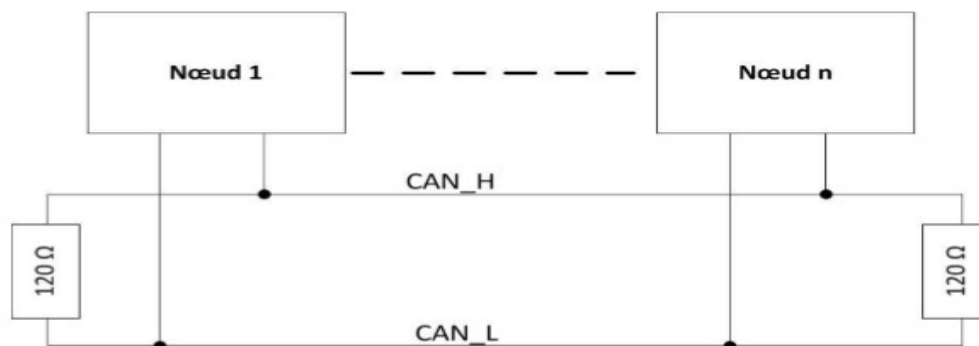


Figure 11: principe de fonctionnement du bus CAN

1.2. Intérêts du bus CAN

Le principal intérêt du Bus CAN est la réduction des coûts. On peut classer ces intérêts en trois catégories :

Réduction des coûts initiaux:

- Un seul câble nécessaire pour tous les équipements au lieu d'en utiliser un par équipement,

- Réutilisation du câblage analogique existant dans certains cas,
- Réduction du temps d'installation,
- Réduction du matériel pour l'installation du système.

Réduction des coûts de maintenance:

- Moins de maintenance due à la fiabilité accrue du système,
- Maintenance plus aisée: réduction du temps de dépannage, localisation des pannes possibles avec des diagnostics à distance,
- Flexibilité de l'extension à un nouveau bus de terrain,
- Simplifications des raccordements.

Performances globales accrues :

- Amélioration de la précision : Il n'y a pas d'erreurs de distorsion ni de réflexions dû à la précision du signal, ce qui n'est pas le cas sur un signal analogique
- Les données et mesures sont généralement disponibles à tous les équipements de terrain,
- Communications possibles entre 2 équipements sans passer par le système de supervision.

1.3 Situation par rapport au modèle OSI

Le Bus CAN étant un protocole réseau, il doit répondre à la norme du modèle OSI qui définit en 7 couches les fonctionnalités nécessaires à la communication et l'organisation d'un protocole en réseau. Les couches (et sous-couches) du modèle OSI liées au CAN sont :

- La couche application (7)
- La couche liaison de données (2)
 - Sous-couche MAC (Medium Access Control)
 - Sous-couche LLC (Logical Link Control)
- La couche physique (1)
 - Sous-couche PLS (Physical Signalling)
 - Sous-couche PMA (Physical Medium Attachment)
 - Sous-couche MDI (Medium Dependant Interface)

❖ Couche applicative

Cette couche est vide, mais sera complétée par l'utilisateur en fonction de l'application à laquelle sera affecté le Bus CAN.

❖ Couche liaison de données

Cette couche va fournir les moyens nécessaires à l'établissement, la libération et au maintien des connexions entre les différentes entités du bus. Elle est aussi en charge de corriger les erreurs du premier niveau.

✓ La sous-couche MAC est en charge de :

- ✚ La mise en trame du message,
- ✚ L'arbitrage,
- ✚ L'acquiescement,
- ✚ La détection des erreurs,
- ✚ La signalisation des erreurs.

✓ La sous-couche LLC est en charge de :

- ✚ Filtrer les messages,
- ✚ Notifier les surcharges,
- ✚ Procéder au recouvrement des erreurs.

❖ Couche physique

La couche physique va définir la façon dont le signal est transmis. Elle assure le transfert physique des bits entre chaque nœud, en accord avec toutes les propriétés du système (électriques, électroniques...). A noter que la couche réseau doit être la même pour chaque nœud. Elle remplit les fonctions suivantes :

- ✚ Synchronisation des bits,
- ✚ La représentation des bits (timing, codage...),
- ✚ Définit les niveaux électriques des signaux, - Définit le support de transmission.

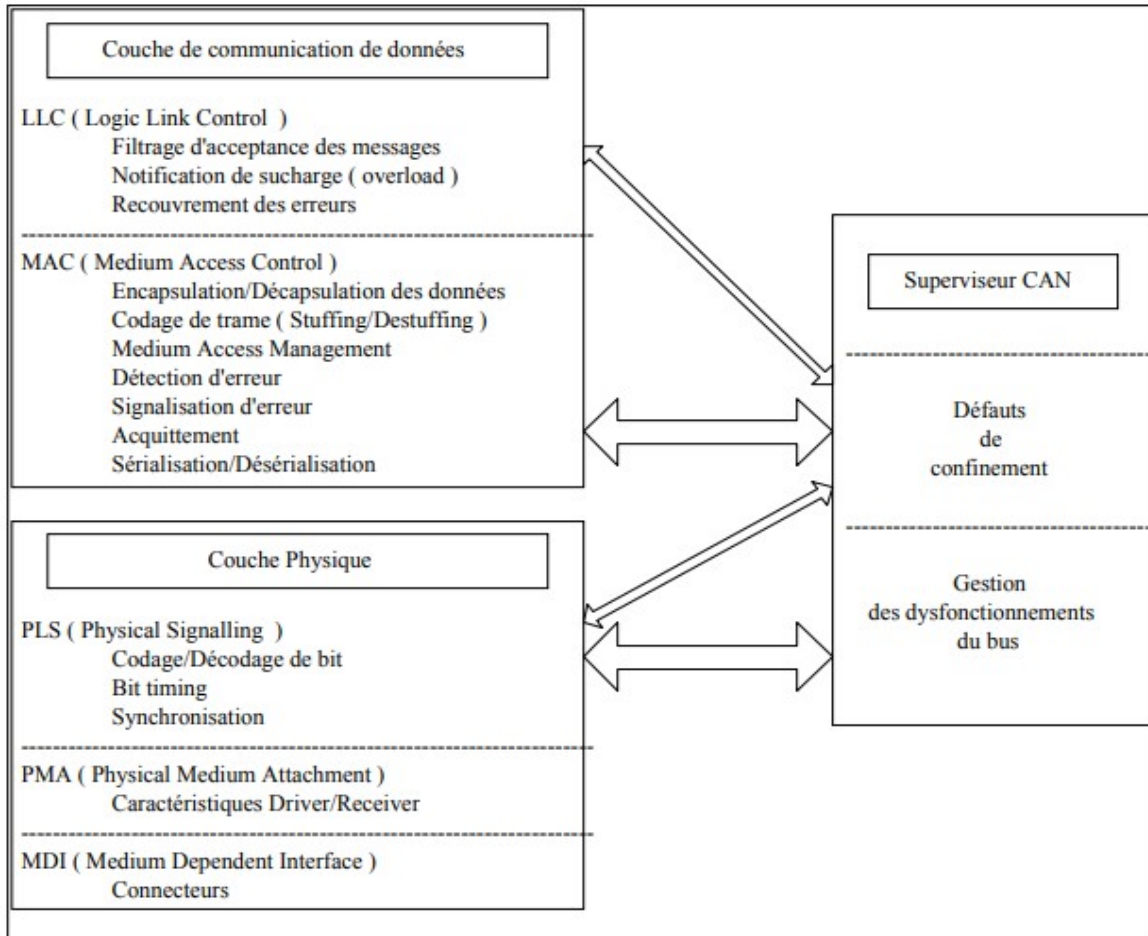


Figure 12: application du CAN au modèle OSI (BOSCH, " Norme CAN")

1.4 Support de transmission

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

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

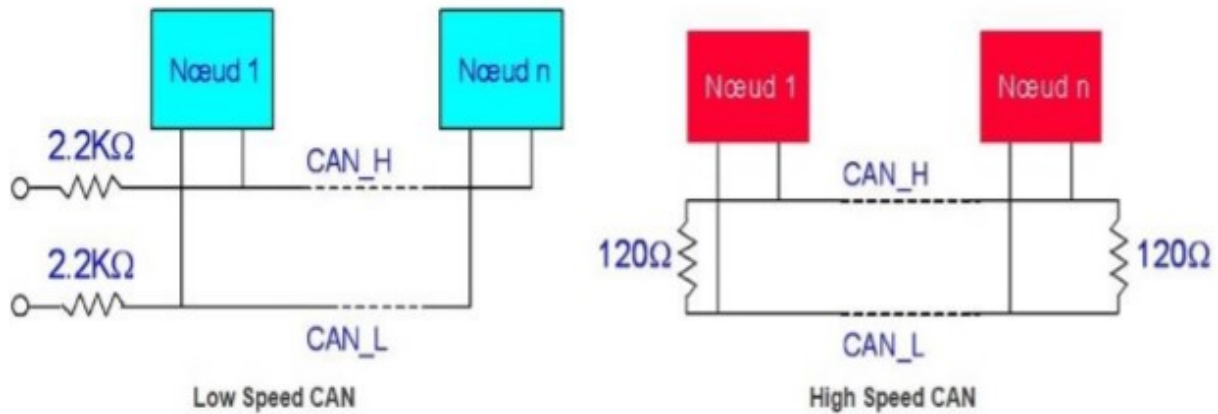


Figure 13: schéma de câblage du bus CAN

Le fait de transmettre en paire différentielle permet de supprimer les parasites qui peuvent être induits sur les lignes de communication. Il existe trois principaux types de transmission possibles en CAN :

- CAN Low Speed,
- CAN High Speed,
- CAN FD (Flexible Dat)

Tableau 3: type de transmission en CAN

	CAN Low Speed	Can High Speed	CAN FD
Débit	125 kb/s	De 125 kb/s à 1 Mb/s	De 500 kb/s à 4 Mb/s
Nombre de nœuds	2 à 20	2 à 30	2 à 30
Courant de sortie	> 1 mA sur 2,2kΩ	25 à 50 mA sur 60Ω	25 à 50 mA sur 60Ω
Niveau dominant	CAN H = 4V CAN L = 1V	CAN H = 3.5V CAN L = 1.5V	CAN H = 3.5V CAN L = 1.5V
Niveau récessif	CAN H = 1.75V CAN L = 3.25V	CAN H = CAN L = 2.5V	CAN H = 2.5V CAN L = 1.5V
Caractéristiques du câble	30 pF entre CAN H et CAN L	Résistance de 120Ω aux deux extrémités du bus entre CAN H et CAN L (=60Ω)	Résistance de 120Ω aux deux extrémités du bus entre CAN H et CAN L (=60Ω)
Tensions d'alimentation	5V	5V	5V

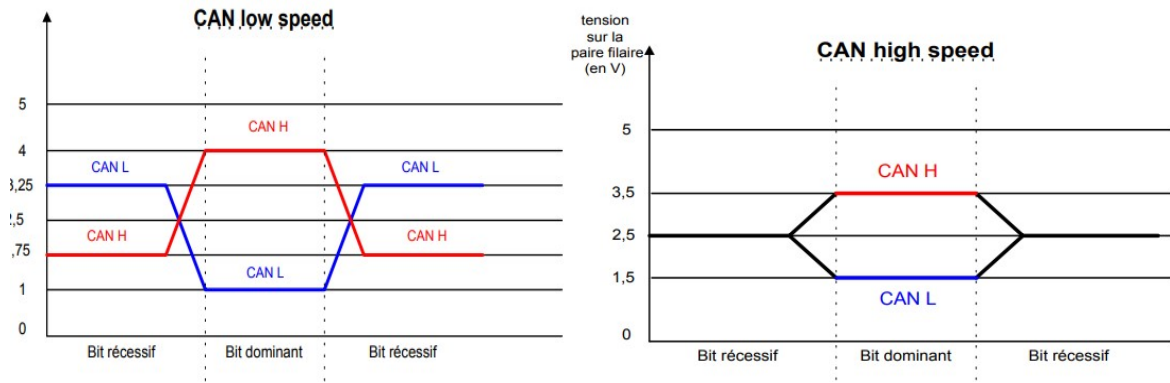


Figure 14: Niveaux de tension en Low Speed & High speed ((BOSCH. Norme CAN)

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,
- Adapter l'impédance en continu de la ligne.

1.5 Effet des longueurs de câble

La longueur du bus dépend des paramètres suivants :

- Le temps de propagation sur les lignes physiques du bus
- La différence due au quantum à cause du temps de propagation défini précédemment. Cela est dû aux différences de cadencement des oscillations des nœuds,
- La variation de l'amplitude du signal en fonction de la résistance du câble et de l'impédance d'entrée des nœuds.

Pour faire fonctionner le Bus CAN avec une longueur de câble supérieure à 200 mètres, il est nécessaire d'utiliser un optocoupleur. Dans le cas où la longueur serait supérieure à 1 kilomètre, il est nécessaire d'utiliser des systèmes d'interconnexion. Un répéteur qui permet de régénérer un signal ou des ponts qui relient des réseaux locaux de même type peuvent être utilisés. Les modules connectés au Bus CAN doivent pouvoir supporter un débit supérieur à 20kbit/sec.

Tableau 5 : propagation dans les câbles

Débit (kb/s)	Longueur (mètres)	Longueur d'un bit (µs)
10	5 000	100
20	2 500	50
62.5	1 000	16
125	500	8
250	250	4
500	100	2
800	50	1,25
1 000	30	1

Le signal qui correspond à un bit émis par un nœud se propage à une vitesse de 200000 km/s sur les lignes électriques et optiques. La plus longue durée de propagation est celle où un bit doit parcourir le bus d'une extrémité à l'autre (on note cette durée t_{bus}). Afin de ne pas créer de conflits entre les nœuds, le temps nominal d'un bit (tn_{bit}) doit être égal à deux fois le t_{bus} .

$$tn_{bus} = 2 * t_{bus} \text{ or } t_{bus} = \text{longueur_du_bus}/200000$$

Sachant que le débit du réseau (Débit_du_réseau) = $1/tn_{bus}$, on obtient la relation suivante :

$$\text{Débit_du_réseau} = 1/tn_{bus} = \frac{1}{2*t_{bus}} = \frac{1}{2*(\frac{\text{longueur_du_bus}}{200000})}$$

1.6 Le Non-Return to Zéro

Dans le cas du Bus CAN, c'est le codage NRZ qui est employé (Non-Return to Zéro). Concrètement, pendant toute la durée de génération du bit, le niveau reste constant, qu'il soit récessif ou dominant. Tous les schémas de trame qui suivent seront représentés de cette manière. Cette méthode de codage possède un inconvénient majeur. Lors de la transmission d'une longue série de bit du même niveau, cela peut entraîner un problème de synchronisation. Pour ce type de séquence, on préférera le codage Manchester qui consiste à effectuer un changement de niveau pour chaque bit transmis. Dans le cas du Bus CAN, on utilise le procédé du « bit stuffing ».

1.7 Trames CAN

Il existe quatre types de trames et un intervalle de temps dans le Bus CAN :

- La trame de donnée : elle est générée par un nœud dit « producteur » qui désire transférer des données, ou comme réponse à la requête d'un autre nœud.
- La trame de requête : elle est générée par un nœud dit « consommateur » ou demandeur de données.
- La trame d'erreur : elle est générée par n'importe quelle entité du bus lors de la détection d'une erreur.
- La trame de surcharge : elle est utilisée pour demander un laps de temps supplémentaire entre les Data Frames ou les Remote Frames lorsque celles-ci sont successives.
- L'intervalle de temps : les Data Frames et Remote Frames sont séparées temporairement par une Interframe.

a. Trames de données

Une trame de donnée se décompose en sept champs différents :

- le début de trame SOF, 1 bit dominant,
- le champ d'arbitrage, 12 bits,
- le champ de contrôle, 6 bits,
- le champ de données, 0 à 64 bits,
- le champ de CRC, 16 bits,
- le champ d'acquittement, 2 bits,
- le champ de fin de trame EOF (End Of Frame), 7 bits récessifs,

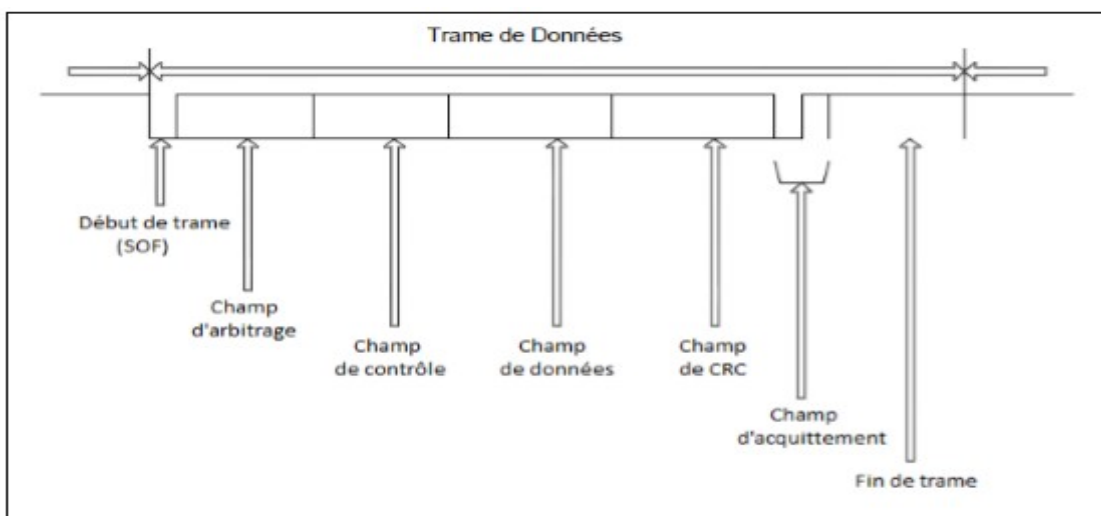


Figure 15: décomposition d'une trame de donnée

b. 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.

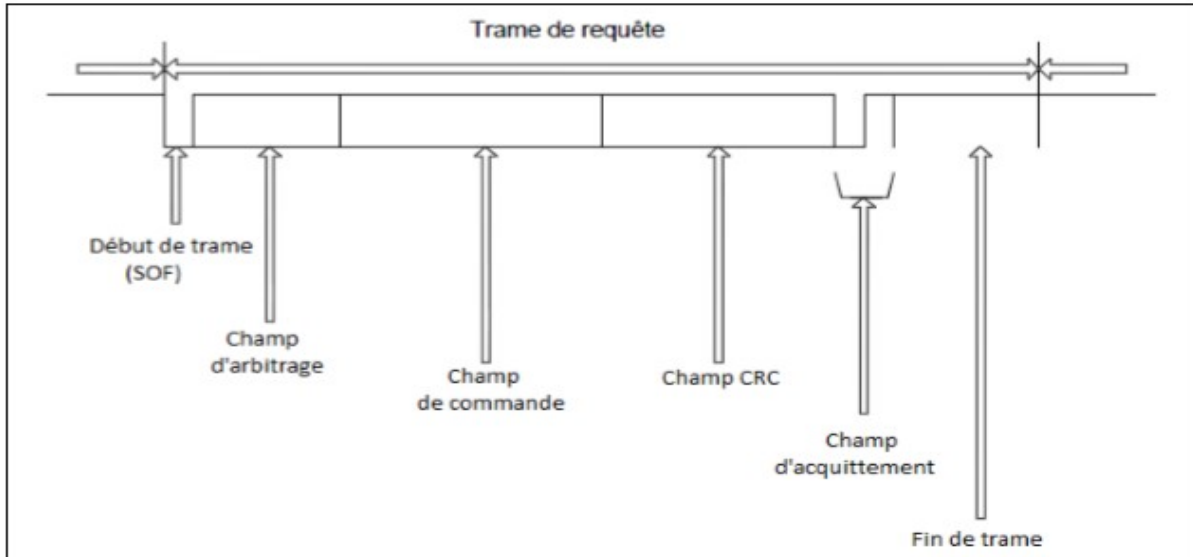


Figure 16: décomposition d'une trame de requête

c. Trames d'erreur

Une trame d'erreur permet de signaler aux autres nœuds CAN la présence d'une erreur (Passive ou Active). La trame d'erreur est constituée de deux champs principaux : le drapeau d'erreur et le délimiteur de champ.

Ci-dessous voici la façon dont se construit une trame d'erreur :

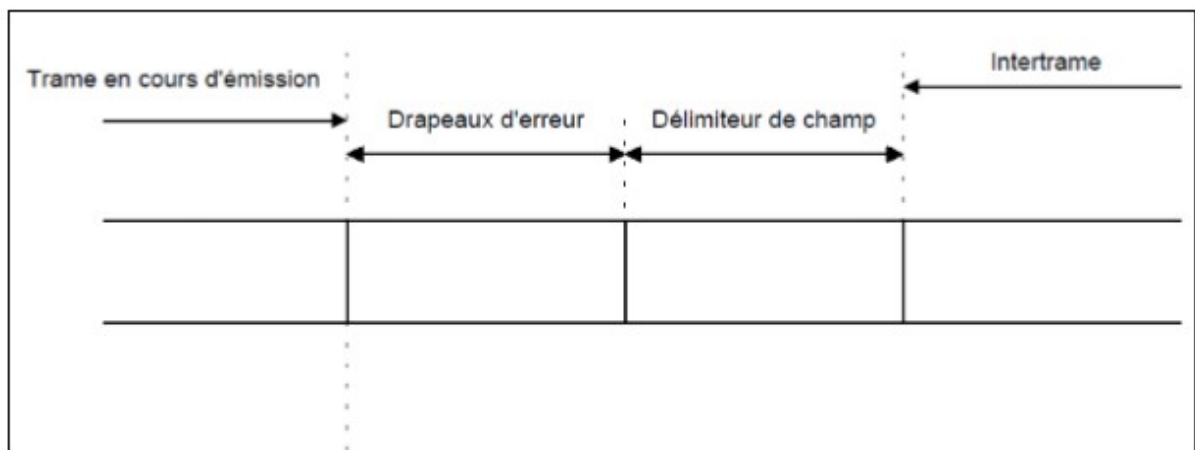


Figure 17: constitution d'une trame d'erreur

d. 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 certain temps avant d'accepter la prochaine trame de données ou de requêtes,
- Lorsqu'un nœud détecte un bit dominant pendant l'inter-trame (3 bits récessifs entre les trames),

Dans le but de ne pas encombrer le bus indéfiniment, seules deux trames de surcharge peuvent être générées de manière consécutive. Elle se compose de deux champs : le champ des flags de surcharge et le champ délimiteur de surcharge.

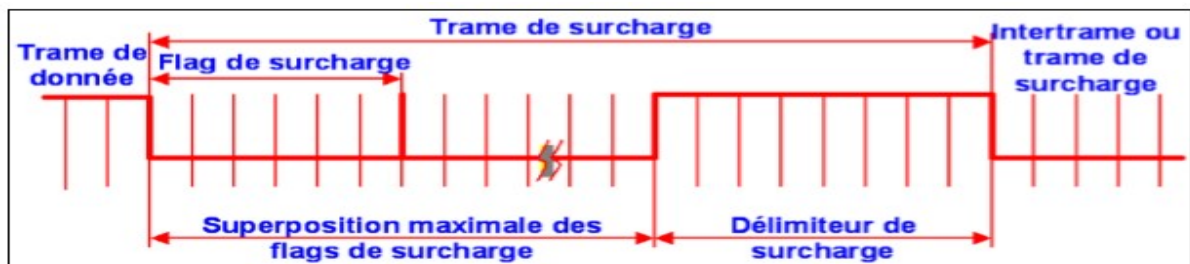


Figure 18: décomposition de la trame de surcharge

1.8 Gestion des erreurs

Afin de comprendre comment se fait la gestion des erreurs, nous allons reprendre toutes les erreurs qui peuvent être rencontrées, les détailler et enfin expliquer comment le Bus CAN gère ces erreurs.

Des erreurs de transmission peuvent perturber le fonctionnement des différents nœuds sur le Bus CAN. Un nœud peut être à l'origine d'une communication perturbée ou impossible sur le BUS. Dans ces cas-là, il existe des méthodes afin de détecter les erreurs de transmissions dans le protocole CAN.

a. Le Bit Error

Lors de l'émission d'un bit sur le bus, une vérification est effectuée afin de surveiller si le niveau émis est le même que celui qui est attendu. S'il n'y a pas de correspondance, il s'agit d'un bit Error. Mais cette erreur n'est pas signalée dans les cas qui suivent :

- Si un bit dominant se situe dans le champ d'arbitrage au lieu d'un bit récessif il s'agit d'une perte d'arbitrage sur le bit dominant.
- Si un émetteur envoie un « flag d'erreur passive » (bit récessif) et reçoit un bit dominant

b. L'erreur de Stuffing

Lorsqu'il y a au-delà de 6 bits consécutifs de même signe sur le bus, il s'agit d'une erreur de Stuffing. Elle est signalée uniquement dans les champs d'identificateurs, de commandes et de CRC. La règle du bit-Stuffing n'est pas déclarée après le CRC. Cette erreur n'est pas signalée dans le champ de fin de trame et le champ d'acquiescement.

c. L'erreur de CRC

Lorsque la valeur reçue par le calcul du nœud récepteur est différente de celle envoyée par le nœud émetteur, il s'agit d'une erreur de CRC.

d. L'erreur d'Acknowledge Delimiter

Lorsque dans le champ d'Acknowledge Delimiter, le nœud récepteur ne voit pas de bit récessif, il s'agit d'une erreur d'Acknowledge Delimiter. Il s'agit de la même erreur pour le CRC Delimiter.

e. L'erreur de Slot Acknowledge

Dans le champ de Slot Acknowledge, lorsque l'émetteur ne lit pas un bit dominant, il s'agit d'une erreur de Slot Acknowledge.

Voici sur la figure 2.10 les différents types d'erreurs ainsi que leurs validités en fonction de l'emplacement dans la trame.

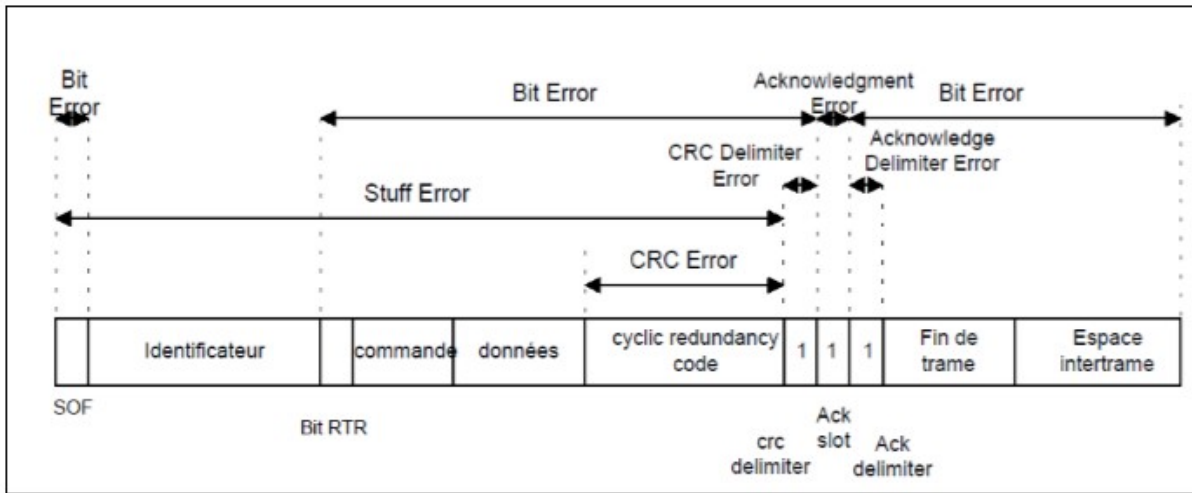


Figure 19: Les sources d'erreur dans la trame CAN

2) Le bus de terrain MODBUS

2.1 Principe général

Le bus est composé d'une station Maître et de stations esclaves. Seule la station Maître peut être à l'initiative de l'échange (la communication directe entre stations Esclaves n'est pas réalisable). Le maître peut s'adresser aux esclaves individuellement ou envoyer un message de diffusion générale à tous les esclaves. Les esclaves renvoient un message (réponse) aux requêtes qui leur sont adressées individuellement. Les requêtes de diffusion générale n'attendent pas de réponses en retour. Un seul équipement peut émettre sur la ligne à tout moment. Le maître gère l'échange et seul lui peut prendre l'initiative. Il interroge chacun des esclaves successivement. Aucun esclave ne peut envoyer de message à moins qu'il ne soit invité à le faire. Le maître répète la question lorsqu'un échange est incorrect et déclare l'esclave interrogé absent si aucune réponse n'est reçue dans un délai donné. Si un esclave ne comprend pas un message, il envoie une réponse d'exception au maître. Le maître peut réitérer ou non la requête.



Figure 20: principe de fonctionnement du bus modBus

Le bus ModBus RS485 peut accueillir 32 nœuds : 1 maître et jusqu'à 31 esclaves

2.2 Liaison série RS485

a. Transmission série asynchrone

En environnement industriel on préfère utiliser la transmission Série asynchrone plus simple à mettre en œuvre et moins coûteuse. La ligne peut ne comporter qu'un fil ; on en utilise en général 3: émission ; réception ; masse. Les éléments binaires d'informations (bits) d'un mot ou caractère sont alors envoyés successivement les uns après les autres (sérialisation) au rythme d'un signal d'horloge. Le récepteur effectue l'opération inverse : transformation Série / parallèle à partir de son horloge ayant la même fréquence que celle de l'émetteur.

- Les informations peuvent être émises de façon irrégulière, cependant l'intervalle de temps entre 2 bits est fixe
- Des bits de synchronisation (START et STOP) encadrent les informations de données.

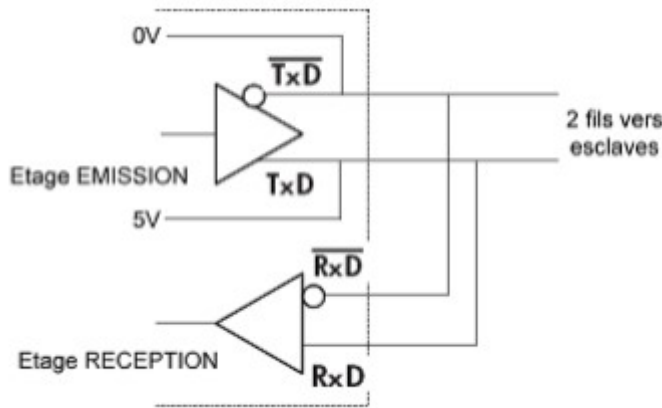
b. Transmission série synchrone

Pour une transmission synchrone, c'est la réception du signal d'horloge sur un fil séparé ou contenu dans les données (code Manchester) qui lance l'exploitation des données par le récepteur.

- Les informations sont transmises de façon continue.
- Un signal de synchronisation est transmis en parallèle aux signaux de données.

c. Câblage liaison ModBus RS485

RS485 est une liaison série, de type asynchrone, différentielle qui permet un débit élevé (jusqu'à 10 Mégabits/secondes) sur une distance importante (jusqu'à 1200 mètres).



Elle dispose de 2 bornes d'émission polarisées notées Tx(+), Tx(-) ou $T \times D$ et de 2 bornes de réception polarisées notées Rx(+), Rx(-) ou $R \times D$.

Figure 21: Cablage liaison ModBus RS485

d. Liaison symétrique (ou différentielle)

Une liaison symétrique comporte deux conducteurs actifs par sens de transfert. L'émetteur possède un amplificateur différentiel qui va transmettre les états logiques à la double ligne de transmission sous forme de deux tensions $V+$ et $V-$ ou $V-$ et $V+$ selon le niveau logique. Le récepteur est un montage à amplificateur opérationnel, il n'est donc concerné que par la différence de tension entre les deux fils de ligne

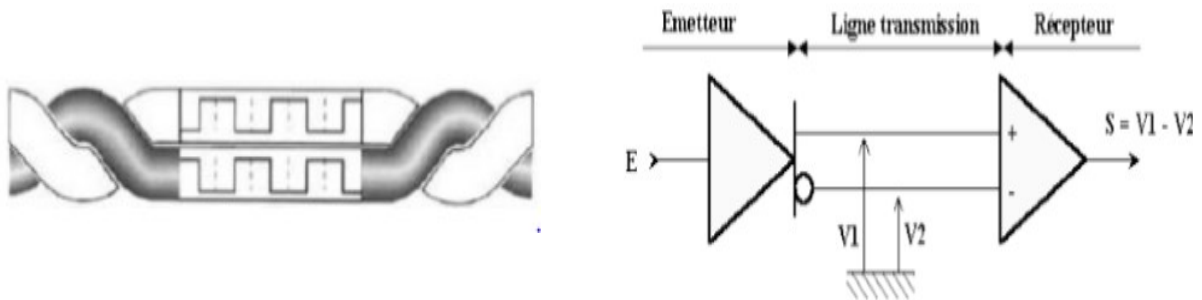


Figure 22 : liaison symétrique

La liaison symétrique ou différentielle permet de transmettre des données sur de grandes distances à des vitesses élevées. Elle est peu sensible aux parasites induits, ceux-ci affectent les deux fils de la ligne et se trouvent inhibés par l'entrée différentielle du récepteur.

E	V1	V2	S	E	V1	V2	Parasites	S = V1 - V2
0	-V	+V	-2V	0	-V	+V	Vp	$(-V + Vp) - (+V + Vp) = -2V$
1	+V	-V	+2V	1	+V	-V	Vp	$(+V + Vp) - (-V + Vp) = +2V$

Sensibilité aux champs magnétiques : Les câbles torsadés sont moins sensibles aux champs magnétiques car les tensions induites par les variations de flux s'annulent mutuellement.

2.3 Les types de transmissions

a. Transmission simplex : monodirectionnel

- Unidirectionnelle.
- Les données sont transmises dans un seul sens.
- Ce mode de communication est utilisé quand il n'est pas nécessaire pour l'émetteur d'obtenir une réponse de la part du récepteur. Un circuit électronique comme un capteur qui envoie régulièrement et de manière autonome des données pourra utiliser une liaison simplex.

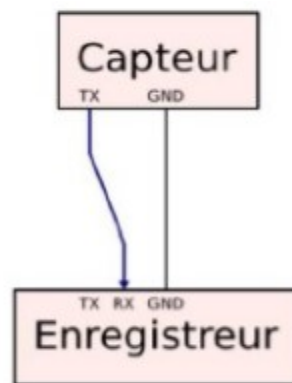


Figure 23: transmission simplex monodirectionnel

b. Transmission half-duplex : bidirectionnel alterné

- Bidirectionnelle.
- La transmission est possible dans les 2 sens, mais pas simultanément.
- Il ne peut y avoir sur la ligne qu'un seul équipement en train d'émettre.
- Dans la communication half-duplex, deux systèmes interconnectés sont capables d'émettre et de recevoir chacun leur tour.

- Il faut que les systèmes communicants soient en mesure de déterminer qui a le droit de parler. Dans le cas contraire, on risque d'avoir une collision (quand les deux systèmes tentent de parler simultanément).

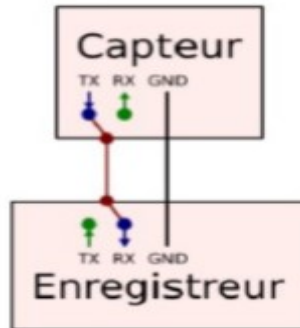


Figure 24: transmission half-duplex bidirectionnelle alternée

c. Transmission full-duplex : bidirectionnel simultané

- Bidirectionnelle.
 - Les données sont reçues ou transmises simultanément dans les 2 sens. Deux systèmes interconnectés sont capables d'émettre et de recevoir simultanément. Pour des raisons liées au coût et à la robustesse, la plupart des réseaux de communication industriels utilisent :
- ✚ Une transmission numérique série asynchrone half-duplex.

L'avantage de ce système de communication par rapport au mode full-duplex est qu'il réduit par deux le nombre de canaux de communication nécessaires.

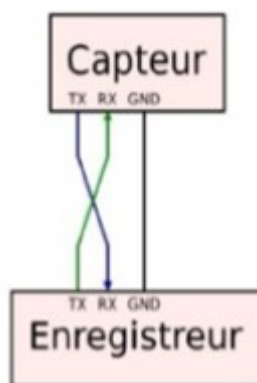


Figure 25: transmission half-duplex série asynchrone

2.4 Les supports physiques

a. Les principaux supports utilisés

Les supports de transmission ou MEDIUMS influent sur :

- ✚ Vitesse,
- ✚ Distance,
- ✚ Immunité électromagnétique.

Mediums les plus utilisés :

- La paire de fils torsadés : Le plus simple à mettre en œuvre, et le moins cher. C'est le plus utilisé sur les bus de terrain.



- Le câble coaxial : Il se compose d'un conducteur en cuivre, entouré d'un écran mis à la terre. Entre les deux, une couche isolante de matériau plastique. Le câble coaxial a d'excellentes propriétés électriques et se prête aux transmissions à grande vitesse. Plus difficile à mettre en œuvre.



- La fibre optique : Ce n'est plus un câble en cuivre qui porte les signaux électriques mais une fibre optique qui transmet des signaux lumineux. Convient pour les environnements industriels agressifs, les transmissions sont sûres, et les longues distances (maxi 10000 mètres).



b. Trame ModBus

Les trames sont de 2 types :

- Mode RTU : les données sont sur 8 bits.

- Mode ASCII2 : les données sont sur 7 bits (les trames sont donc visibles en hexadécimal et il faut deux caractères pour représenter un octet). Ce dernier mode est quasiment tombé en désuétude.

c. Trame MODBUS RTU

Le mode de transmission utilisé est le mode RTU3. La trame ne contient ni octet d'en-tête de message, ni octet de fin de message. Elle est définie de la manière suivante :

START I	Adresse I	Fonction I	Données I	CRC	END
Silence	1 octet	1 octet	n octets	2 octets	Silence

N. esclave	Code fonction	1^{er} paramètre		Autres paramètres	CRC16	
1 octet	1 octet	PF 1 octet	Pf 1 octet	N octets	PF 1 octet	Pf 1 octet

N° esclave : de 1 à 247.

- N° fonction :
 - ✓ 01 : Lecture de n bits de sortie consécutifs,
 - ✓ 02 : Lecture de n bits de sortie consécutifs,
 - ✓ 03 : Lecture de n mots de sortie consécutifs,
 - ✓ 04 : Lecture de n mots d'entrées consécutifs,
 - ✓ 05 : Écriture d'un bit interne ou de sortie,
 - ✓ 06 : Écriture d'un mot interne ou registre...

Il existe 19 fonctions possibles sur ModBus. Ces fonctions sont codées sur 1 octet en hexadécimal, tous les équipements ne supportent pas toutes les fonctions.

- 1er paramètre : Adresse du bit ou du mot adressé.
- 2ème paramètre : Quantité de mots adressés ou valeur du bit ou du mot écrit selon la fonction utilisée.
- Autres paramètres : Données écrites dans plusieurs mots consécutifs.
- CRC16 : Contrôle par redondance cyclique pour détecter les erreurs de transmission.
- La détection de fin de trame est réalisée sur un silence supérieur ou égal à 3 caractères.

d. Trame MODBUS ASCII

Chaque octet composant une trame est codé avec 2 caractères ASCII (2 fois 8 bits).

START	Adresse	Fonction	Données	LRC	END
1 caractère	2 caractères	2 caractères	n caractères	2 caractères	2 caractères

Le mode ASCII permet d'avoir des intervalles de plus d'une seconde entre les différents caractères sans que cela ne génère d'erreurs, alors que le mode RTU permet un débit plus élevé pour une même vitesse de transmission.

II. Application des bus de terrain CAN et ModBus: maquette didactique des feux tricolores et supervision du trafic routier

1. Cahier de charge

L'objectif est de créer une maquette didactique des feux tricolores et de les faire fonctionner de manière synchrone. Cette maquette comporte un carrefour à deux voies (Nord-Sud et Est-Ouest) qui se croisent. L'ensemble est contrôlé par des cartes arduino en communication et pilotées par le logiciel LabVIEW avec supervision à l'aide d'un Interface Homme Machine (IHM). chaque voie est équipée à l'entrée et à la sortie des feux tricolores (voyant rouge, voyant vert, voyant orange) à usage générale (véhicules et piétons). L'intersection des voies est constituée de quatre panneaux disposés sur les extrémités de ses diagonales. Le système est tel que les deux panneaux situés sur la même diagonale reçoivent exactement les mêmes signaux de commande.

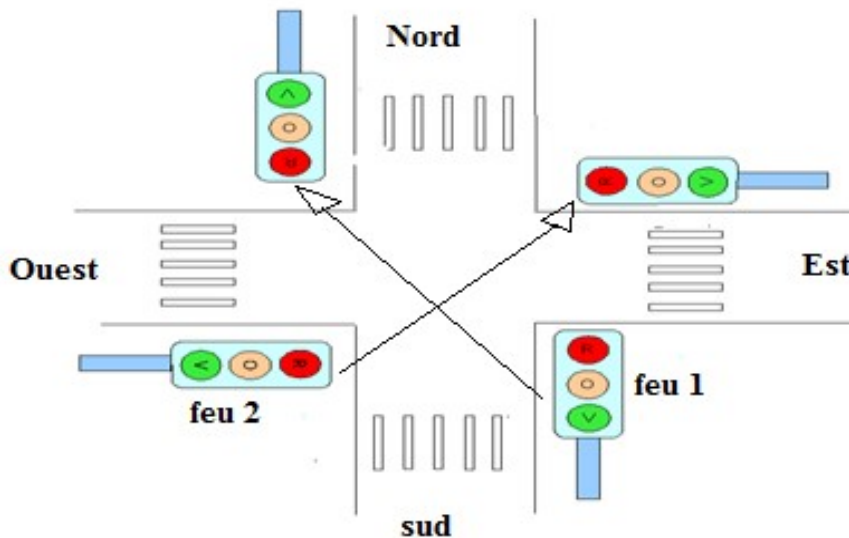
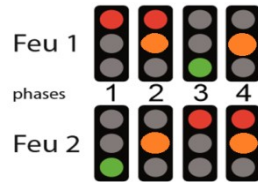


Figure 26: carrefour à deux voies avec croisement

En fonctionnement normal, le système démarre lorsque tout et feux sont éteints et l'on appui sur un bouton poussoir départ cycle (**Dcy**). Pour arrêter le système on appui sur le bouton poussoir arrêt (**Ar**). Les phases de fonctionnement sont les suivants:

La phase 1 dure 10s, la phase 2 dure 5s, la phase 3 dure 10s et la phase 4 dure 5s.



Afin d'assurer un autodiagnostic du système, une partie commande constituée de deux boutons poussoirs est utilisée pour augmenter ou diminuer le temps de fonctionnements de tous les panneaux de façon automatique. Trois LED seront utilisées pour témoigner l'envoi et la réception de la commande.

La communication entre les différents modules (partie opérative, partie commande et l'IHM) sera réalisée à l'aide de deux bus de terrain au choix : le bus ModBus et le bus CAN

2. Matériels et logiciels utilisés

a. Logiciel Arduino (IDE)

Le logiciel Arduino fonctionne sur Mac, Windows et Linux. Il est nécessaire pour coder les programmes et les envoyer sur la carte arduino.il se présente de la manière suivante:



Figure 27: IDE Arduino

b. Le logiciel LabVIEW

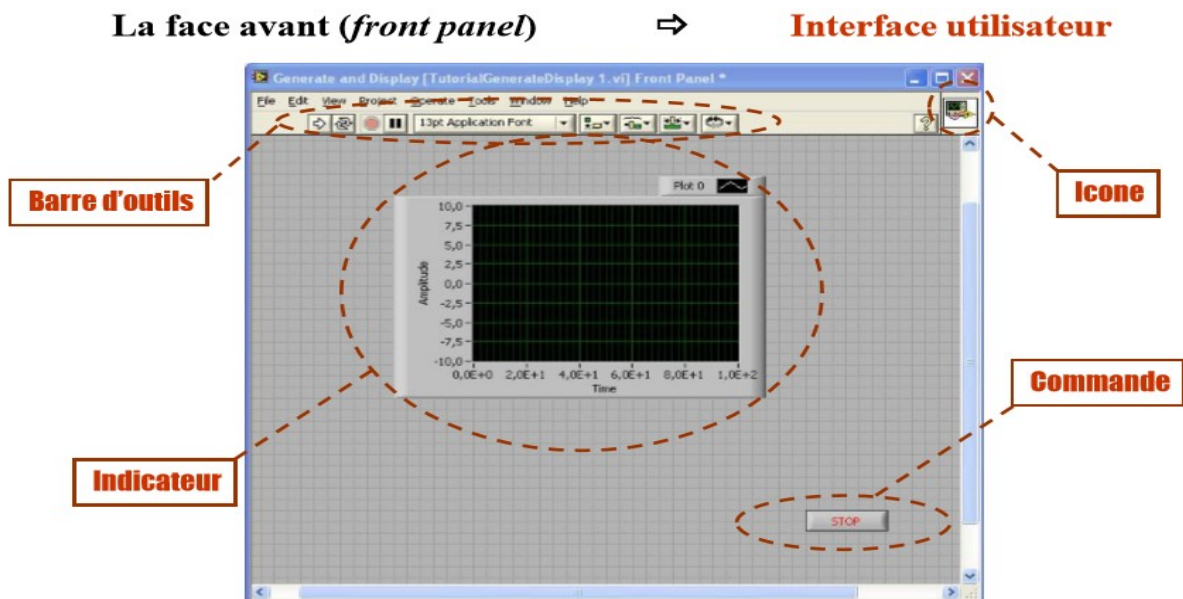


LABVIEW : Laboratory of Virtual Instruments Engineering Workbench, est un logiciel de développement d'applications, comparable à la plupart des systèmes de développement en langage C ou BASIC. Il permet de faire de la programmation graphique pour les mesures et l'automatisation.

L'interface LabVIEW est constituée de deux fenêtres de travail séparées mais associées:

- première fenêtre : face avant (front panel) qui contient les contrôles et les afficheurs de données (Interrupteurs, variateurs, potentiomètres avec curseurs, boîtes de dialogue, compteurs à aiguilles, boîtes de message, graphes....).

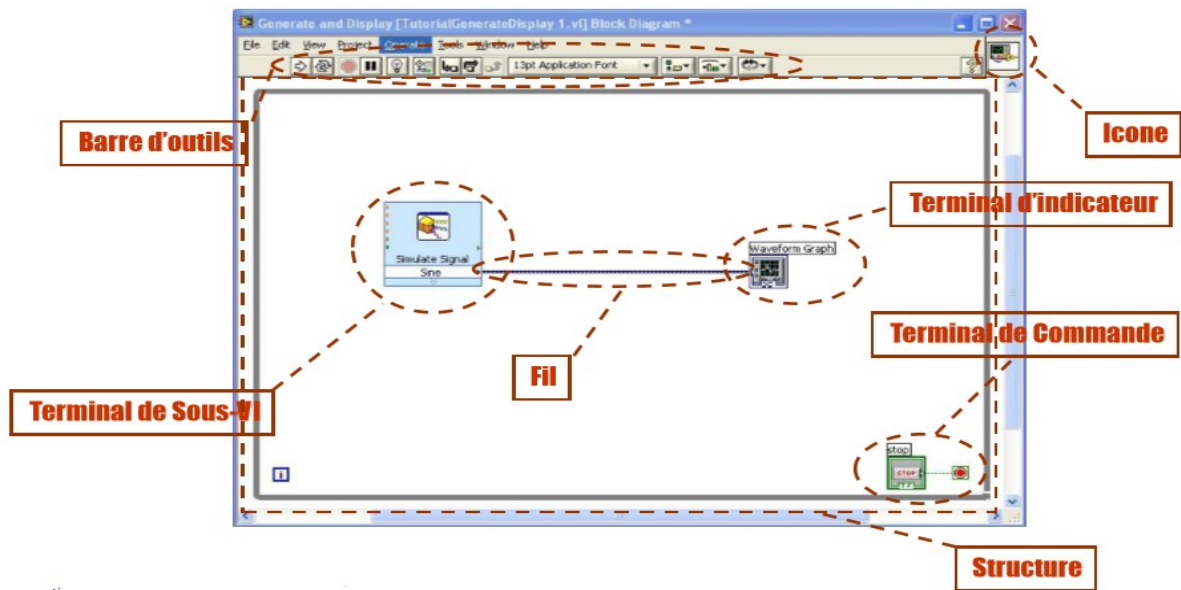
Elle se présente comme suit :



- deuxième fenêtre : diagramme (block diagram) qui contient le programme (Terminaux de commandes, nœuds, constantes, Terminaux d'affichage, des connecteurs, des fils....).

Elle se présente comme suit:

Le diagramme (*block diagram*) ⇒ Interface programmeur



Dans le cadre de notre projet, le logiciel LabVIEW nous permettra de créer un IHM de supervision des feux.

c. La carte Arduino Uno

Arduino fait partie de la famille des platines de développement, et reste par contre l'un des plus abordables et des plus répandus. Une platine de développement est en général un circuit imprimé équipé d'un microprocesseur ou d'un microcontrôleur. Dans le cadre de notre travail, nous nous baserons sur le plus connu: l'Arduino Uno

Le schéma suivant donne la composition d'un Arduino Uno:

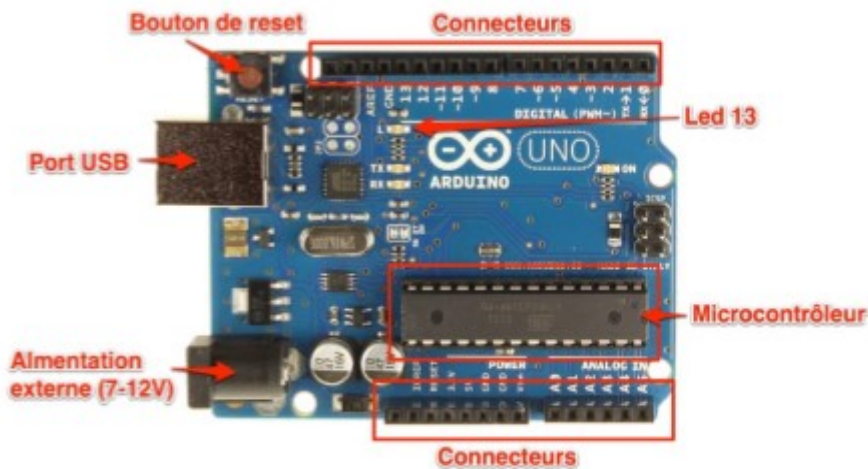


Figure 28: carte Arduino Uno

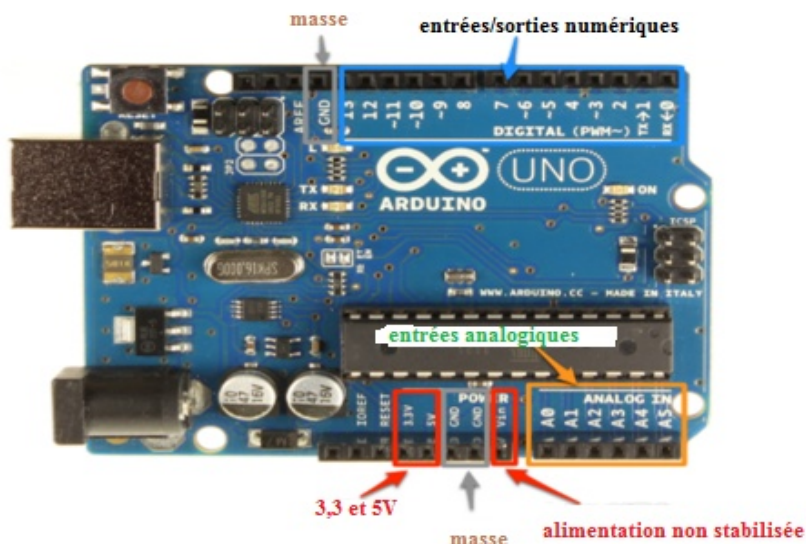
Le microcontrôleur est le cerveau de notre carte. Il va recevoir le programme que nous allons créer et va le stocker dans sa mémoire avant de l'exécuter.

Pour fonctionner, une carte Arduino a besoin d'une alimentation. Le microcontrôleur fonctionnant sous 5V, la carte peut être alimentée en 5V par le port USB ou bien par une alimentation externe qui est comprise entre 7V et 12V. Un régulateur se charge ensuite de réduire la tension à 5V pour le bon fonctionnement.

A part une LED sur la broche 13, la carte Arduino ne possède pas de composants (résistances, diodes, moteurs...) qui peuvent être utilisés pour un programme. Il est nécessaire de les rajouter. Mais pour cela, il faut les connecter à la carte. C'est là qu'interviennent les connecteurs, aussi appelés broches (pins, en anglais).

Exploration des broches Arduino

- 0 à 13 Entrées/sorties numériques
- A0 à A5 Entrées/sorties analogiques
- GND Terre ou masse (0V)
- 5V Alimentation +5V
- 3.3V Alimentation +3.3V
- Vin Alimentation non stabilisée (= le même voltage que celui à l'entrée de la carte)



d. Les diodes

Il est possible de remplacer les ampoules de chaque panneau par des diodes électroluminescentes, aussi appelée LED. Elle a la particularité de ne laisser passer le courant

électrique que dans un sens. Dans notre travail, nous utiliserons 24 LED (8 rouges, 8 oranges et 8 vertes) pour les panneaux et trois LED rouges pour la commande. La description d'une diode électroluminescence est donnée par la figure suivante :

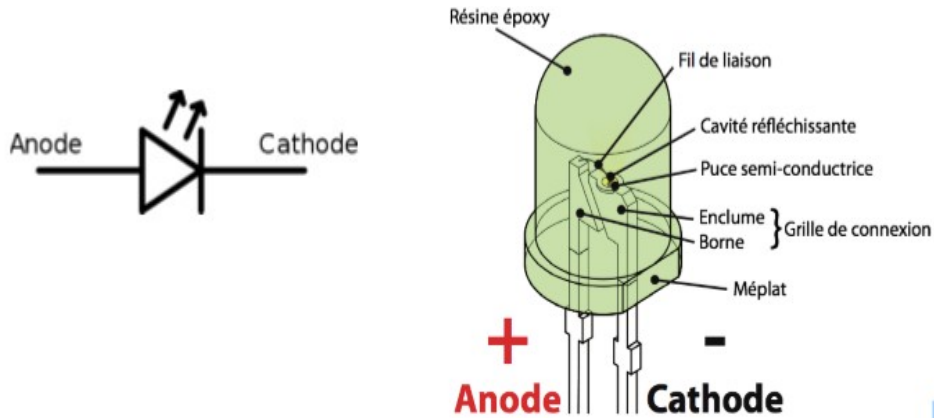


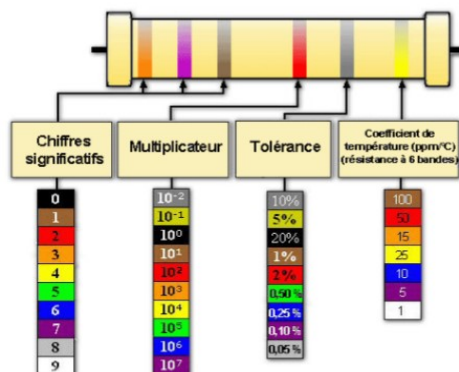
Figure 29: schémas de la LED

Le courant électrique ne peut traverser la diode que dans le sens de l'anode vers la cathode. On reconnaît l'anode, car il s'agit de la broche la plus longue. Lorsque les deux broches sont de même longueur, on peut distinguer l'anode de la cathode, par un méplat du côté de cette dernière.

Attention: le courant produit par l'Arduino est trop important pour y brancher directement une LED dessus. L'utilisation d'une résistance est obligatoire, pour ne pas griller la LED.

e. Les résistances

Une résistance est un composant électronique ou électrique dont la principale caractéristique est d'opposer une plus ou moins grande résistance (mesurée en ohms: Ω) à la circulation du courant électrique. La valeur de la résistance se mesure en Ohms (Ω) et est déterminée par ses bandes de couleurs qui se présentent comme suit:



Dans le cadre de notre projet nous utiliserons des résistances de 120 à 470Ω pour la protection des LEDs et la réalisation de la terminaison CAN.

f. Réalisation du bus CAN

▪ Synoptique

Le synoptique qui sera utilisé pour la réalisation de la communication par bus CAN est le suivant :

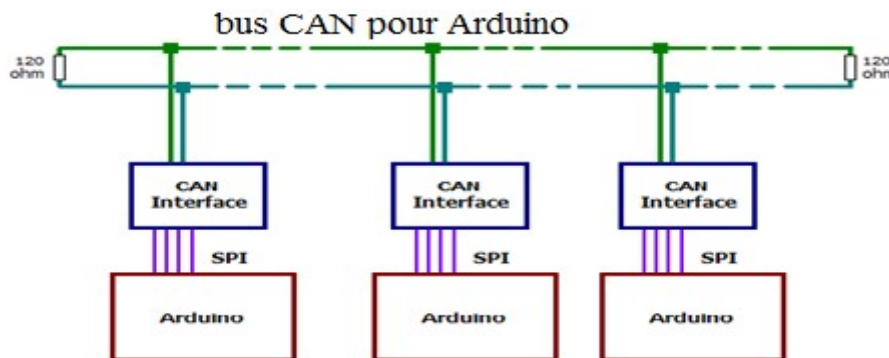


Figure 30: synoptique bus CAN

L'interface utilisée sera le module MCP 2515_CAN qui sera relié à la carte arduino par une liaison de type SPI (Serial Peripheral I interface) pour une communication synchrone.

• Le module MCP 2515_CAN

Cette carte intègre un contrôleur CAN et un émetteur/récepteur et permet d'implémenter un bus CAN standard (2.0A) ou bus CAN étendu (2.0B) avec une vitesse de fonctionnement allant jusqu'à 1 Mbits/s. Elle dispose de deux bornes à vis permettant d'accéder aux lignes CAN-H et CAN-L. Les signaux SCK, MISO, MOSI sont par défaut transmis à la carte shield via le connecteur ICSP. Cette carte se présente de la manière suivante:

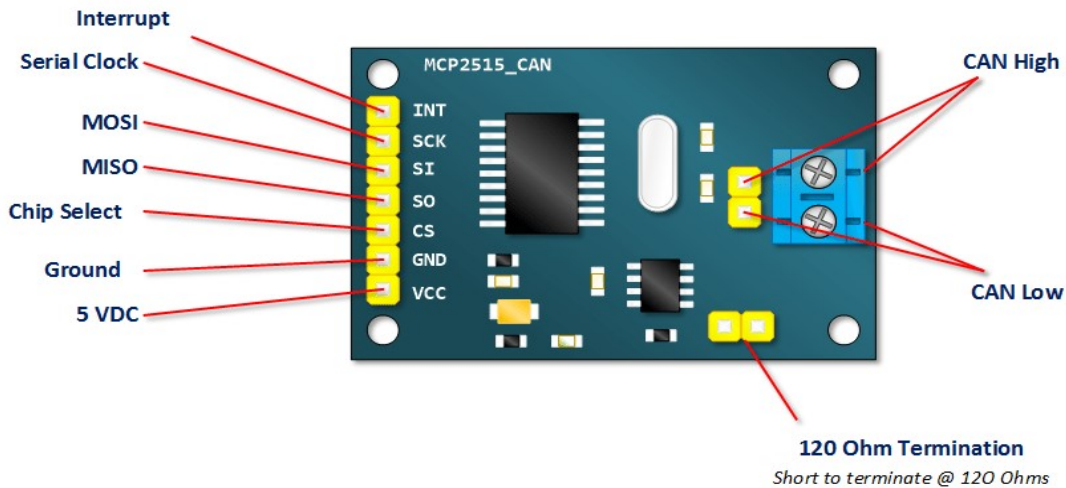


Figure 31: la carte MCP2515-CAN

Le schéma de câblage par défaut entre la carte Arduino et le shield CAN est le suivant :

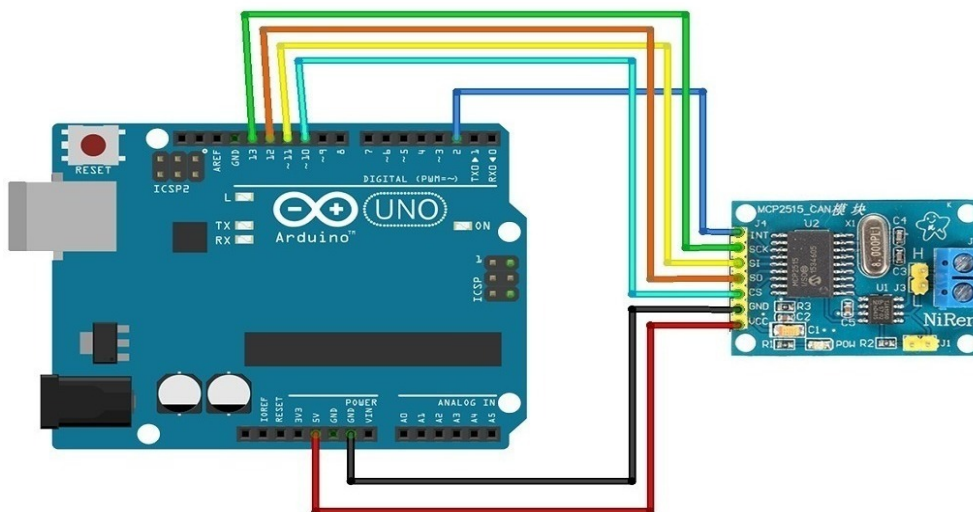


Figure 32: connexion MCP2515_CAN et Arduino

g. Réalisation du bus ModBus

- le module RS485 pour arduino (MAX485)

C'est un module permettant de réaliser une transmission half-duplex entre une application TTL à microcontrôleur sur une ligne de type RS485 jusqu'à 2,5Mbps. sa tension de fonctionnement est de 5V et sa taille est de 44*14(mm). C'est un Protocol industriel largement utilisé pour sa robustesse, sa mise en œuvre facile et ses performances. il est composé d'un circuit MAX485 avec toutes les broches de sorties (RO, RE*, DE et DI) et est idéal pour récupérer une liaison série type RS485 avec une carte arduino. Il se présente comme suit :

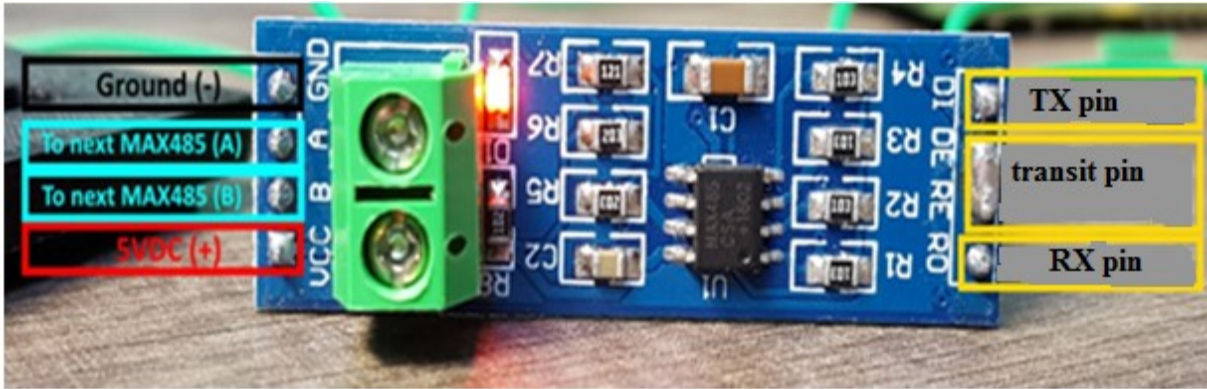


Figure 33: carte MAX485

Le schéma de câblage par défaut entre la carte Arduino et le shield ModBus est le suivant :

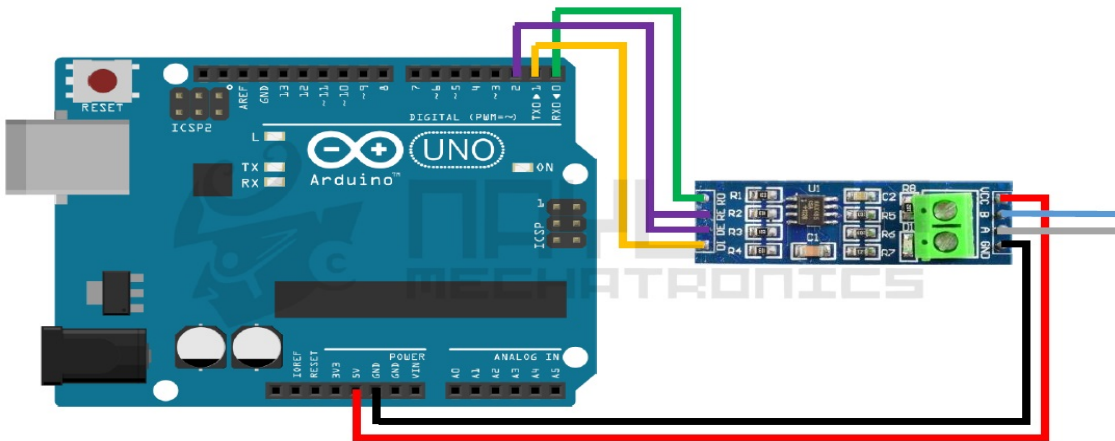


Figure 34: connexion MAX485 Arduino

h. Méthode de mise en œuvre de la maquette

La maquette qui sera mise en œuvre dans le cadre de notre travail comportera quatre blocs à savoir le bloc partie opérative, le bloc de commande, le bloc des bus de terrain et le bloc de supervision. Cette maquette aura pour but principal la communication à travers le bloc des bus de terrain entre les différents blocs restant.

CONCLUSION

Au-delà du matériel et des méthodes que nous allons utiliser dans le cadre de notre conception, nous avons fait l'état des lieux sur les bus de terrain choisis au préalable et avons prévu une application de ces bus à travers un cahier de charge. Le chapitre suivant portera sur la réalisation des différents blocs qui entre dans le cadre de notre maquette et la discussion en ce qui concerne les résultats obtenus.

Chapitre

3

CHAPITRE 3: SIMULATIONS, REALISATION ET DISCUSSIONS

Aperçu

3.1 Descriptions des différents blocs

3.2 Principe de réalisation des différents modules et schémas

INTRODUCTION

Ce chapitre dernier définit les différentes étapes de réalisation de notre maquette et les résultats obtenus afin de pouvoir élaborer une discussion en fonction des objectifs définis au préalable. La conception d'une maquette didactique pour l'étude des RLI comprend l'implémentation de quatre principaux blocs: le bloc de commande, le bloc partie opérative, le bloc bus de terrain et le bloc supervision. L'objectif principal de cette conception étant de pouvoir réaliser une communication entre différents blocs.

3-1) Descriptions des différents blocs

- Le bloc partie opérative est constitué d'un carrefour à deux voies Nord-Sud et Est-Ouest qui se croise et de quatre panneaux situés sur les extrémités des diagonales formés par l'intersection de celles-ci. Chaque panneau est constitué de six LEDs montées trois à trois en série et deux à deux en parallèle pour un éclairage plus considérable. Celui-ci étant protégé par une résistance de 220Ω .
- Le bloc de commande est constitué de deux boutons poussoirs pour l'augmentation et la diminution automatique du temps de commutation entre les différents feux des panneaux de la partie opérative via le module bus de terrain.
- Le bloc bus de terrain est celui qui réalise l'intermédiaire entre les autres blocs et qui permet la communication entre ceux-ci.
- Le bloc supervision est celui constitué d'un ordinateur muni d'un logiciel de supervision (LabVIEW dans notre cas) dans lequel nous avons réalisé un IHM pour la supervision du trafic routier via une communication à travers la carte arduino et les modules bus de terrain.

Il est à noter que le circuit est alimenté par une tension redressé à 5Volt, chaque bloc possède une carte ARDUINO sur lequel sont reliés respectivement le module MCP25-15 CAN à travers une liaison de type SPI pour une communication CAN via les lignes CAN-LOW et CAN-HIGH et le module MAX485 à travers une liaison de type TTL RS485, Half-duplex bidirectionnelle simultanée pour une communication ModBus via les lignes A et B. chaque bloc comporte des connecteurs sous forme de nœuds sur lesquels seront connecté à travers une paire torsadée les autres blocs. Chaque carte sera programmée de façon à établir la fonction réalisée par le bloc concerné.

Le synoptique ci-dessous présente les différents blocs de notre maquette :

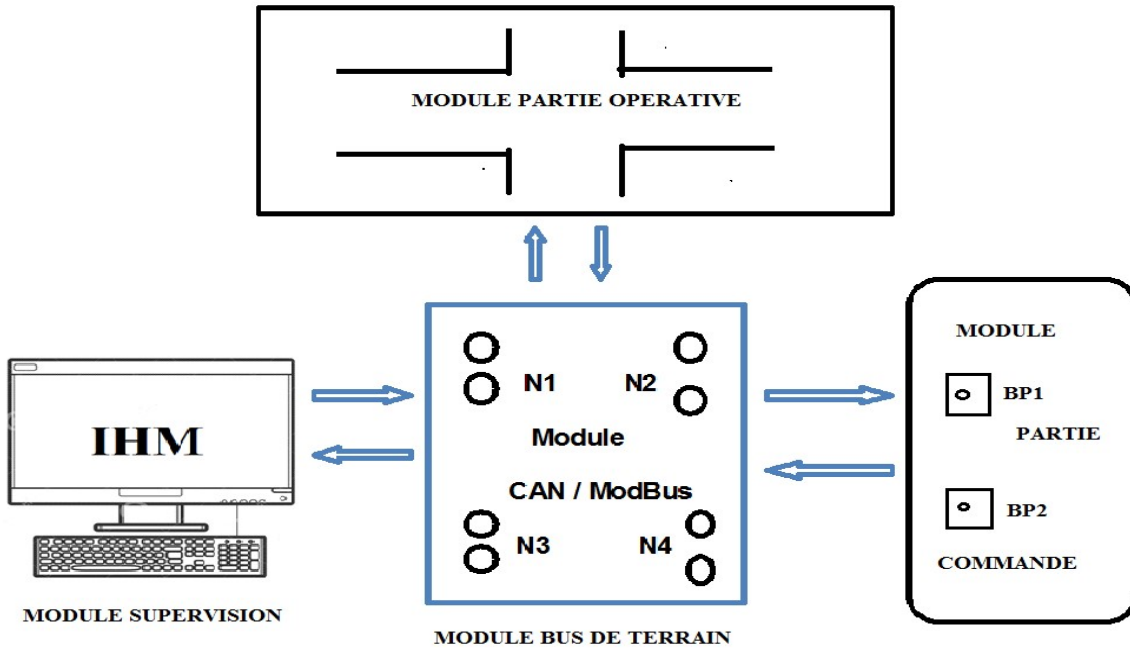


Figure 35: synoptique de la maquette

3-2) Principe de réalisation des différents modules et schémas

1. Le module bus de terrain

Dans ce module la partie bus CAN comporte un contrôleur MCP2515 qui émet le message et un transceiver TJA1050 considéré comme récepteur. Le modèle de coopération utilisé ici est le modèle client/serveur. La partie ModBus comporte un MAX485 qui peut jouer le rôle de maître ou d'esclave en fonction du bloc dans lequel il est utilisé. Le schéma électronique de ce module est donné par la figure suivante.

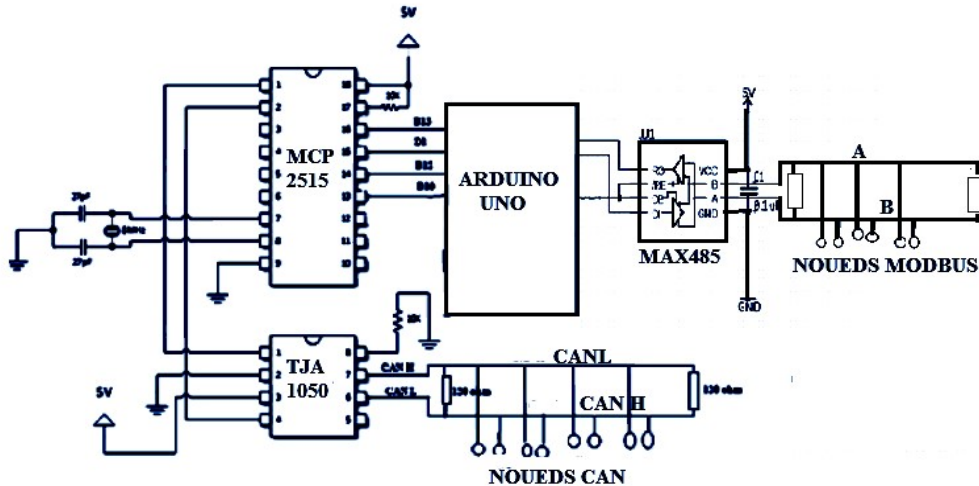


Figure 36: schéma électronique du bloc bus de terrain

2) Le module de commande

Ce module comporte deux boutons poussoirs l'un pour la diminution et l'autre pour l'augmentation du temps de commutation des feux. Le temps de chaque feu va soit augmenter soit diminuer d'une seconde. Un régulateur de tension de type L7805CV sera connecté sur le microcontrôleur ATmega328P pour maintenir la tension à 5V : tension supportable par celui-ci. Sur le schéma suivant seront connectés les modules MCP25-15 CAN et MAX485 pour envoyer la commande à la partie opérative en passant par le module bus de terrain. Trois LED sont utilisé l'une sur le module de commande pour témoigner l'envoi et deux autres sur la partie opérative pour témoigner la réception.

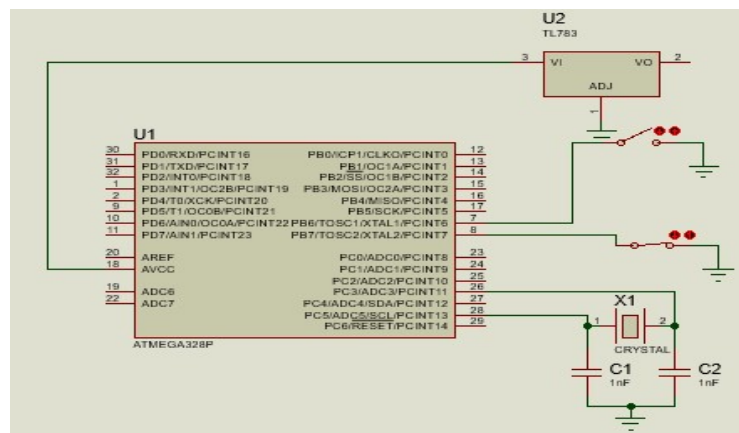


Figure 37: schéma de commande

3) Module partie opérative

Cette partie est celle formée par l'ensemble des organes physiques qui interagissent: ici les panneaux de signalisation. Dans cette partie nous utiliserons comme précédemment un régulateur de tension de type L7805CV qui sera connecté sur le microcontrôleur ATmega328P pour maintenir la tension à 5V. Elle reçoit les ordres de la partie commande à travers les modules MCP25-15 CAN et MAX485 connecter sur le schéma suivant et les exécute.

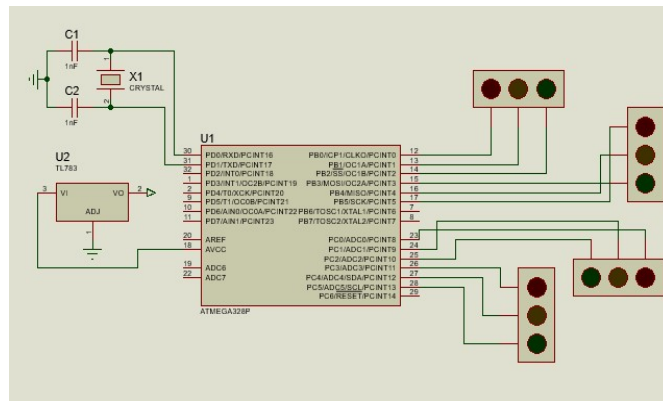


Figure 38: schéma des feux de carrefour

4) Module de supervision

Dans le logiciel de supervision LabVIEW, nous avons réalisé une interface de supervision constituée d'un VI présentant une face avant et une face arrière. Ce VI se présente comme suit :

✓ **La face avant :**

Elle est constituée d'un carrefour, des panneaux de signalisation, de la platine de commande présentant deux modes de fonctionnement (le fonctionnement normal qui définit le temps des feux vert, rouge et orange sur toutes les voies et le fonctionnement personnalisé qui définit le temps des ces feux sur chaque voie), d'une sortie USB pour arduino et d'un bouton d'arrêt. La face avant est donnée par le schéma de la figure suivante:

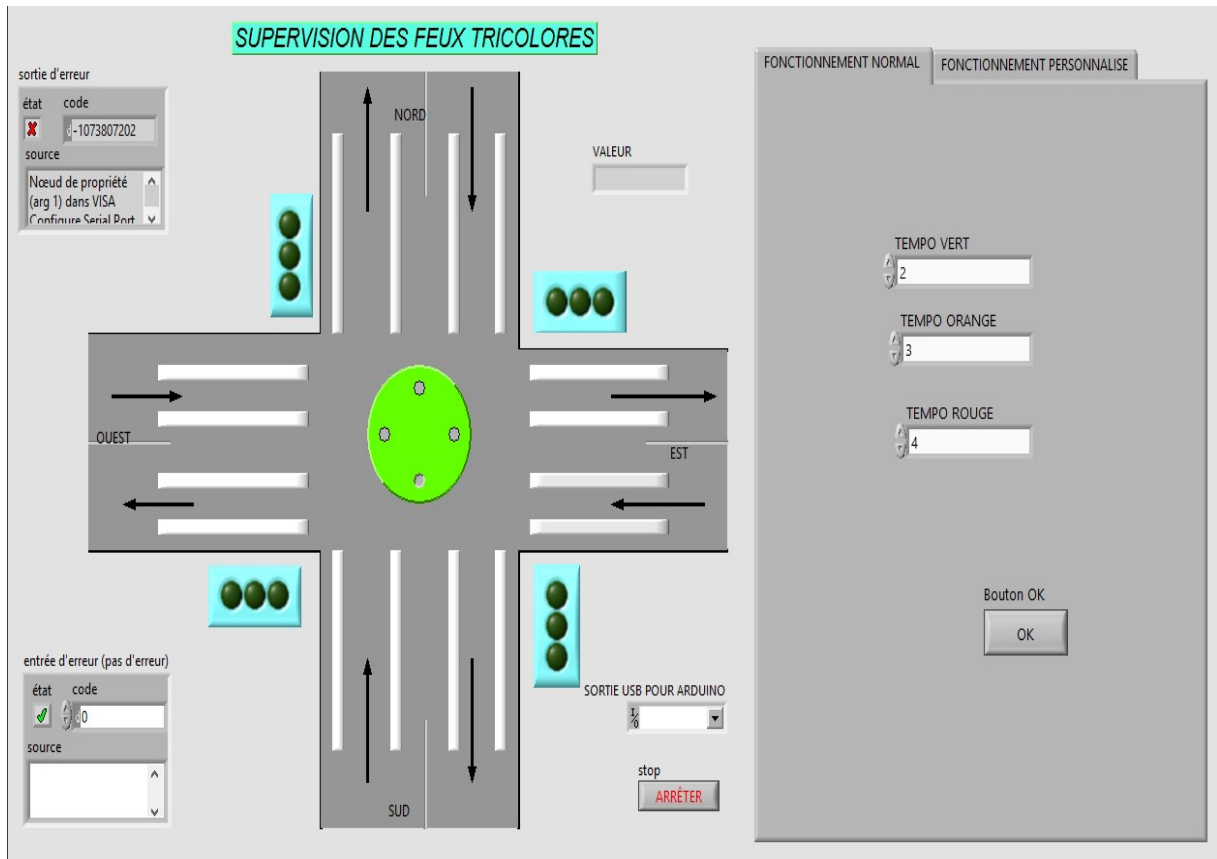


Figure 39: face avant réalisée Labview

✓ **La face arrière :**

C'est celle constituée du diagramme bloc (une programmation graphique) qui a pour rôle de faire fonctionner la face avant. Elle est donnée par la figure suivante:

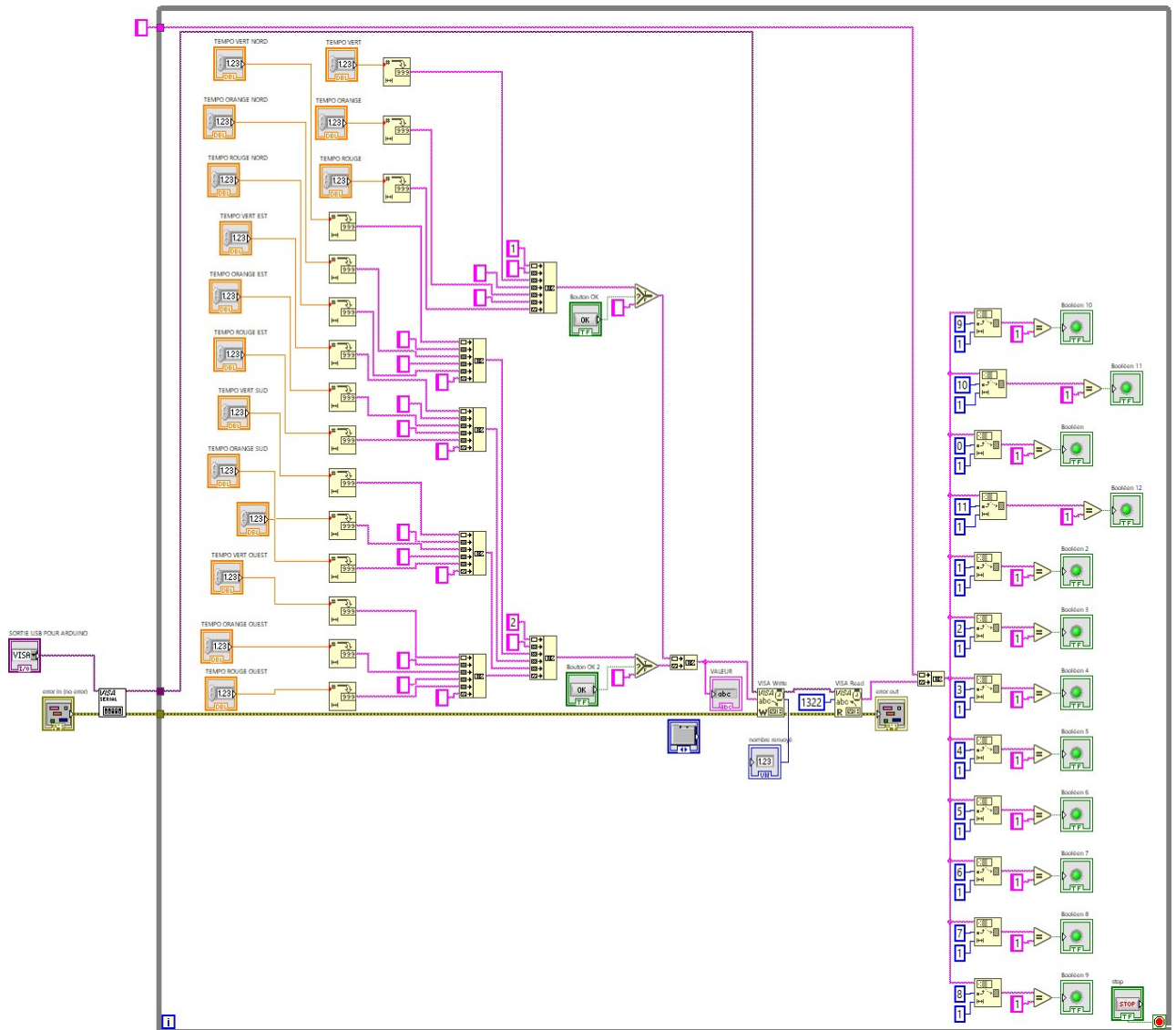


Figure 40: face arrière réalisée LabVIEW

Pour réaliser une interaction entre le module de supervision et la partie opérative à travers le bloc bus terrain constitué par les modules MCP25-15 CAN et MAX485 connectés sur la carte Arduino, nous devrions d'abord faire communiquer notre carte Arduino avec le logiciel LabVIEW.

✓ Communication Arduino LabVIEW

La réalisation de cette communication nécessite l'installation d'un micro logiciel LINX qui va permettre la communication entre les composants d'interface et la plateforme de développement de contrôle intégré (ici Arduino) via le protocole USB et SPI. Nous avons téléchargé le logiciel complémentaire LINX sur le disque dur de notre machine et nous l'avons installé en suivant les instructions d'installation.

Le logiciel d'ajout LabVIEW Maker Hub LINX permet un contrôle direct d'entrées/sorties de notre carte Arduino. Pour terminer l'installation du projet LINX les étapes suivantes ont été nécessaires pour charger le micro logiciel sur la carte arduino :

- Connecter l'Arduino Uno à l'ordinateur à l'aide du câble USB
- Lancer le logiciel LabVIEW
- Cliquer sur **Outils >>Maker Hub >>LINX >> Firmware Wizard**
- Sélectionner le port **COM** associé à notre appareil puis cliquer sur **Suivant**
- Sélectionner la version du micro logiciel (commencer par le micro logiciel série/USB) et cliquer sur suivant.

Remarque : les voyants TX et RX de notre Arduino s'activent pendant quelques secondes. Dans le cas contraire, répétez l'étape précédente.

3.3) Les étapes de réalisation pratique de la maquette finale

1) La soudure

Pour la mise en place de notre maquette, nous avons commencé par souder les LED sur une plaque perforée pour la réalisation de nos panneaux comme indiqué sur la figure suivante :

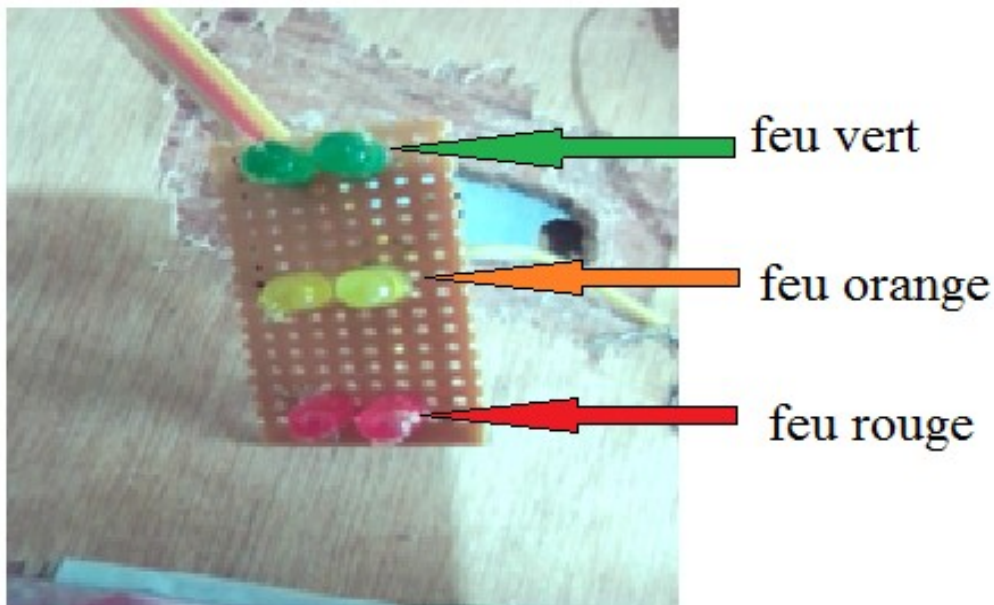


Figure 41: panneau de signalisation

Il faut noter ici que le fait d'associer les LED deux à deux va permettre un éclairage plus considérable. Ces panneaux ont été associés deux à deux pour réaliser la synchronisation entre les feux et répondre ainsi au cahier de charge : les panneaux situés sur la même diagonale reçoivent exactement les mêmes signaux de commande.

Ensuite pour des raisons de cout élevé nous avons simplifié l'utilisation des cartes arduino des blocs de commande et de la partie opérative en utilisant des petits circuits à microcontrôleur ATmega328P soudé comme indiquer sur la figure suivante :

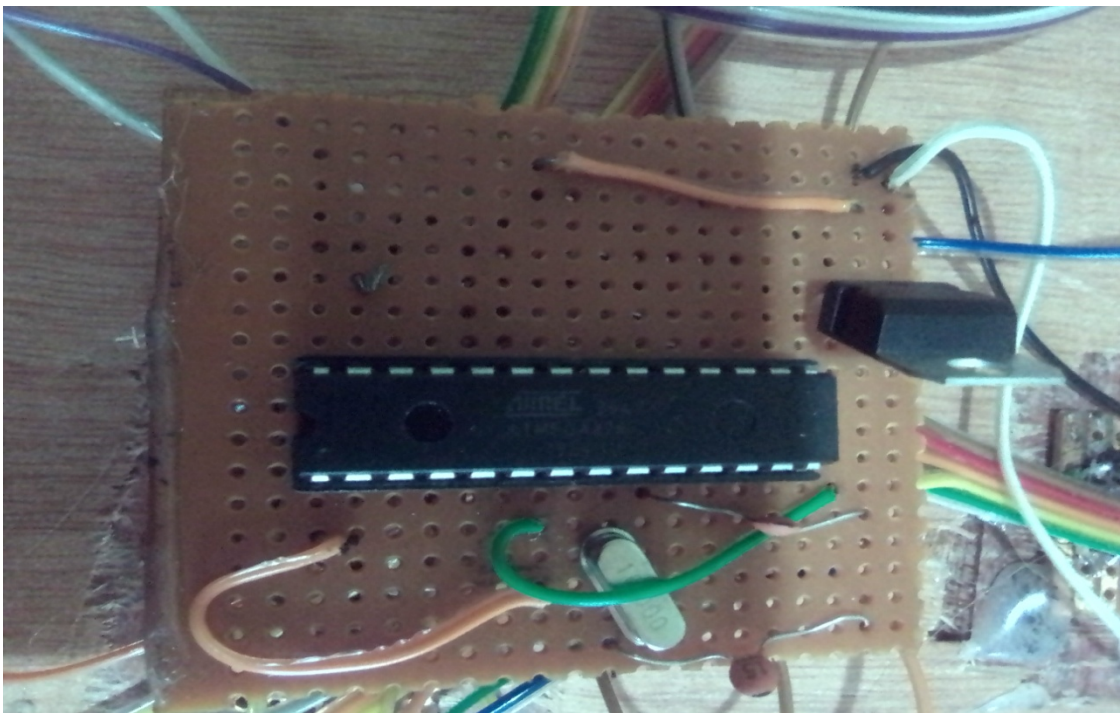


Figure 42: circuit à microcontrôleur ATmega328P

Sur chaque circuit à microcontrôleur ATmega328P sont connectés les modules MCP2515-CAN et MAX 485. Sur chaque module sont prévus des connecteurs pour réaliser la communication. Les lignes de transmission CAN-LOW, CAN-HIGH pour la communication CAN et A, B pour la communication ModBus sont réalisées de la manière suivante :

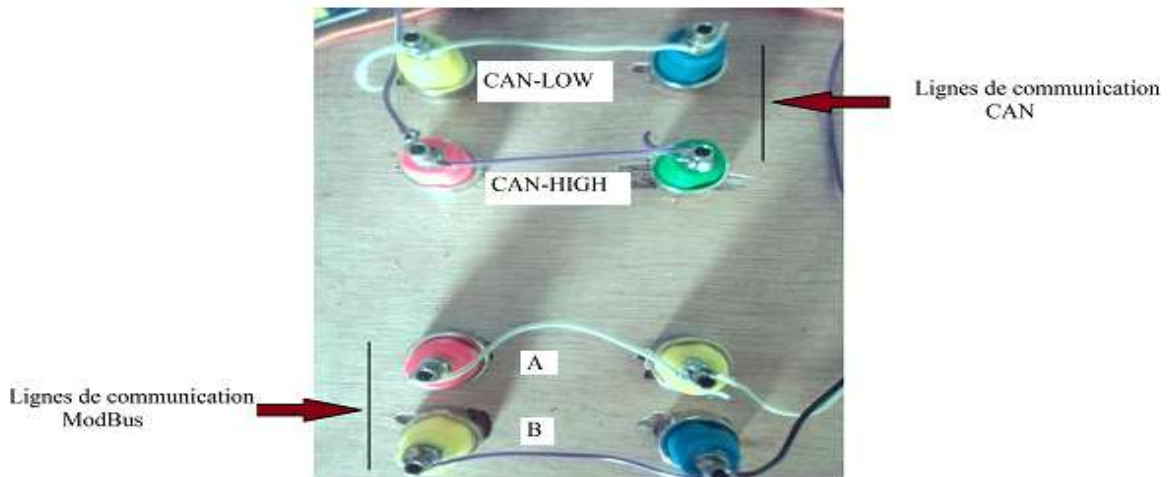


Figure 43: module CAN/ModBus

Dans la suite de la soudure, nous avons augmenté les autres composants comme les MCP2515-CAN, les MAX485, la carte arduino du bloc supervision, le redresseur pont de Graetz pour l'alimentation, l'interrupteur de départ et fin du cycle et les boutons poussoirs de commande.

3) La coque

La coque utilisée dans le cadre de notre travail est une caisse en bois de forme parallélépipédique et ayant une hauteur de 10,5cm, une longueur de 40,7cm et une largeur de 21,6cm. Cette caisse aura pour rôle de protéger notre système contre le durcissement des composants électroniques utilisés dû à l'air. Elle se présente de la manière suivante :



Figure 44: Coque pour la maquette

3.4) Présentation et caractéristiques de la maquette finale

1) Présentation

La maquette finale nommée « WINNER01 » est constituée des quatre blocs cités plus haut et de la caisse ci-dessus. Les fiches bananes sont utilisées sur les lignes physiques des bus de terrain pour réaliser la communication entre différents systèmes qu'intègre celle-ci. Le schéma suivant donne la structure globale :



Figure 45: structure globale de la maquette finale

3) Caractéristiques générales et performances techniques

Les caractéristiques globales de notre maquette sont données par les tableaux suivants :

Caractéristiques de la maquette par transmission CAN	
Type de transmission	CAN High Speed, half-duplex
Nombre de nœuds	3 nœuds
Temps de transmission	3 secondes
Courant de sortie	20mA sur 60 Ω
Tension CAN-L	1,6 V
Tension CAN-H	3,7 V
Modèle de coopération	Client/serveur

Tableau 4: Caractéristiques de la maquette par transmission CAN

Caractéristiques de la maquette par transmission ModBus	
Type de transmission	Half-duplex bidirectionnelle alternée
Nombre de nœuds	3nœuds
Temps de transmission	5 secondes
Modèle de coopération	Maître/esclave
Maître	Microcontrôleur arduino côté PC
Esclaves	Les microcontrôleurs ATmega328P des parties commande et opérative

Tableau 5: Caractéristiques de la maquette par transmission ModBus

Quelques performances techniques sont à relever :

- Puissance d'entrée : monophasé à deux fils $\sim 220V$ plus ou moins 10% /50Hz
- Puissance de sortie du groupe 5V/25mA
- Bouton de départ et fin du cycle
- Taille d'aspect : 40,7*21,6*10,5cm

Ces caractéristiques donnent un temps de transmission plus élevé en ce qui concerne le bus ModBus par rapport au temps de transmission par bus CAN ce qui rend le système plus lent dans ce cas. De ce fait s'il fallait choisir une priorité le bus CAN serait choisit en dépit du bus ModBus.

CONCLUSION

Dans ce chapitre, les différents blocs de notre maquette ont été présentés ainsi que les étapes de sa réalisation. Nous avons donné les caractéristiques et les performances techniques de celle-ci sur lesquelles les commentaires ont été énoncés.

CONCLUSION GENERALE

Dans le cadre de notre projet de mémoire, il nous a été donné de travailler sur le thème «**mise en œuvre d'une plateforme didactique pour l'apprentissage des réseaux locaux industriels et supervision** » pour y parvenir, nous avons subdivisé notre travail en trois chapitres. Le premier était consacré à la revue de la littérature, le deuxième abordait le matériel utilisé ainsi que les méthodes et le troisième présentait les simulations, la réalisation et les discussions sur les résultats obtenus. Nos objectifs spécifiques étaient :

- De mettre sur pied les différents blocs qui entrent dans la conception de notre maquette et les faire fonctionner
- De réaliser la communication entre ces blocs
- De rédiger un cahier de travaux pratiques qui permettra aux étudiants de faire des manipulations sur notre maquette

Pour procéder à l'atteinte des objectifs précédemment énoncés et de l'objectif principal, nous avons grâce aux matériels et logiciels électronique et de supervision développé une maquette intégrant plusieurs blocs en communication via deux bus de terrain (CAN et ModBus) choisis au préalable. Les premiers tests sur notre maquette ont donné des résultats un peu plus intéressants en ce qui concerne le temps de propagation des messages sur les lignes physiques des bus. Pour le bus CAN ce temps est un peu plus faible donc plus intéressant par rapport au bus ModBus. Il est à noter que plus la longueur des câbles sur les lignes de bus augmentent, plus le temps de propagation des messages augmentent aussi. De ce fait s'il fallait étendre notre système sur des distances allant de plusieurs mètres à plusieurs kilomètres, certains dispositifs intermédiaires seront nécessaires.

PERSPECTIVES

- Entreprendre la même initiative en faisant étendre le système sur des distances allant de plusieurs mètres à plusieurs kilomètres tout en minimisant au maximum le temps de propagation du message sur les lignes du bus.
- Développer plus de plateformes didactiques en ce qui concerne les autres disciplines nécessitant des manipulations pratiques pour parfaire la formation.

REFERENCES BIBLIOGRAPHIQUES

- [1] Jean Claude CABIANCA, *Les réseaux locaux industriels synthèse informatique*, 23 pages
- [2] D. Paret, *Le bus CAN*, édition Dunod. 1999, 352 pages
- [3] Jean-Pierre Thomesse, *Les réseaux locaux industriels*, Edition Eyrolles, 1994
- [4] pascal vrignat, *Les réseaux locaux industriels cours et travaux pratiques*.1999
- [5] Eric DECKE, *Réseaux Locaux Industriels et Bus de Terrain*, Janvier 2006
- [6] Thierry Perisse BAUDE | BOUYSSOU | DAZA | FEYSSEL | GAMBLIN, *Projet SPI: Le Bus CAN*, 03/15/2006, 66 pages
- [7] ESKIMON, *Arduino : premiers pas en informatique embarquée*, 1er février 2020, 639 pages
- [8] Hortolland C, *TP bus CAN Arduino*, 2015-2016
- [9] V.Chollet, *Cours Labview 12*, 10/01/2013, 66 pages
- [10] ESKIMON, GLENN, OTANOSHIMINI, *Fabrication numérique*, 12 Août 2019, 365 pages
- [11] *Internationnal Journal for Inovative science, Engineering & Technology*, vol, 5 may 2015
- [12] Corrine Braban/ Florence Boillot, *Les systèmes temps réel de commande de feu en milieu urbain*, Avril 2013
- [13] Fabien. *Initialisation au labview. [En ligne] formation des enseignants*.2014, 14p
- Disponible sur :
- https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwiLi6jj4rNAhWJAsAKHS_BA8sQFggdMAA&url=http%3A%2F%2Feducol.education.fr%2Fsti%2Fsites%2Feducol.education.fr.sti%2Ffiles%2Fressources%2Ftechniques%2F3437%2F3437-1-initiation-labview-levi.pdf&usq=AFQjCNEC93A17pfzubLy9gTuUzlsTmgqdw&sig2=ADTGihgToJ34_87WBKn clw (consulté le 21/05/2019).

[14] Sylvain Chalmin, Bacem ElIdilbi. *Informatique des Systèmes Embarqués : implémentation de protocole USB*. 2009, 25p (consulté le 15/04/2019).

[15] 7Robot. *Electronique : À la découverte de bus CAN [En ligne]* 2011

Disponible sur : <http://www.bde.enseeiht.fr/clubs/robot/node/44> (consulté le 20/04/2019).

[16] Microchip. *Data sheet MCP25-15_CAN*. 2009, 438p.

Disponible sur :

https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&ved=0ahUKEwjxpp3riIrNAhUsIsAKHYJICaYQFgg0MAY&url=http%3A%2F%2Fww1.microchip.com%2Fdownloads%2Fen%2FDeviceDoc%2F39632e.pdf&usg=AFQjCNFNi_QIXM89BfgZ0NeS_PdZ3xLaSQ&sig2=T2IRD4Ghl2HFyi6JqAawnw (consulté le 20/03/2020).

[17] Truillet Aoun, Kacimi, Torguet, *Introduction aux Réseaux Informatiques, Université Paul Sabatier, IRIT- Cours CESI*

[18] Commission générale de terminologie et de néologie - NOR : CTNX0306624X - JO du 14-06-2003, pp. 10047-10050

[19] *The history of standardization and CAN in a nutshell – CiA (Can in Automation) – [En Ligne]* : http://www.can-cia.org/fileadmin/cia/files/Newsletter_02-14/2-14_p3_the-history-ofstandardization-and-can-in-a-nutshell.pdf

[20] CHARLES LERY, *Mise en œuvre d'une communication par Bus CAN [En Ligne]*
http://dirac.epucfe.eu/projets/wakka.php?wiki=P08AB12index/download&file=note_application_co_m_module_mppt.pdf

[21] CIAME, *Réseaux de terrain. Description et critères de choix*, Editions Dunod. 1999.

[22] P. Bonnet, 'introduction à la supervision' P 3

Sites internet :

[23] www.as-interface.net, visité le 25 janvier 2020

[24] www.didalab.fr, visité le 05 décembre 2019,

[25] www.ssedu.com, visité le 04 décembre 2019

ANNEXES

1) Programmes générales dans l'IDE arduino :

a) Module de commande

```
#include <SPI.h>
#include <SoftwareSerial.h>
// specify pins rx and tx respectively
SoftwareSerial mySerial(2, 3);
// pin for send/get rs485
int pin_direction_RX_TX = 4;
// pin led
int pinled = 9;
// value status led
int statusled = 0;
MCP_CAN CAN0(10); // Set CS to pin 10
const int plus=7;
const int moin =6;
const int clic =8;

String inputString0 = "";
// a sign of the end of the resulting string
boolean stringComplete0 = false;
// a sign of correct data
boolean res = false;
void setup()
{pinMode(plus, INPUT_PULLUP);
pinMode(moin, INPUT_PULLUP);
pinMode(clic, OUTPUT);
Serial.begin(9600);
mySerial.begin(9600);
// Initialize MCP2515 running at 16MHz with a baudrate of 500kb/s and the masks and filters disabled.
if(CAN0.begin(MCP_ANY, CAN_500KBPS, MCP_16MHZ) == CAN_OK) Serial.println("MCP2515 Initialized Successfully!");
```

```
CAN0.setMode(MCP_NORMAL); // Change to normal mode to allow messages to be transmitted
}

byte data[8] = {0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07};

void loop()
{ byte sndStat =0;
  digitalWrite(clic,HIGH);
  if(digitalRead(plus)==LOW){digitalWrite(pin_direction_RX_TX, HIGH);
    delay(25);mySerial.println("b");
    data[0]=10;
    sndStat = CAN0.sendMsgBuf(1, 0, 8, data);
  if(sndStat == CAN_OK){
    Serial.println("Message Sent Successfully!");
  } else {
    Serial.println("Error Sending Message...");
  }
  digitalWrite(clic,HIGH);
  delay(100); // send data per 100ms
  digitalWrite(clic,LOW);
  delay(100);
  digitalWrite(clic,HIGH);
  delay(100); // send data per 100ms
  digitalWrite(clic,LOW);
  delay(100);
  digitalWrite(clic,HIGH);
  delay(100); // send data per 100ms
  digitalWrite(clic,LOW);
}
```

b) Module partie opérative

```
const int v1=A0;
const int o1=A1;
const int r1=A2;
const int v2=A3;
const int o2=A4;
const int r2=A5;
long ret=2000;
#include <mcp_can.h>
#include <SPI.h>

#include <SoftwareSerial.h>
// specify pins rx and tx respectively
SoftwareSerial mySerial(2, 3);
// pin for send/get rs485
int pin_direction_RX_TX = 4;
// pin led
int pinled = 9;
// value status led
int statusled = 0;

// to retrieve data from SoftwareSerial
String inputString0 = "";
// a sign of the end of the resulting string
boolean stringComplete0 = false;
// a sign of correct data
boolean res = false;

long unsigned int rxId;
unsigned char len = 0;
, unsigned char rxBuf[8];|
```

```
long unsigned int rxId;
unsigned char len = 0;
unsigned char rxBuf[8];
char msgString[128];           // Array to store serial string
const int plus=8;
const int moin=7;
#define CAN0_INT 2             // Set INT to pin 2
MCP_CAN CAN0(10);             // Set CS to pin 10

void setup() {
  // put your setup code here, to run once:
  pinMode(v1,OUTPUT);
  pinMode(v2,OUTPUT);
  pinMode(o1,OUTPUT);
  pinMode(o2,OUTPUT);
  pinMode(r1,OUTPUT);
  pinMode(r2,OUTPUT);
  pinMode(plus,OUTPUT);
  pinMode(moin,OUTPUT);
  Serial.begin(9600);
  mySerial.begin(9600);
  // Initialize MCP2515 running at 16MHz with a baudrate of 500kb/s and the masks and filters disabled.
  if(CAN0.begin(MCP_ANY, CAN_500KBPS, MCP_16MHZ) == CAN_OK)
    Serial.println("MCP2515 Initialized Successfully!");
  else
    Serial.println("Error Initializing MCP2515...");

  CAN0.setMode(MCP_NORMAL);    // Set operation mode to normal so the MCP2515 sends acks to received data.

  pinMode(CAN0_INT, INPUT);    // Configuring pin for /INT input
```



```
pinMode(pin_direction_RX_TX, OUTPUT);
// for get rs485 data
digitalWrite(pin_direction_RX_TX, LOW);
// led
pinMode(pinled, OUTPUT);
digitalWrite(pinled, statusled);
// reserve 50 bytes for the inputString0:
inputString0.reserve(50);
}

void loop() {
    digitalWrite(pin_direction_RX_TX, LOW);
    serialEvent0();
    long ret2=ret/2;
    // put your main code here, to run repeatedly:
    digitalWrite(v1, HIGH);
    digitalWrite(r2, HIGH);
    digitalWrite(r1, LOW);
    digitalWrite(o2, LOW); digitalWrite(pin_direction_RX_TX, HIGH);
    delay(25); mySerial.println("e"); digitalWrite(pin_direction_RX_TX, LOW);
    delay(ret);
    digitalWrite(v1, LOW);
    digitalWrite(o1, HIGH); digitalWrite(o2, LOW); digitalWrite(pin_direction_RX_TX, HIGH);
    delay(25); mySerial.println("f"); digitalWrite(pin_direction_RX_TX, LOW);

    delay(ret2);
    digitalWrite(o1, LOW);
    digitalWrite(r1, HIGH);
    digitalWrite(r2, LOW);
    digitalWrite(v2, HIGH); digitalWrite(o2, LOW); digitalWrite(pin_direction_RX_TX, HIGH);|

    delay(ret2);
    digitalWrite(o1, LOW);
    digitalWrite(r1, HIGH);
    digitalWrite(r2, LOW);
    digitalWrite(v2, HIGH); digitalWrite(o2, LOW); digitalWrite(pin_direction_RX_TX, HIGH);
    delay(25); mySerial.println("g"); digitalWrite(pin_direction_RX_TX, LOW);
    delay(ret);
    digitalWrite(v2, LOW);
    digitalWrite(o2, HIGH);
    digitalWrite(pin_direction_RX_TX, HIGH);
    delay(25); mySerial.println("h"); digitalWrite(pin_direction_RX_TX, LOW);
    delay(ret2);

    if(!digitalRead(CAN0_INT)) // If CAN0_INT pin is low, read receive buffer
    {
        CAN0.readMsgBuf(&rxId, &len, rxBuf); // Read data: len = data length, buf = data byte(s)

        if((rxId & 0x80000000) == 0x80000000) // Determine if ID is standard (11 bits) or extended (29 bits)
            sprintf(msgString, "Extended ID: 0x%.8lX DLC: %ld Data:", (rxId & 0x1FFFFFFF), len);
        else
            sprintf(msgString, "Standard ID: 0x%.3lX DLC: %ld Data:", rxId, len);

        Serial.print(rxId);

        if((rxId & 0x40000000) == 0x40000000){ // Determine if message is a remote request frame.
            sprintf(msgString, " REMOTE REQUEST FRAME");
            Serial.print(msgString);
        } else {
            for(byte i = 0; i<len; i++){|
```

```
}  
  
void serialEvent0() {  
  if (mySerial.available() > 0) {  
    // get byte:  
    char inChar = (char)mySerial.read();  
    // add to string  
    inputString0 += inChar;  
    Serial.print(inChar);  
    // /n - end of transmission  
    if (inChar == '\n') {  
      stringComplete0 = true;  
    }  
    if (inChar == 'a' && ret > 1000) {  
      digitalWrite(moin, HIGH);  
      delay(1000);  
      digitalWrite(moin, LOW);  
      ret = ret - 1000;  
    }  
    if (inChar == 'b') {  
      digitalWrite(plus, HIGH);  
      delay(1000);  
      digitalWrite(plus, LOW);  
      ret = ret + 1000;  
    }  
  }  
}  
}
```

c) Module supervision

```
#include <SoftwareSerial.h>
// specify pins rx and tx respectively
SoftwareSerial mySerial(2, 3);
// pin for send/get max485
int pin_direction_RX_TX = 4;
// pin led
int pinled = 8;
// value status led
int statusled = 0;

// to retrieve data from SoftwareSerial
String inputString0 = "";
// a sign of the end of the resulting string
boolean stringComplete0 = false;
// a sign of correct data
boolean res = false;
#include <mcp_can.h>
#include <SPI.h>

long unsigned int rxId;
unsigned char len = 0;
unsigned char rxBuf[8];
char msgString[128]; // Array to store serial string
const int plus=7;
const int moin=8;
#define CAN0_INT 2 // Set INT to pin 2
MCP_CAN CAN0(10); // Set CS to pin 10

void setup() {
  Serial.begin(9600);
  // SoftwareSerial begin

  mySerial.begin(9600);
  pinMode(pin_direction_RX_TX, OUTPUT);
  // for get rs485 data
  digitalWrite(pin_direction_RX_TX, LOW);
  // led
  pinMode(pinled, OUTPUT);
  digitalWrite(pinled, statusled);
  // reserve 50 bytes for the inputString0:
  inputString0.reserve(50); // Initialize MCP2515 running at 16MHz with a baudrate of 500kb/s and the masks and filters disabled.
  pinMode(plus, OUTPUT);
  pinMode(moin, OUTPUT);
  Serial.begin(9600);
  digitalWrite(moin, HIGH);
  delay(1000);
  digitalWrite(moin, LOW);
  // Initialize MCP2515 running at 16MHz with a baudrate of 500kb/s and the masks and filters disabled.
  if(CAN0.begin(MCP_ANY, CAN_500KBPS, MCP_16MHZ) == CAN_OK)
    Serial.println("MCP2515 Initialized Successfully!");
  else
    Serial.println("Error Initializing MCP2515...");

  CAN0.setMode(MCP_NORMAL); // Set operation mode to normal so the MCP2515 sends acks to received data.

  pinMode(CAN0_INT, INPUT); // Configuring pin for /INT input

  Serial.println("MCP2515 Library Receive Example...");
}
byte data[8] = {0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07};

void loop() { digitalWrite(pin_direction_RX_TX, LOW);
  serialEvent0();
```



```
if((rxId & 0x40000000) == 0x40000000){ // Determine if message is a remote request frame.
    sprintf(msgString, " REMOTE REQUEST FRAME");
    Serial.print(msgString);
} else {
    for(byte i = 0; i<len; i++){
        sprintf(msgString, " 0x%.2X", rxBuf[i]);

        } Serial.println();Serial.print(rxBuf[0]);

if(rxId==1)
{Serial.print("ghyughgfgfhy");
if(rxBuf[0]==10){    digitalWrite(moin,HIGH);
    delay(1000);
    digitalWrite(moin,LOW);
Serial.print("ghyughgfgfhy");
}
if(rxBuf[1]==11){

}
}
}

Serial.println();
}rxBuf[0]=0;
}

// SerialEvent
void serialEvent0() {
    if (mySerial.available() > 0) {
        //| get byte:

char inChar = (char)mySerial.read();
// add to string
inputString0 += inChar;
Serial.print(inChar);
// /n - end of transmission
if (inChar == '\n') {
    stringComplete0 = true;
}
}
}
```

2) Coût de la maquette

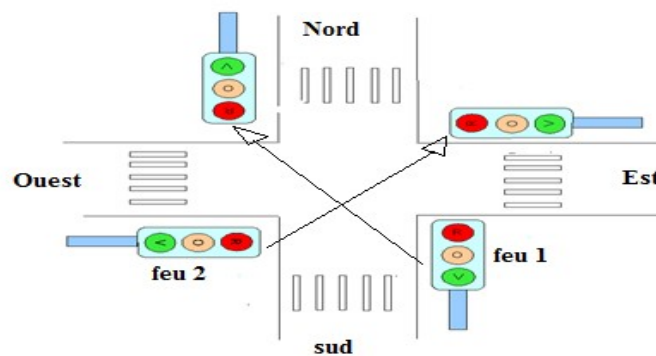
Désignation	Quantité	Prix
Carte arduino Uno	01	8500 F
Microcontrôleur Atmega328P	02	5000 F
Quartz	02	1000 F
Régulateur LM7805	02	1000F
Condensateurs	05	500F
Resistances	10	1000 F
LED	27	3000 F
Plaque perforée	01	500F
Fer à soudé+étain	01	5000F
MCP2515_CAN	03	12000F
MAX485	03	9000F
Fils de connexion	01	2500F
Transformateur	01	3500F
Diodes	04	800F
Boutons poussoirs	03	600F
Connecteurs	16	9000F
Fiche banane +coque	04+01	2000+10000F
Activation de la licence LabVIEW	//	50000F
VI LabVIEW	//	50000F

TOTAL : 175000 F main d'œuvre exclut

3) Cahier de travaux pratiques : maquette didactique « Winner01 »

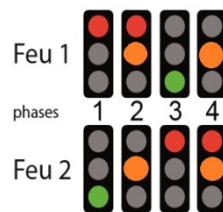
CAHIER DE CHARGE :

L'objectif est de créer une maquette didactique des feux tricolores et de les faire fonctionner de manière synchrone. Cette maquette comporte un carrefour à deux voies (Nord-Sud et Est-Ouest) qui se croisent. L'ensemble est contrôlé par des cartes arduino en communication et pilotées par le logiciel LabVIEW avec supervision à l'aide d'un Interface Homme Machine (IHM). chaque voie est équipée à l'entrée et à la sortie des feux tricolores (voyant rouge, voyant vert, voyant orange) à usage générale (véhicules et piétons). L'intersection des voies est constituée de quatre panneaux disposés sur les extrémités de ses diagonales. Le système est tel que les deux panneaux situés sur la même diagonale reçoivent exactement les mêmes signaux de commande.



En fonctionnement normal, le système démarre lorsque tout et feux sont éteints et l'on appuie sur un bouton poussoir départ cycle (**Dcy**). Pour arrêter le système on appuie sur le bouton poussoir arrêt (**Ar**). Les phases de fonctionnement sont les suivants:

La phase 1 dure 10s, la phase 2 dure 5s, la phase 3 dure 10s et la phase 4 dure 5s.



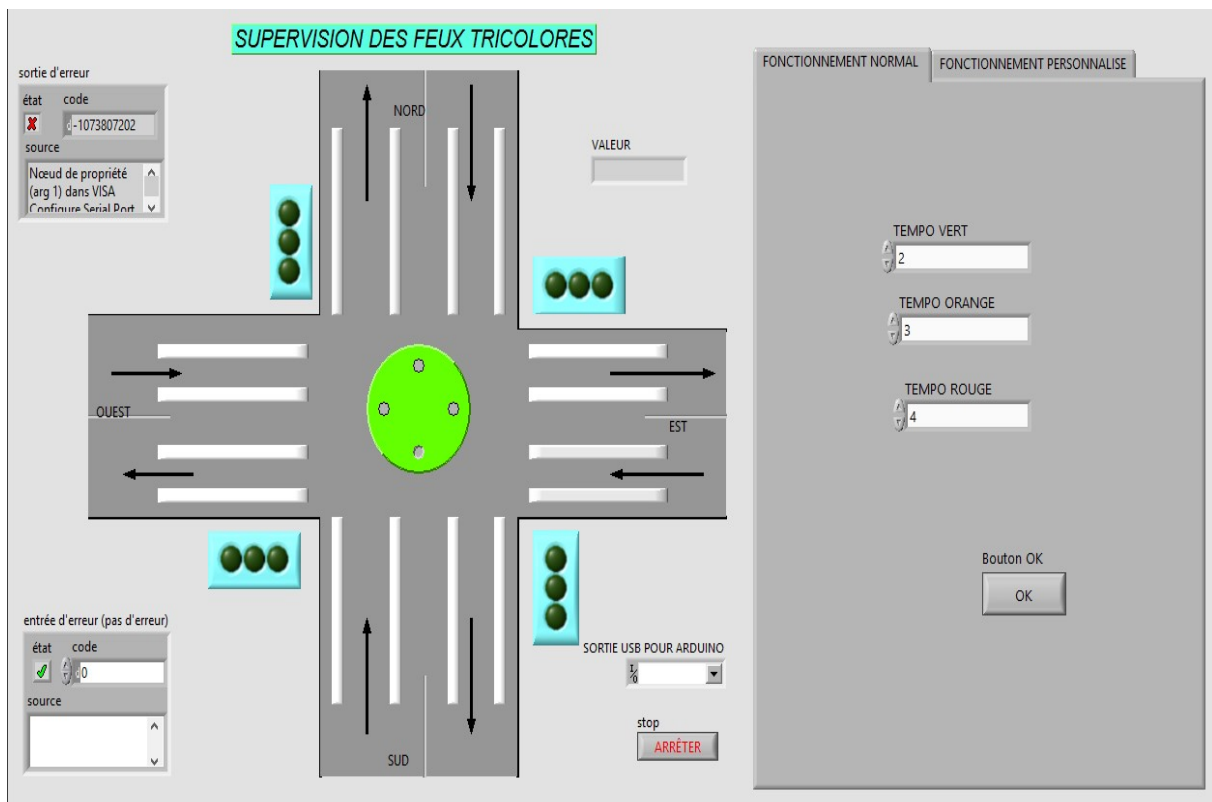
Afin d'assurer un autodiagnostic du système, une partie commande constituée de deux boutons poussoirs est utilisée pour augmenter ou diminuer le temps de fonctionnements de tous les panneaux de façon automatique. Trois LED seront utilisées pour témoigner l'envoi et la réception de la commande.

La communication entre les différents modules (partie opérative, partie commande et l'IHM) sera réalisée à l'aide de deux bus de terrain au choix : le bus ModBus et le bus CAN

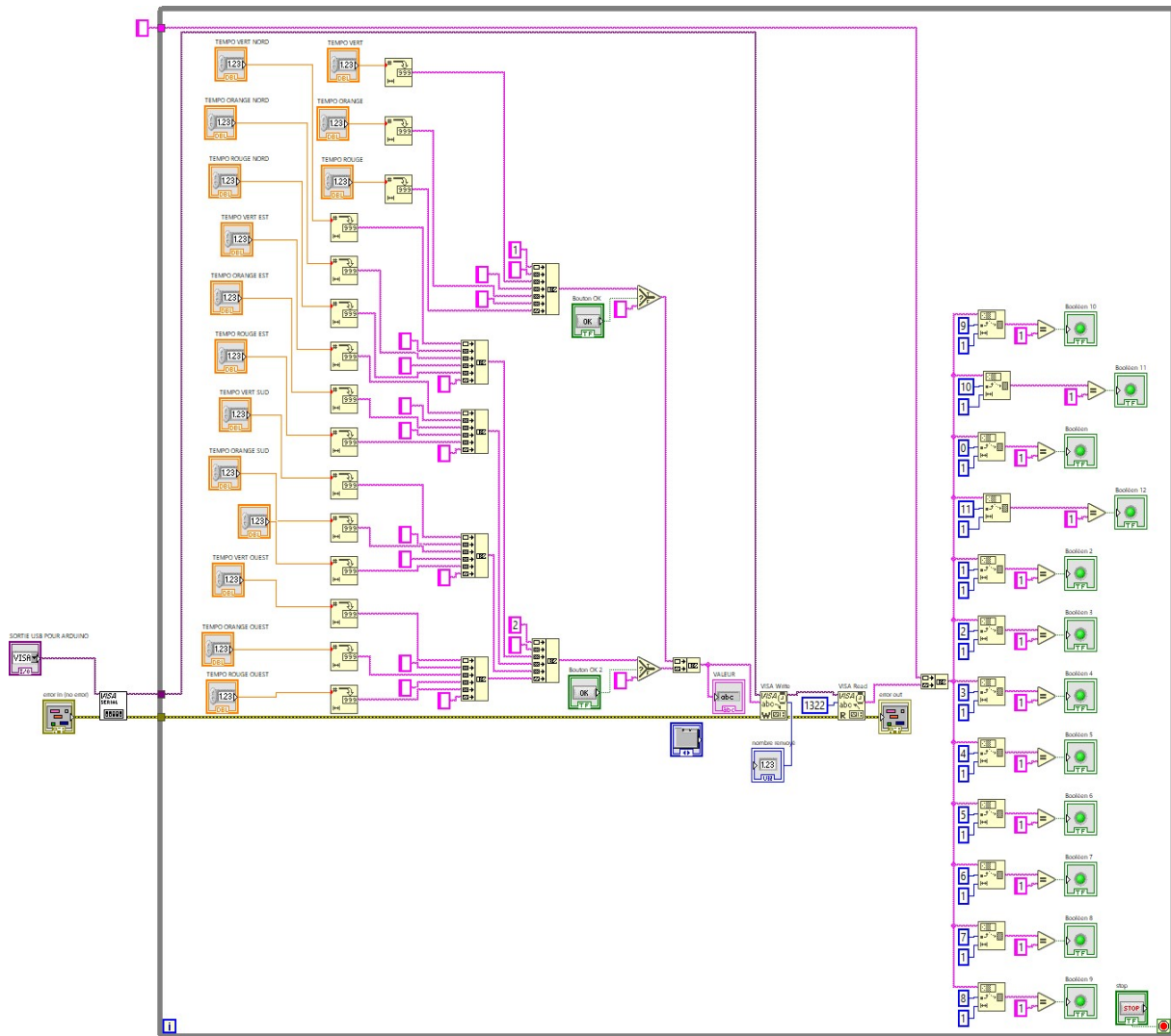
TP N°1 : prise en main de LabVIEW et réalisation de l'interface de supervision

La supervision d'un système automatique une interface dite de supervision réalisée dans un logiciel dit de supervision. Dans notre cas, le logiciel de supervision est LabVIEW. Suivez les étapes suivantes et réaliser l'interface de supervision sur LabVIEW.

- Télécharger et installer LabVIEW 2017 sur votre PC
- Ouvrir le logiciel LabVIEW 2017 et à l'aide des commandes et des décorations, réaliser la face avant de supervision de notre système donné par la figure suivante :



- Après réalisation de la face, avant, aller sur la face arrière et réaliser le diagramme bloc suivant :



- Après réalisation du VI pour la supervision, connecter la sortie USB de la carte arduino présente sur notre maquette à la machine et réaliser la communication arduino-LabVIEW en suivant les étapes suivantes :
 - Lancer le logiciel LabVIEW
 - Cliquer sur **Outils >>Maker Hub >>LINX >> Firmware Wizard**
 - Sélectionner le port **COM** associé à notre appareil puis cliquer sur **Suivant**
 - Sélectionner la version du micro logiciel (commencer par le micro logiciel série/USB) et cliquer sur suivant.
- NB : le Lancer le logiciel LabVIEW
- Cliquer sur **Outils >>Maker Hub >>LINX >> Firmware Wizard**
- Sélectionner le port **COM** associé à notre appareil puis cliquer sur **Suivant**
- Sélectionner la version du micro logiciel (commencer par le micro logiciel série/USB) et cliquer sur suivant.

NB : il faut au préalable installer le VI package et le NI VISA.

TPN°2 : étude de la maquette et manipulation

Après avoir fait une auto observation de la maquette :

- 1) Faire ressortir avec description à l'appui les différents blocs que présente celle-ci
- 2) Faire ressortir s'il y en a la liste des capteurs et actionneurs. A quels niveaux de la pyramide SIM appartiennent-ils ?
- 3) Identifier les deux bus de terrain utilisés pour la communication sur cette maquette, identifier leurs lignes de transmission, la typologie des canaux de transmission et le type de transmission utilisé par chacun d'eux.
- 4) Utiliser l'un après l'autre des bus de terrain pour la communication en reliant les connecteurs des lignes de transmission.
- 5) A chaque fois, appuyer sur les boutons poussoirs « dim » et « aug » et observer le fonctionnement du système
- 6) Relever à chaque fois à l'aide d'un chronomètre le temps de transmission des données sur les lignes physique des bus et tirer une conclusion
- 7) En fonction de ce temps, faire un choix en ce qui concerne le type de bus le mieux adapter pour notre système
- 8) Observer l'interface de supervision et conclure en ce qui concerne la supervision du système tout en faisant ressortir les organes de supervision utilisés.

Eléments de correction TP N°2:

- 1) La maquette comporte quatre parties :
 - La partie opérative qui est constituée d'un carrefour à deux voies Nord-Sud et Est-Ouest qui se croise et de quatre panneaux situés sur les extrémités des diagonales formés par l'intersection de celles-ci. Chaque panneau est constitué de six LEDs montées trois à trois en série et deux à deux en parallèle.
 - La partie de commande qui est constituée de deux boutons poussoirs pour l'augmentation et la diminution automatique du temps de commutation entre les différents feux des panneaux de la partie opérative via le module bus de terrain.
 - La partie bus de terrain qui est celui qui réalise l'intermédiaire entre les autres blocs et qui permet la communication entre ceux-ci.

- La partie supervision qui est celui constituée d'un ordinateur muni d'un logiciel de supervision (LabVIEW dans notre cas) dans lequel nous avons réalisé un IHM pour la supervision du trafic routier via une communication à travers la carte arduino et les modules bus de terrain.

2) listes des capteurs et actionneurs

Capteurs : LED de réception de la commande

Actionneurs : les boutons poussoirs BP aug, BP dim et Dcy/Ar

Ces capteurs et actionneurs appartiennent au premier niveau de la pyramide SIM

3)

Bus de terrain	Lignes de transmission	Type de transmission	Typologie des canaux de transmission
CAN	CAN-L CAN-H	Half-duplex bidirectionnelle simple	Paire torsadée
ModBus	A B	Half-duplex bidirectionnelle alternée	Paire torsadée

6) Temps de transmission

Bus CAN : 3 secondes

ModBus : 5 secondes

Conclusion : Pour le bus CAN le temps de transmission est un peu plus faible donc plus intéressant par rapport au bus ModBus

7) le bus de terrain le mieux adapter pour notre système est donc le bus CAN