Réalisation du banc de test DeviceNet
Nicolas Charanton

To cite this version:

HAL Id: dumas-01280359
https://dumas.ccsd.cnrs.fr/dumas-01280359
Submitted on 29 Feb 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
CONSERVATOIRE NATIONAL DES ARTS ET METIERS
CENTRE REGIONAL ASSOCIE DE GRENOBLE

MEMOIRE
présenté en vue d'obtenir
le DIPLOME D'INGENIEUR CNAM

SPECIALITE : ELECTRONIQUE-AUTOMATIQUE
OPTION : AUTOMATIQUE

par

Nicolas CHARANTON

Réalisation du banc de test DeviceNet

Soutenu le 11 juin 2012

JURY

PRESIDENT : Monsieur André PLISSON

MEMBRES :
Madame Nathalie GAUDIN
Monsieur Bernard DESCOTES-GENON
Monsieur Gabriel BUCHE
Monsieur Philippe LE POUHAER
Monsieur Stéphane HURY

Directeur du CNAM Grenoble

Directrice d'études
Responsable de la filière automatique
Enseignant CNAM
Responsable du serviceingénierie40-30
Responsable du service R&D 40-30
Réalisation du banc de test DeviceNet

Remerciements

Je tiens à remercier Pierre DELALEZ directeur de 40-30 qui m’a fait confiance en me permettant de réaliser mon mémoire d’ingénieur dans sa société ainsi qu’au responsable du site de Bernin Jose-Manuel Fernandez et à mon tuteur de stage Philippe Le Pouhaer qui ont participé quotidiennement au bon déroulement du projet.

Je remercie également les membres du jury, non seulement pour avoir accepté de consacrer du temps pour la lecture de ce document mais également pour me permettre de leur présenter le résultat de mes recherches.

Je remercie tout particulièrement Monsieur Bernard Descotes-Genon tuteur du mémoire d’ingénieur en automatismes. J’exprime aussi toute ma gratitude à tous les collègues de l’entreprise 40-30, qui m’ont apporté leur aide dans la réalisation de mon projet et dans la relecture de ce rapport.

J’associe à mes remerciements toutes les personnes, qui directement ou indirectement, ont contribué au bon déroulement de mon stage.
Table des matières

INTRODUCTION .................................................................................................. 6

PRESENTATION DE LA SOCIETE 40-30.................................................................. 8
  INFORMATIONS GENERALES ................................................................................ 9
  HISTORIQUE ........................................................................................................ 9

CHAPITRE 1 ETAT DE L’ART DES BUS DE TERRAIN ........................................... 10
  1.1 - LES BUS DE TERRAIN .............................................................................. 10
    1.1.1 - Les principaux avantages des bus de terrain ....................................... 11
    1.1.2 - Les protocoles génériques d’accès .......................................................... 12
  1.2 - RESEAUX INDUSTRIELS ......................................................................... 16
    1.2.1 - La communication sur un réseau ......................................................... 16
    1.2.2 - Le modèle OSI ..................................................................................... 16
    1.2.3 - Le bus terrain est une version simplifiée du modèle OSI ....................... 23
  1.3 - L’ODVA ..................................................................................................... 24
    1.3.1 - A propos de l’ODVA ............................................................................ 24
    1.3.2 - Common Industrial Protocol ................................................................. 24

CHAPITRE 2 ETAT DE L’ART DU PROTOCOLE DEVICENET ............................... 36
  2.1 - LE PROTOCOLE DeviceNet ....................................................................... 36
  2.2 - CONTROL AREA NETWORK ..................................................................... 41
    2.2.1 - Principe de fonctionnement .................................................................. 42
    2.2.2 - Principe de l’arbitrage ......................................................................... 43
    2.2.3 - Formats de trames .................................................................................. 44
    2.2.4 - Détectio des erreurs .............................................................................. 46
  2.3 - CARACTERISTIQUES SPECIFIQUES AU RESseau DeviceNet ................. 49
    2.3.1 - ECHANGE DE DONNEES .................................................................... 50
    2.4 - DESCRIPTION DES TRAMES ECHANGEES ............................................. 52
      2.4.1 - Les messages explicites ...................................................................... 54
      2.4.2 - Les messages E/S Poll ....................................................................... 60
  2.5 - ASPECTS ECONOMIQUES ET STRATEGIQUES ......................................... 62
      2.5.1 - Comparatif entre DeviceNet et Profibus ............................................ 63

CHAPITRE 3 LE BANC DE TEST DEVICENET ...................................................... 65
  3.1 - CHOIX DES LOGICIELS ET DU MATERIEL ADAPTE ................................. 65
  3.2 - PROGRAMME DE TEST DES GENERATEURS RF ET DES VANNES MKS 683 ......... 67
    3.2.1 - Réalisation du programme ................................................................. 68
    3.2.2 - Programme de test du générateur RF ............................................... 68
    3.2.3 - Programme de test de la vanne MKS 683 ............................................ 70
  3.3 - REDACTION DU MODE OPERATOIRE .................................................... 71
    3.3.1 - Configuration des outils ...................................................................... 71
    3.3.2 - Installation des logiciels Rockwell automation .................................... 72
    3.3.3 Le banc de test ....................................................................................... 85
  3.4 - REFERENT TECHNIQUE DES BUS DE TERRAIN .................................... 91
  3.5 - LIMITES ET PERSPECTIVES .................................................................... 91
    3.5.1 - Simplifier l'utilisation du banc ........................................................... 92
    3.5.2 - Communication analogique entre le PLC et l'équipement .................... 93
    3.5.3 - Communication avec l'interface AMPS .............................................. 93

CONCLUSION .................................................................................................... 94

LEXIQUE .......................................................................................................... 96

BIBLIOGRAPHIE .............................................................................................. 98
Index des Illustrations

Figure 1 : Classification des bus de terrain [2] ............................................. 11
Figure 2 : Exemple d’accès orienté connexion ......................................... 12
Figure 3 : Exemple d’accès par polling ......................................................... 13
Figure 4 : Exemple d’accès par multiplexage temporel .............................. 13
Figure 5 : Exemple d’accès par jeton sur anneau ....................................... 14
Figure 6 : Carrier Sense Multiple Access [21] ............................................ 15
Figure 7 : Le modèle OSI [20] ................................................................. 17
Figure 8 : Le modèle OSI (Envoi de données) [20] .................................... 22
Figure 9 : Classification des techniques d’accès MAC [4] ............................ 23
Figure 10 : Le CIP et les couches OSI [13] .............................................. 26
Figure 11 : Exemple d’adressage CIP [14] ................................................... 27
Figure 12 : Application Object et Connection ID [14] .................................. 28
Figure 13 : Producer-Consumer I/O Connection [14] ................................ 28
Figure 14 : Explicit Messaging Connection [14] ........................................... 29
Figure 15 : DeviceNetObjects [13] .......................................................... 31
Figure 16 : Adressage DeviceNet [18] ......................................................... 38
Figure 17 : Couches OSI du protocole DeviceNet [18] ................................ 40
Figure 18 : CAN avec la norme ISO 11898 [5] ........................................... 41
Figure 19 : Couche physique du réseau CAN [5] ........................................ 42
Figure 20 : Arbitrage par compétition [14] .................................................. 44
Figure 21 : Trames CAN [4] .................................................................. 45
Figure 22 : Trame de données CAN et le bit stuffing [21] ............................. 47
Figure 23 : Compteur d’erreur et état d’un nœud [13] ................................ 48
Figure 24 : Topologie unique DeviceNet [13] ............................................. 49
Figure 25 : Mappage PLC E/S [24] ......................................................... 51
Figure 26 : Classement des objets - The DeviceNet Specification [13] ........... 52
Figure 27 : Priorités des groupes [13] ......................................................... 53
Figure 28 : Explicit Request Message [13] ................................................. 54
Figure 29 : Explicit Message Response [13] .............................................. 56
Figure 30 : Explicit Message Error Response [13] ....................................... 58
Figure 31 : I/O Poll Request Message [13] ................................................... 60
Figure 32 : I/O Poll Response Message ....................................................... 61
Figure 33 : Connecteurs DeviceNet [13] ...................................................... 65
Figure 34 : Synoptique du banc de test ...................................................... 66
Figure 35 : Topologie du réseau DeviceNet (cf. le chapitre 2.3) .................... 67
Figure 36 : Extrait du programme de test : Générateur RF ......................... 69
Figure 37 : Extrait de la séquence des commandes puissance ...................... 69
Figure 38 : Affichage de la puissance Forward et Reverse ......................... 70
Figure 39 : Extrait du programme de test : Vanne MKS 683 ...................... 71
Figure 40 : Communication entre l’automate et le terminal [24] .................. 73
Figure 41 : BOOTP/Server 2.3 ............................................................... 73
Figure 42 : Acquittement de l’adresse IP sur BOOTP/DHCP ...................... 74
Figure 43 : Lien entre RSLinx et RSLogix 5000 [24] ................................. 75
Figure 44 : Configuration du pilote RSLinx ................................................. 75
Figure 45 : Vérification du pilote RSLinx ..................................................... 76
Figure 46 : Configuration OPC Topic ........................................................ 76
Figure 47 : Advanced Communication OPC Topic ..................................... 77
Figure 48 : Configuration adresse IP statique ......................................... 77
Figure 49 : Réseau visualisé avec RSNetWorx for DeviceNet ..................... 78
Figure 50 : Configuration du mappage .......................................................79
Figure 51 : Configuration des modules sur RSLogix 5000 .........................80
Figure 52 : Affichage de RSNetWorx sur RSLogix 5000 ..........................81
Figure 53 : Vérification du réseau actif.....................................................82
Figure 54 : FactoryTalk Activation Manager.............................................83
Figure 55 : Find Available Activations.......................................................83
Figure 56 : Get New Activations ...............................................................84
Figure 57 : L’automate CompactLogix L23E .............................................86
Figure 58 : Descriptif du connecteur DeviceNet de l’automate [13] ............87
Figure 59 : Le concentrateur DeviceNet ....................................................87
Figure 60 : Configuration de la vanne MKS 683 avec RSNetWorx ..............88
Figure 61 : Vérification de la configuration en sélectionnant le nœud ..........88
Figure 62 : Automate CompactLogix L23E et ses connecteurs .................89
Figure 63 : Vanne MKS 683 ..................................................................90
Index des tables

Tableau 1 : Exemple des principaux équipements gérés par l’ODVA et son numéro Device Profile correspondant ................................................................. 32


Tableau 5 : Le débit en fonction de la longueur [13] ........................................ 43


Tableau 7 : CAN Identifier Field for Predefined Master/Slave Connection Set [13] ........................................................................................................ 53

Tableau 8 : CAN Data Field [13] ......................................................................... 53
Introduction

Une des activités de la société 40-30 est la réparation des sous-ensembles entrant dans des machines de production de semi-conducteurs. Une prestation de réparation comprend le diagnostic, la réparation et le test. Les matériels ou sous-ensembles dans ces machines communiquent avec un port analogique ou avec le protocole RS232 ou encore avec des bus de terrain. Jusqu’à présent, 40-30 a su communiquer avec les ports analogiques et le protocole RS232 pour valider le fonctionnement d’un matériel. Les protocoles de bus de terrain sont quant à eux une nouvelle étape dans les phases de maintenance afin de garantir le bon fonctionnement des matériels lors des tests finals. Le protocole, qui est apparu le plus urgent (demande client) à traiter est le protocole DeviceNet.

Ce stage a donc pour objet la réalisation d’un banc de test DeviceNet et la création des programmes de test permettant de valider les fonctionnalités des équipements. Ce test sera exécuté dès que la phase de réparation est terminée ou que la détection d’un défaut de pilotage est suspectée par le client. Puisque les compétences de l’entreprise ne sont pas suffisantes pour la réalisation de ce banc de test, il a fallu commencer par une étude des principaux bus de terrains existants dans l’industrie afin de choisir pertinemment les outils selon les spécificités à tester.

Ce projet est traité en trois parties : l’état de l’art des réseaux de terrain, l’étude du protocole DeviceNet et la réalisation du banc de test.

Dans la première partie, les bus de terrain présents dans l’industrie sont définis ainsi que les principales méthodes d’accès au medium. Cela nous expose les avantages et les inconvénients de chaque protocole selon ses choix techniques. On mettra en évidence le rôle du modèle théorique OSI appliqué dans les réseaux industriels.

La deuxième partie est l’étude du protocole DeviceNet avec les fondamentaux du protocole Control Area Network. Dans l’échange des données, il est montré le rôle des objets et les types de connexion DeviceNet. Open DeviceNet Vendor Association est l’organisme qui a pour principal objectif de garantir les spécifications du protocole et il est le seul à fournir la dernière mise à jour. Toutes les spécifications rencontrées dans différentes sources de ce rapport ont été comparées avec la dernière publication ODVA (The DeviceNet Specification - novembre 2011).

Enfin dans la troisième partie, les équipements adaptés à la réalisation du banc sont choisis en se basant sur l’étude précédente. On décrit la mise en place de ce dernier : la configuration du matériel et la programmation.
de la séquence de test en tenant compte des ses fonctionnalités. Les tests des sous-ensembles et la mise au point de ces programmes sont également exposés. Les modes opératoires et les programmes de test ont été finalisés une fois que le banc fut opérationnel.

Afin que le travail de recherche et de développement soit capitalisé la rédaction des documents techniques, détaillant les démarches à suivre, sont rédigés : Le mode opératoire pour l’utilisation du banc de test et le mode opératoire pour la prise en charge d’un nouvel équipement. Ce dernier est un support qui aide à la mise en place d’un banc de test pouvant tester tout appareil utilisant le protocole DeviceNet.

Le projet a été réalisé au sein de la société 40-30 dans le site de Bernin (38).
Présentation de la société 40-30

Crée en 1986 par Pierre Delalez et Yves Cotte, 40-30 est basée à Seyssinet-Pariset et assure la maintenance de systèmes industriels. Elle se positionne principalement sur trois domaines techniques:

- La technologie du vide (détecteurs de fuite, métrologie des basses pressions, pompes...).
- Electronique (automatisme, régulation de température, transfert d’équipements et radiofréquences industrielles), contrôles et essais non destructifs.
- Reconditionnements ultra propres.

40-30 est présent dans tous les secteurs d’activités industriels (Semi-conducteurs, aéronautique, agroalimentaire, automobile, chimie, pharmacie, photovoltaïque...) et de la recherche scientifique. La société définit son offre en s’adaptant aux problématiques clients, tels que la pérennisation et l’amélioration de leurs équipements. 40-30 est composé de cinq Business Unit :

- La maintenance
- Les formations
- L’ingénierie
- Spares Center
- Applications Informatiques
Informations générales

Marché clients :

Air Liquide, AREVA, CEA, CNRS, Thales, STMicroelectronics, Schneider Electric…

40-30 en chiffres :

- 210 employés
- 25 000 interventions par an
- 20 millions d’euros de CA

Certifications et accréditations :

- Certification **ISO 14001** depuis 2001
- Accréditation **COFRAC Étalonnage** depuis 1998.
- Centre de formation agréé par la **Cofrend**
- Attestation de capacité **Quali Clima Froid**.
- Agréments constructeurs : Adixen Service Center (level 3), Inficon Service Center, LAM Research

Historique

1986 : Création de 40-30 à Grenoble (France)
1994 : Métrologie de basse pression.
1998 : Cleaning et reconditionnement ultra propre ; Chiller.
2000 : Consolidation des bases arrières ; Radiofréquences industrielles ; Cryogénie.
2005 : Contrôles et essais non destructifs ; Formations.
        40-30 crée 40-30 SEA (South East Asia), filiale 100% implantée à Singapour.
2009 : 40-30 met au point AMPS (Advanced Management Procedures System), un outil informatique de création, de supervision et de contrôle du respect des procédures.
2010 : Création des Business Units : Ingénierie et Sourcing

Actuellement les réseaux de terrain connaissent une forte progression en industrie, comment ces réseaux s’imposent? Comment peut-on définir les bus de terrain?
CHAPITRE 1 ETAT DE L’ART DES BUS DE TERRAIN

1.1 - Les bus de terrain

Un bus de terrain est un système de communication entre plusieurs ensembles communicants (capteurs, microcontrôleurs, actionneurs, mémoires...). Actuellement ils remplacent les boucles analogiques 4-20mA et ils s’adaptent aux réseaux existants dans un périmètre donné. Il existe plus de 50 types de réseaux de terrain et leur finalité est de fournir des informations pertinentes d’entrées/sorties en temps réel. Ce sont des architectures décentralisées dont les objectifs principaux sont : la simplification du câblage et la baisse de coûts. Des tableaux comparatifs des protocoles disponibles sur le marché se trouvent en annexe. Il existe plus d’une cinquantaine de spécifications différentes selon des critères techniques et stratégiques.

Parmi les critères techniques on tient principalement compte du temps (vitesses de transmission, temps de réponse, temps de mise à jour des variables...) et de la topologie (la longueur maximale avec répéteurs, la distance maximale entre équipements...) [1]. D’autres critères techniques peuvent être pris en compte tel que le mode d’accès au réseau, la gestion des priorités, le mode de coopération (Producteur/Consommateur), le mode de transmission, l’efficacité du protocole, la sécurité...).

Finalement voici les critères stratégiques tels que la position par rapport aux standards, la disponibilité des fournisseurs, l’évolutivité, le domaine d’application typique, la compatibilité avec d’autres réseaux, la maintenance, etc. Comment peut-on classifier ces réseaux?

Classification des bus industriels

Les bus de terrain sont classés en trois grandes familles [2], le niveau de complexité des données échangées est leur principal critère :

- Cellule : Il intervient sur les équipements plus complexes tels que des robots et les trames échangées sont plus importantes. Les échanges de 10 à 256 octets se font en quelques secondes.


La pyramide CIM ci-dessous (figure 1) illustre les familles décrites. Elle compare les bus de terrain en mettant en évidence les temps de réaction sur une distance donnée et la taille des données utiles échangées [3]. Chaque protocole est spécifique à certaines applications, comme par exemple ici le protocole DeviceNet gère la communication des modules E/S et le protocole ASI gère les capteurs avec des spécifications et des contraintes différentes.

**Pyramide CIM** : C.I.M. : Computer Integrated Manufacturing

![Diagramme de la pyramide CIM](image)

Figure 1 : Classification des bus de terrain [2]

### 1.1.1 - Les principaux avantages des bus de terrain

Il existe trois principaux avantages par rapport à un réseau classique.

**Economies d’installation** : Moins de fils sont nécessaires par rapport à une connexion directe fil à fil sans utilisation d’un protocole de communication. Il est possible d’utiliser les câbles de l’ancien réseau s’ils sont adaptés aux nouvelles spécifications du futur réseau.
Economies d’entretien : Ces bus sont moins complexes que les conventionnels (connexions fil à fil). Cela les rend plus fiables avec une interaction rapide. Le débogage global peut s’effectuer en ligne.

Amélioration des performances des systèmes : Flexibilité accrue dans la conception du système de bus. La collecte d’information est simplifiée.

Il existe plusieurs moyens d’accès à la couche de liaison de données suivant le protocole de communication [4].

1.1.2 – Les protocoles génériques d’accès

- Accès avec protocole orienté *connexion*

La communication s’établit (établie au préalable par l’acquittement des nœuds actifs par exemple). Un échange de données (envoi d’un message) s’effectue entre le nœud 1 et le nœud 2.

Le message a la possibilité de passer au travers plusieurs nœuds. Voir la figure 2.

Cet accès est utilisé dans les réseaux actuels.

![Figure 2 : Exemple d’accès orienté connexion](image)

- Accès par *polling*

Un message est envoyé aux esclaves (tour à tour) qui répondent si nécessaire. Voir figure 3.

Peu efficace : Problème du nombre d’esclaves.

La communication entre esclave-esclave est difficile.

La robustesse est limitée (en cas de panne du maitre).

Utilisé dans le système PROFIBUS.
• Accès par **multiplexage temporel** : *Time Division Multiple Access*.

Chacun parle tour à tour avec un temps de parole donné.
Plus efficace que le *polling* :
Moins d’activité sur le bus pour la synchronisation.
Taille des données fixe.

• Accès par **jeton sur anneau** (Token Ring).

Topologie en anneau.
Un jeton (*token*) circule sur l’anneau :
Lorsqu’un nœud veut parler, il attend le jeton.
Quand le nœud émet, il garde le jeton.
Inconvénient : ne fonctionne plus si l’anneau est coupé.
Utilisé dans le bus FDDI.
Accès par **dominance bit** (*Binary Countdown*). Utilisé par exemple par le bus de terrain LonWorks

Accès **CSMA/CD** (*Carrier Sense Medium Access/ Collision Detection*)

- Emission lorsqu’il y a un temps mort sur le réseau

  **Carrier Sense** : L’émetteur observe si un message passe sur le réseau.
  **Multiple Access** : Quand il n’y a aucune donnée qui transite plusieurs émetteurs peuvent communiquer.

  - Spécificité

  **Collision Detection** : En cas de collision, on interrompt le transfert et on recommence plus tard avec un délai aléatoire.

    - Utilisé dans les réseaux Ethernet.
    - Non utilisable sur les réseaux sans-fil.

Accès **CSMA/CA** (*Collision Avoidance*) :

- Le nœud attend un blanc avant d’émettre

  **Carrier Sense** : L’émetteur observe si un message passe sur le réseau.
  **Multiple Access** : Quand rien ne transite, plusieurs émetteurs envoient un message RTS (*Ready To Send*).

  - Spécificité
**Collision Avoidance** : Le récepteur choisit un émetteur et bloque les autres jusqu’à la fin de la transmission.

Permet des communications lorsque les émetteurs ne sont pas à portée (sans-fil) : WIFI, Zigbee et LON (*Local Operating Network*).

- **Accès CSMA/BA** (*Carrier Sense Medium Access / Bitwise Arbitration*) :

  **Principe de base** : Lorsque le réseau ne transmet pas de message.
  **Carrier Sense** : L’émetteur observe si un message passe sur le réseau.
  **Multiple Access** : Quand rien ne transite, plusieurs émetteurs envoient un message RTS (*Ready To Send*)

  ▪ Spécificité

  **Bitwise Arbitration** : Chaque nœud a un identifiant et le transmet. En cas de collision, le nœud avec l’identifiant prioritaire émet.

  Utilisé dans le bus CAN (automobile).

Cette dernière technique d’accès est utilisée par le protocole DeviceNet car il utilise une base CAN (le protocole CAN a été inventé par l’entreprise Bosch en 1986) : pour plus de détails se référer au chapitre
2.2). Sur le dessin ci-dessus (figure 6) on compare l'accès CSMA avec le CSMA/NBA dans le cas d'une collision : le premier constat est que CSMA n'est pas déterministe donc le délai de réception n'est pas garanti car CSMA prévoit un temps d'attente aléatoire. Ce qui n'est pas le cas du CSMA qui utilise le moyen d'accès CAN : l'accès ne dépend que de la priorité du message. Ce dernier sera détaillé par la suite (chapitre 2.2.2).

1.2 – RESEAUX INDUSTRIELS

1.2.1 - La communication sur un réseau

Le fondement d'un bon réseau est basé sur les capacités de son système d'exploitation :

- De gérer la transmission de données.
- De fournir aux applications des interfaces standard pour leur permettre d'exploiter les ressources du réseau.

C'est le cas de tous les systèmes d'exploitation à jour. A priori, rien ne devrait obliger les plateformes client et serveur à fonctionner avec le même système d'exploitation. C'était le cas pour les solutions propriétaires mais c'est impensable aujourd'hui. Même si un réseau Microsoft dispose d'outils qui lui sont spécifiques, les hôtes de ce réseau peuvent tout de même dialoguer avec ceux d'un réseau Unix.

Pour arriver à cette interopérabilité, il faut que les divers protagonistes se mettent d'accord sur les fonctionnalités à implanter dans leurs applications et leurs fonctions réseau. C'est le rôle des RFC (Request For Comment) et des normes que de définir ces critères [7].

C'est l'objectif du modèle théorique OSI (1983) qui décrit comment l'OSI réseau, encore appelé NOS19 doit être construit. Il décrit l'architecture en 7 couches logicielles présentant chacune des interfaces standard pour communiquer entre elles.

1.2.2 - Le modèle OSI

Considérations générales

Même si ce modèle reste très théorique [8], il a le mérite d'être le plus méthodique.

Il y a deux points qu'il convient de bien comprendre avant tout :
- Chaque couche est conçue de manière à dialoguer avec son homologue, comme si une liaison virtuelle était établie directement entre elles.

- Chaque couche fournit des services clairement définis à la couche immédiatement supérieure, en s'appuyant sur ceux, plus rudimentaires, de la couche inférieure, lorsque celle-ci existe.

**Description succincte des couches**

![Image de la couche OSI](image)

**Figure 7 : Le modèle OSI [20]**

**La couche physique : 1**

C'est la couche spécifique à la « tuyauterie » du réseau. Elle permet de transformer un signal binaire en un signal compatible avec le support choisi (cuivre, fibre optique, HF etc.) et réciproquement. C'est donc sur cette couche qu'on tient compte de la topologie du réseau et du nombre des nœuds. Le nombre total de connexions est limité par le temps de retard (due à la propagation).

Le délai de propagation : La vitesse de transfert dépend directement de la longueur du réseau. L'atténuation de la propagation est accentuée avec des fréquences élevées (Effet de peau ~ $1/f^2$ : cf. *annexe F*) et avec une désadaptation d’impédance de la ligne de transmission (Impédance caractéristique).

Les critères à tenir en compte avant la mise en place du réseau :
- Topologie (bus, boucle, arbre, étoile, etc.),
- type de support de transmission (paire torsadée, câble coaxial, ...),
- redondance du support de transmission,
- alimentation des nœuds par les lignes de communication,
- type d’isolement,
- précaution d’installation,
- conditions environnementales (température, feu, nucléaire, choc thermique ou mécanique, corrosion, ...),
- type de codage de bit (NRZ, NRZI, Manchester, etc.),
- type de transmission (en bande de base ou avec modulation),
- type de modulation (amplitude, phase, fréquence, ...),
- longueur maximale du réseau avec ou sans répéteurs,
- distance maximale et minimale entre deux nœuds voisins,
- débit maximum,
- sécurité de transmission [9].

Cette couche fournit des outils de transmission de bits à la couche supérieure, qui les utilisera sans se préoccuper de la nature du médium utilisé.

La couche liaison : 2

Cette couche assure le contrôle de la transmission des données. Une trame doit être envoyée ou reçue en s'affranchissant d'éventuels parasites sur la ligne. Le contrôle est effectué au niveau du paquet de bits (trame), au moyen d'un "checksum". C'est à ce niveau que se situe l'adresse MAC (Medium Access Control).

Elle est elle-même divisée en deux sous-couches.

• La sous-couche MAC (Medium Access Control). C'est à ce niveau que l'on trouve le protocole de diffusion de l'information: Ethernet, Token Ring, ATM, etc. Ethernet est utilisé pour un réseau domestique.

• Par ailleurs, cette couche fournit des services de base pour la transmission de données, via LLC (Logical Link Control) ou HDLC (High level Data Link Control).

Ces services peuvent être classés en trois groupes :

• Les services sans connexion et sans acquittement.
• Les services sans connexion, mais avec acquittement.
• Les services orientés connexion.

Nous aurons l'occasion de développer ces notions par la suite. Le délai d'attente est la principale caractéristique de cette dernière.
Voici les caractéristiques disponibles de cette couche à choisir selon le besoin spécifique :

- méthode d’accès au réseau (maître/esclave, jeton, etc.),
- type de communication (point à point, diffusion),
- type d’adressage (objet ou nœud),
- taille des messages et de l’information utile dans les messages,
- garantie ou non de temps de communication borné,
- mode de connexion (avec ou sans),
- mécanisme d’acquittement, mécanisme de contrôle d’erreur,
- mécanisme de contrôle de flux.

Cette couche fournit des outils de transmission de paquets de bits (trames) à la couche supérieure. Les transmissions sont "garanties" par des mécanismes de contrôle de validité.

**La couche Réseau : 3**

Cette couche assure la transmission des données sur les réseaux. C'est ici que la notion de routage intervient, permettant l'interconnexion de réseaux différents. C'est dans le cas de TCP/IP20 la couche IP (*Internet Protocol*). En plus du routage, cette couche assure la gestion des congestions. Il faudrait développer davantage ce chapitre pour être clair. Disons simplement que lorsque les données arrivent sur un routeur, il ne faudrait pas que le flot entrant soit plus gros que le flot sortant maximum possible, sinon il y aurait congestion. Une solution consiste à contourner les points de congestion en empruntant d'autres routes (phénomène bien connu des vacanciers sur les routes).

Le problème de la congestion est un problème épineux, auquel il nous arrive assez souvent hélas d’être confronté. Cette couche est la plus haute dans la partie purement « réseau ».

Cette couche fournit des outils de transmission de paquets de bits (trames) à la couche supérieure. Les transmissions sont routées et la congestion est contrôlée.

**La couche Transport : 4**

Cette couche apparaît comme un superviseur de la couche réseau. Qu'est-ce que cela signifie? Il n'est par exemple pas du ressort de la couche réseau de prendre des initiatives si une connexion est interrompue. C'est la couche transport qui va décider de réinitialiser la connexion et de reprendre le transfert des données.

Son rôle principal est donc de fournir à la couche supérieure des outils de transport de données efficaces et fiables.
La couche Session : 5

La notion de session est assez proche de celle de connexion. Il existe cependant quelques détails qui peuvent justifier la présence de ces deux concepts.
Une seule session peut ouvrir et fermer plusieurs connexions, de même que plusieurs sessions peuvent se succéder sur la même connexion.
Comme cette explication n'est pas suffisamment claire, essayons de prendre quelques exemples [6] :

• Vous avez un message à transmettre par téléphone à un de vos amis, votre épouse doit faire de même avec celle de ce même ami.
Vous appelez votre ami (ouverture d'une connexion), vous discutez avec lui un certain temps (ouverture d'une session), puis vous lui dites que votre épouse voudrait parler à la sienne (fermeture de la session).
Les épouses discutent un autre certain temps (ouverture d'une seconde session), puis n'ont plus rien à se dire (fermeture de la seconde session) et raccrochent (fin de la connexion).

Dans cet exemple, deux sessions ont eu lieu sur la même connexion.

• Vous avez un travail à réaliser avec un collègue, par téléphone. Vous l'appelez (ouverture de la connexion et ouverture de la session). Il vous demande des informations qui nécessitent de votre part une recherche un peu longue, vous raccrochez après lui avoir dit que vous le rappellerez ultérieurement (fermeture de la connexion, mais pas de la session).
Votre recherche effectuée, vous rappelez votre collègue (ouverture d'une seconde connexion pour la même session), vous lui transmettez les informations demandées, vous n'avez plus rien à vous dire (fermeture de la session), vous raccrochez (fermeture de la connexion). Dans cet exemple une session s'étend sur deux connexions.

Cette couche fournit donc à la couche supérieure des outils plus souples que ceux de la couche transport pour la communication d'informations, en introduisant la notion de session.

La couche Présentation : 6

Cette couche est un peu un « fourre tout » de la conversion entre représentation interne et externe des données. Là encore, cette explication n'est pas d'une grande clarté... Prenons donc quelques exemples.

Le format de codage interne des données

Les « mots » sont une suite d'octets. Un mot de 32 bits est donc une collection de 4 octets. Le fondeur de silicium Intel, numérote les octets
des ses composants de droite à gauche, alors que le fondeur Freescale utilise la numération de gauche à droite. Il s'en suit que si une machine à base Intel envoie des mots à une machine à base Freescale, il vaut mieux tenir compte de ce détail pour ne pas s'y perdre...
D'autres exemples des spécifications comme la complémentation à 1 ou à 2 pour les représentations négatives, les divers dialectes ASCII, etc.

La compression des données

Certains transferts de données se font par le biais d'algorithmes de compression. C'est intéressant pour certains types de documents comme le son, l'image ou la vidéo. Dans le modèle OSI, ces algorithmes sont fournis par la couche présentation.

Le cryptage des données

Même chose pour la cryptographie. Ce ne sont que des exemples, le modèle définissant bien d'autres fonctions.

La couche Application : 7

Il s'agit de services offerts au niveau application. A ce niveau, certains réseaux de terrain n’offrent aucun service, d’autres offrent des services de base de type lecture ou écriture de variable à distance et d’autres encore offrent des applications quasiment toutes faites et qu’il faut seulement paramétrer pour un usage donné. Les intérêts et besoins au niveau application par (ex : Le délai de traversée) sont très diversifiés, de sorte qu’on est encore loin d’un consensus pour avoir une norme de services de niveau application pour les réseaux de terrain. Par ailleurs, les services application fournis par réseaux de terrain sont très diversifiés et peu stables pour les utiliser aujourd’hui comme critères de comparaison sans rentrer dans les détails de fonctionnement de ces services.

A priori, cette couche pourrait être la plus simple à comprendre, ce n’est pas obligatoirement le cas. En effet, dans le modèle OSI, cette couche propose également des services: Principalement des services de transfert de fichiers, (FTP), de messagerie (SMTP) de documentation hypertexte (HTTP) etc.
Dans le modèle, les applications ayant à faire du transfert de fichiers utilisent le service FTP fourni par la couche 7 [8].
La figure 8 schématisé le parcours d’un message : c’est sur la couche applicative que la commande ou la requête est produite. Ensuite ce message traverse toutes les couches où s’ajoute un entête (encapsulation d’un message) qui sera retiré progressivement à la réception afin d’avoir uniquement la partie utile sur la couche applicative (la traversé du message est représenté para le trait rouge).

Ce modèle théorique, extrêmement détaillé est fait pour que chaque couche puisse être construite indépendamment des couches qui sont immédiatement au dessus et au dessous d’elle. Mais en pratique les protocoles de communication n’utilisent pas la totalité des couches.
1.2.3 - Le bus terrain est une version simplifiée du modèle OSI

Ces bus n’utilisent pas d’interconnections vers d’autres réseaux. Et cela représente un gain de performances (réduction du nombre de formatages successifs). Les couches 1, 2 et 7 représentent typiquement un bus de terrain.

La présence d’une couche utilisateur et d’un bloc de supervision (norme ISA SP50) permettent une répartition des données et un contrôle distribué. Ainsi que la configuration et le monitoring à distance.

Le schéma nous illustre les différentes techniques d’accès au MAC des réseaux de terrain. C’est en fonction de ces options qu’un protocole est privilégié par rapport à un autre. A titre d’exemple le protocole DeviceNet utilise un accès non contrôlé, car il est aléatoire. Et les résolutions des collisions s’effectuent par forçage. Elle est résolue par compétition c’est-à-dire en tenant compte de la priorité la plus élevée [9].

Figure 9 : Classification des techniques d’accès MAC [4]

La figure 9 nous expose les différentes techniques d’accès des protocoles bus de terrain. Dans le type d’accès il peut être non contrôlé ou contrôlé,
c’est-à-dire que le message peut apparaître d’une façon aléatoire ou périodique avec un intervalle de temps prédéfini. Les messages non contrôlés sont soumis directement à une résolution de collision avec forçage (cf. CAN chapitre 2.2) ou non (cf. chapitre 1.2 proto- 
coles génériques). C’est la technique d’accès par **compétition** : réseau multi-
maîtres de type producteur/consommateur, au sein duquel des infor-

**Dans la technique d’accès contrôlé par **élection** avec le droit de parole géré par une station ou par coopération entre stations obéit au principe de fonctionnement suivant : réseau à arbitre de bus de type producteurs/distributeur/consommateurs, au sein duquel l’arbitre pilote les échanges selon sa table de scrutin. L’intégrité des données est assurée par datation. Une donnée à une durée de vie limitée, pas de file d’attente mais un mécanisme d’actualisation (anciennes valeurs remplaçées par nouvelles, si non corrompues).**

**1.3 – L'ODVA**

**1.3.1 - A propos de l'ODVA**

L'ODVA est une association internationale des leaders mondiaux de l'automatisation. Ensemble, l'ODVA et ses membres soutiennent les technologies des réseaux basées sur le protocole industriel commun (CIP). Elles sont actuellement DeviceNet, EtherNet/IP ainsi que les principales extensions au CIP, à savoir CIP Safety, CIP Sync et CIP Motion. L'ODVA gère le développement de ces technologies ouvertes et aide fabricants et utilisateurs de technologies CIP par le biais d'outils, de stages de formation et d'activités de marketing. En outre, l'ODVA propose des essais de conformité pour assurer que les produits construits selon ses spécifications fonctionnent dans les systèmes multimarques.

L'organisme est également actif dans de nombreux groupes et consortiums industriels travaillant à l'évolution des normes de communication ouverte.

**1.3.2 - Common Industrial Protocol**

*Control Industrial Protocol (CIP)* est une technologie orientée objet couvrant les couches Session, Présentation et Application du modèle OSI. Pour l’utilisateur, cela se traduit par une transparence du type de réseau utilisé. Les développeurs d’application n’ont même pas à se soucier de la nature de réseau sur lequel sont connectés les équipements qu’ils programment. CIP définit également des profils d’équipements standards, qui identifient les objets, options de
configuration et formats de données d’E/S pour différents types d’équipements. Les équipements possédant le même profil standard répondront aux mêmes commandes, et auront le même comportement sur le réseau.

Un autre élément caractéristique important des réseaux basés sur CIP est leur capacité à générer des messages à partir d’un réseau, puis à les transmettre vers un autre réseau de façon transparente. Cela signifie qu’un ensemble d’objets inclus dans la spécification des CIP définissent les mécanismes pouvant être utilisés par un équipement de routage ou par une passerelle. Ils peuvent transmettre un message d’un port réseau vers un autre sans agir sur le contenu du message lui-même. L’un de ses éléments distinctifs est sa capacité à atteindre de grandes vitesses pour la transmission de données de contrôle d’entrées-sorties, tout en restant capable d’échanger des messages explicites avec les couches de niveaux supérieurs.


CompoNet, lancé en 2007, est la dernière génération. Il utilise le Time Division Multiple Access (TDMA) qui permet de gérer la présence de plusieurs signaux sur un même canal. Le multiplexage temporel découpe le temps disponible sur un cycle pour qu’il soit reparti avec plusieurs utilisateurs. Le temps de réponse d’une tâche est ainsi optimisé à 1 ms. Toute la gamme ControlLogix de Rockwell Automation utilise le protocole CIP. Le diagramme ci-dessous (figure 10) montre la relation entre les différentes couches de la famille CIP et les couches ISO. Il nous montre en couleur grise un exemple d’équipement géré par trois différents types de protocole CIP. Les couches supérieures CIP (violet, bleu et rose) sont communes à ces derniers. Par contre les couches inférieures sont spécifiques à chaque protocole : EtherNet (vert), ControlNet (bleu) et DeviceNet (rouge). C’est l’organisme ODVA qui certifie ces spécifications.
Le CIP a été conçu pour palier aux besoins d’automatisation dans l’industrie. Voici les spécifications garanties par l’ODVA :

- **Object modelling**
- **Messaging protocol**
- **Communication objects**
- **General object library**
- **Device profiles**
- **Device configuration**
- **Services**
- **Data management**

**Le modèle Objet**

Le réseau CIP utilise des objets (objects) pour modéliser les nœuds et les services de communication. Pour chaque nœud modélisé on associe des objets. Ils appartiennent à une classe d’objets (class of objects) qui partagent la même liste d’attributs (attributes) et des comportements (behaviours).
Un objet est une instance de cette classe avec ses propres et uniques valeurs d’attributs. Un nœud peut contenir plus d’un objet de la même classe.

Les objets et les nœuds utilisent un adressage standard avec les éléments suivants :

- **MAC ID** - désigne chaque nœud dans le réseau CIP
- **Class ID** - désigne chaque classe de l’objet dans le réseau
- **Instance ID** - désigne une instance spécifique (objet) de la classe
- **Attribute ID** - désigne un attribut de la classe ou l’objet
- **Service Code** – identifie un comportement spécifique de la classe ou de l’objet.

La figure 11 nous montre en rouge la représentation type d’un objet CIP dans un réseau actif : Le service 2 (son numéro de série par exemple) de l’équipement 4 se trouve dans l’objet MAC ID 4/Class 5/Instance 1/Service 2.

**Figure 11: Exemple d’adressage CIP [14]**

**Le protocole Messaging**

Une connexion CIP fournie la « clé » (*Path*) entre de multiples applications. C’est le chemin qui permet d’obtenir la donnée demandée. Quand la communication est établie un *CAN Connection ID* est attribué. Dans le cas d’un échange bidirectionnel deux
Connections ID seront assignés. Le graphique nous montre la relation entre les Application Objects, les Connections Objects et les Connection IDs.

Figure 12 : Application Object et Connection ID [14]

Les connexions E/S fournissent un Path dédié pour des applications spécifiques entre une application de type producteur et un consommateur (plusieurs consommateurs peuvent intervenir). Les messages explicites utilisent un Path générique entre deux équipements pour une communication orientée réponse-requête, référencés comme des messagings connections.

Figure 13 : Producer-Consumer I/O Connection [14]

**Communication objet**

Ils sont utilisés pour un échange de messages. Chaque communication object contient un link producer et ou un link consumer. Les connexions Explicit messaging sont soit productrices soit consommatrices.
Les valeurs des attributs lors d’une connexion objet décrivent des paramètres. Ils déterminent le type de connexion E/S et l’Explicit messaging. Comme par exemple la taille maximum d’un message, la source et la destination. Elles donnent aussi l’état de la connexion, comme le changement d’état d’un équipement ou éventuellement un intervalle de temps prédéterminé dépassé [10].

**General object library**

Les protocoles de la famille CIP contiennent une large gamme d’objets prédéfinis.

- *General use objects* : Librairie généraliste utilisée par de nombreux équipements.
- *Application specific objects* : Utilisés exclusivement par des applications spécifiques.
- *Network specific objects* : Utilisés exclusivement sur un seul type de réseau.

Voici la liste des objets prédéfinis existants :

**General Use Objects**:

<table>
<thead>
<tr>
<th>Identity Object</th>
<th>Parameter Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Router Object</td>
<td>Parameter Group Object</td>
</tr>
<tr>
<td>Assembly Object</td>
<td>Acknowledge Handler Object</td>
</tr>
<tr>
<td>Connection Object</td>
<td>Connection Configuration Object</td>
</tr>
<tr>
<td>Connection Manager Object</td>
<td>Port Object</td>
</tr>
</tbody>
</table>
Application Specific Objects:

<table>
<thead>
<tr>
<th>Discrete Input Point</th>
<th>Position Controller Object Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Object</td>
<td>Sequencer Object</td>
</tr>
<tr>
<td>Discrete Input Point Object</td>
<td>Command Block Object</td>
</tr>
<tr>
<td>Discrete Output Point Object</td>
<td>Motor Data Object</td>
</tr>
<tr>
<td>Analog Input Point Object</td>
<td>Control Supervisor Object</td>
</tr>
<tr>
<td>Analog Output Point Object</td>
<td>AC/DC Drive Object</td>
</tr>
<tr>
<td>Presence Sensing Object</td>
<td>Overload Object</td>
</tr>
<tr>
<td>Group Object</td>
<td>Softstart Object</td>
</tr>
<tr>
<td>Discrete Input Group Object</td>
<td>Selection Object</td>
</tr>
<tr>
<td>Discrete Output Group Object</td>
<td>S-Device Supervisor Object</td>
</tr>
<tr>
<td>Discrete Group Object</td>
<td>S-Analog Sensor Object</td>
</tr>
<tr>
<td>Analog Input Group Object</td>
<td>S-Analog Actor Object</td>
</tr>
<tr>
<td>Analog Output Group Object</td>
<td>S-Single Stage Controller Object</td>
</tr>
<tr>
<td>Analog Group Object</td>
<td>S-Gas Calibration Object</td>
</tr>
<tr>
<td>Position Sensor Object</td>
<td>Trip Point Object</td>
</tr>
<tr>
<td>Position Controller Supervisor Object</td>
<td></td>
</tr>
</tbody>
</table>

La liste complète des **Object Specifications** se trouve dans :

Volume 1 : Common Industrial Protocol Specification  
Chapter 5 : Object Library  
Table 5-1.1 Object Specifications in the CIP Object Library  
Page 181/1418 du CD *The DeviceNet Specification November 2011*  
[13]

**Network Specific Objects**:

DeviceNet Object (specific to DeviceNet)  
ControlNet Object (specific to ControlNet)  
ControlNet Keeper Object (specific to ControlNet)  
ControlNet Scheduling Object (specific to ControlNet)  
TCP/IP Interface Object (specific to EtherNet/IP)  
Ethernet Link Object (specific to EtherNet/IP)

**Identity Object Attributes :**

<table>
<thead>
<tr>
<th>Mandatory</th>
<th>Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vendor ID</td>
<td>State</td>
</tr>
<tr>
<td>Device Type</td>
<td>Configuration Consistency Value</td>
</tr>
<tr>
<td>Product Code</td>
<td>Heartbeat Interval</td>
</tr>
<tr>
<td>Revision</td>
<td>Languages Supported</td>
</tr>
<tr>
<td>Status</td>
<td></td>
</tr>
<tr>
<td>Serial Number</td>
<td></td>
</tr>
<tr>
<td>Product Name</td>
<td></td>
</tr>
</tbody>
</table>
Device profiles

Des équipements similaires qui présentent quelques différences dans la structure interne et dans le fonctionnement ont été groupés. Ils sont associés à un Device Profile. Ce profil permet de faciliter les applications CIP sur l’équipement. Il contient toute la description de l’objet tel que la structure et le comportement. Voici un extrait de la table des différents types d’équipements et leur profil associé :

Tableau 1 : Exemple des principaux équipements gérés par l’ODVA et son numéro Device Profile correspondant

<table>
<thead>
<tr>
<th>Device Profile</th>
<th>Device Type No</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC Drives</td>
<td>02kex</td>
</tr>
<tr>
<td>Communications Adapter</td>
<td>0Chex</td>
</tr>
<tr>
<td>Contactor</td>
<td>15kex</td>
</tr>
<tr>
<td>DC Drives</td>
<td>13kex</td>
</tr>
<tr>
<td>DC Power Generator</td>
<td>1Fhex</td>
</tr>
<tr>
<td>General Purpose Discrete I/O</td>
<td>07kex</td>
</tr>
<tr>
<td>Generic Device</td>
<td>00kex</td>
</tr>
<tr>
<td>Human-Machine Interface</td>
<td>18kex</td>
</tr>
<tr>
<td>Inductive Proximity Switch</td>
<td>05kex</td>
</tr>
<tr>
<td>Limit Switch</td>
<td>04kex</td>
</tr>
<tr>
<td>Mass Flow Controller</td>
<td>1Ahex</td>
</tr>
<tr>
<td>Motor Overload</td>
<td>03kex</td>
</tr>
<tr>
<td>Motor Starter</td>
<td>16kex</td>
</tr>
<tr>
<td>Photoelectric Sensor</td>
<td>06kex</td>
</tr>
<tr>
<td>Pneumatic Valve(s)</td>
<td>1Bhex</td>
</tr>
<tr>
<td>Position Controller</td>
<td>10kex</td>
</tr>
<tr>
<td>Process Control Valve</td>
<td>1Dhex</td>
</tr>
<tr>
<td>Residual Gas Analyzer</td>
<td>1Ehex</td>
</tr>
<tr>
<td>Resolver</td>
<td>09kex</td>
</tr>
</tbody>
</table>

Chaque profil présente une liste d’objets (obligatoires et/ou optionnels) et un comportement associé à un type d’équipement. La plupart des profils définissent plusieurs formats de données E/S.

La liste complète des Device Profiles se trouve dans :

Volume 1 : Common Industrial Protocol Specification
Chapter 6 : Device Profiles
Table 6-7.1 Device Profiles in this Chapter
**Device configuration**

Le CIP utilise différentes méthodes pour configurer un équipement :

- *Printed data sheet* – C’est l’outil qui fourni à l’utilisateur des informations nécessaires à la configuration. Il relaye les informations de l’utilisateur.


- *Electronics Data Sheet (EDS)* – Contient toute l’information des paramètres objets. Il peut être utilisé simultanément avec le *parameter object stub*, afin de permettre toutes les fonctionnalités d’un objet sans que l’équipement soit surchargé avec tous les entêtes.

- *Configuration Assembly* – Une fois que la configuration a été transférée à l’équipement ou chargée (avec les paramètres de configuration déjà connus) la *configuration assembly* peut être utilisée en conjonction avec toutes les autres méthodes décrites ci-dessus.
**Services**

Ils définissent les actions d’un message reçu correspondant à une partie de l’objet ou à sa totalité. Outre les fonctions de lecture et d’écriture, un ensemble de services communs CIP (**CIP Common Services**) a été défini pour les objets. Il existe aussi un certain nombre de codes de service spécifiques à l’objet (**object-specific service codes**) qui peuvent avoir différentes fonctionnalités. Les fournisseurs peuvent obtenir des codes spécifiques (**vendor-specific service code**), bien qu’ils ne soient pas réutilisables par d’autres équipements.

**Data management**

Les spécifications CIP, décrivent les modes d’adressage du CIP et les structures des données de l’équipement. Les adresses sont référencées avec des « segments » ; ils se divisent en deux catégories : le premier octet d’un segment CIP détermine le type : le segment logique **logical segment** (0x20-0x3F) et le type de donnée **data type** (0xA0-0xDF). Les segments logiques ce sont des adresses segmentées, c’est-à-dire que chaque segment (class, instance et attribut) correspond à une spécification. Elles sont utilisées pour adresser les objets et ses attributs. Voici la structure type :

```
[Class ID] [Instance ID] [Attribute ID (if required)]
```

Chaque élément de la structure peut contenir jusqu’à quatre octets. Ce type d’adressage est communément utilisé pour paramétrer un équipement ou simplement pour établir une connexion. C’est l’adressage le plus répandu des EDS et des **Unconnected Messages**. Les données peuvent être de type structuré ou élémentaire. Les données structurées sont représentées avec des tableaux contenant des données de type élémentaires, ou avec tout autre tableau des données (**assembly**). Les données de type élémentaires sont utilisées dans les fichiers EDS. Elles spécifient le type de données des paramètres et d’autres informations **[11]**.
Avantages du CIP

Le CIP utilise un langage de programmation à objet, il décrit les nœuds et les services dans le réseau CIP, à condition qu’il soit implémenté avec une technologie compatible (DeviceNet, ControlNet, EtherNet/IP, CompoNet, Modbus/TCP). Cela favorise le transfert du savoir-faire et de l’expertise ainsi que le remplacement ou la mise à jour du réseau existant. Il apporte finalement la réduction des coûts liée au temps de développement et à la formation du personnel. Le firmware (programme de l’automate) peut être réutilisé sans recourir à des changements conséquents. L’interconnexion des systèmes utilisant la base CIP est relativement simple, ils utilisent le même langage et les profils des équipements sont identiques sur tous les systèmes. Il est très évolutif, le mécanisme producteur-consommateur et l’architecture open object dans la famille des protocoles CIP rendent l’utilisation de la bande passante plus efficace sur les couches inférieures du réseau. L’ODVA garantit toutes les spécifications à condition qu’elles soient communiquées par ses membres. L’ODVA distribue les spécifications du protocole exclusivement à ses membres mais une demande justifiée d’un projet R&D m’a permis d’obtenir *The DeviceNet Specification November 2011*.

Ce chapitre nous donne une vue d’ensemble des bus de terrain et des leurs spécificités, cela m’a permit de situé DeviceNet et les protocoles équivalents. Le chapitre suivant expose ce protocole avec toutes ses principales spécificités. Un tableau comparatif des bus de terrain est en *annexe A*. Ce dernier nous expose les protocoles existants avec leurs principales particularités. Il peut être très utile pour la mise en place d’un réseau car les spécificités suivantes sont comparées : technique d’accès, débit maximum, taille de trames, topologie, support de transmission, nombre maximum de nœuds, distance maximale, constructeur, standards et principaux secteurs d’utilisation.
CHAPITRE 2 ETAT DE L’ART DU PROTOCOLE DEVICENET

2.1 - Le protocole DeviceNet

Crée en 1994 par Rockwell Automation (Allen Bradley), DeviceNet couvre les bus de terrain et les bus de capteurs/actionneurs. Il est capable de gérer simultanément des E/S distribuées et des équipements intelligents (variateurs de vitesse...), aux PLC. Les applications se situent principalement dans les secteurs de l’automobile, la métallurgie, l’agro-alimentaire. DeviceNet permet d’interconnecter jusqu’à 64 stations et propose 3 vitesses de transmission avec une longueur de bus comprise entre 100 et 500m. Le support est un câble 4 fils (2 pour la communication, 2 pour l’alimentation 24V, 8A).

DeviceNet fait appel à la technologie CAN pour la transmission des données. Sur DeviceNet, les débits de transmission disponibles sont de 125, 250 ou 500 K bauds. Le protocole divise l’identifiant CAN en 4 groupes de messages et leur alloue des tâches basées sur le principe client-serveur. La procédure est orientée connexion. Cela signifie qu’avant de pouvoir échanger quelque information que ce soit, il faut impérativement qu’une connexion soit établie entre 2 nœuds. Chaque équipement DeviceNet possède donc pour cela soit un Unconnected Message Manager (UCMM pour connexions multiples ou connexions sans port de communication), soit un port non-connexe. Ces deux types réservent des fonctions au moyen de certains identifiants CAN selon l’adresse équipement. Cet identifiant permet d’ouvrir le dénommé « canal explicite » qui permet exclusivement un échange entre les nœuds ayant la même priorité. Des informations sont ensuite échangées via ce canal sur la connexion d’E/S à établir [12].

**Volume 1** : Le protocole de communication et la couche applicative.
- CAN : La couche de liaison de données
  - La couche physique

**Volume 2** : Description de profils des équipements

DeviceNet connecte tout type d’équipements : Les plus simples tels que des capteurs et des actionneurs ainsi que les plus complexes tels que des PLC. L’adresse de données (*I/O data format*) comporte quatre parties : *None Address* (MAC ID), *Class*, *Instance* et *Attribute*. Cela nous indique la nature de la commande.

- **NodeAddress** : [0…63]
- **Class ID** : [1…255]
- **Instance ID** : [0…255]
- **Attribute ID** : [1…255]

Les variables dans un nœud communiquent en utilisant une clé (*Path*). Elle est composée par une *classe ID*, une *instance ID* et un *attribute ID* [13]. La figure 16 représente l’adressage DeviceNet qui est l’équivalent de la figure 11 (adressage CIP). L’attribut #1 peut représenter par exemple la position du clapet d’une vanne 683. Un réseau DeviceNet avec ses équipements actifs est représenté : l’équipement est identifié avec le MAC ID #4 (vanne MKS 683 par exemple). Le chemin pour atteindre la
valeur de l’attribut#1 est le suivant MAC ID #4 : Object class #5 / Instance #2 / Attribute #1.

Figure 16 : Adressage DeviceNet [18]

MAC ID représente chaque nœud dans le réseau. 
Class ID représente chaque classe d’objet. 
Instance ID représente une instance (objet) spécifique de la classe. 
Attribute ID représente un attribut de la classe ou l’objet.

Exemple :

Voici la configuration (Data Path) d’un paramètre utilisé par un onduleur moteur.

Param2=

0, 6, "20 28 24 01 30 06", 0x0004, 2, 2, 
"Motor Rated Curr", "A", 
10, 2000, 36,

$ Group : Motor Data, Code Num : 2 
$ Reserved 
$ Path size, Logical address(Class:40, Instance:1, Attribute:6) 
$ Descriptor (Using Scaling, ) 
$ Data type (UINT), Data size 
$ Parameter name 
$ Unit 
$ Help 
$ Min, Max, Default data values
Banc de test DeviceNet

1, 10, 1, 0, $ Mul, Div, Base, Offset scaling
0, 0, 0, 0, $ Mul, Div, Base, Offset link
1; $ Decimal precision

Les octets class ID, instance ID et attribute ID nous indiquent le format de sa valeur (un octet) et la class, instance et attribute ce sont ses valeurs. Dans le paramètre 2 on le retrouve dans le logical adresse (20 28 24 01 30 06), c’est le chemin à suivre pour retrouver la valeur du courant moteur. Le reste des valeurs représente les unités utilisées et les limites de ce courant.

Le fichier EDS commenté se trouve en annexe B. Tout fichier de ce type peut être visualisé avec le logiciel Wordpad de Windows ou avec le logiciel EZ-EDS de l’ODVA.

DeviceNet est capable de gérer simultanément des communications client-serveur (master-slave) et producteur/consommateur (peer to peer). Tous les équipements esclaves se trouvent sur un seul bus. Ce qui élimine considérablement la quantité de câbles utilisés par rapport à une connectique standard [14].

Voici dans les tableaux 3 et 4 les caractéristiques succinctes de l’implémentation des nœuds esclave et maitre qui servent à une mise en place rapide du protocole.


<table>
<thead>
<tr>
<th>Caractéristiques</th>
<th>Valeur</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connexions</td>
<td>Bit-strobe, Change of state, Cyclic, Polling, Explicit peer-to-peer messaging</td>
</tr>
<tr>
<td>UCMM</td>
<td>non pris en charge</td>
</tr>
<tr>
<td>Fragmentation</td>
<td>Explicite et E/S</td>
</tr>
<tr>
<td>Image process</td>
<td>max. 510 octets</td>
</tr>
<tr>
<td>Connexion explicite</td>
<td>Services reset, Get and Set-Attribute, jusqu’à 240 octets par message transférés à l’application</td>
</tr>
</tbody>
</table>

Tableau 4 : Caractéristiques de l’implémentation d’un nœud maitre DeviceNet [13]

<table>
<thead>
<tr>
<th>Caractéristiques</th>
<th>Valeur</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equipements</td>
<td>63 maxi.</td>
</tr>
<tr>
<td>Connexion</td>
<td>Bit-strobe, Change of state, Cyclic, Polling, Explicit peer-to-peer messaging</td>
</tr>
<tr>
<td>UCMM</td>
<td>pris en charge</td>
</tr>
<tr>
<td>Fragmentation</td>
<td>Explicite et E/S</td>
</tr>
<tr>
<td>Connexion explicite</td>
<td>Services reset, Get and Set-Attribute, jusqu’à 255 octets par connexion E/S</td>
</tr>
<tr>
<td>Image process</td>
<td>max. 7 Koctets</td>
</tr>
</tbody>
</table>
En tenant compte du modèle OSI, on constate que le protocole DeviceNet ne couvre qu’une partie de la couche physique et la totalité de la couche applicative. Les spécifications du protocole CAN seront développées, car c’est la base du DeviceNet [15].

La couche 1 est principalement spécifique à DeviceNet (avec sa connectique et l’alimentation du bus 24VDC). La partie de la couche 1 proche de la couche 2 et la totalité de la couche suivante appartient au protocole CAN avec ses principales spécificités tel que sa détection d’erreurs et son moyen d’accès par priorité qui constitue la force. Ces deux fonctions garantissent une fiabilité de transfert de données en temps réel et le déterminisme de DeviceNet. La couche 7 est exclusivement applicative, elle est constitué de la sémantique CIP commune à plusieurs protocoles (cf. le chapitre 1.3.2).

Figure 17 : Couches OSI du protocole DeviceNet [18]
2.2 - Control Area Network

A l’origine, le protocole CAN a été développé par la société Robert BOSCH GmbH en 1983 pour l’industrie automobile européenne avec pour ambition de remplacer le câblage relativement coûteux des automobiles par un câble de réseau beaucoup moins onéreux. Le protocole a été officiellement lancé en 1986 et les premiers contrôleurs CAN, produits par Intel et Phillips, ont été commercialisés en 1987. Le protocole CAN se distingue par un temps de réaction extrêmement rapide et une fiabilité de haut niveau, caractéristiques indispensables pour assurer, par exemple, le déclenchement du système ABS ou des airbags. A partir de ce protocole, une norme de communication a été développée pour une application en milieu industriel et notamment dans les automatismes.

Le protocole CAN (Control Area Network) est un protocole de communication série qui supporte des systèmes temps réel avec un haut niveau de fiabilité. Ses domaines d’application s’étendent des réseaux moyens débits aux réseaux de multiplexages faibles coûts. Il est avant tout à classer dans la catégorie des réseaux de terrain utilisé dans l’industrie pour remplacer la boucle analogique 4-20mA [16].

La figure 18 nous montre la couche physique de ce protocole qui est la même sur DeviceNet : le niveau récessif « 1 » et le niveau dominant « 0 » avec les tensions correspondantes ; CAN H (high) et CAN L (low) sont les lignes de transmission différentielles où tous les messages transitent, DeviceNet utile deux fils supplémentaires pour que l’alimentation 24VDC.
Avec le bus CAN, les stations ayant les mêmes droits (organes de commande, capteurs ou actionneurs) sont reliées par un bus série. Le protocole CAN de base leur permet d'échanger 2048 variables. Ce protocole, ainsi que les paramètres électriques de la ligne de transmission, sont fixés par la norme 11898. La transmission physique s'effectue avec une paire torsadée, par liaison infrarouge, par liaison hertzienne ou par fibre optique.

**Couplage physique du réseau CAN**

![Couplage physique du réseau CAN](image)

La *figure 19* représente (par rapport à la figure 18) plus en détail les composants contenus des nœuds CAN pris en charge par le réseau.

À la différence du faisceau de câbles, le réseau détecte et corrige, grâce à son protocole, les erreurs de transmissions induites par les radiations électromagnétiques. L'organisation en réseau apporte aussi une configuration aisée du système et la possibilité d'établir un diagnostic central. Elle permet à chaque station de communiquer avec les autres sans charger le calculateur des organes de commande.

**2.2.1 - Principe de fonctionnement**

Du type multi-maître, orienté messages courts, le bus CAN est bien adapté à la scruutation de variables émises par des stations déportées. La norme ISO 11898 spécifie un débit maximum de 1Mbit/s. La longueur maximum du bus est déterminée par la charge capacitve et le débit. Les configurations recommandées sont les suivantes (*tableau 5*):
Tableau 5 : Le débit en fonction de la longueur [13]

<table>
<thead>
<tr>
<th>Data Rate</th>
<th>Thick Cable</th>
<th>Thin Cable</th>
<th>Flat Cable</th>
<th>Maximum</th>
<th>Cumulative</th>
</tr>
</thead>
<tbody>
<tr>
<td>125 kBaud</td>
<td>500 meters</td>
<td>100 meters</td>
<td>420 meters</td>
<td>6 meters</td>
<td>156 meter</td>
</tr>
<tr>
<td>250 kBaud</td>
<td>250 meters</td>
<td></td>
<td>200 meters</td>
<td></td>
<td>78 meters</td>
</tr>
<tr>
<td>500 kBaud</td>
<td>100 meters</td>
<td></td>
<td>75 meters</td>
<td></td>
<td>39 meters</td>
</tr>
</tbody>
</table>

Le protocole est basé sur le principe de diffusion générale : lors de transmission, aucune station n'est adressée en particulier, mais le contenu de chaque message est explicité par une identification reçu de façon univoque par tous les abonnés. Grâce à cet identificateur, les stations, qui sont en permanence à l'écoute du réseau, reconnaissent et traitent les messages qui les concernent; elles ignorent simplement les autres. L'identificateur indique aussi la priorité du message, qui détermine l'assignation du bus lorsque plusieurs stations émettrices sont en concurrence. En version de base, c'est un nombre de 11 bits, ce qui permet de définir jusqu'à 2048 messages plus ou moins prioritaires sur le réseau. Chaque message peut contenir jusqu'à 8 octets de données, ce qui correspond par exemple à l'état de 64 capteurs. L'adressage par le contenu assure une grande flexibilité de configuration. Il est possible d'ajouter des stations réceptrices à un réseau CAN sans modifier la configuration des autres stations.

2.2.2 - Principe de l'arbitrage

Afin d'être traitées en temps réel, les données doivent être transmises rapidement. Cela suppose non seulement une voie physique de transmission atteignant jusqu'à 1 Mbit/s, mais encore exige une assignation rapide du bus dans les cas de conflits, lorsque plusieurs stations souhaitent transmettre simultanément des messages.

L'urgence des informations échangées sur le bus peuvent être très diverses : une valeur variant rapidement, comme l'état d'un capteur ou l'asservissement d'un moteur, doit être transmis plus souvent avec un retard moindre que d'autres valeurs comme la température du moteur, qui évolue lentement. Sur le réseau CAN, l'identificateur de chaque message, qui est un mot de 11 bits (format standard) ou 29 bits (format étendu), détermine sa priorité. Les priorités sont attribuées lors de l'analyse conceptuelle du réseau, au moyen de valeur binaire, et ne peuvent donner lieu à aucune modification dynamique.
Le procédé d'attribution du bus est basé sur le principe de l'arbitrage bit à bit, selon lequel les nœuds en compétition, émettant simultanément sur le bus, comparent bit à bit l'identificateur de leur message avec celui des messages concurrents. Les stations de priorité moins élevées perdront la compétition face à celle qui a la priorité la plus élevée.

Les stations sont câblées sur le bus par le principe du « OU câblé », en cas de conflit c'est à dire émission simultanée, la valeur 0 écrase la valeur 1. On appel donc l'état dominant l'état logique 0, et l'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 la compétition et arrête d'émettre. Tous les perdants deviennent automatiquement des récepteurs du message, et ne tentent à nouveau d'émettre que lorsque le bus se libère. Illustration ci-dessous : figure 20.

![Figure 20 : Arbitrage par compétition](image)

2.2.3 - Formats de trames

La norme CAN définit deux formats de protocole : Standard (Version2.0 A) et Extended (Version2.0 B) voir la figure 21. Spécification publiée par Bosch en 1991. La différence résulte seulement dans la longueur de l'identificateur (ID) qui est de 11 bits de base et 18 bits supplémentaires en mode Extended. Cette extension permet l'augmentation du nombre de variables échangées, et le nombre de stations sur le réseau. Le nombre d'octets de données échangées à chaque trame reste inchangé.
Figure 21 : Trames CAN [4]

Une trame est composée des champs suivants :

- 1 bit SOF (*Start Of Trame*)

- 11 bits dans la zone d'arbitrage (*CAN Identifier* ou *Message ID*)

- 1 bit RTR (*Remote Transmission Request*) : détermine s'il s'agit d'une trame de données ou d'une trame de requête.

- 1 bit IDE qui établit la distinction entre format standard (état dominant) et format étendu (état récessif)

- 1 bit réservé pour une utilisation future

- 4 bit DLC (*Data Lenght Code*) : nombre d'octets contenus dans la zone de données

- Zone de données comprise entre 0 et 8 octets

- 15 bits CRC (*Cyclic Redundancy check Code*) : Ces bit sont recalculés à la réception et comparés aux bits reçus. S'il y a une différence, un bit d'erreur CRC est déclaré.

- La zone ACK est composée d'un bit ACK Slot à l'état récessif ainsi qu'un bit séparateur : *delimiter bit*. Le premier bit doit être forcé à l'état dominant par les stations ayant bien reçu cette trame.

- 7 bits dans la zone EOF (*End Of Frame*) : Ils permettent d'identifier la fin de la trame.
- 3 bits d’intermission : ce sont des bits récessifs qui notifient la libération du bus juste après l’EOF;

Les contrôleurs CAN qui admettent le format étendu peuvent aussi émettre et recevoir des messages au format standard. En revanche, dès que l’on utilise sur le réseau des contrôleurs ne maitrisant que le format standard, les messages étendus seront mal interprétés [17].

### 2.2.4 - Détection des erreurs

La trame du protocole CAN intègre plusieurs mécanismes de détection d’erreur. Afin de sécuriser la transmission des messages on utilise la méthode dite de *Bit-Stuffing* (bit de transparence). Voir l’exemple sur la **figure 22**. Cette méthode consiste, dès que l’on a émis 5 bits de même polarité sur le bus, à insérer un bit de polarité contraire pour casser des chaînes trop importantes de bits identiques. On obtient ainsi dans le message un plus grand nombre de transitions ce qui permet de faciliter la synchronisation en réception par les nœuds. Cette technique est uniquement active sur les champs de SOF, d’arbitrage, de contrôle, de *Cyclic Redundancy check Code* (délimiteur exclu). Pour un fonctionnement correct de tout le réseau, cette technique doit être implémentée aussi bien à la réception qu’à l’émission.

De plus, comme tous les nœuds de réseau surveillent simultanément le bus, ils détectent des différences entre les bits reçus et les bits émis. Dès qu’une erreur est détectée, la transmission en cours est interrompue par l’émission d’un indicateur d’erreur : *Error Flag*. L’émetteur peut donc recommencer à émettre son message.

Tout ce système de gestion des erreurs est complètement transparent pour le développeur et l’utilisateur. Le système est capable de gérer automatiquement ses conflits et ses erreurs en émettant des trames d’erreurs pour renseigner l’émetteur du message sur le type de faute qu’il a commis. Une station est capable de faire la distinction entre les perturbations temporaires et les défauts permanents. Les stations en défaut permanent sont déconnectées automatiquement du réseau.
Ce système de gestion d’erreur fait toute la puissance du réseau CAN, certains constructeurs démontrent que la probabilité d’erreur résiduelle reste inférieur 4,6 $10^{-11}$. Cela veut dire que le système détecte une erreur pour 1000 ans de fonctionnement.

**Recouvrement des erreurs**

Le recouvrement des erreurs est assuré par la retransmission automatique de la trame incriminée jusqu’à ce que l’émission de cette trame s’effectue sans erreur. La validité du message est acquise s’il n’y a aucune erreur depuis le SOF (*Start Of Frame*) jusqu’à la fin de trame.

Si l’émetteur n’arrive pas à émettre sa trame correctement, il essaye de nouveau de l’émettre jusqu’à ce que son compteur d’erreur passe en mode d’erreur passive.

**La gestion des modes d’erreur :**

Suivant le nombre d’erreurs qu’un nœud comptabilise, l’état du mode de ce nœud peut différer. Un compteur mémorise le nombre d’erreurs rencontrées lors de la transmission des trames sur le bus. Deux compteurs séparés régissent respectivement le nombre d’erreurs en émission et en réception. Il se nomme :

- *Transmit Error Counter* pour l’émission
- *Receive Error Counter* pour la réception

Lorsque le nombre d’erreur devient trop important et que le gestionnaire est déjà en erreur passive, le nœud se met en *Bus Offet* et se déconnecte.
du bus. Il ne reçoit ni émet à ce moment là aucune trame circulant sur le bus CAN. L’erreur de confinement fait la différence entre la perturbation de courte durée et les dysfonctionnements permanents [18].

Le passage dans les différents modes s’effectue suivant la valeur des compteurs comme le montre la figure 23.

![Figure 23 : Compteur d’erreur et état d’un nœud][1]

La structure du protocole du bus CAN possède implicitement les principales propriétés suivantes :

- hiérarchisation des messages.
- garantie des temps de latence.
- souplesse de configuration.
- réception de multiples sources avec synchronisation temporelle.
- fonctionnement multi maître.
- détections et signalisations d’erreurs.
- retransmission automatique des messages altérés dès que le bus est de nouveau au repos.
- distinction d’erreurs : d’ordre temporaire ou de non-fonctionnalité permanente au niveau d’un nœud.
- déconnexion automatique des nœuds défectueux.

En étudiant la norme BOSCH, on se rend compte que le protocole CAN ne couvre seulement que deux des sept couches du modèle d’interconnexion des systèmes ouverts OSI de l’ISO.
2.3 - Caractéristiques spécifiques au réseau DeviceNet


La vitesse de transfert de données varie en fonction de la longueur des câbles et de leur positionnement sur le réseau.

Voici un exemple montrant les performances d’une implémentation DeviceNet (1 Mbaud, mode polling) :

<table>
<thead>
<tr>
<th>Durée de cycle du bus</th>
<th>4,528 ms</th>
<th>8,0697 ms</th>
<th>20,821 ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temps de réaction</td>
<td>1,92 ms</td>
<td>2,84 ms</td>
<td>à 16 esclaves avec 2 octets</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>à 16 esclaves avec 8 octets</td>
<td>à 16 esclaves avec 16 octets</td>
<td></td>
</tr>
</tbody>
</table>

Ce bus de terrain possède une Connection ID (identifiant du protocole CAN), qui permet de hiérarchiser les priorités des messages. L’état des entrées et des sorties ont toujours la plus forte priorité. Ceci est réalisé grâce à un mécanisme électrique par lequel les états dominants écrasent...
les états récessifs. L’identifiant ayant la valeur binaire la plus petite aura la priorité la plus élevée. C’est la spécificité de ces bus [19]. Voici la classification par des groupes de priorités : Tableau 6. Dans les Identity Usage on constate que les Messages Group 2 sont les plus utilisés

Tableau 6 : Identification DeviceNet [13]

<table>
<thead>
<tr>
<th>CONNECTION ID = CAN Identifier (bit 10 : 0)</th>
<th>HEX RANGE</th>
<th>Identity Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 9 8 7 6 5 4 3 2 1 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0  Message ID  Source ID</td>
<td>000-3FF</td>
<td>Message Group 1</td>
</tr>
<tr>
<td>1 0 MAC ID  Message ID</td>
<td>400-5FF</td>
<td>Message Group 2</td>
</tr>
<tr>
<td>1 1  Message ID  Source MAC ID</td>
<td>600-7BF</td>
<td>Message Group 3</td>
</tr>
<tr>
<td>1 1 1 1 1 Message ID</td>
<td>7C0-7EF</td>
<td>Message Group 4</td>
</tr>
<tr>
<td>1 1 1 1 1 1 1 1 1 1 1</td>
<td>700-7FF</td>
<td>Invalid CAN Identifiers</td>
</tr>
</tbody>
</table>

2.3.1 - Echange de données

Le processus de lecture des entrées et d’écriture des sorties (dans un PLC 32 bits par exemple) se nomme échange de données (I/O data exchange). Les données E/S de chaque équipement sont « mappées » sur une zone de mémoire du PLC. Le scrutateur DeviceNet Rockwell Automation permet la configuration automatique ou manuelle. La table de mappage contient 90 registres de 32 bits. Cette architecture favorise l’augmentation de la vitesse de transfert de données [20]. Dans la figure 25 le schéma Mappage PLC/IO nous montre un exemple d’emplacement mémoire configuré dans le PLC. Sur les deux premières lignes du tableau (32 bits chaque case) sont affectées à l’équipement 1 (Slave 1) et les deux dernières lignes du tableau à l’équipement 4. Ces emplacements dans la table restent inchangés une fois la configuration effectuée. Le schéma « Câblage conventionnel » nous expose un câblage fil à fil entre un automate et un équipement où chaque entrée et sortie correspond à une E/S de l’automate. Ce câblage n’utilise pas de protocole donc il n’a pas tous ses avantages (cf. le chapitre 2.5).
Voici les différents paramètres utilisés par le protocole :

**Bit-Strobe Command** : C’est un message E/S envoyé par le maître. Il fourni un bit de donnée à chacun des esclaves du réseau avec un intervalle de temps fixe.

**Bit-Strobe Response** : Il est transmis au maître après réception du *Bit-Strobe Command*. L’esclave a la possibilité d’envoyer au moins 8 octets de données au maître.

**Change-of-State/Cyclic** : Il peut être envoyé soit par le maître soit par l’esclave. Il sera reçu uniquement par le nœud concerné qui retournera par la suite un message ACK.

**Poll Command** : C’est un message E/S envoyé par le maître et dirigé vers un seul et unique nœud. Le message peut contenir jusqu’à 256 octets et il utilise un intervalle de temps prédéfini.

**Poll Response** : C’est le message de retour envoyé par l’esclave après avoir reçu le message *Poll Command* par le maître. Le message peut contenir jusqu’à 256 octets.

**Explicit Request** : C’est le message envoyé par le maître à un nœud spécifique. Il est utilisé pour demander des données (lecture ou l’écriture de données) et la réinitialisation de l’équipement esclave. Ils peuvent être demandés par le maître à tout moment.
Explicit Response : Le message de retour envoyé par l’esclave après réception du message Explicit Request. Il est utilisé pour retourner les résultats du message Explicit Request [21].

Poll Command et Poll Response, ce sont les paramètres les plus utilisés.

Figure 26 : Classement des objets - The DeviceNet Specification [13]

2.4 - Description des trames échangées

Lors d’une connexion, les messages explicites et les E/S Poll sont constitués au moins de deux segments CAN : Le CAN Identifier Field et le CAN Data Field. Le seul segment propre à DeviceNet est défini par le Predefined Master/Slave Connection Set. Ce sont des attributs prédéfinis et ils ne sont pas modifiables. Les tableaux mettent en évidence toutes les combinaisons possibles des identifiants CAN et des données CAN [22].
La construction des identifiants selon DeviceNet permet de définir 3 groupes distinct, fonction des bits de poids fort:

<table>
<thead>
<tr>
<th>Bit 10</th>
<th>Bit 9</th>
<th>Groupe</th>
<th>Priorité</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>x</td>
<td>groupe 1</td>
<td>haute</td>
<td>Entrées/sorties</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>groupe 2</td>
<td>moyenne</td>
<td>Maître/Esclave</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>groupe 3</td>
<td>basse</td>
<td>messages</td>
</tr>
<tr>
<td></td>
<td></td>
<td>groupe 4</td>
<td>très faible</td>
<td>non définie</td>
</tr>
</tbody>
</table>

Figure 27 : Priorités des groupes [13]

Tableau 7 : CAN Identifier Field for Predefined Master/Slave Connection Set [13]

<table>
<thead>
<tr>
<th>CAN IDENTIFIER</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Group 1 Message</td>
</tr>
<tr>
<td>0 0 1 1 0 0</td>
<td>Source MAC ID</td>
</tr>
<tr>
<td>0 0 1 1 0 1</td>
<td>Source MAC ID</td>
</tr>
<tr>
<td>0 1 1 1 1 1</td>
<td>Source MAC ID</td>
</tr>
<tr>
<td>1 0 1 1 1 1</td>
<td>Source MAC ID</td>
</tr>
<tr>
<td>1 0 Source MAC ID</td>
<td>0 0 0</td>
</tr>
<tr>
<td>1 0 Destination MAC ID</td>
<td>0 0 1</td>
</tr>
<tr>
<td>1 0 Destination MAC ID</td>
<td>0 1 0</td>
</tr>
<tr>
<td>1 0 Source MAC ID</td>
<td>0 1 1</td>
</tr>
<tr>
<td>1 0 Destination MAC ID</td>
<td>1 0 0</td>
</tr>
<tr>
<td>1 0 Destination MAC ID</td>
<td>1 0 1</td>
</tr>
<tr>
<td>1 0 Destination MAC ID</td>
<td>1 1 0</td>
</tr>
<tr>
<td>1 0 Destination MAC ID</td>
<td>1 1 1</td>
</tr>
</tbody>
</table>

Tableau 8 : CAN Data Field [13]

<table>
<thead>
<tr>
<th>Byte</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 MESSAGE HEADER</td>
<td>Fragment Bit (0/1)</td>
<td>Transaction ID (XID) Bit (0/1)</td>
<td>Nodes Address 6-bit Hexadecimal Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 MESSAGE BODY</td>
<td>Service Code Calculate the combined hexadecimal value of the request/response bit (bit 7) and the assigned Service ID hex value (bits 6 to 0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Request Bit (0) or Response Bit (1)</td>
<td>Service ID Assigned hexadecimal value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Instance ID Assigned hexadecimal value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2.4.1 - Les messages explicits

Le protocole des messages DeviceNet est représenté en base hexadécimal (base 16). Les messages explicites utilisent différents formats de requête/réponse qui permettent un accès direct à tout attribut. Les messages suivants illustrent une requête (constructeur de l’équipement : Vendor ID) et l’interprétation de la réponse du réseau [23]. Voici les messages situés dans le contexte suivant :

Représentation des messages explicites
Les trames ne sont pas fragmentées
L’adresse du nœud esclave est le 05
L’adresse du nœud maître est le 01
Les espacements sont représentés uniquement par souci de clarté, ils n’apparaissent pas dans la trame.

Explicit Request Message

<table>
<thead>
<tr>
<th>CAN Identifier Field</th>
<th>CAN Data Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>42C</td>
<td>01 0E 01 01 01</td>
</tr>
</tbody>
</table>

**Group 2 Message ID**

**Destination MAC ID**

**Message Type ID** : EXPLICIT REQUEST

**CAN Identifier Field** : La valeur décimale est $42_{10}$. Cet identifiant représente : le message ID du groupe 2 (bits 10 et 9), l’adresse de destination du nœud (bits 8 à 3) et le message type ID du message explicit (bit 2 à 0).

$10 000101 100 = 42_{10}$
**CAN Data Field :**

**Message Header:** Avec 01\textsubscript{hex}. Les bits 7 et 6 représentent respectivement la fragmentation (bit à 0 = non fragmenté) et le XID (Placeholder) avec une valeur fixe de 0. L’adresse du nœud source (le maître) ce sont les bits restants.

\[0 \quad 0 \quad 0 \quad 000001 \quad = \quad 01\textsubscript{hex}\]

**Message body :**

*Service code:* Le bit 7 est celui de la requête et le reste de bits représentent le *Get Attribute Single* du service ID.

\[0 \quad 0001110 \quad = \quad 0E\textsubscript{hex}\]

Explicit Message Response

En tenant compte du message de requête voici la réponse sans erreur:

<table>
<thead>
<tr>
<th>42B</th>
<th>01</th>
<th>8E</th>
<th>36 00</th>
</tr>
</thead>
<tbody>
<tr>
<td>GROUP 2 MESSAGE ID</td>
<td>MESSAGE HEADER</td>
<td>SERVICE CODE</td>
<td>DATA</td>
</tr>
<tr>
<td>SOURCE MAC ID</td>
<td>(Frag Bit, XID Bit, Destination MAC ID)</td>
<td>(Response Bit, Service ID)</td>
<td></td>
</tr>
</tbody>
</table>

**CAN Identifier Field**

La valeur décimale est 42B\(_{\text{hex}}\). Cet identifiant représente : Le message ID du groupe 2 (bits 10 et 9), l’adresse du nœud source (bits 8 à 3) et le message type ID du message explicit (bits 2 à 0).

![Diagram](image)

**CAN Data Field**

Message Header : Avec 01\(_{\text{hex}}\). Les bits 7 et 6 représentent respectivement la fragmentation (bit à 0 = non fragmenté) et le XID (**placeholder**) avec une valeur fixe de 0. L’adresse de destination du nœud (le maître) ce sont les bits restants.
**Message body :**

*Service code*: Le bit 7 est la réponse et le reste de bits représentent le *Get Attribute Single* du *Service ID*. $8E_{\text{hex}}$ est la réponse à ce service ID.

Le *class code*, l’*instance ID* et l’*attribute ID* ne constituent pas le message de réponse.

*Data*: Ce sont les informations de réponse à la requête : *Vendor ID* (UNIT=16 bits). Le *Vendor ID* des équipements MKS est 36.
**Explicit Message Error Response**

C’est la réponse avec une erreur à la requête du même message explicit.

![Diagram](image)

**Figure 30 : Explicit Message Error Response [13]**

**CAN Identifier Field** : La valeur décimale est $42B_{\text{hex}}$. Cet identifiant représente : Le message ID du groupe 2 (bits 10 et 9), l’adresse du nœud source (bits 8 à 3) et le message type ID du message explicit (bits 2 à 0).

![CAN Identifier Field Diagram](image)

**CAN Data Field** :

**Message Header** : Avec $01_{\text{hex}}$. Les bits 7 et 6 représentent respectivement la fragmentation (bit à 0 = non fragmenté) et le XID (*Placeholder*) avec une valeur fixe de 0. L’adresse de destination du nœud (le maître) ce sont les bits restants.
**Message body** :

*Service code* : Le bit 7 est la réponse et le reste de bits représentent le signalement d’erreur du service ID. $14_{\text{hex}}$ est la réponse à ce *Service ID*.

1 0010100  = $94_{\text{hex}}$

**General error code** : C’est la réponse générique à l’erreur (désignée par l’ODVA). L’erreur $08_{\text{hex}}$ correspond à un service inexistant dans le nœud.

**Additionnal error code** : C’est l’erreur spécifique d’un *Object Class* de la jauge MKS. Si le message ne comporte pas d’erreur prédéfinie la valeur FF$_{\text{hex}}$ est envoyée par défaut.

Le *class code*, l’*instance ID* et l’*attribute ID* ne constituent pas le message de réponse.
2.4.2 - Les messages E/S Poll

Les messages E/S Poll utilisent un format qui se charge de ressembler et rapporter des données provenant de plusieurs objets en utilisant un seul type de commande. Ils sont souvent utilisés pour des retours rapides des données (Run-time). Les messages E/S Polls transitent plus rapidement par rapport aux messages explicites, mais ils se limitent aux attributs (ex : Exception status et pressure valve). Les données, lors d’une connexion, sont définies par le Predefined Master/Slave Connection Set. Voici une série de messages situés dans le contexte suivant :

Représentation des messages E/S Poll
Les trames ne sont pas fragmentées
L’adresse du nœud esclave est le 05
L’adresse du nœud maître est le 01
Les espacements sont représentés uniquement par souci de clarté, ils n’apparaissent pas dans la trame.

**I/O Poll Request Message**

<table>
<thead>
<tr>
<th>42D</th>
</tr>
</thead>
<tbody>
<tr>
<td>GROUP 2 MESSAGE ID</td>
</tr>
<tr>
<td>DESTINATION MAC ID</td>
</tr>
<tr>
<td>MESSAGE TYPE ID</td>
</tr>
<tr>
<td>0 BYTES</td>
</tr>
<tr>
<td>(Empty)</td>
</tr>
</tbody>
</table>

**CAN Identifier Field** : La valeur décimale est 42D_{hex}. Cet identifiant représente : le message ID du groupe 2 (bits 10 et 9), l’adresse de destination du nœud (bits 8 à 3) et le message type ID de la requête E/S Poll (bit 2 à 0).
Le *CAN Identifier Field* est le seul segment qui définit le type de connexion. Un seul message de requête E/S Poll reçu par le nœud est suffisant car il se base sur le *Predefined Master/Slave Connection Set*. Aucune autre information supplémentaire n’est nécessaire à la requête [24].

**I/O Poll Response Message**

![I/O Poll Response Message](image)

*CAN Identifier Field*: La valeur décimale est $42D_{\text{hex}}$. Cet identifiant représente : le message ID du groupe 2 (bits 10 et 9), l’adresse source du nœud (bits 8 à 3) et le message type ID de la réponse E/S Poll (bit 2 à 0).
Poll Response #1 : Le status byte indique que les exceptions ne sont pas gérées. La pression de la jauge affichée est de 50%.

2.5 - Aspects économiques et stratégiques

La liste des critères de comparaison parmi les différents protocoles industriels est donc très longue. De plus, les informations correspondant à certaines spécifications sont parfois difficiles à obtenir ou à élaborer (c’est le cas des critères liés à la facilité ou à la difficulté de mise en œuvre des applications ou à la maintenance) ou changeantes dans le temps (c’est le cas des prix). Le tableau suivant se limite à 8 critères qui permettent de situer globalement les réseaux de terrain les uns par rapport aux autres.

- coûts des différents éléments du réseau (installation des câbles, cartes réseau, logiciels, formation ...),
- facilité/difficulté de développement des applications qui vont utiliser le réseau,
- facilité/difficulté de maintenance du réseau et des applications,
- ouverture du réseau pour rajouter des équipements, augmenter les distances, modifier des protocoles, etc.,
- interoperabilité avec d’autres réseaux (en général, dans un système complexe, il y a plusieurs réseaux),
- existence de normes nationales ou internationales pour le réseau,
- pérennité des investissements,
- notoriété des constructeurs et vendeurs des composants du réseau.

DeviceNet provient à l’origine des États-Unis (cf. le chapitre 2) mais actuellement il s’est largement implanté en Europe où il n’est plus le seul
dans ce secteur d’activité. Profibus (développé par Siemens en 1994) est devenu un concurrent sérieux dans ce domaine d’application des réseaux industriels. Il est le bus de terrain le plus répandu au monde avec plus de 40 millions équipements reliés (cf. le site de l’organisme de standardisation http://www.profibus.com).

L’impératif de ce projet a été de créer un banc de test DeviceNet opérationnel mais cette base servira pour le futur banc de test Profibus.

2.5.1 - Comparatif entre DeviceNet et Profibus

Chaque protocole utilise exclusivement comme support de programmation le PLC de son constructeur (Allen-Bradley pour DeviceNet et Siemens pour Profibus) et ils ne sont pas interchangeables.

DeviceNet apporte un avantage dans la programmation :

La manipulation des objets CIP permet une programmation très aisée des commandes DeviceNet et par la même occasion ces programmes sont transférables d’un support à un autre. CIP est composé des couches applicatives communes à plusieurs protocoles (voir figure 10). Par contre Profibus utilise une plateforme unique (langage de programmation propre à Profibus) en se basant sur les emplacements physiques (slots) des PLC. Donc il doit être adapté au programme de pilotage si le modèle de PLC ou l’affectation physique change. C’est un inconvénient pour ce protocole, la manipulation des programmes demande une modification des plusieurs paramètres par rapport à DeviceNet.

Profibus, les temps de réponse est plus rapide :

Les équipements gérés par le protocole Profibus ont un temps de réponse moyen de 10 ms dû à la taille réduite de ses lignes de commande traitées par le microprocesseur. Pour que DeviceNet puisse atteindre les mêmes performances pour une requête similaire il faut que le microprocesseur du PLC soit bien plus puissant et plus rapide par rapport à celui utilisé par Profibus.

Coût réduit des microprocesseurs CAN utilisés par DeviceNet :

Les microprocesseurs CAN utilisés par DeviceNet sont produits à grand échelle (production d’environ 10 millions de pièces en 2011 ; voir liste des microprocesseurs existant sur le site de l’entreprise CiA [19]). Ils sont aussi utilisés dans l’industrie automobile et dans d’autres protocoles tels que CAN Open par exemple. Cela entraîne une réduction de prix conséquente : 5 à 10 fois moins cher par rapport à une puce d’autre réseau. Les puces Profibus sont spécifiques et fabriquées à moindre
échelle. La conséquence du tout cela est son prix très élevé par rapport aux puces DeviceNet.
Le comparatif entre DeviceNet et Profibus nous illustre le cas concret de deux protocoles similaires avec ses différences. Ces différences permettent d’anticiper sur les besoins en pilotage des équipements et les performances des bus de terrain. Il faut faire un compromis à faire entre les coûts du matériel, le temps de réponse et le temps de programmation.

L’étude des bus de terrain est clôturée avec ces deux chapitres. Maintenant cette base d’étude est utilisée pour que le banc de test corresponde au besoin : le choix des outils adaptés et les tests des fonctionnalités conformes au protocole DeviceNet.
CHAPITRE 3 LE BANC DE TEST DEVICE NET

Le banc de test est composé des logiciels et du matériel adaptés au besoin, les critères de ces choix sont exposés ainsi que les sources utilisés pour la comparaison des différentes modèles et marques. Une fois que le choix des éléments est finalisé la mise en place du banc est décrite sous forme de mode opératoire (demande postulée par l’entreprise 40-30) afin que ce dernier puisse être installé et exploité par des opérateurs ayant peu de connaissances des bus de terrain.

3.1 - Choix des logiciels et du matériel adapté

Les fichiers EDS, avec tous les paramètres des commandes, sont rarement disponibles sur les équipements traités par la société 40-30. Le choix des éléments du banc de test a été principalement effectué en fonction de la couche physique utilisée par le client. Les logiciels associés doivent nous permettre une mise en place aisée des commandes. Les options de configuration sont indispensables. Elles doivent nous permettre de retrouver aisément le mappage adapté et nous révéler un aperçu global en temps réel des nœuds présents sur le réseau.

Figure 33 : Connecteurs DeviceNet [13]

Les connecteurs les plus utilisés, par les clients de l’entreprise, sont de la famille Sealed Mini-Style [25].
L’alimentation 24V DC du réseau permet d’alimenter également des capteurs et des actionneurs. La spécification précise que le réseau est capable de supporter une consommation inférieure à 8 ampères. Pour les applications du banc une alimentation de 4 ampères satisfait aisément nos besoins.
RSNetWorx for DeviceNet de Rockwell Automation est le logiciel le plus performant en tenant compte des spécifications décrites et du besoin. Il permet de scruter le réseau en temps réel. La configuration des paramètres des nœuds peut être consultée et modifiée aisément. Le reste du banc contient des sous-ensembles de la même marque. Tous les autres logiciels ont été choisis afin que la comptabilité soit totale. L’automate est compatible avec le scanner DeviceNet ainsi qu’avec tous les logiciels de configuration.

Liste des sous-ensembles du banc de test :

- Concentrateur : DeviceNet Hub with voltage monitor
- Automate : CompactLogix L23E avec Compact I/O DeviceNet Scanner module
- Câble réseau avec le connecteur RJ45
- Alimentation 24VDC/4A

Le câblage du banc doit respecter le synoptique ci-dessous (figure 34). L’ordinateur doit être suffisamment puissant pour que les logiciels puissent configurer l’automate qui simule le serveur maître DeviceNet. L’automate gère le bus de terrain d’une façon autonome (configuration préalable décrite dans les pages qui suivent : cf. le chapitre 3.3), l’ordinateur ne fait que communiquer avec celui-ci mais ils capable aussi de changer des paramètres, comme par exemple imposer des niveaux haut et bas des E/S des sous-ensembles. Cela a été très utile pour la conception des logiciels de test des équipements (cf. le chapitre 3.2).

Figure 34 : Synoptique du banc de test

![Diagram of DeviceNet network topology](image)

**Figure 35 : Topologie du réseau DeviceNet (cf. le chapitre 2.3)**

Voici la liste des logiciels utilisés pour le développement du banc :

- BOOTP-DHCP server
- RSLinx
- RSNetWorx for DeviceNet
- RSLogix5000
- EZEDS

Les logiciels ci-dessus sont du même fabricant ce qui assure une compatibilité idéale entre eux. Le banc de test actuel comporte les mêmes logiciels (le descriptif de ces logiciels et les fonctionnalités utilisés sont décrites dans le chapitre 3.3.2), mais par la suite il suivra une évolution : L’interface homme-machine sera ajoutée.

### 3.2 - Programme de test des Générateurs RF et des vannes MKS 683

Le but recherché étant de faire réagir correctement l’équipement en utilisant le protocole DeviceNet. Il faudrait que toutes les fonctions utilisées pendant la production puissent être testées. Le logiciel RSLogix 5000 est un logiciel très performant et reconnu. IL permet la programmation de l’automate avec des éditeurs de diagramme à relais (seule option comprise dans la licence achetée), de diagramme de blocs fonctionnels, de graphe de fonctionnement séquentiel et de texte structuré. Ils fournissent une interface de programmation standard.
3.2.1 - Réalisation du programme


3.2.2 - Programme de test du générateur RF

La spécification des registres utilisés a été fournie par le constructeur du matériel Advanced Energy. Les tests fonctionnels tiennent compte des points de contrôle AE. Comme par exemple la conformité de la puissance (puissance Forward et Reverse) par rapport à la consigne. Cela vérifie le bon fonctionnement de la boucle de régulation du générateur selon l’impédance de la charge utilisée. Et la prise en compte des alarmes telles que l’ouverture de la boucle de sécurité du générateur (interlock) ou l’activation d’un disfonctionnement intempestif du générateur (Overtemp par exemple) qui entraîne un arrêt immédiat de ce dernier afin qu’il ne soit pas endommagé pendant son cycle de régulation. Sur la figure 36 le programme gère le statut de la puissance et les alarmes sur les quatre premiers échelons. L’autorisation de la puissance (RF ON) et l’activation du scanner DeviceNet sont validées simultanément tant que la temporisation est active. Le statut de la puissance délivrée et des alarmes sont représentés.
La temporisation rythme une consigne croissante avec un temps prédéfini avec des comparateurs. Elle augmente la consigne (manipulation des registres avec la commande \textit{move}) de 5\% à chaque comparaison validée par l'accumulateur du temporisateur comme le montre la \textbf{figure 37}.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure36.pdf}
\caption{Extrait du programme de test : Générateur RF}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure37.pdf}
\caption{Extrait de la séquence des commandes puissance}
\end{figure}

Des variables ont été créées pour manipuler les bits. La puissance \textit{Forward} et la puissance \textit{Reverse} se trouvent sur le même registre (32bits). L'instruction BTD déplace ces informations dans des nouvelles variables. Ensuite les données sont adaptées pour faciliter la lecture de puissance par l'opérateur. Dans le cas d'un générateur avec une puissance nominale de 1000W par exemple la valeur décimale correspondante à sa puissance maximale est de 4096 car le retour puissance est donné par l'équipement sur 11 bits. Donc on peut en déduire par une simple règle de trois (en ayant converti les bits utilisés en un nombre décimal) que le retour de
puissance (puissance *Forward*) en valeur décimale multiplié par un coefficient de 0.31 correspond à la puissance réelle affichée par un Wattmètre. Voir figure 38.

**Figure 38 : Affichage de la puissance Forward et Reverse**

### 3.2.3 - Programme de test de la vanne MKS 683

Le programme est basé sur une séquence de mouvements qui doit être reproductible sur toutes les vannes du même type. Deux temporisateurs ont été utilisés simultanément pour que les bits puissent suivre les changements de position du test. Les spécifications du constructeur étaient indisponibles donc il a fallu trouver les commandes de position du clapet avec une méthode empirique. Les mouvements et les positions de la vanne sont rythmés avec le même mécanisme utilisé pour la consigne puissance du générateur.
3.3 - Rédaction du mode opératoire

3.3.1 - Configuration des outils

Les modes opérateurs seront utilisés comme un support par les opérateurs au moment des tests. La capitalisation du savoir-faire est la raison principale. Par la même occasion ils serviront éventuellement dans des futures études. Comme par exemple la réalisation d’un nouveau programme de test ou l’étude d’un autre réseau de terrain.

La première étape de la mise en place du banc de test DeviceNet est la configuration des logiciels en tenant compte du hardware (port de l’automate et carte réseau) et du système d’exploitation du PC.
3.3.2 - Installation des logiciels Rockwell automation

Il existe quatre types de logiciels, du même constructeur, utilisés simultanément pour mettre en place un test fonctionnel de l’équipement en utilisant le protocole DeviceNet (cf. le chapitre 3.3.2).

- BOOTP-DHCP server
- RSLinx
- RSNetWorx for DeviceNet
- RSLogix5000

Le mode opératoire de l’installation et de la configuration de ces logiciels est détaillé ci-dessous, les figures sont pour la plupart des saisies d’écran montrant la configuration recherchée par étapes. Ce mode opératoire pourra être utilisé par des opérateurs ayant peu des connaissances sur les bus de terrain.

Configuration de la carte réseau du PC utilisée

Le terminal, qui est le PC (Windows XP) dans notre cas, doit être configuré de la façon suivante :

Le port de communication utilisé par l’automate Allen-Bradley CompactLogix1769-L23E est le port TCP-IP.
Ouvrir la fenêtre Connexions réseau et sélectionner la carte réseau utilisée.
Dans Propriétés de connexion au réseau local cliquer sur Protocol Internet TCP/IP et ensuite sur Propriétés.
Cocher la case Obtenir l’adresse IP suivante et rentrer les adresses tels que :

<table>
<thead>
<tr>
<th>Adresse IP</th>
<th>192.168.9.111</th>
</tr>
</thead>
<tbody>
<tr>
<td>Masque de sous-réseau</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>Passerelle par défaut</td>
<td>192.168.9.1</td>
</tr>
</tbody>
</table>

Voici le synoptique qui représente la communication entre les différents sous-ensembles : l’ordinateur exécute les logiciels RSLinx, RSLogix 5000 et BOOTP simultanément afin que la communication avec l’automate soit optimale. C’est la liaison physique entre l’ordinateur avec sa carte réseau et le connecteur Ethernet/IP de l’automate qui permet l’échange de données. L’adresse MAC de la carte réseau et du PLC est unique par contre les adresses IP choisis doivent être très proches. Voir figure 40.
L’adresse IP configurée doit être très proche de l’adresse de l’automate (ici 192.168.9.110) L’adresse IP de l’automate est choisie de sorte qu’il n’ait pas de conflit avec d’autres équipements.

**Configuration du logiciel BOOTP-DHCP server**

BOOTP-DHCP server permet d’attribuer une adresse IP à l’automate en fonction de son adresse MAC. L’automate aura systématiquement une nouvelle adresse IP à chaque connexion si l’option adresse dynamique (par défaut) reste activée. Il faut s’assurer de la liaison physique du câble réseau (connecteur RJ45) entre l’automate et la carte réseau du PC. Configurer dans **Tools** les paramètres réseau dans **Network Settings** : **Subnet Mask** et **Gateway** de l’automate (vérifier que la configuration en comparant l’affichage avec la figure 41).
Sélectionner une adresse Ethernet MAC dans la liste affichée : **Request History.** L’adresse MAC est unique à l’automate. Remplir l’adresse IP choisie ou déjà préétabli de l’automate et puis valider. L’adresse IP associée à l’adresse MAC apparaît dans la **Relation List.** Sélectionner cette ligne et cliquer sur **Disable BOOTP/DHCP** pour la rendre active. Uniquement après cela l’automate aura son adresse IP « en dur ». Ainsi la manipulation qui vient d’être effectué ne sera plus nécessaire lors de la prochaine connexion (vérifier la configuration avec la figure 42).

**Figure 42 : Acquittement de l’adresse IP sur BOOTP/DHCP**

**Configuration du logiciel RSLinx**

Ce logiciel permet de configurer le réseau existant et de le visualiser. L’automate et ses sous-ensembles seront visibles avec leur état respectif. Pour configurer les drivers du port Ethernet/IP activer le menu **Communications** et sélectionner **Configure driver** puis ajouter **AB_ETHIP-1.** Cette appellation est proposée par défaut quand on sélectionne **Ethernet/IP Driver** dans le menu déroulant **Available Driver Types.** La figure 43 et 44 ci-dessous nous illustre un chemin d’accès utilisant le port Ethernet/IP avec le logiciel RSLinx.
Vérifier que la carte réseau sélectionnée reste inchangée, ainsi que son adresse IP dans le menu **Configurated Drivers**. Le statut affiché sera : **Running**. Les fenêtres ci-dessous (figure 45, 46 et 47) doivent correspondre à la configuration recherchée et acquittée.
Figure 45 : Vérification du pilote RSLinx

Figure 46 : Configuration OPC Topic
Pour simplifier les prochaines connexions et ne plus avoir à répéter systématiquement la procédure ci-dessus il faut rendre « statique » la configuration du réseau. Sélectionner, dans la fenêtre RSWho, l’adresse IP 192.168.9.1101769-L23E-QB1 Ethernet Port (qui correspond à l’automate), en cliquant sur AB-ETHIP-1, Ethernet. Ensuite ouvrir Module configuration avec un clic gauche et finalement choisir Static dans l’onglet Network Configuration Type (figure 48).
Configuration du logiciel RSNetWorx for DeviceNet

Ce logiciel permet de visualiser le réseau en temps réel (figure 49) et de configurer les équipements détectés. Il définit sa configuration sur le réseau avec les fichiers EDS et il contrôle le mappage entre le PLC et les nœuds. La configuration du réseau est sauvegardée dans un fichier ***.dnt. Le 1769 Scanner DeviceNet (coupleur DeviceNet) est la passerelle (Gateway) entre les équipements et l'automate. Parmi les principales fonctionnalités on retrouve la définition de la configuration des nœuds sur le réseau (il utilise les fichiers EDS), la configuration du scrutateur avec les librairies Allen Bradley, le contrôle du mappage (information échangée entre le contrôleur et les nœuds) et la configuration des propriétés des équipements présents à l'aide d’une simple fenêtre.

Figure 49 : Réseau visualisé avec RSNetWorx for DeviceNet

Les fichiers EDS (Electronic Data Sheet) : Chaque appareil sur le réseau a une fiche de configuration électronique (EDS) contenant des paramètres de fonctionnement pour cet appareil. Ils fournissent l’identification de l’équipement nécessaire à l’implémentation du réseau DeviceNet. Ce fichier, donné par le constructeur du matériel, est souvent incomplet. La plupart des paramètres Entrées/Sorties ne sont pas pris en charge.


Le Mappage des registres E/S (affectation de la zone mémoire) des équipements est réalisé avec ce logiciel. Le Mappage est effectif uniquement quand le chargement de la configuration du nœud à travers le
PLC est effectué. La fenêtre ci-dessous (figure 50) nous montre le scanner DeviceNet (nœud 00) et deux équipements actifs. Le nœud 01 (Vanne) occupe dans la table l’emplacement 1:I.Data[0] et 1:I.Data[1]. Cet ordre peut être configuré manuellement mais il doit impérativement correspondre avec les bits utilisés dans le programme de test (RSLogix 5000). L’affectation du numéro de nœud (MAC ID) se trouve directement sur l’équipement (commutateur Node Address).

En cas de problème de la prise en charge du réseau :

Il est possible que le RSNetworx détecte le réseau (scoutateur et nœuds) d’une façon aléatoire sur un automate neuf. Réinstaller à nouveau les firmware du contrôleur CompactLogix 5000 et du scanner DeviceNet peut solutionner le problème. Le logiciel utilisé est le Flash Programming Tools / Control FLASH, on peut le trouver dans le DVD RSLogix.

**Configuration du logiciel RSLogix5000**

C’est le logiciel de programmation de l’automate. Logix 5000 est un contrôleur de 32 bits. Créer ou charger un projet est indispensable pour que la communication s’établisse. Lors de la création d’un **Nouveau projet** les informations suivantes seront demandées : Fabricant de l’automate, type du contrôleur, révision du pilote (firmware), nom du projet, la description du projet et le répertoire utilisé pour les sauvegardes.
Une fois que le projet est créé (nom et autres informations remplis), le contrôleur propose une mise à jour du Firmware : **Mettre à jour le Firmware du module.**

Attention : ne pas interrompre la mise à jour dans aucun cas.

La déclaration du Scanner DeviceNet est effectuée en ajoutant un nouveau module dans la fenêtre **Organisateur du Controller** et suit ce chemin : **Configuration des E/S / Système CompactLogix 5323E-QB1 / CompactBus Local / I/O d’extension.** Le modèle qui correspond au scanner se trouve dans **Communications \ 1769-SDN/B.** Il ne faut pas oublier de spécifier la taille des entrées et des sorties (90 registres de 32 bits) ; L’emplacement dans le fond de panier et la dernière révision du logiciel. La manipulation sera effectuée « hors ligne ». Une fois la manipulation terminée vérifier que le scanner apparait dans la fenêtre **Organisateur du Controller** (chemin : **Configuration des E/S / Système CompactLogix 5323E-QB1 / CompactBus Local / I/O d’extension / [3] 1769-SDN/B SDN**). Vérifier que la fenêtre du logiciel RSLogix 5000 correspond avec les **figures 51 et 52**.

**Figure 51 : Configuration des modules sur RSLogix 5000**
Points à vérifier avant de lancer le programme de test :

Vérifier que le chemin du contrôleur vers le terminal est bien **AB_ETHIP1\192.168.9.110\Backplane\0** dans la case Chemin : (voir figure 53).

Si ce n’est pas le cas, la case indique un autre type de chemin ou tout simplement le message **aucun**.

Voici la configuration à suivre pour que le contrôleur prenne en compte le chemin utilisé :

Dans l’onglet **Communications** sélectionner **Qui actif**;
La fenêtre **Qui actif** apparaît, sélectionner le chemin : **00, CompactLogix Processor, Vanne_683_MKS** ou autre chemin recherché (Figure 53).
Finalement cliquer sur **Aller en ligne**.
Le logiciel Factory Talk Activation Manager permet de valider les licences des logiciels Rockwell Automation utilisés sur un même poste (PC).

Voici quelques conseils utiles avant de démarrer la procédure de déverrouillage des logiciels sur le PC utilisé ou sur un nouveau poste où les licences ne sont pas actives.

Les fenêtres affichées ci-dessous (figures 54 et 55) nous montrent les configurations à obtenir pendant les manipulations.

**Factory Talk Activation Manager**
En cliquant sur le cadre **Find Available Activations** on peut visualiser les licences prises en charge.

---

**Figure 54 : FactoryTalk Activation Manager**

**Figure 55 : Find Available Activations**
Avant toute manipulation des licences s’assurer que ce logiciel est correctement installé sur l’ordinateur.

**Activation d’un nouveau logiciel avec une nouvelle licence**

**Attention** : Cette démarche est uniquement utile quand le logiciel n’a jamais été enregistré auprès de Rockwell Automation auparavant.

Le document **Activation Certificate** est livré avec celui-ci. Il contient les informations nécessaires à l’activation du produit. Tels que le **Serial Number**, le **Product Key** et le **License Type**.

Le **host ID** est une identifiant unique au PC. L’ordinateur sera identifié par défaut avec sa carte réseau. Suivre les indications sur l’écran du PC une fois qu’on a cliqué sur **Get New Activations** (figure 56).

![Figure 56 : Get New Activations](image)

**Transfert de la licence sur un autre poste**

La licence achetée avec ce logiciel ne peut être activé que sur un seul PC à la fois. Mais elle peut être désactivée et transférée sur un autre poste. Pour cela il faut cliquer sur **Rehost Activation set** suivre les instructions
indiquées. Le principe de cette manipulation est de dissocier la licence au Host ID du PC utilisé et l’associer au nouveau Host ID du PC cible.

Sur la Figure 55, on constate l’activation réussie du logiciel RSLogix 5000. Le logiciel RSNetWorks for DeviceNet n’est pas pris en charge par l’activation. On aperçoit un point d’exclamation rouge et une date d’expiration antérieure à la date d’utilisation: 09/06/2011.

Les logiciels Rockwell Automation mettent à disposition des vidéos avec les séquences complètes d’activation. Elles sont disponibles sur le logiciel RSLogix 5000 dans la rubrique Aide ou sur la Page de démarrage. On retrouve le Centre d’apprentissage avec toutes ces vidéos. Celle qui correspond à l’activation des logiciels se nomme : Gérer les activations.

3.3.3 Le banc de test

Avant de prendre en charge un matériel il faut s’assurer que le format des trames et la description des bits de commande soient disponibles sur le manuel d’utilisation ou sur tout autre support. Dans le cas contraire, le format des trames échangées sera suffisant pour commencer l’étude. Le bon format des trames assure un échange sans alarmes.

Spécification du matériel

- Fichiers EDS (Electronic Data Sheet)
- Descriptif des bits utilisés (Poll Command Message et Poll Response Message)
- Format des Poll Messages

Les fichiers EDS

Ils fournissent l’identification de l’équipement nécessaire à l’implémentation du réseau DeviceNet. Le logiciel RSNetWorx for DeviceNet affiche les fichiers EDS qui se trouvent sur un répertoire prédéfini. Ils sont classés par constructeur (Vendor) et par type (Category). Pour les visualiser il suffit de cliquer sur View et puis sur Hardware.

Poll Messages

Voici quelques exemples de format des messages type :

<table>
<thead>
<tr>
<th>Polled Input</th>
<th>Polled output</th>
<th>Equipement</th>
</tr>
</thead>
<tbody>
<tr>
<td>9 octets</td>
<td>5 octets</td>
<td>Générateur RF Comdel CX1250s</td>
</tr>
<tr>
<td>8 octets</td>
<td>8 octets</td>
<td>Throttle Valve MKS 683</td>
</tr>
<tr>
<td>3 octets</td>
<td>6 octets</td>
<td>Ultrasonic générateur Kaijo</td>
</tr>
</tbody>
</table>
L’intégrateur Applied Materials configure la communication, suivant la catégorie des sous-ensembles, généralement d’une façon standard. Comme par exemple les générateurs RF Advanced Energy utilisent le même format d’échange que les générateurs RF Comdel.

La configuration du logiciel du matériel utilisé est indispensable avant de lancer le test (se référer au fichier d’« Installation des logiciels Rockwell automation » pour avoir plus de détails sur le mappage et la recherche de bits de commande utilisés par les Poll Messages).

Le câblage du réseau série multi-maître doit respecter la spécification constructeur (Figure 35).

*Mise en place du banc de test*

Ce banc se présente comme la Figure 34. Les sous-ensembles et leur configuration seront décrits par la suite.

**Eléments du banc :**

- Alimentation DC 24V/4A
- DeviceNet Hub with voltage monitor. (Réf. JBBS 57-E811-X2/S630)
- Automate CompactLogix L23E avec Compact I/O DeviceNet Scanner module
- Câble réseau avec le connecteur RJ45
- Câbles DeviceNet avec les connecteurs Sealed Micro Style et le connecteur Combicon cinq broches standard

**Alimentation DC 24V**

Cet automate doit être alimenté en externe : 24VDC.

Par contre le réseau DeviceNet utilise la même tension DC. Il est fortement conseillé d’utiliser deux alimentations séparées. Prévoir deux alimentations DC : Une pour l’automate et l’autre pour le réseau. Cela
empêche que le fonctionnement de l'automate soit dégradé à cause d'une surconsommation sur le réseau suite au disfonctionnement d'un ou plusieurs nœuds. Le contraire est aussi envisageable. Le connecteur sur le scanner de l'automate est décrit ci-dessous. On trouve le descriptif des pins avec leur couleur respective ainsi que l'emplacement de la résistance de ligne à ajouter.

<table>
<thead>
<tr>
<th>Wire Color</th>
<th>Wire Identity</th>
<th>Usage Round</th>
<th>Usage Rat</th>
</tr>
</thead>
<tbody>
<tr>
<td>white</td>
<td>CAN_H</td>
<td>signal</td>
<td>signal</td>
</tr>
<tr>
<td>blue</td>
<td>CAN_L</td>
<td>signal</td>
<td>signal</td>
</tr>
<tr>
<td>bare</td>
<td>drain</td>
<td>shield</td>
<td>n/a</td>
</tr>
<tr>
<td>black</td>
<td>V-</td>
<td>power</td>
<td>power</td>
</tr>
<tr>
<td>red</td>
<td>V+</td>
<td>power</td>
<td>power</td>
</tr>
</tbody>
</table>

**Figure 58: Descriptif du connecteur DeviceNet de l’automate [13]**

La résistance de ligne 121 ohms ¼ W du connecteur Combicon est câblée entre le fils CAN_H et le fils CAN_L. Par contre sur le Hub DeviceNet il existe un emplacement réservé (voir **figure 59**).

**Figure 59 : Le concentrateur DeviceNet**

_**Automate CompactLogix L23E avec Compact I/O DeviceNet Scanner module**_

Le logiciel RSNetworx for DeviceNet nous présente le réseau comme le dessin ci-dessous. On aperçoit le scanner DeviceNet ainsi que les
équipements détectés. La Throttle Valve (TV) MKS se trouve sur le nœud 01 du réseau. C’est dans le menu **Properties** que les formats des I/O sont configurables. La **Scanlist** nous présente les équipements pris en charge (figure 60).

![Figure 60: Configuration de la vanne MKS 683 avec RSNetWorx](image)

En sélectionnant chaque nœud on peut vérifier que le format des messages *Polled* est configuré correctement (figure 61). La **TV** dans le
nœud 01 gère les messages *Polled* avec le format suivant : 8 octets Input et 8 octets Output.

**Automate CompactLogix L23E**

C’est avec ce contrôleur que le programme de test est exécuté. La communication entre l’automate et le PC est faite à travers le connecteur Ethernet/IP (*figure 62*). La gestion du réseau avec l’OS Windows utilise une adresse IP proche de celle de l’automate afin qu’il soit reconnu. La carte réseau du PC est l’interface de liaison. L’automate seul est capable d’effectuer le test. L’utilisation du PC permet de suivre l’évolution des entrées/sorties en temps réel.

*Figure 62: Automate CompactLogix L23E et ses connecteurs*

**La vanne MKS Type 683**

On distingue sur cet équipement le connecteur DeviceNet ainsi que les voyants utiles.
La configuration du numéro du nœud désigné et de la vitesse de transfert est réalisée hors tension. Les LED nous permettent de suivre la transition de l’acquittement du nœud par le réseau et son état en temps réel.

**Node Address et le Baud Rate**

Tout équipement géré par le protocole DeviceNet peut être configuré hors tension de la sorte : La vitesse de transfert (Baud rate) et le numéro du nœud (Node Address). Tous les connecteurs utilisés dans les sites de production des semi-conducteurs sont du même type que la vanne ci-dessus (Sealed Micro Style-Connecteur).

**LED Status**

Les nœuds sur un réseau DeviceNet actif peuvent être connectés ou déconnectés à tout moment. Le LED Status nous renseigne sur l’état du réseau. Si les leds NET et MOD sont allumées (Lumière verte) cela veut dire que l’équipement fait parti du réseau et qu’il est acquitté. Dans le cas contraire vérifier la configuration du réseau ou de l’équipement. L’initialisation du nœud, quand il vient d’être branché sur le réseau, suit la séquence suivante : La led Mod clignote alternant deux couleurs (rouge et vert) jusqu’au moment que la led Net s’allume. Comme par exemple la vanne MKS 683 dans la figure 63.
3.4 - Référent technique des bus de terrain

La plupart des constructeurs d'équipements et de fournisseurs pour les automatismes industriels et les systèmes de contrôle-commande proposent un support utilisateur qui se présente généralement sous la forme de documents techniques (manuels utilisateurs, manuels d'installation, guides de développement etc.), de logiciels et de drivers téléchargeables ou de forums. Les manuels d'utilisation disponibles sont très succincts et le support technique proposé par les représentants locaux (Faure Technologie est le représentant de Rockwell Automation en Rhône Alpes) est une prestation payante. Le budget alloué à ce projet ne le permettait pas donc j'ai dû chercher des référents techniques ailleurs. Les blogs sur internet des bus de terrains et les sites internet des entreprises prestataires de services dans ce domaine ont été un support idéal. En effet ils proposent sur leur site un vue d'ensemble des dernières technologies disponibles sur le marché et par la même occasion ils l'utilisent comme une vitrine pour ses clients. Ainsi ils exposent leur savoir-faire avec un vaste inventaire des spécificités de bus de terrain et du matériel associé [6] [14] [17]. Ils proposent aussi des blogs de discussion sur différents thèmes d'automatisation et ils sont disposés à répondre par téléphone ou par email à toute question qui fait référence à leur domaine de compétence. Le site de l’entreprise américaine RTA Automation [17] a été le plus utile à la réalisation de mon projet. L'entreprise RTA est la référence mondiale dans les bus de terrain. Ils proposent des solutions sur mesure adaptées à ses clients en leur proposant toute la gamme des bus de terrain existants. Un exemple de la pertinence de l’information disponible sur ce site est le blog RTA (John Rinaldi’s BLOG) sur le modèle CIP (Ethernet/IP EDS) du 21 février 2012 : ils commentent que l’ODVA permet que les nouveaux fichiers EDS peuvent être utilisés librement sur les équipements sans plus avoir besoin d’être certifiés par ce dernier auparavant. Cette fonctionnalité sera possible principalement avec le logiciel en cours de réalisation du logiciel RSLogix 5000 version 20 (bientôt disponible). Cela donnera une ouverture supplémentaire aux protocoles CIP et un gain de temps important aux utilisateurs.

3.5 - Limites et perspectives

Le manque d’information constructeur freine la compréhension et la prise en main du matériel.

Les fonctionnalités sur les logiciels Rockwell nécessitent une période d’adaptation, et leur mise en œuvre n’est pas simple. Mettre en évidence une consigne, et interpréter les retours du matériel testé, reste difficile d'accès.
Aujourd'hui le logiciel de programmation est également utilisé pour effectuer le test. Le programme n'étant pas protégé en écriture une mauvaise manipulation peut entraîner un disfonctionnement ou dans le pire des cas la casse du matériel.

En tenant compte de la difficulté de pilotage, et de la sécurité des lignes de code, nous devrons penser à un autre moyen de pilotage. En intégrant un module externe par exemple. Plusieurs interfaces sont disponibles sur le marché. L'opérateur n'aura ainsi aucune possibilité de modification du code, seules les informations utiles seront accessibles.

**3.5.1 - Simplifier l’utilisation du banc**


Par contre le logiciel LabVIEW propose une solution globale : Soit en intégrant le PLC CompactLogix avec un pilote adapté ou en modélisant tout le banc avec des éléments propres à National Instruments.
Les solutions NI limitent l’étude des nouveaux équipements. L’accès au mappage et l’interopérabilité entre les logiciels Rockwell Automation n’est pas égalée par ses concurrents.
3.5.2 - Communication analogique entre le PLC et l’équipement

L’échange des données analogiques et numériques entre le PLC et l’équipement permet une meilleure fiabilité du test. Certaines machines utilisent les retours analogiques dans la boucle de régulation. Pour récréer ce fonctionnement il faut ajouter un module qui contient un convertisseur analogique/numérique. Il transformera les signaux pour que le programme de test puisse l’intégrer dans le programme. Par exemple, la puissance Forward d’un générateur est donnée par une recopie analogique, elle sera comparée avec la consigne puissance du programme test. Cela permettra de tester simultanément les E/S analogiques et numériques du nœud. Ainsi le test de la commande analogique est possible. Il suffit que les retours analogiques du nœud soient comparés avec la consigne du PLC.

Recherche du module qui s’ajoutera à l’automate afin que les recopies analogiques soient prises en compte par ce dernier : Le module 1769 IF4FXOF2F/A Allen Bradley semble convenir. Il est composé de quatre canaux d’entrée et de deux canaux de sortie. La précision du convertisseur analogique/numérique sera choisie par rapport à l’application.

3.5.3 - Communication avec l’interface AMPS

40-30 pour la maintenance des sous-ensembles utilise AMPS. Cet outil assiste les opérateurs en les informant des opérations à appliquer au cours d’une intervention. AMPS utilise un serveur qui met à jour toutes les procédures. Une application envisagée serait de lui intégrer le programme de test DeviceNet pour qu’il puisse utiliser le réseau AMPS pour une éventuelle mise à jour ou tout simplement comme support de transmission.
Conclusion

L'objectif de l'étude consistait à la réalisation d'un banc de test permettant de réaliser un test fonctionnel des sous-ensembles pris séparément afin de confirmer toutes ses fonctionnalités en les pilotant exclusivement par le port DeviceNet. C'est dans ce sens que le mémoire repose sur une étude des bus de terrain pour ensuite se concentrer sur le protocole DeviceNet. Le choix des outils adaptés et les programmes de test fonctionnels des équipements ont été effectués en s'appuyant sur l'étude précédente. Ce qui a permis de faire un choix judicieux des outils par rapport au besoin en tenant en compte toutes les spécifications utilisées. Finalement, une fois que le banc DeviceNet a été mis au point et opérationnel, des modes opératoires pour la prise en main du banc ont été rédigés. Ils pourront servir également comme base pour l'étude des nouveaux protocoles.

Le banc de test est fonctionnel. Il exploite partiellement deux sous-ensembles (une vanne et un générateur RF). Les générateurs RF sont testés intégralement car l'agrément obtenu par 40-30 auprès du fabricant Advanced Energy nous donne accès à toute la documentation technique ce que n'est pas le cas avec le fabricant MKS. Ainsi, pour une société de services comme 40-30, la dépendance aux informations complètes provenant du fabricant du sous ensemble reste un handicap majeur. Ce stage m'a permis d'acquérir des nouvelles compétences et une connaissance du fonctionnement des bus de terrain existants. Une plateforme commune de test des sous-ensembles en utilisant le protocole DeviceNet a été créée et mise au point pour permettre le développement aisé des programmes de test. Cela a demandé une recherche et une étude des outils existants dans l'industrie. Une fois le choix effectué j'ai dû me familiariser avec les outils de test. Rockwell Automation (Automate et logiciels associés) est le créateur du protocole DeviceNet. En France ses distributeurs fournissent le matériel ainsi que l'assistance technique car les notices d'utilisation sont très succinctes. J'ai du me passer de cet assistance car le budget alloué ne le permettait pas. Ce qui m'a permis d'acquérir une connaissance plus approfondie du matériel et c'est pour cela que le mode opératoire a été rédigé. Cela représente un gain de temps et financier pour l'entreprise. Internet a été d'une aide indispensable pour la recherche des spécificités des bus de terrain actualisés. En effet elles ont été complémentées par les sites internet des entreprises prestataires de services dans ce domaine tel que RTA Automation.

Une perspective à court terme est l’implémentation d’une interface IHM et des convertisseurs analogique/numérique. Cela permettra un pilotage aisé par l’opérateur et une prise en compte des données analogiques. A moyen terme une dizaine d’équipements sont à relier à ce protocole. Par la suite le protocole Profibus (il est le plus répandu parmi les protocoles de communication industriels - 40 millions d'équipements actuellement dans
le monde), qui est son équivalent, sera étudié. La réalisation du banc sera accélérée car la pré-étude des bus de terrain a déjà été capitalisée.

A long terme, 40-30 envisage d’intégrer dans la base de données de son outil AMPS (logiciel interactif d’assistance à la maintenance) les programmes de test DeviceNet. Ces derniers pourront ainsi piloter des sous-ensembles en utilisant le réseau de l’entreprise déjà existant.
Lexique

**API** : Application Programming Interface

**ARINC** : Aeronautical Radio Inc.

**ARCNET** : (Attached Resource Computer NETwork)

**ARP** : Address Resolution Protocol

**ASI** : Actuator Sensor Interface

**CAN** : Control Area Network

**CEI** : Commission Electrotechnique Internationale

**COS** : Change Of State

**CRC** : Cyclic Redundancy Code

**CSMA/CA** : Carrier Sense Multiple Access/Collision Avoidance

**CSMA/CD** : Carrier Sense Multiple Access/Collision Detection

**DHCP** : Dynamic Host Configuration Protocol

**DLC** : Data Length Code

**EDS** : Electronic Data Sheet.

**EOF** : End of Frame

**HMI** : Human Machine Interface

**IEC** : International Electrotechnical Commission

**ISA** : Industry Society of America

**ISO** : International Standardization Organization

**LLC** : Logic Link Control

**LonWorks** : Local Operating Network

**MAC** : Medium Access Control
ODVA : Open DeviceNet Vendor Association
OSI : Open System Interconnection.
PLC : Programmable Logic Controller
PPP : Point to Point Protocol.
RTR : Remote Transmission Request
SERCOS : Serial Real Time Communication System
SMTP : Simple Mail Transport Protocol
SOF : Start Of Frame
TDMA : Time Division Multiple Access
UCMM : Explicit Unconnected Message Manager
WorldFIP : Factory Instrumentation Protocol
Bibliographie

Tous les sites internet ont été consultés pour vérification le 02 janvier 2012.

Partie 1

[1] Intersections
Magazine mensuel Schneider Electric de l’enseignement technique et professionnel
Guide technique
Les bus et les réseaux de terrain en automatisme industriel
(Novembre 2002)

[2] Region Mag
Magazine sur internet
Cours : les réseaux de terrain
http://www.regionmag.com/cours-reseau-de-terrain-pdf.html#a1

[3] Philippe Hoppenot
Maître de conférences à l’université d’Evry
Cours : Introduction aux Réseaux Locaux Industriels (1999)

Guers Jérôme
Guinchard Sébastien
Professeurs de l’école d’Ingénieurs 2000
Marne-la-Vallée
Cours : Les réseaux VAN-CAN (2005)

[5] Patrice Kadionik
Maître de Conférences à l’ENSEIRB Bordeaux
Cours : LE BUS CAN (2001)
http://www.enseirb.fr/~kadionik

[6] Bruno Hénon
Automaticien indépendant
Bus de terrain industriels pour les systèmes automatisés
http://www.bh-automation.fr/Ressources/Pour-les-automaticiens/Communications/
[7] Bertrand Dupouy  
Maître de Conférences à l'ENST Paris  
Cours : Introduction aux bus et réseaux temps réel (2010)  

[8] Diou Camille  
Maitre de conférences  
Université de Lorraine  
Maitrise EEA  
Cours : Le bus de terrain : Système de communication embarqué (2002)  

[9] Lusteau Franck  
Probatoire filière Informatique  
CNAM Bretagne  
Réseaux de terrain : La couche physique (1999)  
http://tvaira.free.fr/reseaux/rli_couche_physique.pdf

[10] Valentin Gies  
Enseignant IUT GEII Sud Protée  
http://www.vgies.com/downloads/CoursEME31_2.pdf

Maitre de conférences au Polytech Montpellier  
Cours : Réseaux Locaux Industriels (2006)  

[12] Viktor Schiffer  
Site internet ODVA  
The common Industrial protocol and family of CIP network  
http://www.odva.org

[13] ODVA  
COMPACT DISK English language Version  
The DeviceNet™ Specification (November 2011)

[14] Technology UK  
Site Internet de l’entreprise  
Common Industrial Protocol  
http://www.technologyuk.net/telecommunications/industrial_networks/cip.shtml
Partie 2

Site Internet : portail en automatique
Ethernet/CompoNet
http://www.automation.com

[16] Hilscher France
Site internet de l’entreprise
Products Bus DeviceNet
http://www.hilscher.fr/products_bus_devicenet.html

[17] John Rinaldi
Real Time Automation
Site internet de l’entreprise
DeviceNet Introduction
http://www.rtautomation.com/devicenet/

[18] Patrick Monassier
Maitre de conférences à l’Université Lyon 1
Cours : Les couches applicatives de CAN
http://patrick.monassier.free.fr/cours/canappli.pps

[19] CiA
Site internet de l’entreprise
CAN in Automation
http://www.diakom.ru/el/communication/can/can_org.pdf

[20] François Birling
Enseignant à la Haute Ecole d’Ingénierie et de Gestion du Canton de Vaud (Suisse)
Cours : Introduction aux bus de terrain
http://iai.heig-vd.ch/fr-ch/Enseignement/Supports/Bus%20de%20terrain%20%28BUT%29/BUT%20Introduction%20aux%20bus%20de%20terrain.pdf

[21] Yoshihiro Miyazaki
Industrial Systems Division, Industrial Computer Systems Dept.
Hiroaki Fukumaru
Omika Control Product Systems Division, Control Systems Development Dept.
Hitachi
Open Control Systems and DeviceNet Applications (Juin 1999)
[22] ODVA
Object and device profile definition (1998)
http://www.odva.org

[23] Eurotherm Drives Technical manual
DeviceNet Communications Interface

[24] Rockwell Automation
Site internet de l’entreprise
Compact I/O DeviceNet Scanner Module
http://www.rockwellautomation.com

[25] Cégep du Vieux Montréal
Enseignement collégial
Cours : Concurrent Time Domain Multiple Access
http://www.cvm.qc.ca/charbono/perso/Cours/Reseaux/ControlNet/CN_Tutor/web_pages/Important_concepts/supported%20data/IC_supported_CTDMA.htm

[26] ODVA
Everything you need to know to use DeviceNet Explicit Messaging
http://www.odva.org

[27] MKS Baratron Operating Manual
Site de l’entreprise
MKS Baratron Type DMA Capacitance Manometer with DeviceNet Digital Interface (2009)
http://www.mksinst.com

[28] ODVA
DeviceNet - CIP on CAN Technology (2008)

[29] Jessica Couturier
Enseignant à IUT GEII Bordeaux1
Cours : Les lignes de transmission (juillet 2009)
http://www.iut.u-bordeaux1.fr/geii/images/stories/Cours_CEM/Lignes_de_transmission_2.pdf

Carrier sense multiple access with collision detection
Simplified Algorithm of CSMA/CD (2012)
http://en.wikipedia.org/wiki/Carrier_sense_multiple_access_with_collision_detection

[31] Zoubir Mammeri
Enseignant chercheur à l’IRIT de Toulouse
Cours : Communication dans les systèmes temps réel et embarqués (2011)
http://www.irit.fr/~Zoubir.Mammeri/Cours/M2PRLI/M2PRLI.html
ANNEXES

A Tableaux comparatifs des bus de terrain
B Fichier EDS
C Caractéristiques des câbles réseau
D Topologies des réseaux
E Impédance caractéristique d’une ligne de transmission
F L’effet de peau
G Principaux protocoles CIP
H Algorithme Carrier Sense Multiple Access
I Principaux domaines d’application du protocole DeviceNet
J Comparatif entre DeviceNet et les principaux bus de terrain
K Spécifications DeviceNet
### A Tableaux comparatifs des bus de terrain [5] [20] [31]

<table>
<thead>
<tr>
<th>Réseau</th>
<th>Technique d’accès</th>
<th>Débit maximum</th>
<th>Taille maximum de données</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARCNET</td>
<td>Passage de jeton</td>
<td>10 Mb/s</td>
<td>507 octets</td>
</tr>
<tr>
<td>ARINC 429</td>
<td>Réseau ayant un seul émetteur</td>
<td>100 kb/s</td>
<td>Mot de 32 bits</td>
</tr>
<tr>
<td>ARINC 629</td>
<td>TDMA</td>
<td>2 Mb/s</td>
<td></td>
</tr>
<tr>
<td>ARINC 659</td>
<td>TDMA</td>
<td>30 Mb/s</td>
<td>1024 octets</td>
</tr>
<tr>
<td>ASI</td>
<td>Maître/esclave avec scrutation cyclique</td>
<td>167 kb/s</td>
<td>8 bits par nœud</td>
</tr>
<tr>
<td>Batibus</td>
<td>CSMA/CA</td>
<td>4,8 kb/s</td>
<td></td>
</tr>
<tr>
<td>BitBus</td>
<td>Maître/esclave</td>
<td>2,4 Mb/s</td>
<td>248 octets</td>
</tr>
<tr>
<td>ControlNet</td>
<td>Multiplexage temporel (CTDMA)</td>
<td>5 Mb/s</td>
<td>510 octets</td>
</tr>
<tr>
<td>CAN</td>
<td>CSMA/CR</td>
<td>1 Mb/s</td>
<td>8 octets</td>
</tr>
<tr>
<td>EHS</td>
<td>CSMA/CA</td>
<td>48 kb/s</td>
<td></td>
</tr>
<tr>
<td>EIB</td>
<td>CSMA/CA</td>
<td>10 Mb/s</td>
<td>256 octets</td>
</tr>
<tr>
<td>Fieldbus Fondation H1 cyclique</td>
<td>Maître-esclave avec scrutation cyclique</td>
<td>31,25 kb/s</td>
<td>128 octets</td>
</tr>
<tr>
<td>Fieldbus Fondation HSE</td>
<td>CSMA/CD</td>
<td>100 Mb/s</td>
<td></td>
</tr>
<tr>
<td>HSDB</td>
<td>Passage de jeton</td>
<td>50 Mb/s</td>
<td>9192 octets</td>
</tr>
<tr>
<td>IEC/ISA SP50 Fieldbus</td>
<td>Passage de jeton ou maître</td>
<td>5 Mb/s</td>
<td>256 octets</td>
</tr>
<tr>
<td>Interbus</td>
<td>monomaître + TDMA</td>
<td>500 kb/s</td>
<td></td>
</tr>
<tr>
<td>LonWorks</td>
<td>CSMA persistante</td>
<td>1,25 Mb/s</td>
<td>228 octets</td>
</tr>
<tr>
<td>MIL-STD 1553</td>
<td>Maître/esclave</td>
<td>1 Mb/s</td>
<td>64 octets</td>
</tr>
<tr>
<td>MIL-STD-1773</td>
<td>Maître/esclave</td>
<td>20 Mb/s</td>
<td>64 octets</td>
</tr>
<tr>
<td>Modbus+</td>
<td>Passage de jeton</td>
<td>1 Mb/s</td>
<td>256 octets</td>
</tr>
<tr>
<td>P-Net</td>
<td>Multi-maître/Multi-esclave avec passage de jeton virtuel entre maîtres.</td>
<td>76,8 kb/s</td>
<td>56 octets</td>
</tr>
<tr>
<td>Réseau</td>
<td>Topologie</td>
<td>Support de transmission</td>
<td>Nombre maximum de nœuds</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>---------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>PROFIBUS</td>
<td>Multi-maître/Multi-esclave avec passage de jeton virtuel entre maîtres.</td>
<td>12 Mb/s</td>
<td>244 octets</td>
</tr>
<tr>
<td>SERCOS</td>
<td>Maître/esclave</td>
<td>2 Mb/s</td>
<td></td>
</tr>
<tr>
<td>VAN</td>
<td>CSMA/CA</td>
<td>10 Mb/s</td>
<td>28 octets</td>
</tr>
<tr>
<td>WorldFIP</td>
<td>Contrôle par un arbitre de bus</td>
<td>5 Mb/s</td>
<td>128 octets</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Réseau</th>
<th>Topologie</th>
<th>Support de transmission</th>
<th>Nombre maximum de nœuds</th>
<th>Distance maximale</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARCNET</td>
<td>Etoile, bus</td>
<td>Coaxial, paire torsadée, Fibre optique</td>
<td>255</td>
<td>Dépend du débit. Par exemple : 6 km sur câble coaxial ; 2 km sur fibre optique</td>
</tr>
<tr>
<td>ARINC 429</td>
<td>Bus, étoile</td>
<td>Paire torsadée</td>
<td>20</td>
<td>300 pieds (90 m)</td>
</tr>
<tr>
<td>ARINC 629 Bus</td>
<td>Bus</td>
<td>Paire torsadée, Fibre optique</td>
<td></td>
<td>100m</td>
</tr>
<tr>
<td>ARINC 659</td>
<td>Bus</td>
<td>Paire torsadée, Fibre optique</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASI</td>
<td>Libre</td>
<td>Paire torsadée</td>
<td>1 maître + 31 esclaves</td>
<td>100 m (sans répéteur), 300 m (avec répéteur)</td>
</tr>
<tr>
<td>BatiBus</td>
<td>Bus, étoile, anneau</td>
<td>Paire torsadée</td>
<td>75</td>
<td>2 km</td>
</tr>
<tr>
<td>BitBus</td>
<td>Bus, arbre</td>
<td>Paire torsadée blindée, Fibre optique</td>
<td>32 (sans répéteur) 251 (avec répéteur)</td>
<td>Dépend du débit et du support. Par exemple : 13,2 km répéteurs et à 62,5 kb/s</td>
</tr>
<tr>
<td>CAN</td>
<td>Bus, Arbre</td>
<td>Paire torsadée, Fibre optique, Coaxial</td>
<td>30 (sans répéteur) 127 (avec répéteur)</td>
<td>25m à 1000m en fonction du débit</td>
</tr>
<tr>
<td>ControlNet Libre</td>
<td>Coaxial,Fibre optique</td>
<td>99</td>
<td>20 km (dépend du nombre de nœuds et du type de support)</td>
<td></td>
</tr>
<tr>
<td>EHS</td>
<td>Bus, étoile</td>
<td>Tout type</td>
<td>256 par sous-</td>
<td></td>
</tr>
<tr>
<td>Banc de test DeviceNet</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>EIB</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bus, arbre, étoile</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paire de fils, Réseau d’alim, Radio, Infrarouge</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>réseau</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64 k</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dépend du support. 1 km sur paire de fils.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>FieldbusFoundation H1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arbre, bus</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paire torsadée, Fibre optique</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>240 par segment</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dépend du débit. Par exemple : 1,9 km à 31,25 kb/s</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>FieldbusFoundation HSE</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Etoile</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paire torsadée, Fibre optique</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>illimité</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dépend du débit et du support. Par exemple : 2 km à 100 Mb/s sur fibre optique.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>HSDB</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bus, étoile</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paire torsadée, Fibre optique</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>IEC/ISA SP50 Fieldbus</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Etoile, bus</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paire torsadée, Fibre optique, radio</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>128</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dépend du débit. Par exemple : 500 m à 5 Mb/s</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Interbus</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anneau</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paire torsadée, Fibre optique</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>512</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>400 m par segment, 12,8 km au total</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>LonsWorks</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bus, boucle, étoile</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paire torsadée, Fibre optique, ligne de tension</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>127 par segment, 255 segments</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dépend du débit et du support. Par exemple: 1,25 Mb/s sur 125 m sur paire torsadée.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>MIL-STD-1553 Bus</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bus</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paire torsadée blindée</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>MIL-STD-1773 Bus</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bus, étoile</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fibre optique</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Modbus+</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bus</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paire torsadée blindée</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>450 m par segment, 1800 m au maximum</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>P-Net</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bus, anneau</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paire torsadée blindée</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>125 (dont 32 maîtres)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1,2 Km sans répéteur. 6 Km avec répéteurs.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Réseau</td>
<td>Constructeur</td>
<td>Année</td>
<td>Standards</td>
<td>Principaux secteurs d'utilisation</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------</td>
<td>-------</td>
<td>----------------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>ARCNET</td>
<td>Datapoint</td>
<td>1977</td>
<td>ANSI/ATA 878</td>
<td>Systèmes automatisés</td>
</tr>
<tr>
<td>ARINC 429</td>
<td>SBS Techno.</td>
<td>1977</td>
<td>Aeronautical Radio Inc.</td>
<td>Avionique civil</td>
</tr>
<tr>
<td>ARINC 629</td>
<td>Ballard Techno.</td>
<td>1989</td>
<td>Aeronautical Radio Inc.</td>
<td>Avionique civil</td>
</tr>
<tr>
<td>ARINC 659</td>
<td>Honeywell Inc</td>
<td>1991</td>
<td>Aeronautical Radio Inc.</td>
<td>Avionique civil</td>
</tr>
<tr>
<td>ASI 1993</td>
<td>ASI Consortium</td>
<td>1993</td>
<td>IEC 62026-2</td>
<td>Capteurs et actionneurs</td>
</tr>
<tr>
<td>BatiBus</td>
<td>Merlin Gerin</td>
<td>1980</td>
<td>NFC 46.620-629</td>
<td>Bâtiment</td>
</tr>
<tr>
<td>BitBus</td>
<td>Intel</td>
<td>1983</td>
<td>IEEE-1118, RS485</td>
<td>Systèmes automatisés, bâtiment</td>
</tr>
<tr>
<td>CAN</td>
<td>Bosch - Intel</td>
<td>1995</td>
<td>ISO 11898 &amp;11519</td>
<td>Automobile, Systèmes automatisés</td>
</tr>
<tr>
<td>ControlNet</td>
<td>Rockwell auto.</td>
<td>1995</td>
<td></td>
<td>Systèmes automatisés</td>
</tr>
<tr>
<td>EHS</td>
<td></td>
<td>1992</td>
<td></td>
<td>Domotique</td>
</tr>
<tr>
<td>EIB</td>
<td>ABB, Siemens</td>
<td>1980’s</td>
<td>ENV 50090</td>
<td>Bâtiment et domotique</td>
</tr>
<tr>
<td>Protocole</td>
<td>Organisation</td>
<td>Année</td>
<td>Norme</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------</td>
<td>-------</td>
<td>---------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>Fieldbus H1</td>
<td>FieldbusFoundation</td>
<td>1995</td>
<td>ISA SP50/IEC 61158</td>
<td>Systèmes automatisés</td>
</tr>
<tr>
<td>Fieldbus HSE</td>
<td>FieldbusFoundation</td>
<td>2000</td>
<td>IEEE 802.3, IEC 61158</td>
<td>Systèmes automatisés</td>
</tr>
<tr>
<td>HSDB</td>
<td>Jane’sDefense</td>
<td>1991</td>
<td></td>
<td>Avionique militaire</td>
</tr>
<tr>
<td>IEC/ISA SP50 Fieldbus</td>
<td>ISA &amp; FieldbusFoundation</td>
<td>1992</td>
<td>IEC 1158; ANSI/ISA-50</td>
<td>Systèmes automatisés</td>
</tr>
<tr>
<td>Interbus</td>
<td>Phoenix Contact</td>
<td>1984</td>
<td>IEC-1131, EN 50.254</td>
<td>Systèmes automatisés</td>
</tr>
<tr>
<td>LonWorks</td>
<td>Echelon Corp</td>
<td>1991</td>
<td>ANSI/EIA 709.1</td>
<td>Systèmes automatisés, bâtiment</td>
</tr>
<tr>
<td>MIL-STD-1773</td>
<td>Condor Engin.</td>
<td>1989</td>
<td>STD 1773 (DoD US)</td>
<td>Avionique militaire</td>
</tr>
<tr>
<td>Modbus+</td>
<td>AEG Schneider</td>
<td>1980’s</td>
<td></td>
<td>Systèmes automatisés</td>
</tr>
<tr>
<td>P-Net Systèmes automatisés</td>
<td>Process Data</td>
<td>1983</td>
<td>EN 50170, RS485</td>
<td>Systèmes automatisés</td>
</tr>
<tr>
<td>PROFIBUS</td>
<td>Siemens</td>
<td>1988</td>
<td>EN 50170 / DIN 19245, IEC 1158-2</td>
<td>Systèmes automatisés</td>
</tr>
<tr>
<td>SERCOS</td>
<td>SGS-Thomson</td>
<td>1990’s</td>
<td>IEEE 1491</td>
<td>Systèmes automatisés</td>
</tr>
<tr>
<td>VAN</td>
<td>Sous-traitants de PSA</td>
<td>1980</td>
<td>ISO 11519-2</td>
<td>Automobile</td>
</tr>
<tr>
<td>WorldFIP</td>
<td>WorldFIP</td>
<td>1988</td>
<td>IEC 1158-2, EN 50170 NF UTE C-46-xxx</td>
<td>Systèmes automatisés</td>
</tr>
</tbody>
</table>
B  Fichier EDS [13]

Il est unique à chaque type d’équipement. Dans l’exemple ci-dessous on retrouve un extrait de la configuration d’un variateur de moteur AC. Les valeurs sont représentées en décimal mais elles peuvent être aussi affichées en hexadécimal (format : 0 x 00). Voici comment est partialement présenté le fichier EDS avec les commentaires qui expliquent les différentes configurations :

[File]
DescText = "IS7 EDS File";

Nom du fichier EDS

CreateDate = 11-30-2006;
CreateTime = 15:57:30;
ModDate = 08-09-2010;
ModTime = 17:09:42;
Revision = 2.1;                            $ Major_revision.Minor_revision

[Device]
VendCode = 259;                               $ Vendor Code

Numéro attribué par l’ODVA qui correspond au fabricant, disponible sur le site internet ODVA :
Advanced Energy  198
Comdel  197
LS Industrial Systems  259
MKS  36

ProdType = 2;                                 $ Product Type
Type de l’équipement donné par l’ODVA : AC Drive Device (2)
Liste complète dans le CD ODVA volume I PAGE 825/1418
ProdCode = 11;                                $ Product Code

Défini par le constructeur. ( Exemple : Baratron Type MDA ; Product Code 3)
MajRev = 2;                                   $ Major Revision
MinRev = 2;                                   $ Minor Revision
VendName = "LS Industrial Systems Co. Ltd";
ProdTypeStr = "AC Drive";
ProdName = "IS7 DeviceNet";
Catalog = "";
Icon = "LSISInvDnet.ico";

[IO_Info]
Default = 0x0001;                             $ Bit0:Poll 1:Strobe
2:COS 3:Cyclic
PollInfo = 0x0001,
1,                                            $ Poll Only
$ Default Input = Input1
1;                                  $ Default Output=Output1

$Input Connections - To Inverter

Input1 = 4, $ 4 bytes
0, $ 0 bits are significant
0x0001, $ Poll Only Connection
"70(RPM)Basic Speed Status", $ Name
6, "20 04 24 46 30 03",
" ", $ Help

Input2 = 4, $ 4 bytes
0, $ 0 bits are significant
0x0001, $ Poll Only Connection
"71(RPM)Extended Speed Status", $ Name
6, "20 04 24 47 30 03",
" ", $ Help

Input3 = 4, $ 4 bytes
0, $ 0 bits are significant
0x0001, $ Poll Only Connection
"110(Hz)Basic Speed Status", $ Name
6, "20 04 24 6E 30 03",
" ", $ Help

Input4 = 4, $ 4 bytes
0, $ 0 bits are significant
0x0001, $ Poll Only Connection
"111(Hz)Extended Speed Status", $ Name
6, "20 04 24 6F 30 03",
" ", $ Help

Input5 = 2, $ 2 bytes
0, $ 0 bits are significant
0x0001, $ Poll Only Connection
"141 User Define 1 Status", $ Name
6, "20 04 24 8D 30 03",
" ", $ Help

Output6 = 4, $ 4 bytes
0, $ 0 bits are significant
0x0001, $ Poll Only Connection
"122 User Define 2 Cmd", $ Name
6, "20 04 24 7A 30 03",
" ", $ Help

Output7 = 6, $ 6 bytes
0, $ 0 bits are significant
0x0001, $ Poll Only Connection
"123 User Define 3 Cmd", $ Name
6, "20 04 24 7B 30 03",
" ",
" ",
Output8 = 8,
0,
0x0001,
"124 User Define 4 Cmd",
6, "20 04 24 7C 30 03",
" ",

[ParamClass]
MaxInst = 596;
$ Number of parameters described in this file
Descriptor = 0x03;
$ Contains bit flags that describe the behavior of the
device's parameter objects
$ 0-bit : Supports Parameter Instance
$ 1-bit : Supports Full Attributes
$ 2-bit : Must do non-volatile Storage Save Command
$ 3-bit : Params are stored in non-volatile Storage
CfgAssembly = 0;
$ We don't support a configuration assembly

[Params]

Param1= 0, "20 28 24 01 30 03",
6, "20 28 24 01 30 03",
0x0032,
$ Path size, Logical address(Class:40, Instance:1, Attribute:3)
8,1,
Format des données: BOOL 0x00020,4,1
USINT 0x00020,8,1
Byte 0x00030,24,1
UINT 0x00020,2,2
INT 0x00030,3,2
REAL 0x00020,11,4

"Motor Type",
" ",
" ",
0, 10, 7,
1, 1, 1, 0,
0, 0, 0, 0,
$ Min, Max, Default data values
$ Mul, Div, Base, Offset scaling
$ Mul, Div, Base, Offset link

$ Data type (USINT), Data size
$ Parameter name
$ Unit
$ Help

$ Class:Assembly, Instance:123, Attribute:3
$ Help

$ 8 bytes
$ 0 bits are significant
$ Poll Only Connection
$ Name

$ Class:Assembly, Instance:124, Attribute:3
$ Help
Param2=

0;                              $ Decimal precision

$ Group : Motor Data, Code Num : 2

$ Reserved

6, "20 28 24 01 30 06",
$ Path size, Logical address(Class:40, Instance:1, Attribute:6)
0x0004,

$ Descriptor (Using Scaling, )

2,2,

$ Data type (UINT), Data size

"Motor Rated Curr",
$ Parameter name
"A",
$ Unit
" ",
$ Help

10, 2000, 36,
$ Min, Max, Default data values
1, 10, 1, 0,
$ Mul, Div, Base, Offset scaling
0, 0, 0, 0,
$ Mul, Div, Base, Offset link
1;

Param3=

$ Group : Motor Data, Code Num : 3

0;                              $ Decimal precision

$ Reserved

6, "20 28 24 01 30 07",
$ Path size, Logical address(Class:40, Instance:1, Attribute:7)
0x0004,

$ Descriptor (Using Scaling, )

2,2,

$ Data type (UINT), Data size

"Motor Rated Volt",
$ Parameter name
"V",
$ Unit
" ",
$ Help

0, 480, 0,
$ Min, Max, Default data values
1, 1, 1, 0,
$ Mul, Div, Base, Offset scaling
0, 0, 0, 0,
$ Mul, Div, Base, Offset link
0;

Param4=

$ Group : Control Supervisor, Code Num : 1

0;                              $ Decimal precision

$ Reserved

6, "20 29 24 01 30 03",
$ Path size, Logical address(Class:41, Instance:1, Attribute:3)
0x0000,

$ Descriptor ()

4,1,

$ Data type (BOOL), Data size

"Forward Run Cmd.",
$ Parameter name
" ",
$ Unit
" ",
$ Help

0, 1, 0,
$ Min, Max, Default data values
0, 0, 0, 0,
$ Mul, Div, Base, Offset scaling
C Caractéristiques des câbles réseau [13]

<table>
<thead>
<tr>
<th>paire torsadée</th>
<th>câble coaxial</th>
<th>fibre optique</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>non blindée</td>
<td>blindée</td>
</tr>
<tr>
<td>débit max</td>
<td>~100 Mbit/s</td>
<td>x 100 Mbit/s</td>
</tr>
<tr>
<td>longueur max.</td>
<td>x 100 m</td>
<td>1 km</td>
</tr>
<tr>
<td>coût de mise en œuvre</td>
<td>faible</td>
<td>moyen</td>
</tr>
<tr>
<td>CEM et fiabilité</td>
<td>faible</td>
<td>moyenne</td>
</tr>
<tr>
<td>complexité</td>
<td>simple</td>
<td>moyenne</td>
</tr>
</tbody>
</table>
D  Topologie des réseaux [1]

E  Impédance Caractéristique d’une ligne de transmission [28]
L’effet de peau [28]

\[ \delta = \frac{\rho}{\pi f \mu_0 \mu_r} \]

\( \delta \) : Epaisseur de peau en m  
\( f \) : Fréquence en Hz  
\( \sigma \) : Conductivité électrique en Ω/m  
avec la résistivité \( \rho = 1/\sigma \) (Cuivre = 1.673 Ω/m)  
\( \mu \) : Perméabilité en H/m

Principaux protocoles CIP [13]
H  Algorithme Carrier Sense Multiple Access [29]

I  Principaux domaines d’application du protocole DeviceNet [20]
### J Comparatif entre DeviceNet et les principaux bus de terrain [20]

<table>
<thead>
<tr>
<th>Item</th>
<th>DeviceNet</th>
<th>LONWORKS</th>
<th>PROFIBUS(-DP)</th>
<th>Foundation Fieldbus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. No. of Nodes (including master node)</td>
<td>64</td>
<td>127 / subnet</td>
<td>256</td>
<td>256</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(255 subnet / domain)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extension Distance (extendible with optical fiber)</td>
<td>500m</td>
<td>2km</td>
<td>1.2km</td>
<td>1.9km</td>
</tr>
<tr>
<td>Max. Distance between Nodes</td>
<td>Not stipulated</td>
<td>2km</td>
<td>200m</td>
<td>Not stipulated</td>
</tr>
<tr>
<td>Transmission Medium (optical fiber also possible)</td>
<td>Twisted pair</td>
<td>Twisted pair</td>
<td>Twisted pair</td>
<td>Twisted pair</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wireless Infrared rays</td>
<td>Optical fiber</td>
<td>Wireless Optical fiber</td>
</tr>
<tr>
<td>Transmission Rate</td>
<td>125kbps/</td>
<td>9.6kbps~1.25kbps/</td>
<td>9.6kbps~12Mbps</td>
<td>31.25kbps~38.75kbps</td>
</tr>
<tr>
<td></td>
<td>250kbps/</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>500kbps (cable only)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bus Access Mode</td>
<td>CSMA/NBA Mode</td>
<td>Predictive CSMA Mode</td>
<td>Polling Mode</td>
<td>Link Active Scheduler Mode</td>
</tr>
<tr>
<td>Data Length of Frame</td>
<td>0~8 Bytes</td>
<td>2~32 Bytes</td>
<td>1~246 Bytes</td>
<td>1~256 Bytes</td>
</tr>
<tr>
<td>Standardization Organization</td>
<td>ODVA (USA)</td>
<td>~ (ECHelon Co.)</td>
<td>PROFIBUS International (Germany)</td>
<td>IEC / Fieldbus Foundation</td>
</tr>
<tr>
<td>Main Application Field</td>
<td>F.A (North America, Japan)</td>
<td>Building Management (North America, Japan)</td>
<td>F.A (Europe)</td>
<td>P.A (North America)</td>
</tr>
</tbody>
</table>

### K Spécifications DeviceNet [13]

<table>
<thead>
<tr>
<th>Data Rates</th>
<th>125 Kbps</th>
<th>250 Kbps</th>
<th>500 Kbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thick Trunk Length</td>
<td>500 m (1,640 ft)</td>
<td>250 m (820 ft)</td>
<td>100 m (328 ft)</td>
</tr>
<tr>
<td>Thin Trunk Length</td>
<td>100 m (328 ft)</td>
<td>100 m (328 ft)</td>
<td>100 m (328 ft)</td>
</tr>
<tr>
<td>Maximum Drop Length</td>
<td>6 m (20 ft)</td>
<td>6 m (20 ft)</td>
<td>6 m (20 ft)</td>
</tr>
<tr>
<td>Cumulative Drop Length</td>
<td>156 m (512 ft)</td>
<td>78 m (256 ft)</td>
<td>39 m (128 ft)</td>
</tr>
</tbody>
</table>
Réalisations du banc de test DeviceNet

Nicolas CHARANTON

Grenoble, le 11 juin 2012

Résumé

40-30 prévoit d’acquérir des nouvelles compétences sur le protocole DeviceNet et les réseaux de terrain. L’objectif de ce banc est de tester les sous-ensembles avec ce protocole afin de valider leur fonctionnement. L’étude a débuté par l’état de l’art des bus de terrain puis l’étude du protocole DeviceNet. A partir de cette base les outils adaptés pour les tests ont été choisis. Rockwell Automation est le créateur de DeviceNet et par la même occasion il a développé les automates et les logiciels qui prennent en charge toutes ses fonctionnalités. La mise en place du banc a demandé une familiarisation avec ces différents nouveaux éléments. Le plus grand obstacle à la réalisation des programmes de test a été le manque de spécifications. Ce manque a été palié partiellement avec une connaissance générique des équipements et par une méthode de recherche empirique sur les commandes et les retours. Le programme consiste en une séquence de consignes réproductibles d’un équipement à un autre. Cela permet de vérifier le comportement du celui-ci en temps réel. Des modes opératoires et une compilation des documents sont actuellement à disposition pour faciliter l’utilisation du banc de test et pour qu’ils soient utilisables dans les futurs projets. AMPS, est une interface 40-30 de tests de maintenance, sera implémentée par ces programmes de test.

Mots clés : Réseaux de terrain, protocole DeviceNet, Automate, temps réel, spécifications, programmes de test.

Abstract

The DeviceNet bench is a 40-30 project request from its main customer (ST Microelectronics). The objective is the final test performed with this protocol. 40-30 does not have any experience in industrial networks, so I had to start by defining the condition of the Fieldbus and then the DeviceNet protocol. After that the right tools could have been chosen for the bench. Rockwell Automation created DeviceNet protocol and at the same time it developed controllers and software that support its full functionality. The establishment of the bench requested familiarity with these new tools. The biggest obstacle to write test programs was the lack of specifications. It was partially overcome with knowledge of generic devices and the empirical research of the commands and returns. This test code is a cyclic device instructions sequence. The program checks the behavior in real time. Procedures and Fieldbus library are currently available to help the test bench users and it can be useful for others Fieldbus projects. AMPS, which is 40-30 interface test maintenance, will be implemented by these test programs.

Key Words: Fieldbus, DeviceNet protocol, controller, real time, specifications, program test.