République Algérienne Démocratique et Populaire
Ministère de l'Éducation Supérieure et de la Recherche Scientifique

École Nationale Polytechnique

Département : Automatique

Projet de fin d'études

Commande et supervision d'un système technologique distant en exploitant le réseau informatique

Proposé par :
Pr. H. Chekireb

Présenté par :
Mohamed Yanis Hadj-Said
Houssam Eddine Gougam

juin 2010

Remerciement

Avant tout, nous remercions ALLAH, le tout puissant, pour nous avoir offert la force, la patience et la détermination nécessaire à la réalisation de ce modeste travail.

Nous exprimons notre gratitude à notre promoteur, le P^r. H. Chkireb, et le remercions pour son aide, sa présence et sa perpétuelle bonne humeur, même dans les moments les plus difficiles.

Nos remerciements vont aussi aux membres du jury qui ont aimablement accepté de participer à l'évaluation de notre travail.

Au final, nous remercions toute personne ayant participer de près ou de loin à l'achèvement de notre mémoire, par un conseil, un encouragement ou tout autre moyen.

Dédicaces

- *À mes parents.*
- *À ma famille.*
- *À mes amis.*
- *Aux membres de l'auto-proclamé « labo A106 ».*
- *À toute la promotion automatique 2010.*

Houssem.

Dédicaces

Je dédie ce modeste travail :

- *À mes très chers parents et ma sœur que j'aime par dessus tout, qui par leurs sacrifices, leur amour, leur patience, leur soutien et leurs encouragements m'ont permis de toujours persévérer, de viser haut et surtout de toujours donner le meilleurs de moi-même malgré tous ;*
- *À mes oncles Ahmed et Arezki ainsi qu'à son épouse avec une attention particuliaire aux petits Anis et Serine ;*
- *À mes grand parents ainsi qu'à mon oncle Abdallah ;*
- *À ma tante unique ainsi qu'à toute sa petite famille ;*

Yanis.

ملخص

العمل المطروح في هذه المذكرة يتعرض لأنظمة المراقبة و التحكم عن بعد. قمنا بعرض عدة تقنيات مستعملة في هذا المجال. أولا، طبقنا هذه الطرق علي نظام بسيط لآدخال القاريء تدريجيا في صلب الموضوع و هذا بالفصل بين الجزء الشبكي و الجزء النظامي. اخيرا، ختمنا بنظام اكثر تعقيدا و واقعية بافق اقتصادي و هو نظام القياس عن بعد.

كلمات مفتاحية، مراقبة تحكم و اكتساب معطيات، الواجهة، وحدة تحكم قابلة للبرمجة، وحدة تحكم طرفية، بروتوكول شبكي، قياس عن بعد.

Résumé

Le travail présenté dans ce mémoire aborde les systèmes de supervision et de commande à distance. On a fait le point sur les différentes technologies utilisée dans cette optique en essayant d'être le plus complet possible. On a, dans un premier temps, appliquée ces méthodes sur un système simple et cela pour introduire le lecteur progressivement dans le sujet en séparant la partie réseau de la partie système. On a finalement conclu par un système plus complexe — mais aussi plus réel — et à perspective économique qu'est le système de télé-relève.

Mots clés : SCADA, interface homme-machine, automate programmable, terminal distant, protocoles réseau, télé-relève,

Abstract

The work presented in this paper deals with remote supervision and control. We have exposed different technologies usually used in this field. First, we applied these methods to a simple system. To progressively introduce the reader to the subject, we have separated the network part from the system part. Finally, we have concluded by a more complex — but more real — system with economical perspective, this system is the remote meter.

Keywords : SCADA, human-machine interface, programmable logic controller, remote terminal unit, network protocols, remote meter.

Table des matières

Table des figures

Introduction générale

Internet, réseau informatique mondial constitué d'un ensemble de réseaux nationaux, régionaux et privés — ce qui explique l'appellation réseau des réseaux — et qui sont *reliés* par le protocole de communication TCP/IP coopèrent dans le but d'offrir une interface unique à leurs utilisateurs.

Cette définition, aussi sommaire soit elle, résume tout l'intérêt d'internet et explique son incroyable succès auprès du grand publique, le rendant ainsi un standard *de facto* pour les échanges de tout genre.

L'expansion fulgurante réalisée par internet comme moyen de communication international d'un coté, et les proportions importantes prises par les systèmes technologiques distribués et embarqués de l'autre, nous amènent tout naturellement à réfléchir à un moyen nous permettant de bénéficier des avantages d'une éventuelle combinaison des deux.

D'un autre coté des systèmes appelés SCADA pour la télégestion et le monitoring à distance qui ont vu leur développement et donc leur utilisation exploser ces dernières années notamment dans les pays développés où leur utilisation s'est avérée plus que nécessaire car de plus en plus la télégestion des équipements devient inévitable vu l'importance qu'a pris la technologie dans notre vie quotidienne. Les systèmes SCADA permettent ainsi de diminuer du nombre d'opérateurs ou plutôt d'augmenter leur capacité de gestion sur un système mais aussi cela a permis de mettre à disposition la main d'œuvre la plus approprié pour un travail donné sans prendre en considération les contraintes géographiques et de mobilité des individu vis a vis des implantations des systèmes.

Même si internet est largement adoptée dans le domaine de la communication — échange de messages, textes, sons, vidéos, etc. — l'idée de son utilisation pour contrôler des systèmes industriels distants est relativement récente. Les quelques recherches menées seront exposées ultérieurement.

L'objectif du projet s'en trouve ainsi doublé. D'un coté, nous voulons développer une solution globale qui se servira des deux tendances précédemment citées — l'expansion d'internet et des systèmes distribués. D'un autre coté, on doit appliquer la solution développée à un système donnée afin de tester sa faisabilité.

Le présent document s'articule autour de cinq chapitres, comme suit :

Chapitre I, Objectif : Ce chapitre présente plus en détail le problème qui nous a été poser, à savoir les systèmes SCADA en général mais aussi quelques solutions déjà

apportées par d'autres études.

Sur ce dernier point nous expliquerons brièvement une technologie aujourd'hui très répondue et qui est en l'occurrence la télésurveillance par GSM, notamment utilisée dans les applications domotiques pour la transmission d'alarme mais aussi de commande tel l'allumage de lanternes ou déclenchement d'un appareil électroménager.

Chapitre II, Les systèmes SCADA : Dans ce chapitre, nous définissons, d'une manière exhaustive, les spécifications de base d'un système SCADA, ses domaines d'application et enfin la manière et les étapes par lesquelles passe la conception d'un système SCADA.

Chapitre III, Supports et protocoles de communication : Ici, on fera l'exposé des différents supports physiques de transmission lesquels sont utilisés dans les transmissions que ce soit locales ou distantes, nous verrons aussi les différents protocoles qui existent pour ces transmissions dites dédiées.

Chapitre IV, Exposé des solutions technologiques retenues : Nous exposerons l'architecture de la solution, les logiciels, langages de programmation et protocoles de communication choisis. On justifiera nos orientations en énumérant les critères de sélection.

Chapitre V, implémentation logicielle : Ce chapitre présente l'implémentation pratique des solutions théoriques développées tout au long du document. Nous donnerons des exemples applicatifs simples — voir simplistes — mais suffisants pour comprendre les rouages de fonctionnement du système.

Chapitre VI, système réel : Contrairement au chapitre précédent, où les exemples étaient plus didactiques que pratiques. Dans ce chapitre, on testera l'applicabilité des solutions sur un système technologique réel et complexe.

Conclusions et perspectives : On dressera un bilan générale du travail effectué. Et étant donné la relative jeunesse du domaine, on exposera les axes de recherche qu'il aurait était intéressant d'explorer et les différents problèmes rencontrés.

Chapitre 1

Objectif

Introduction

Commande et supervision d'un système technologique distant en exploitant le réseau informatique ! !

Qu'est ce qu'on veut dire par commande distante, car techniquement, même le fait de changer de chaine de télévision avec une télécommande peut être considéré comme de la commande distante.

Nous allons, dans ce chapitre, expliciter le sujet de notre étude et lever l'ambiguïté sur certains aspects de l'intitulé. Cela nous permettra de bien définir la problématique, ce qui, au final, nous aidera à mieux cerner les différentes facettes du problème, et ainsi facilitera notre travail.

1.1 L'intitulé

Même si l'idée générale du sujet peut paraître claire, il n'en reste pas moins que l'intitulé de l'étude peut prêter à confusion, pour éviter cela nous commencerons par la définition des différentes parties qui le constituent.

Commande et supervision : ces deux notions ne doivent pas être étrangères à l'esprit d'un automaticien, puisque c'est les deux activités qui constituent l'essence même de l'automatique.

La supervision est la mesure de certaines grandeurs physiques — celles qui nous intéressent — à l'aide de capteurs — physiques ou logiciels — et leur injection dans le régulateur à des fins de commandes automatique, ou bien simplement leur affichage sur un moniteur, pour qu'un opérateur puisse prendre les décisions adéquates dans l'éventualité d'une commande manuelle.

La commande — automatique ou manuelle — est l'envoi de consignes à un système pour réaliser certaines tâches, et cela en exploitant — en général — les informations

recueillies au cours de la supervision.

Système technologique : toute machine dont la commande et la supervision nous intéresserait.

Distant : cette notion est relative, puisque la majorité des commandes sont implémentées sur des régulateurs indépendants du système commandé et donc distants. Nous admettrons dans notre étude qu'un système technologique est dit *distant* s'il est dans un site géographique différent de celui d'où la consigne est envoyée. Au minimum, les deux organes seront situés dans des salles différentes. Mais l'étude ne révèlera son véritable intérêt que si les deux systèmes sont séparés par une grande distance.

Le réseau informatique : c'est un ensemble d'équipements reliés entre eux pour échanger des informations. Cette définition ne lève pas vraiment le voile sur la nature du réseau qu'on utilisera au cours de notre étude. Notre choix s'est arrêté sur *Internet* puisque c'est le réseau le plus étendu et le plus démocratique — dans le sens populaire, et non dans le sens démocratie ! ! — et sa réputation n'est plus à faire tant son adoption s'est généralisée au sein de secteurs aussi sensibles qu'importants.

1.2 La problématique

La définition des différentes parties de l'intitulé, les unes indépendamment des autres nous aide, certes, à mieux comprendre le problème, mais cela ne nous dispense pas de faire le lien entre ces notions pour ainsi avoir une vision d'ensemble du travail qui sera nécessaire d'effectuer et avoir les idées claires pour proposer une solution adaptée et innovante.

Que voulons nous réaliser ? !

Le but final de notre étude est la réalisation d'un système distribué nous permettant de contrôler et d'observer une machine distante — convenablement équipée — et cela, à travers le réseau Internet. Tout ceci n'aura véritablement d'intérêt que si notre implémentation est générique, c.-à-d. qu'on pourrait l'appliquer à un grand nombre de systèmes avec des adaptations minimes pouvant être réaliser par quiconque, sans avoir besoin de connaissances approfondies du sujet.

1.3 Commande à distance via le réseau GSM

M2M — acronyme de Machine To Machine ou bien Man To Machine ou Mobil To Machine — est une technologie qui s'est développée ces dernières années d'une manière exponentielle car la demande étant grande, la technologie suit nécessairement.

Le M2M proprement dit est l'infrastructure qui permet ces communications, elle permet en outre de gagner du temps car en général certaines machines et équipement devant

être commander à des milliers de kilomètres alors quoi de plus utile qu'un moyen qui permettra à son détenteur d'effectuer sa tâche sans avoir à se déplacer.

Un exemple d'application typique est les feux de circulation installés à diffèrents endroits de la ville et où il est inconcevable de mettre un agent devant chaque feu — sinon a quoi bon de les installer. Un tel système permettra une totale autonomie tout en ayant un œil et surtout un privilège de commande sur l'organe asservi.

Cette technologie peut être implémenté sur plusieurs support de communications telle les WAN, le réseau téléphonique, le réseau cellulaire, les réseaux radio privés ainsi que les communications par satellite et bien d'autres technologies. Dans cette partie nous présenterons une solution apportée par une étude antérieure — pour avoir une de ce qui se fait dans le monde — et opérant sur un réseau cellulaire particulier qu'est le GSM.

1.3.1 Le GSM

Le GSM pour — global system for mobile communication — est le réseau cellulaire qui offre la meilleure couverture dans la majeur partie du globe et qui permet en outre la disponibilité presque tout le temps du média en question. Aussi, le GSM est un réseau très sécurisé et il ne permet aucune intrusion de la part d'un étranger aux données envoyées via ce réseau, une dernière raison pour sa forte popularité est le coût d'utilisation qui est très bas relativement au service rendu et c'est pour ces raisons que beaucoup de systèmes de communication se sont appuyés sur cette technologie.

Le GSM offre trois différents services de connexion.

Le Dual Tone Multi Frequency — DTMF —

Qui est un service couramment utilisé et qui consiste en l'envoi d'une paire de sons aux fréquences audibles par l'oreille humaine — 1209 – 1633 Hz/697 – 941 Hz — celles-ci seront perçu par des serveurs vocaux qui pourrons ainsi fournir le service attendu.

Ses applications sont très répondues et nous pouvons citer l'exemple des centres d'appel qui utilisent ce service afin de guider le client vers le téléopérateur le plus à même de lui rendre service.

Une autre application dans le domaine des M2M est l'appel vers sa banque pour connaitre son solde en tapant le numéro de son compte sur le clavier de son téléphone.

Le short message Service — SMS —

Ce service offre au consommateur la possibilité d'envoyer 160 caractères alphanumériques, cette solution est populaire car elle est le moyen de communication préféré de la jeune génération mais aussi elle trouve très bien sa place dans les applications M2M qui permet ainsi l'envoi et la réception d'informations sous forme alphanumérique entre machine.

Le General packet radio service — GPRS —

Le GPRS est un service qui consiste en l'envoi de paquet de données sur le réseau GSM, il a la particularité d'offrir des débits très élevés relativement aux deux premiers services proposés, celui-ci trouve ses applications dans les système complexes mais surtout critiques qui on besoin d'envoi de données en continu telle des logs — fichier journal, cependant ce service à l'inconvénient d'être un peu plus cher que les deux premiers.

1.3.2 Structure proposée pour un système M2M

Une structure des M2M a été définie comme le montre la figure 1.1 où il y a trois sous-systèmes.

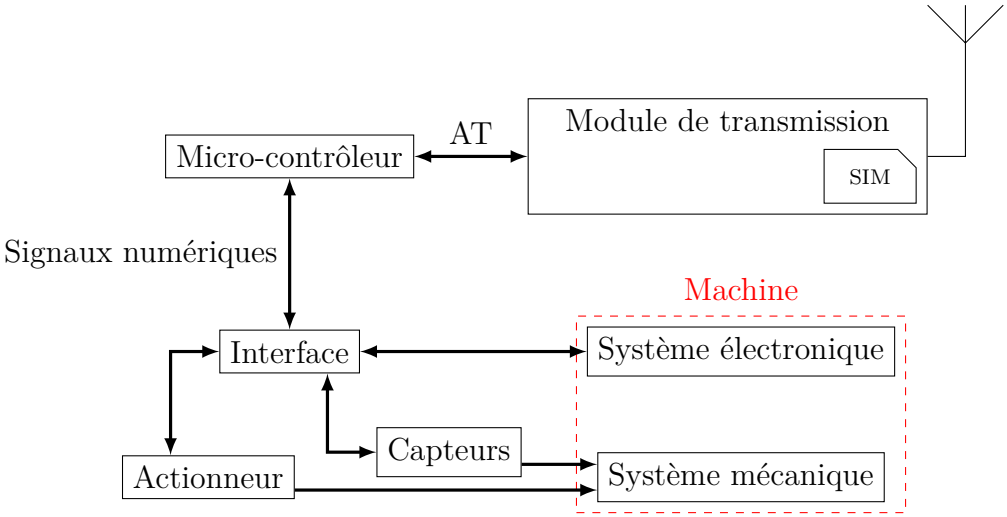


FIGURE 1.1 – Structure possible pour un système M2M.

La machine

C'est le système à commander ou à superviser, il peut être soit mécanique ou électronique.

Dans le premier cas, il doit y avoir des capteurs ainsi que des actionneurs pour traduire les signaux et/ou les états physiques en leurs finalités respectives.

Dans le second cas, l'opération est encore plus simple car généralement les machine électrique et électronique sont équipées de port spécialement conçu pour les communications et qui sont programmable à la guise de l'utilisateur.

Le microcontrôleur

C'est l'organe dit d'intelligence car c'est lui qui décrypte les signaux, paquet ou message qui lui parviennent pour les transformer en signaux électriques adaptés aux organes d'action tout comme il transcrit les informations qui lui parviennent depuis les capteurs

dans des langages, paquets ou autre signaux pour les transférer vers le module de transmission.

Le microcontrôleur peut être soit embarqué sur une plaque dédiée, soit de type PC — pour personal computer.

Un module de transmission GSM

Le module de transmission doit impérativement être équipé d'une carte SIM — Subscriber Identity Module card — qui lui permet l'authentification dans le réseau GSM afin d'avoir le privilège d'emmètre et de recevoir des flux d'informations, le module en question peut en tout cas être assimilé a un téléphone portable équipé d'une antenne radio qui émet et reçoit d'une cellule qui appartient a l'opérateur téléphonique.

1.3.3 Les algorithmes de communication

Un exemple typique de ce système a fait l'objet d'un développement au sein de l'université de King Saud.

Voici les algorithmes implémentés sur ce type de systèmes standards.

Algorithme de commande

1. **Initialisation des modules :** tout appareil à intelligence artificielle nécessite une étape d'initialisation qui en l'occurrence commence par vérifier l'état des périphérique un à un en envoyant des requêtes vers ces derniers puis d'analyser leurs réponses en commençant par les périphérique internes du PC puis de la connexion RS232 vers la carte d'acquisition et du port COM avec le module de transmission Puis vient l'étape d'initialisation des headers : deftypes.h, RS232.h, SerComm.h, and ATCommand.h. Et enfin, celle des librairies.
2. **Vérification et lecture du message :** le service utilisé est le SMS qui est envoyé par l'utilisateur distant pour l'exécution d'une tâche.

   Dans le projet en question la tâche a été simplifiée à un relai pour illustrer le fonctionnement. Le message étant chargé depuis le module de transmission et vérifié qu'il a été bien reçu par le le destinataire grâce à un algorithme de détection d'erreur de trame.
3. **Décodage :** le message en question est ensuite décoder par une application qui le transforme en lignes de code pour l'application de commande.
4. **Application de commande :** l'application est le programme réalisé afin d'exécuter les tâches que demande l'utilisateur.

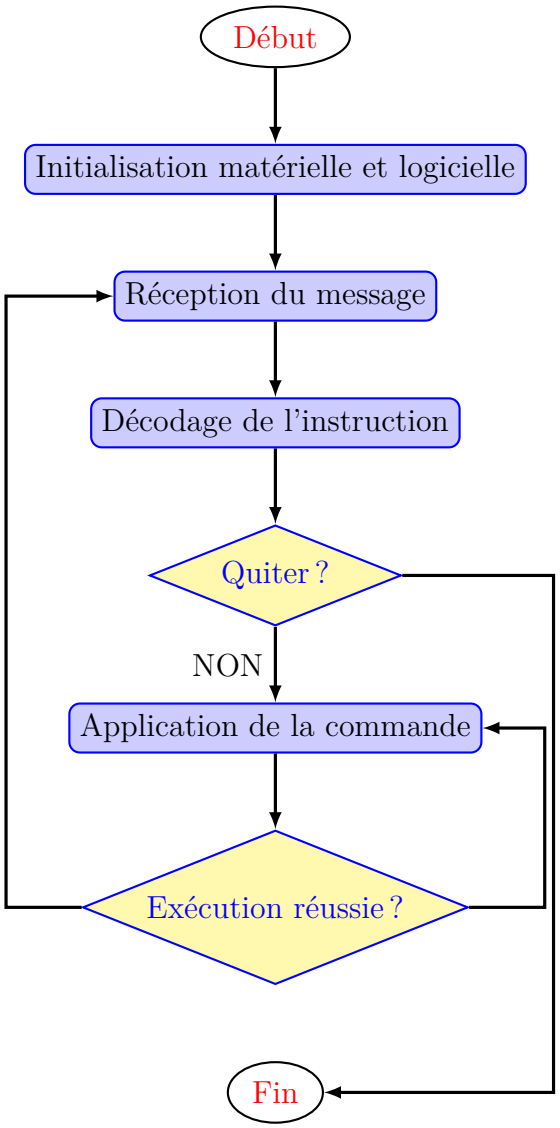


FIGURE 1.2 – Algorithme de commande.

Algorithme de supervision

1. **Initialisation du module.**
2. **Requête pour capture :** la requête de capture est une commande qui demande aux capteurs d'effectuer des relevés de grandeur.
3. **Encodage de l'information recueillie :** c'est l'opération inverse du décodage expliqué précédemment.
4. **Envoi d'un message contenant l'information.**

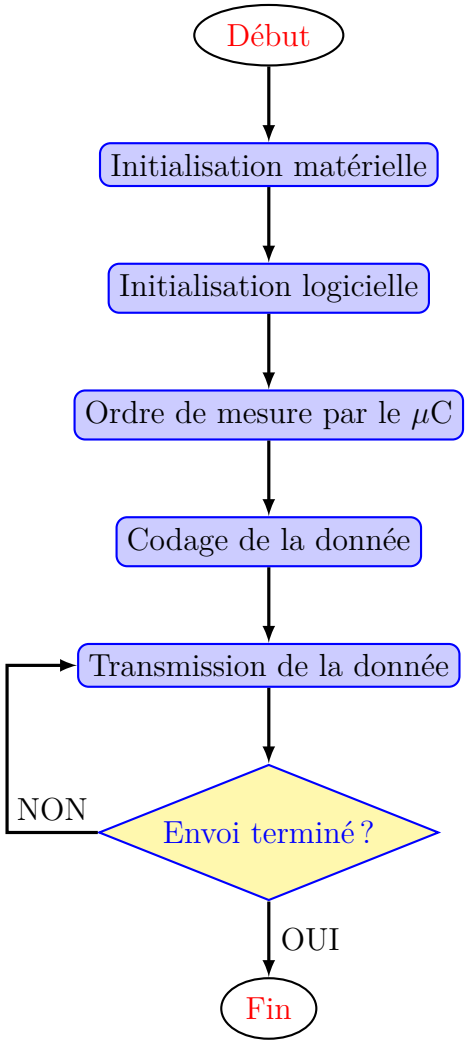


FIGURE 1.3 – Algorithme de supervision.

1.3.4 Inconvénients de la méthode

- Le prix d'installation d'un tel système reste moyennement abordable pour ce genre d'application, cependant à l'exploitation cela risque d'être un peu plus difficile à rentabiliser car généralement les opérateurs facturent la prestation a l'unité et donc plus il y a de communication plus la facture sera élevée.
- Le débit de la technologie GSM est actuellement très bas en rapport avec les application de télécommande même qu'il existe des algorithmes de compression de l'information capable d'optimiser les communications cependant il faut noter qu'en moyenne il faut prévoir trente secondes comme étant le temps d'une communication dont vingt secondes rien que pour la numérotation.
- La technologie GSM est vulnérable d'un point de vu physique car il est très simple de mettre en quarantaine le système — une cage de faraday peut le faire ou bien un brouilleur de signal comme celui utilisé dans les mosquées.
- La technologie GSM est vulnérable d'un point de vu logiciel, puisque récemment l'algorithme de chiffrage utilisé a été cassé.

Conclusion

Notre objectif ainsi précisé, nous pouvons, désormais, entrer dans le vif du sujet, en exposant l'architecture générale des systèmes de commande et de supervision à distance.

Chapitre 2

Les systèmes SCADA

Introduction

Dans ce chapitre, nous expliquerons ce que c'est qu'un système SCADA et nous exposerons ses caractéristiques, avantages, inconvénients, ainsi que ses différents domaines d'application.

2.1 Définition

SCADA — pour Supervisory, Control And Data Acquisition, ce qui veut dire supervision, commande et acquisition de données — est un type de technologie industrielle dans le domaine de l'instrumentation. On peut donner cette appellation à tout type d'application recevant des données depuis un système physique et y envoyant des commandes.

2.2 Composants d'un système SCADA

Un système SCADA se compose généralement des éléments suivants :

- une interface homme machine — IHM — pour présenter les états du système à l'opérateur humain et lui permettre de contrôler le processus ;
- un système — ordinateur — de supervision pour l'acquisition de données et l'envoi de commandes au processus ;
- des terminaux distants se connectant au capteurs et convertissant les signaux reçus en données numériques exploitables par le système de supervision ;
- des automates programmables — PLC — utilisés sur le terrain car plus économiques, versatiles, flexibles et configurables que des terminaux distants dédiés ;
- infrastructure de communication permettant de connecter le système de supervision aux terminaux distants.

2.2.1 L'interface homme machine

Une interface homme machine présente les données du processus à l'opérateur humain, et lui permet de commander le système.

Une interface homme machine est généralement liée à la base de données du système SCADA. Elle affiche les données pour permettre le diagnostic et la gestion des information, comme les procédures de maintenance, les informations logistiques, les schémas détaillés pour un capteur donné, etc.

L'interface homme machine présente l'information à l'opérateur *graphiquement*, sous forme de diagrammes mimiques. Cela veut dire que l'opérateur peut voir une représentation schématique des éléments constituant le processus. Par exemple, un pictogramme d'une pompe connectée à une canalisation peut montrer à l'opérateur que la pompe est en état de marche et la quantité de fluide qu'elle est entrain de pomper instantanément. L'opérateur peut alors éteindre la pompe. L'interface homme machine montrera le flux du fluide diminuer en temps réel. Un diagramme mimique consiste en des symbole schématique pour représenter les éléments du processus, ou peut aussi être une simple photo de l'élément — mais c'est moins courant.

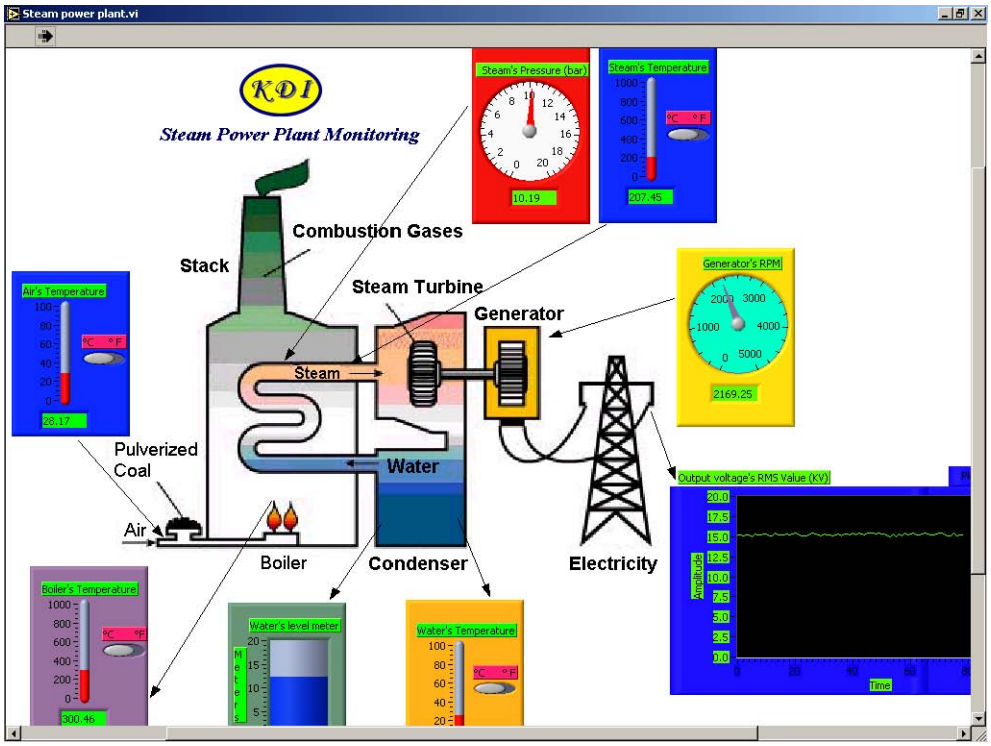


FIGURE 2.1 – Exemple d'interface homme machine basique.

La figure 2.3 montre bien l'intérêt d'une interface graphique claire. Même si l'interface est en perse, un ingénieur peut facilement la manipuler pour la simple raison que les symboles utilisés sont *universels*.

Une des parties les plus importantes dans n'importe quelle implémentation d'un sys-

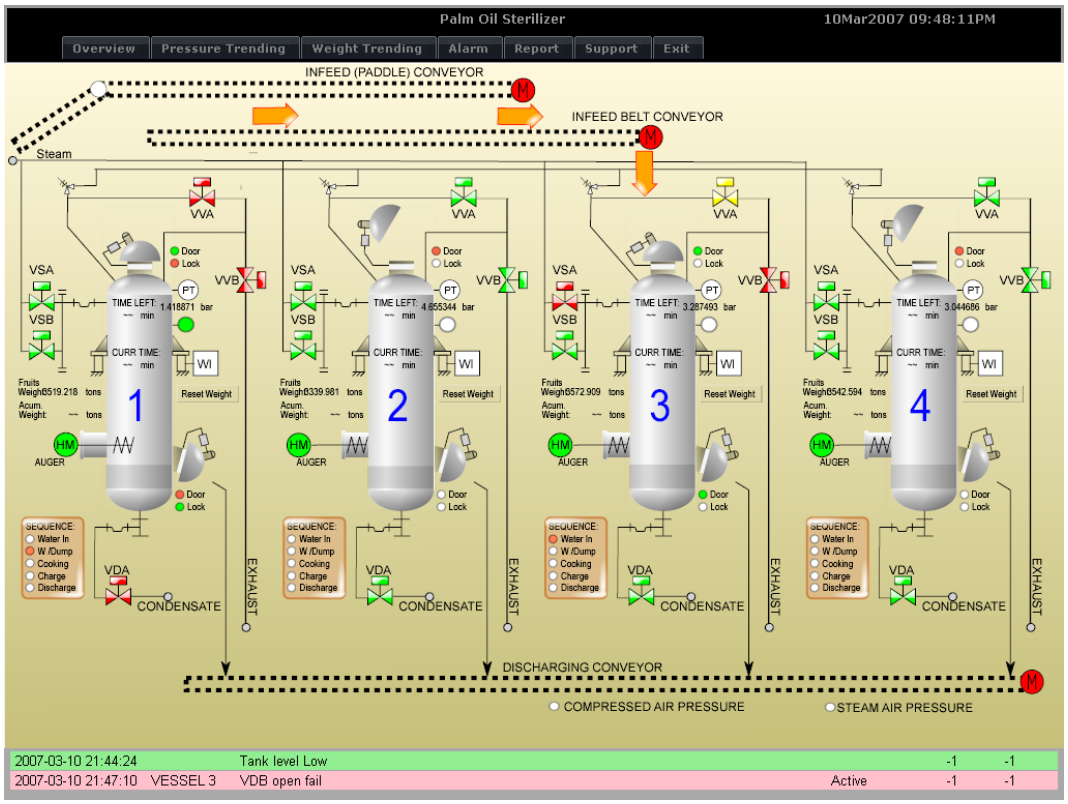


FIGURE 2.2 – Exemple d'interface homme machine plus moderne.

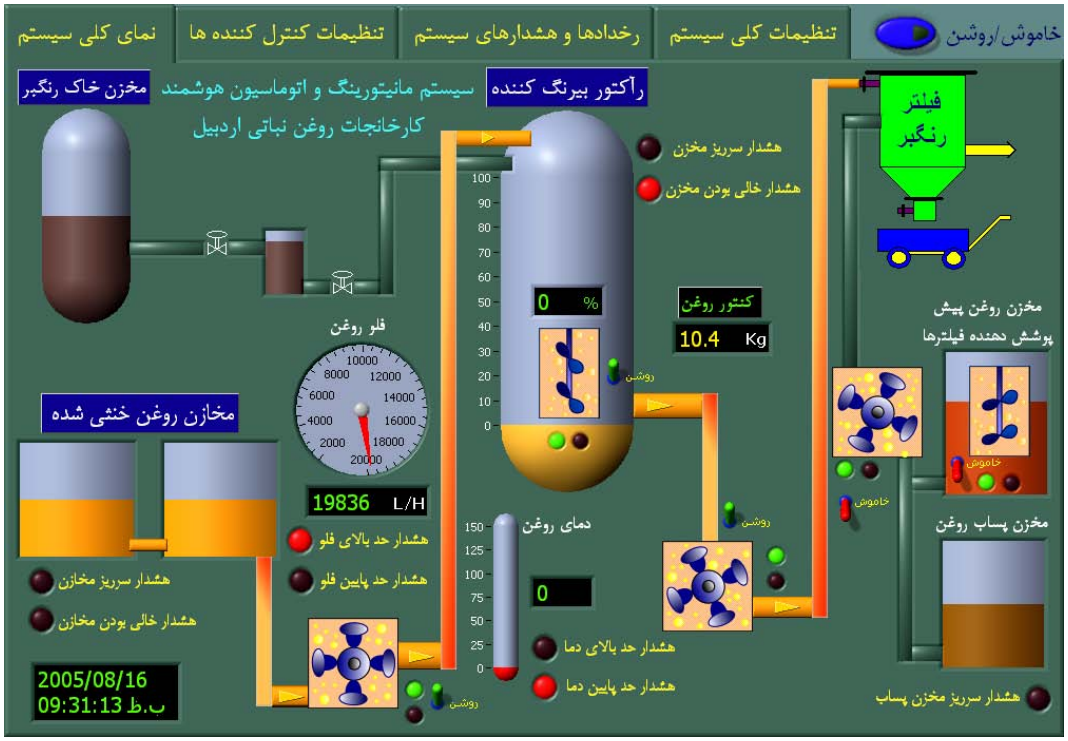


FIGURE 2.3 – Exemple d'interface homme machine en langue exotique.

tème SCADA est la gestion de l'alarme. Le système surveille si certaines conditions sont satisfaites, pour déterminer quand une alarme doit être déclenchée. Dans ce cas, on peut

entreprendre plusieurs actions — par exemple, allumage d'un voyant, déclenchement d'une sirène, etc. Les conditions de déclenchement de l'alarme peuvent être explicites — par exemple, une alarme est une variable binaire qui accepte les valeurs NORMAL ou ALARME qui est calculée par une formule mathématique basée sur les valeurs des autres points de mesure — ou implicites : le système SCADA peut automatiquement surveiller si la valeur d'un point de mesure analogique sort de certaines limites associées à ce point.

Une alarme peut être représentée de plusieurs manières différentes, par l'émission d'un son, l'apparition d'une fenêtre pop-up ou l'allumage d'un voyant ou d'une partie de l'écran. Dans tous les cas, le rôle de l'alarme est d'attirer l'attention de l'opérateur pour qu'il puisse prendre les décisions appropriées.

Dans la conception d'un système SCADA, une attention particulière est prêtée à la gestion d'une cascade d'alarmes survenant dans une courte période de temps, sinon, l'alarme la plus dangereuse — qui n'est pas nécessairement la première déclenchée — peut être noyée dans la masse d'information.

Malheureusement, le mot « alarme » est mal utilisé dans l'industrie. Au lieu de représenter un état — problématique — du système, il représente l'indicateur déclenché dans ce cas.

2.2.2 Le système de supervision

C'est les dispositifs matériels et logiciels s'occupant de faire le lien entre les équipements de terrain — automates, terminaux, etc. — et l'interface homme machine du centre de contrôle. C'est le chainon du milieu du système SCADA. Dans les petites installations, ça peut être *un seul* ordinateur. Par contre, dans une installation conséquente — plus probable sur un plan industriel — ça consistera en un réseau de serveurs, des logiciels distribués, et des postes de sauvegarde.

Pour préserver l'intégrité des données, on utilise souvent des architectures redondantes. Cela permet de se prémunir contre d'éventuelles pannes du système et ainsi assurer un contrôle continu.

2.2.3 Le terminal distant

Le terminal distant se connecte à un équipement physique. Il convertit les signaux provenant de l'équipement vers des valeurs binaires — par exemple, état ouvert/fermé d'un relai ou d'une vanne — ou bien en valeurs numériques — pression, flux, voltage, courant, etc.

Par la conversion et l'envoi de ces signaux électriques, le terminal distant peut contrôler un équipement — ouvrir un relai ou une vanne ou régler la vitesse d'un moteur.

Ils sont plus appropriés aux installations conséquentes et environnements hostiles, par exemple : les plateformes pétrolières.

FIGURE 2.4 – Un terminal distant Kadtronix.

Les terminaux distants jouent aussi le rôle d'interface de conversion, puisqu'ils peuvent envoyer les mesures voulues sous forme de texte cela simplifie le système en l'allégeant d'un composant supplémentaire et facilite le traitement de ces mêmes données.

2.2.4 L'automate programmable

Ce sont des régulateurs généralistes basés sur un microprocesseur. Ils fournissent une panoplie de fonctions très intéressantes pour la commande : opérations logiques, timers, compteurs et prédisposition à la mise en réseau.

FIGURE 2.5 – Un automate programmable AutoLog.

C'est une combinaison de modules — ou cartes — montés sur un support physique disposant d'infrastructure d'interconnexion électrique appelé *rack*. Comme son nom l'indique il est programmable, donc, adaptable à des systèmes aussi divers que variés.

Avantages

– Ils sont spécialement développés pour les applications industrielles ce qui assure une fiabilité et une tolérance aux éléments extérieurs — chaleur, vibrations, interférences électromagnétiques, etc.

- Leur large pénétration du marché garantie leur disponibilité et le support technique qui va avec.
- Ils fournissent des vitesses de traitement élevées, ce qui a une grande importance dans les applications sensibles comme la génération d'énergie.
- Ils supportent les configurations redondantes ce qui est un gage de fiabilité.

2.3 Transmission et formatage des données

La transmission des données dans un système SCADA à une grande importance, elle fera l'objet d'un chapitre dans notre étude, intéressons nous, donc, au formatage pour le moment.

Une donnée informatique n'est en fait qu'une suite de bits mis à 1 ou 0. Ça signification n'est connue qu'après que les deux systèmes — émetteur et récepteur — se soient mis d'accord sur une convention d'interprétation. Cette *convention* est appelée un *format* de fichier, de communication, etc.

C'est exactement le même principe pour la commande et la supervision, on doit pouvoir représenter l'état du système et les commandes à lui envoyer. Pour cela, il existe plusieurs possibilités :

Créer son propre format : c'est la solution la plus flexible et la plus difficile à mettre en œuvre. Non seulement il faut penser à tous les cas possibles mais on doit en plus du format créer son parser — une application qui extrait les données utiles.

Créer un format XML : cette solution élimine le besoin de coder son propre parser, puisque XML étant un méta-langage d'organisation de données très adopté, il existe un choix conséquent de librairies informatiques pouvant remplir ce rôle. Seulement, cette solution exige de bien penser aux différents aspects du problème d'un point de vue logiciels ce qui est synonyme de temps supplémentaire.

Utiliser RCXML : le Remote Control XML, c'est la solution précédente, mais conçu par des ingénieurs chevronnés spécialisés dans les standards industriels, cela nous garantie la fiabilité du langage tout en nous évitant l'étape de réflexion — supplémentaire — que nous aurions été obligés d'effectuer si nous avions conçu notre propre langage — même basé sur XML.

Nous pensons, donc, que la dernière solution est la meilleur pour des systèmes réels complexes. Puisqu'elle nous assure l'exhaustivité dans un temps record.

2.4 Les concepts des systèmes SCADA

Le terme SCADA se réfère généralement à un système centralisé qui supervise et contrôle un site industriel entier, ou des complexes industriels éparpillés sur une large

zone — large zone voulant dire n'importe quelle surface supérieure à un site industriel. La plupart des décisions sont prises automatiquement par un terminal distant ou un automate programmable. Le rôle de l'opérateur humain est souvent restreint au niveau de la supervision et d'un contrôle basique. Par exemple, un automate programmable peut commander le flux d'un liquide de refroidissement dans un processus industriel, mais le système SCADA peut autoriser l'opérateur à changer la consigne, déclencher une alarme — en cas de perte de pression, ou d'augmentation anormale de température, bien que ces opérations soient aussi prises en compte par le terminal distant ou l'automate programmable.

L'acquisition de données commence au niveau du terminal distant ou de l'automate programmable, elle inclut la lecture des valeurs des différents capteurs et rapporte l'état actuel du processus qui est transmis au système SCADA si nécessaire. Les donnée sont alors traitées et formatées de manière à ce que l'opérateur du centre de contrôle puisse prendre des décisions pour ajuster ou redéfinir la décision prise par le terminal distant ou l'automate programmable. Les données peuvent aussi être sauvegarder dans une base de données, pour ainsi permettre une analyse ultérieure.

Les systèmes SCADA utilisent généralement une base de données distribué et redondante, qu'on appelle une base de données de *tag*. Elle contient des données élémentaire appelées *tag* ou *point*. Un point représente une seule valeur d'entrée ou de sortie surveillée ou commandée par le système. Un point peut être réel ou virtuel. Un point réel représente une entrée ou une sortie du système, tandis qu'un point virtuel est le résultat d'une combinaison logique ou mathématique de points réels. Pour plus de simplicité, on peut s'affranchir de la distinction entre point virtuel et point réel en considérant tous les points comme étant virtuels, qui seront égaux à des points réels dans le plus simple des cas. Les tags sont enregistrés sous forme de « valeurs - temps d'accès ». Une série de « valeurs - temps d'accès » donne l'historique d'un tags donné. Il est aussi possible d'enregistrer des données supplémentaires sous forme de méta données.

2.5 Domaines d'application

SCADA nous entoure ! !

Partout où nous allons, une déclinaison du système est présente — surtout dans les pays développés. Voici une liste non exhaustive des domaines d'application :

- surveillance de processus industriels ;
- transport de produits chimiques ;
- systèmes municipaux d'approvisionnement en eau ;
- commande de la production d'énergie électrique ;
- distribution électrique ;
- canalisations de gaz et de pétrole ;

2.5.1 L'énergie électrique

C'est un domaine où les systèmes de télégestion sont très présents, et cela dans toutes les étapes du processus.

La production

Dans une centrale électrique, un système SCADA sert à visualiser les paramètres des machines auxiliaires, telles que les pompes du liquide de refroidissement, la température des chambres de combustion, les débits de combustible et d'aire à différents stades du processus, etc.

Toutes ces données sont critiques dans la gestion d'une centrale électrique. Et pour cela, la notion de temps réel est très importante et est souvent déterminante dans le choix des technologies pour la mise en œuvre du système SCADA. Cependant, pour des raisons d'ergonomie notamment, des calculs additionnels sont fait en aval et en parallèle à la réception des données pour générer d'autres informations qui peuvent aider le responsable de la centrale dans ses prises de décision, notamment dans la gestion des programmes de maintenance.

Le transport

« L'électricité n'est pas stockable sur le long terme, tout ce qu'est produit doit être rapidement consommé ». Cette contrainte illustre parfaitement la nécessité d'un système automatique de gestion.

Ce que l'on nomme *partie transport* commence à la sortie des génératrices des centrales et se termine au début de la partie distribution — que nous exposerons ultérieurement. Elle se fait à très haute tension et les paramètres gérés par le système sont si complexes qu'il est difficile pour un être humain ni même une équipe de s'en occuper.

En effet, d'un côté la consommation varie chaque seconde et n'obéit à aucune loi mathématique déterministe. D'un autre côté, la production elle même — mais dans une moindre mesure — varie, s'ajoute à cela les pannes que le système doit palier instantanément en ordonnant à d'autres centrales d'augmenter la production pour éviter des blackouts qui coutent énormément cher.

La distribution

Cela concerne la moyenne et basse tension, c'est-à-dire ce qui arrive chez le client.

Le système SCADA associé peut varier beaucoup en terme de complexité car cela dépend des moyens de l'entreprise, aussi — faut-il le préciser ?! — cette partie n'est pas critique et peut être plus ou moins géré efficacement par des opérateurs humains.

Mais pour plus de confort d'utilisation, on peut imaginer des systèmes plus évolués tels que l'allumage et l'extinction *automatique* de l'éclairage publique, la télé-relevé des compteurs clients, etc.

Cela étant des options superficielles — mais très intéressantes à moyen et à long terme d'un point de vue financier — l'essentiel du système réside dans la communication avec la partie transport. Et notamment pour ce qui concerne la consommation. Car dans certain cas, la partie distribution du système SCADA peut prendre le relai et entreprendre des opérations de délestage pour priver une certaine partie du réseau de l'électricité selon une hiérarchie afin de limiter la consommation momentanément.

2.5.2 Les infrastructures de transport individuel et en commun

Là aussi le système SCADA est très présent sur tous les types de transport et pour différentes tâches.

Le transport sur voies routière

Le système gère le trafic, les itinéraires ainsi que les relevés des employés dans les bus, les camions et tout type de véhicules roulants. Généralement équipés de système de détection par satellite ou GSM, certaines flottes publiques ou privées peuvent être tracées à tout moment et partout dans les villes ou le GSM est bien établi. Cela sert généralement à planifier des itinéraires et à suivre des livraisons. Cela permet d'éviter des zones de congestions aux véhicules de la flotte quand le système est connecté à un autre système qui, lui, gère le trafic routier.

Le tramway et le métro

Dans ce type de systèmes il y a la gestion classique des transports, mais aussi, l'alimentation du réseau en électricité. En plus du changement de voie qui doit être extrêmement synchronisé.

L'aviation

Le radar et les outils de télécommunication entre la tour de contrôle et les avions en l'air constituent un système SCADA.

2.5.3 Dans les usines

Tout ce qui est équipement dans une usine moderne comporte une sortie dite de « communication » qui est en réalité une prédisposition à la mise en réseau ou bien à être connecté à un terminal local dans les petits ateliers.

Dans une usine, par exemple, une des tâches que peut accomplir un système SCADA dédié est la synchronisation entre les différents organes et machines tout en visualisant les états en temps réel.

2.6 Architecture des systèmes

Il existe plusieurs classes de systèmes de commande :

2.6.1 Commande locale

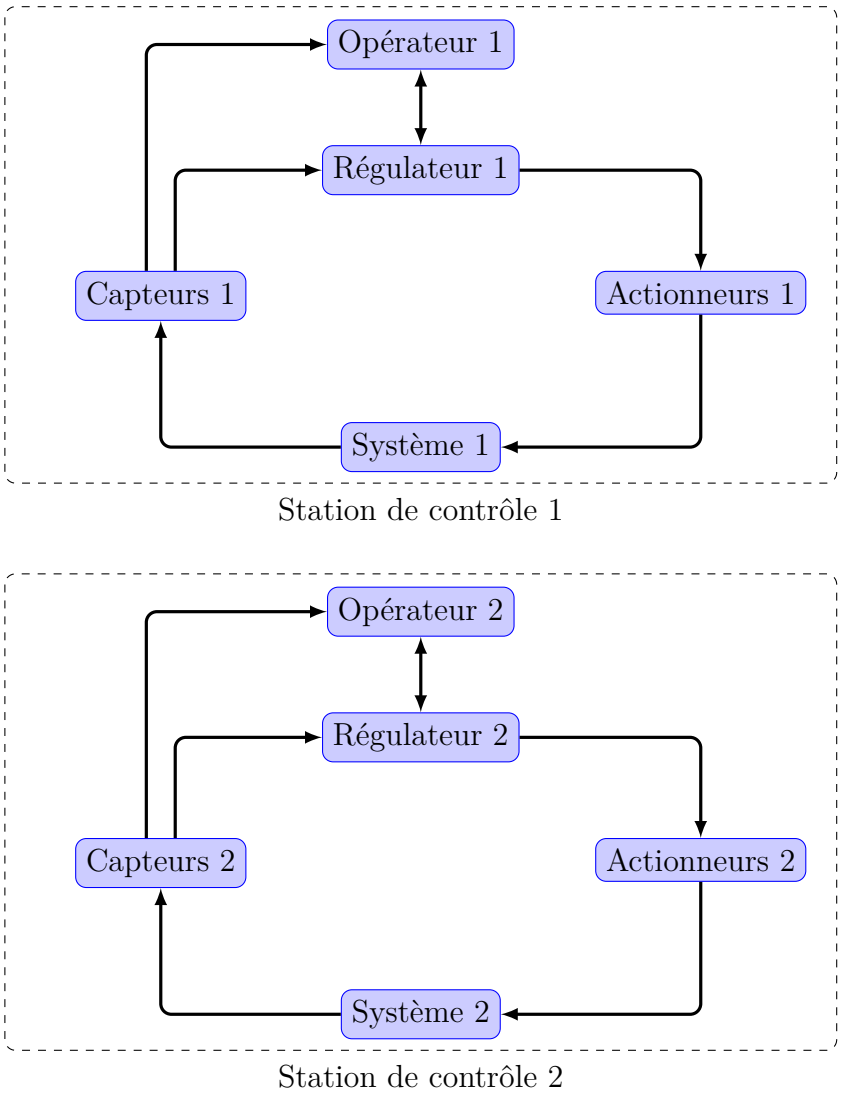


FIGURE 2.6 – Représentation schématique d'un système à commande locale

Dans ce genre de systèmes, tous les composants de la boucle de commande — opérateur, régulateur, actionneurs, capteurs et système — se situent à proximité les uns des autres. Et la portée de chaque régulateur se limite à *un* système ou sous-système. La figure 2.6 représente le cas de deux systèmes dont chacun est commandé localement.

Avantages

- Indépendance des processus les uns des autres, c.-à-d. la défaillance de l'un des sous-systèmes n'influe aucunement sur les autres sous-systèmes.
- La rapidité de transmission des commandes et la réception des donnée, puisque les liaisons sont exclusivement réservées à ces tâches.
- Facilité de dépannage accrue puisque tous les composants se trouvent dans le même site.

Inconvénients

- La multiplication des régulateurs — calculateurs — donc l'augmentation du coût total du système.
- La présence d'un opérateur est obligatoire pour chaque sous-système.

2.6.2 Commande centralisée

Tous les organes de la boucle de commande sont reliés à un seul régulateur situé dans la station de contrôle, qui peut donc contrôler un nombre quelconque de sous-système — plus précisément, cela dépend des capacités du calculateur, et du nombre d'entrées et de sorties dont il dispose. Ce type d'architecture était très répondu dans l'industrie lourde et les premiers régulateurs numériques, mais maintenant elle est largement supplantée par les systèmes distribués à cause de problèmes de coût et de fiabilité évidents.

Ce système constitue un exemple de commande de système à distance, puisque les systèmes commandés par le régulateur centrale peuvent — et le sont généralement — être éloignés géographiquement de lui. La figure 2.7 présente un schéma possible pour ce genre de configuration.

Avantages

- La diminution du nombre d'opérateur nécessaire pour le bon fonctionnement du système à *un*. Puisque tous les sous-systèmes sont contrôlés depuis le même régulateur.
- On a besoin que d'un seul régulateur, pour tout le système. Bien que cet avantage soit relative, puisque ce même régulateur doit être plus puissant qu'un régulateur qui ne s'occuperait que d'un seul sous-système.

Inconvénients

- La défaillance du régulateur central entrainera l'arrêt du système total, puisque aucun sous-système n'est indépendant.
- La soumission aux aléas des réseaux, latence, défaillance, etc.
- Problèmes de sécurité du réseaux.

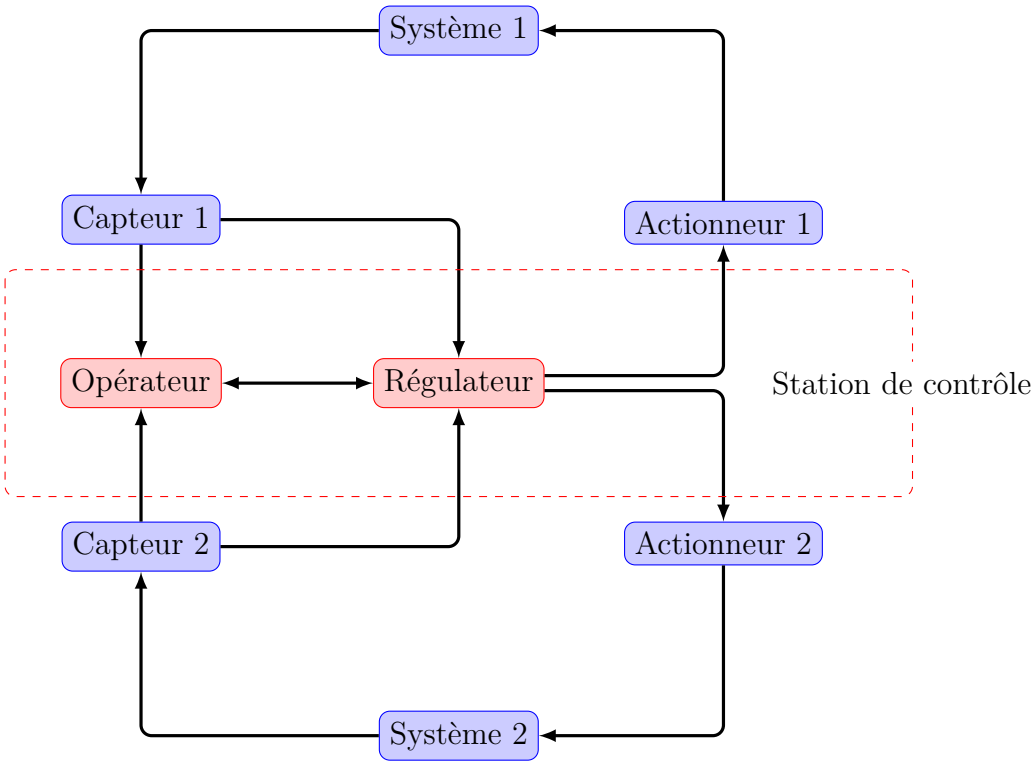


FIGURE 2.7 – Représentation schématique d'un système à commande centralisée

2.6.3 Commande distribuée

C'est la combinaison des avantages des deux configurations précédentes, d'un coté chaque sous-système est doté de son propre régulateur local, de l'autre tous les régulateurs sont connecté à la station de contrôle. Donc, la commande pour chaque sous-système est développée localement au niveau du régulateur, mais cela n'empêche pas la station de contrôle centrale de connaitre l'état du système global à tout instant, et de pouvoir agir sur celui-ci dans les limites permises par la configuration. La figure 2.8 schématise un exemple de ce genre de système.

Avantages

- Indépendance des processus, puisque la défaillance de l'un n'implique pas l'arrêt du système global.
- Robustesse : une coupure de la liaison avec la station de contrôle centrale n'est pas critique puisque les régulateurs étant locaux, ils peuvent assurer le bon fonctionnement pour un certain temps.
- Main d'œuvre restreinte, puisque un seul opérateur est nécessaire pour faire tourner le système et le superviser.
- Facilité d'entretien et de dépannage, car les organes principaux de la boucle de commande de chaque sous-système se situent dans le même local.

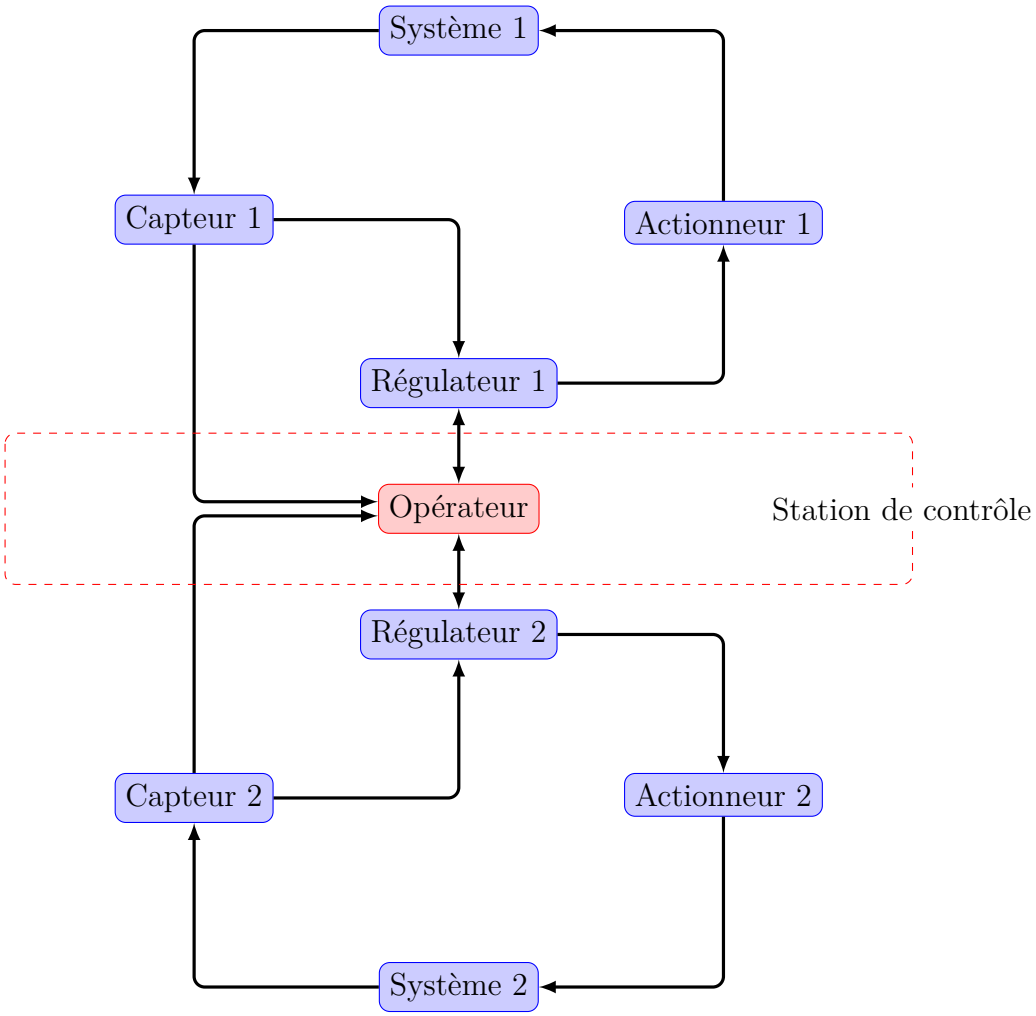


FIGURE 2.8 – Représentation schématique d'un système à commande distribuée

Inconvénients

- Temps de retard indéterminé — dans le cas d'Internet — lors de la transmission des signaux.
- Gestion de la sécurité et de l'authenticité des signaux de commandes.

2.7 Les types de systèmes de commande distribuée

2.7.1 DCS

Bien que l'appellation DCS — Distributed Control System — s'applique en général à tout système dont le régulateur est distribué — en opposition à un régulateur centralisé, dans les centrales électriques et l'industrie pétrochimique, on préfère généralement utiliser le terme de DCS pour décrire un type spécifique de régulateurs capable d'exécuter des algorithmes complexes de commande analogique à des vitesses très élevées, en plus de fournir des fonctions de surveillance et de journalisation. Dans la plupart des applications,

les modules d'entrées/sorties du système sont *distribués*, tandis que les régulateurs sont centralisés à proximité du centre de commande.

Ces systèmes utilisent du matériel, des logiciels et des protocoles de communication propriétaires, nous imposant un prestataire *unique* pour les pièces de rechange et la maintenance.

SCADA vs. DCS ! !

Il existe une certaine confusion, dans l'industrie, concernant la différence entre un système SCADA et un système DCS. Un système SCADA coordonne mais ne commande pas en temps reel. Mais cela n'est plus vraiment valable, au vue des nouvelles technologies émergente qui permettent des communications fiables, rapides, à latences faible sur des distances considérables. La différence entre ces deux systèmes s'estompe, donc, progressivement.

2.7.2 DDC

Les systèmes DDC — Direct Digital Control — sont utilisés dans les systèmes de chauffage d'immeuble, ventilation et de climatisation, pour maintenir les conditions environnementales désirées. Ils consistent en des régulateurs locaux connectés par un réseau à une station centrale — basée sur un ordinateur — qui offre les fonctions de surveillance, stockage de données et des possibilités de programmation. Les régulateurs sont optimisés pour des commandes économiques, qui ne requièrent — généralement — pas une grande vitesse d'exécution.

Le matériel et les logiciels sont propriétaires tandis que l'infrastructure et les protocoles de communication peuvent être ouverts ou fermés.

2.7.3 Systèmes SCADA à base de terminaux distants

Ils sont utilisés dans le domaine de l'électricité, le gaz et la distribution de l'eau, où la surveillance et le contrôle doivent être mis en œuvre sur des surfaces géographiques larges. Ils communiquent avec une station centrale en exploitant les lignes téléphoniques, la fibre optique ou les ondes radio. Les terminaux distants sont principalement utilisés pour la surveillance mais dans une moindre mesure pour la commande.

Le matériel et les logiciels sont propriétaires tandis que les protocoles de communication sont soit ouverts soit fermés.

2.7.4 Systèmes SCADA à base d'automate programmable

Les automates programmables peuvent être connectés ensembles pour partager des données, en plus de fournir des possibilités de surveillance et de contrôle. Les systèmes à

base d'automates programmables surpassent les DCS et les systèmes à base de terminaux distants dans beaucoup d'applications industrielles. Ils ont été développés pour l'automatisation des usines et sont traditionnellement très rapides dans la commande numérique, mais sont de plus en plus munis de possibilité de commande analogique.

Le matériel pour ces systèmes est propriétaire, mais les logiciels et les protocoles de communication sont ouverts, offrant ainsi une liberté de choix à l'utilisateur final.

2.8 Étapes de conception d'un système SCADA

Pour concevoir un système SCADA, un certain nombre de questions doit être posé.

2.8.1 Quelle est la taille de la structure à contrôler ?

Cette notion étant relative, des normes de grandeur ont été émises en ce sens afin d'harmoniser le langage.

Les systèmes SCADA sont divisés en trois catégories :

Les petits systèmes

Ce sont des systèmes qui ne requièrent pas beaucoup de matériel et sont sur un site géographiquement restreint. Dans ce type de système, la supervision et la commande est basée sur la technologie PLC plutôt que celle des RTUi, car étant autonomes et sans trop d'entrées/sorties il est plus concevable d'utiliser des PLC plutôt que de gros équipement pour un niveau d'efficacité et de sécurité égale voir même plus intéressant avec les PLC.

Dans ces système généralement un terminal de type PC es relié directement au PLC qui fourni une interface graphique.

Les systèmes de taille moyenne

Ce sont des systèmes à mis chemin entre les petits systèmes et le système de taille conséquente. — à titre indicatif, une turbine à gaz double redondante est considérée comme étant un système de moyenne taille.

Les systèmes de grande taille

C'est la, la mise en association de plusieurs systèmes de moyenne taille à des endroits géographiquement indépendant les uns des autres. La topologie étant la même, la différence est qu'a chaque sous système est associé un RTU qui s'occupe de la transmission des données vers le centre de contrôle.

2.8.2 Est-il nécessaire de maitre en œuvre la redondance ?

Redondance d'équipements de production

La disposition du matériel lié au SCADA dans ce contexte peut être défini de deux façons, la première étant de mettre à chaque équipement sont propre PLC puis de relier tout les PLC par un bus de terrain au centre de contrôle.

Cette configuration peut être intéressante dans la mesure où si un PLC tombe en panne il y aura toujours un relai au niveau du système de production en lui-même donc basculement vers le second bras de production.

Équipement de production avec redondance PLC à chaud

Dans cette configuration il y a toujours redondance au niveau des équipements mais les PLC sont disposés de manière à travailler en paire c'est-à-dire que chaque PLC travaille pour les deux équipements de production simultanément. Si le premier PLC tombe en panne, la charge de travaille se transmet sans interruption au second.

Bien que cette méthode semble plus séduisante que la première il ne faut pas négliger les deux facteurs suivant :

- la redondance a chaud surcharge considérablement les PLC et donc diminue leur durée de vie.
- cette redondance a chaud peut induire certain disfonctionnements lors de la transmission de charge a chaud, ces disfonctionnements peuvent induire des dégâts importants sur les équipements.

Donc, chacune des deux méthodes trouvent son application dans l'industrie.

2.8.3 Quel est le degré d'importance de chaque équipement ?

Afin d'émettre un mappage de capteurs à installer dans le système, plusieurs facteurs doivent être pris en compte.

La faisabilité : car installer un capteur à un niveau donné ne peut être sans conséquences pour le système entier.

Donc cette influence doit être limitée et sans risques, cela en choisissant les endroits appropriés dans le système mais aussi les capteurs adéquats au niveau de contrainte auquel ils seront soumis.

Le nombre : le nombre de capteur doit être proportionnel à l'ensemble de la tâche, et si une redondance au niveau des capteurs est nécessaire, celle-ci sera plutôt logicielle que matérielle.

Le coût : l'investissement total doit être bien réparti entre la partie SCADA et le système lui-même.

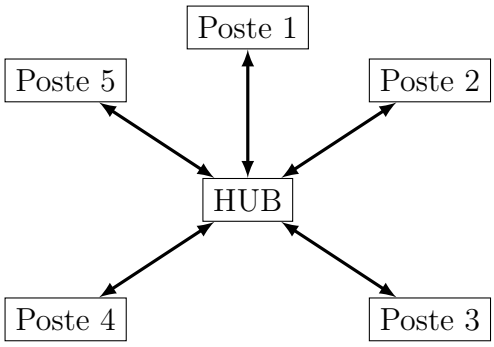


FIGURE 2.9 – Exemple de topologie — topologie étoile.

2.8.4 Quel type de communication doit-on utiliser ? Sur quel support ? Sous quel protocole ?

Les topologies des réseaux jouent un rôle dans la sécurité, la fiabilité et le coût d'un système. Par exemple, Dans la topologie étoile, si une branche tombe en panne ou le câble le reliant au reste du système est sectionné, le système continue de fonctionner normalement, ce qui n'est pas le cas de la topologie en mode bus qui est l'équivalent d'une cascade. Donc, si sectionnement il y a quelque part dans e réseau, c'est toute la structure qui se retrouve sans connexion physique.

À noter que ces topologies sont prises en compte pour des réseaux locaux. Généralement pour les réseaux distants, cette topologie est à un niveau logiciel et des programmes de routage configurent à chaque instant les chemins par lesquels doit transiter l'information, ce qui implique que les craintes sont beaucoup plus orientées vers la perte de l'information en elle-même que vers la perte d'un support physique.

Maintenant que nous avons les différentes topologies utilisées dans la mise en réseaux du matériel, intéressons nous aux critères de choix des supports physiques.

D'ordre mécanique : on doit toujours penser à la température que le support doit subir, mais aussi les contraintes de potence qui s'exerceront sur lui.

D'ordre électrique : cela inclut l'impédance du matériel qui détermine entre autre la longueur maximale du support, mais aussi les interférences électromagnétiques qui peuvent être induites par le câblage sur l'environnement extérieur et vis versa.

Le débit : en connaissant le débit théorique de chaque type de support et en connaissant le ratio de perte de l'information sur ce même support nous pouvons lui associer un débit dit réel — pour la quantité d'information correctement acheminer.

A la troixieme question nous repondrons par un tableau pour recapituler les protocols existants et leur domaine d'utilisation.

2.9 Libération des systèmes SCADA

Les systèmes SCADA tendent à être de plus en plus hétérogènes — c.-à-d. utilisation de pièces et composants provenant de différents fournisseurs.

Dans la première moitié des année 90, les constructeurs des composants utilisés dans la conception de systèmes type SCADA fournissaient un équipement communicant par des protocoles propriétaires. Donc l'utilisateur final qui a investi dans le matériel d'un certain constructeur se retrouve ainsi piégé, puisque limité à un choix restreint aussi tôt qu'il voudra changer de composants — pour cause de panne, d'augmentation de performance, etc. Pour palier à ce problème différents *standards ouverts* ont vu le jour — IEC 60870-5-101, IEC 61850, DNP3, etc. Cela a permis aux utilisateurs de monter leurs propres systèmes — en combinant des composants de différentes provenances — qui sont plus performants que les éventuelles solutions offertes par un constructeur unique.

La fin des années 90 a vu l'accentuation de la migration vers les standards ouverts, puisque même les « petits » constructeurs ont suivi la tendance.

De nos jours, l'ethernet et les protocoles basés sur TCP/IP remplacent progressivement les anciens standards propriétaires. Même s'il reste quelques domaines spécifiques qui n'ont pas encore fait le pas, pour différentes raisons.

2.10 Les problèmes de sécurité

Comme exposé dans la section « *domaines d'application* », les systèmes SCADA ont un champ d'application très large qui va de l'acheminement de pétrole et de gaz au transports en commun en passant par la production d'électricité. Ces application sont sensibles et très importantes dans les modèles sociales actuelles. Une éventuelle panne ou attaque visant ces installations aurait des répercussions graves sur notre confort de vie. Par exemple, un blackout causé par un système SCADA électrique compromis aurait un impact sur tous les consommateurs reliés à cette source.

Pendant longtemps, les industriels pensaient *contourner* les problèmes de sécurité des système SCADA en « cachant » les spécifications de leurs logiciels et protocoles de communication. Mais il s'est avéré que la stratégie inverse était beaucoup plus intéressante. Si on utilise des technologies standardisées, on profitera des améliorations qui leur est apportées aussitôt qu'elles sont publiées.

Les systèmes SCADA ont deux points faibles liés à la sécurité :

- les accès non autorisés au système ;
- la modification des trames en cours d'acheminement, puisque l'utilisation d'internet implique nécessairement le passage par un nombre inconnu de routers, switch et autres joyeusetés du réseau mondial.

Heureusement — ou malheureusement, ça dépend du point de vu — ses mêmes pro-

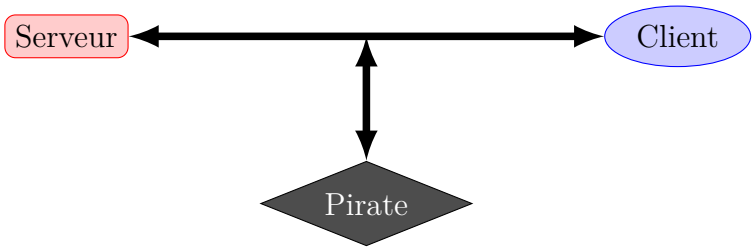


FIGURE 2.10 – Schéma d'une attaque man in the middle.

blèmes sont rencontrés dans d'autres domaines, le plus évident étant le *web*. Donc des solutions prêtes à être utilisées existent.

Pour la première faiblesse, on utilise ce qu'on appelle *l'authentification*, c'est le simple fait de demander un nom d'utilisateur et un mot de passe à chaque client qui veut se connecter.

Quant au problème de modification des données *après envoi*, on peut le régler par le chiffrage — faussement appelé cryptage dans le langage commun — si un utilisateur malintentionné se place entre le client et le serveur et modifie la trame — qui est chiffrée — son déchiffrage par le serveur conduit à une information *invalide*. Le serveur prend alors les mesures qui s'imposent, en ignorant tout simplement cette commande ou bien en déclenchant une alarme spéciale intrusion, etc.

Pour ces deux techniques, il existe des logiciels et des protocole qui on fait leurs preuves. On peut en citer quelques uns comme le SSL/TLS pour l'authentification et les algorithmes asymétriques comme RSA pour le chiffrage.

Conclusion

Les systèmes distribués en général et les systèmes SCADA en particulier font de plus en plus partie de notre quotidien sans même qu'on le remarque. Leur utilité n'est plus à prouver tant ils sont indispensables. Mais reste à résoudre les problèmes liés à leur sécurisation qui avancent à mesure que les technologies du net progressent.

Chapitre 3

Supports et protocoles de communication

Introduction

Dans un système SCADA réel, nous devons fournir deux types de réseaux de communication :

- un réseau local, pour permettre la communication — et donc, l'échange de données — entre les différents organes du sous-système et éventuellement entre les sous-systèmes eux-mêmes ;
- un réseau large, pour transmettre l'état du sous-système à la station de contrôle et recevoir ses commandes.

Nous exposerons dans ce chapitre les technologies les plus utilisés dans l'industrie pour chaque cas, ainsi qu'un bref aperçu de l'infrastructure matérielle nécessaire.

3.1 Les systèmes de communication

3.1.1 Définition

Ce sont des entités capables de transmettre l'information depuis un émetteur — homme, machine, etc. — à travers un support — fil, air, etc. — afin d'atteindre un récepteur donné.

Dans les systèmes de communications, le soucis principal étant le bon acheminement de l'information sans perte ni piratage en cours de transmission, plusieurs moyens ont été mis en œuvre pour palier ces problèmes, ces solutions sont généralement déployées en fonction des distances à parcourir.

3.1.2 Les communications locales

Ce sont des échanges de données sur courte distance — quelques centaines de mètres au plus.

Ce genre de système est utilisé en domotique pour la gestion et la supervision ainsi que la diffusion de la connexion à l'intérieur des bâtiments et autres types d'habitations.

3.1.3 Les communications distantes

Comme leur nom l'indique, elles se déploient sur des surfaces beaucoup plus étendues — ça peut aller jusqu'à l'échelle mondiales.

À titre d'exemple nous pouvons citer le réseau téléphonique fixe, le réseau GSM, internet, etc.

3.2 Les techniques de communication

Les technologies utilisées — niveau industriel ou particulier — pour communiquer en réseau sont diverses et variées. On peut les classer selon les distances qu'elles couvrent ou le support qu'elles utilisent. La figure 3.1 présente une classification possible de quelques techniques très connues.

3.2.1 RTC

Le Réseau Téléphonique Commuté — ou RTC — est le réseau du téléphone dans lequel un poste d'abonné est relié à un central téléphonique par une paire de fils alimentée en batterie centrale — la boucle locale. Les centraux sont eux-mêmes reliés entre eux par des liens offrant un débit de 2 Mb/s : ce sont les Blocs Primaires Numériques — BPN.

Dans le cas d'un réseau construit par un opérateur public, on parle parfois de Réseau Téléphonique Commuté Public — RTCP — ou PSTN, de l'anglais Public Switched Telephone Network.

3.2.2 xDSL

Digital Subscriber Line, DSL ou encore xDSL — que l'on peut traduire par « ligne numérique d'abonné » — renvoie à l'ensemble des techniques mises en place pour un transport numérique de l'information sur une ligne de raccordement téléphonique.

Il s'agit d'un mode d'exploitation étendu des lignes en cuivre existantes partant du principe suivant : une ligne téléphonique permet de diffuser des ondes comprises dans un certain spectre de fréquences ; or la voix n'utilise qu'une partie très restreinte de ce spectre, et il est même possible de la réduire encore sans gêner les communications. L'idée est donc de mettre à profit la partie non utilisée du spectre pour transporter des données.

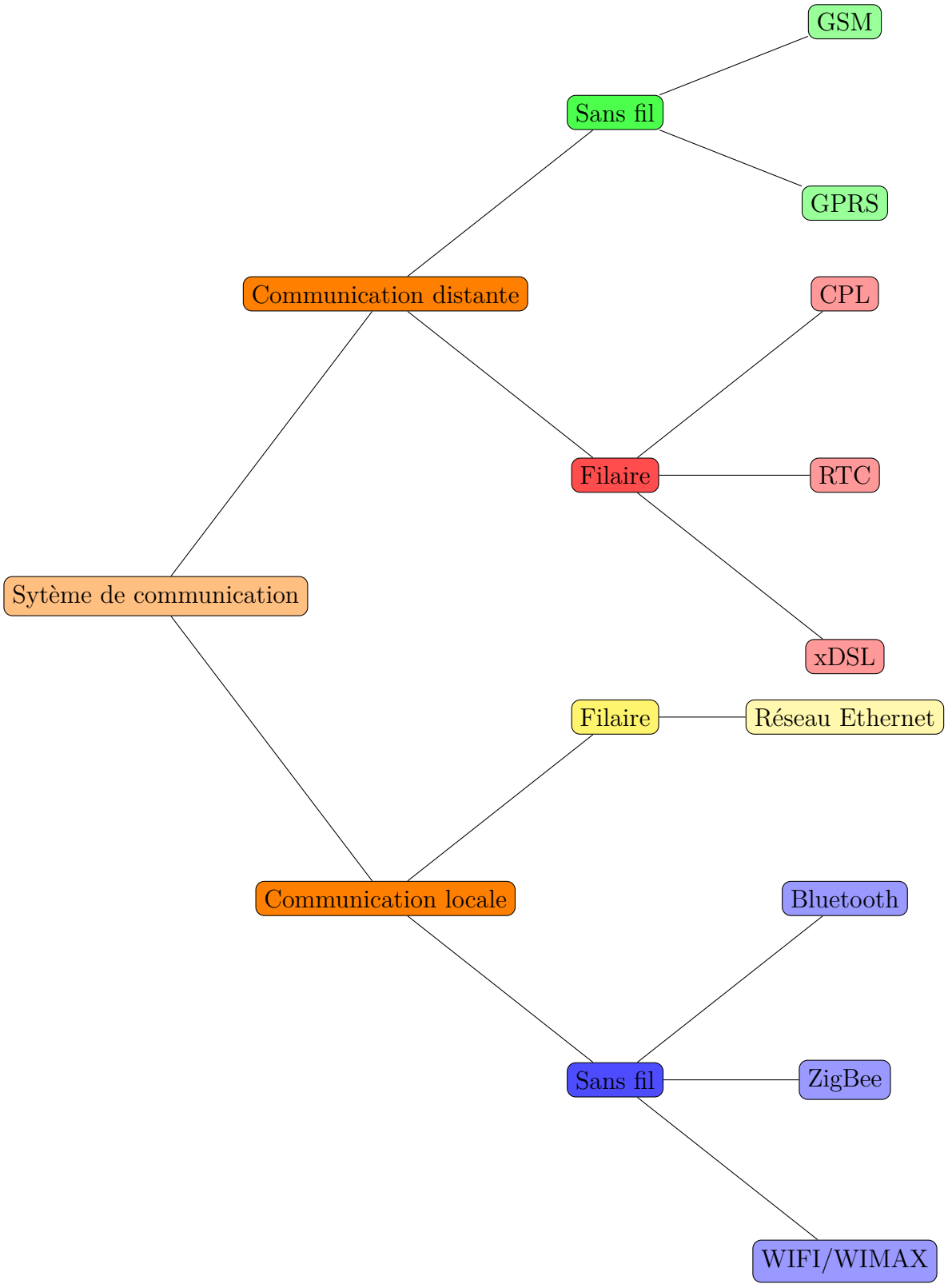


FIGURE 3.1 – Classement des techniques de communication.

3.2.3 CPL

CPL pour Courant Porteur en ligne — ou PLC pour PowerLine Communications — est une technologie qui permet de transmettre des données numériques sur un réseau électrique déjà existant.

Principe de fonctionnement

Le principe des CPL consiste à superposer au courant électrique de 50 ou 60 Hz un signal à plus haute fréquence et de faible énergie. Ce deuxième signal se propage sur l'installation électrique et peut être reçu et décodé à distance. Ainsi, le signal CPL est reçu par tout récepteur CPL qui se trouve sur le même réseau électrique.

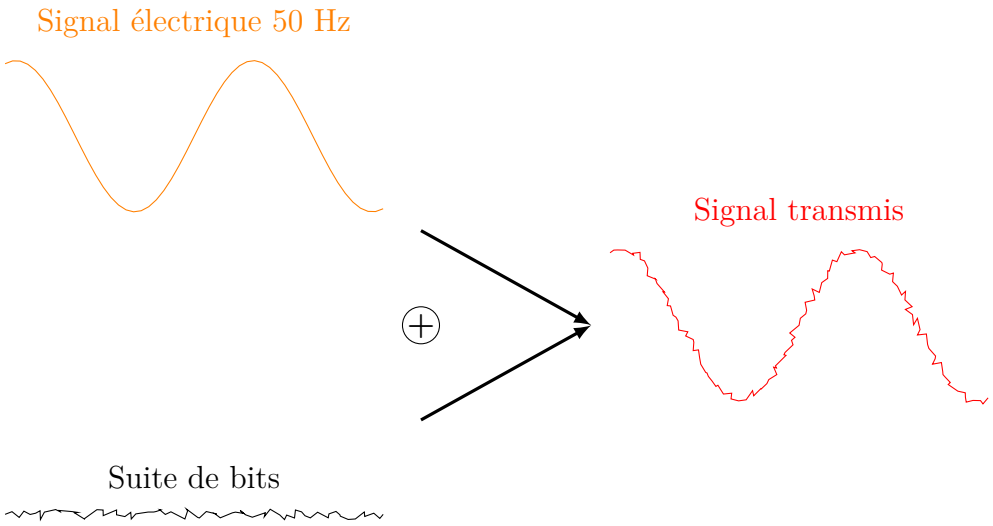


FIGURE 3.2 – La superposition du signal CPL et du signal électrique.

En ce qui concerne la transmission de données proprement dite le modèle OSI est pris en compte mais en fusionnant plus ou moins certaines couches entre elle selon le type d'application, cependant les deux premières couches à savoir la couche physique et la couche liaison de données restent principalement standard dans les transmissions par CPL.

Avantages :

- Support déjà établi — le réseau électrique ;
- une large diffusion — surtout dans les pays développés ;
- temps de réponse très rapide.

Inconvénient :

- Émission d'ondes perturbatrices pour les équipements avoisinants, pour la simple raison que les fils n'étaient pas initialement prévus pour transmettre des signaux à haute fréquence.

– sensibilité aux ondes électromagnétiques émis par les systèmes proches ce qui implique des problèmes de fiabilité.

3.2.4 Le GSM

Le Global System for Mobile Communications est une norme numérique de seconde génération pour la téléphonie mobile. Elle fut établie en 1982 par la Conférence européenne des administrations des postes et télécommunications.

Principe de fonctionnement

Le poste d'un abonné permet l'accès au réseau. Ce terminal est aussi appelé « station mobile » dans le cadre du GSM. Une station mobile est à la fois un poste téléphonique sans fil sophistiqué et un terminal de données qui transmet et reçoit des messages du réseau.

La « Base Transceiver Station » — BTS — est l'équipement terminal du réseau vers les « stations mobiles », Une BTS est un groupement d'émetteurs et de récepteurs fixes qui permet d'échanger des messages avec les stations mobiles présentes dans la cellule qu'elle contrôle. La BTS utilise des canaux radio différents selon le type d'information échanges, données utilisateur ou signalisation, et selon le sens de l'échange abonné vers réseau ou réseau vers abonné.

Dans le réseau, après la « Base Transceiver station », nous trouvons le contrôleur de station de base nommé « Base Station Controller » ou — BSC —. Il dialogue avec une ou plusieurs BTS. Cet équipement est a la fois un concentrateur du trafic issu des stations de base et une passerelle vers le sous-système réseau. L'équipement suivant, la « Base Station Controller » est le commutateur du réseau GSM, le « Mobile Switching Centre » — MSC. D'une part il connecte un réseau GSM avec le réseau téléphonique public RTCP/RNIS, d'autre part, il est l'interface des bases de données du réseau GSM avec le sous-système radio. Ces bases de données, outre qu'elles permettent de contrôler les droits d'accès des usagers au réseau, enregistrent la localisation des abonnés.

Les bases de données sont l'enregistreur des visiteurs « Visitor Location Register » — VLR —, le « Home Location Register » — HLR — du commutateur, et le « Authentication Centre » — AUC —. La base de données relative aux visiteurs du réseau VLR stocke des informations se rapportant à des abonnés qui sont en transit, Le HLR d'un abonné d'un réseau GSM est une banque de données. Elle renferme les originaux des informations relatives à cet abonné, notamment le profil de son abonnement. Quand cet abonné entre dans le réseau, ou quand il demande l'accès à un service, un équipement du réseau qui veut contrôler la validité des privilèges du demandeur interroge le HLR de l'abonné. Le HLR d'un abonné contient des informations permanentes. En revanche, un VLR enregistre les informations temporaires, dynamiques, relatives à une station mobile. Un canal de

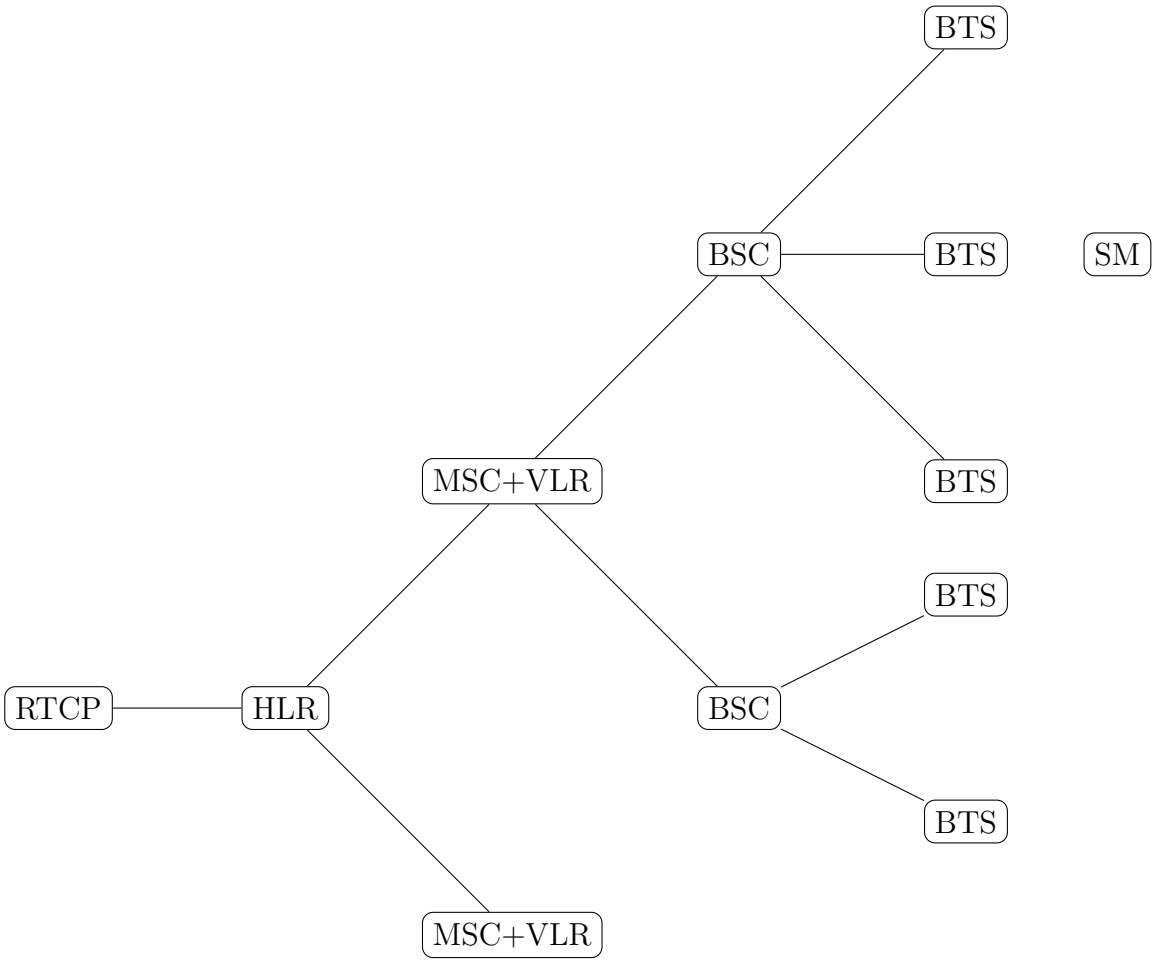


FIGURE 3.3 – Structure simplifiée d'un réseau GSM.

transmission offre un débit brut de 270 Kb/s mais le débit maximum utile pour un abonné est de 13 Kb/s.

Pour utiliser les GSM dans la supervision et la commande à distance, on se sert généralement de la fonction SMS. Ça nous permet de recevoir des notifications de l'état du système ainsi que d'envoyer des instructions.

3.2.5 Le GPRS

Le General Packet Radio Service ou GPRS est une norme pour la téléphonie mobile dérivée du GSM permettant un débit de données plus élevé. On le qualifie souvent de 2,5G.

Le GPRS est une extension du protocole GSM : il ajoute par rapport à ce dernier la transmission par paquets. Cette méthode est plus adaptée à la transmission des données. En effet, les ressources ne sont allouées que lorsque des données sont échangées, contrairement au mode « circuit » en GSM où un circuit est établi — et les ressources associées — pour toute la durée de la communication.

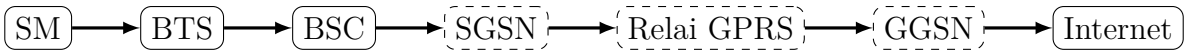
De ce fait, le GPRS est beaucoup plus approprié au besoin de la commande et de la supervision.

Principe de fonctionnement

Le GPRS permet de fournir une connectivité IP constamment disponible à une station mobile, mais les ressources radio sont allouées uniquement quand des données doivent être transférées, ce qui permet une économie de la ressource radio. Les utilisateurs ont donc un accès bon marché, et les opérateurs économisent la ressource radio. De plus, aucun délai de numérotation n'est nécessaire.

Avant le GPRS, l'accès à un réseau se faisait par commutation de circuits, c'est-à-dire que le canal radio était réservé en continu à la connexion — qu'il y ait des données à transmettre ou pas. La connexion suivait le chemin suivant :

Le GPRS introduit lui de nouveaux équipements. La connexion suit le cheminement suivant :



La connexion entre le SM et le BTS + BSC fait intervenir un protocole de couche 2 — MAC, Medium Access Control — et un protocole de couche 3 — RLC, Radio Link Control. Ces deux couches ont pour mission de gérer les procédures de connexion/déconnexion et de gérer le partage de la ressource radio entre plusieurs utilisateurs. RLC gère la segmentation, le réassemblage et la retransmission des trames erronées.

La connexion entre le BSC et le SGSN — Serving GPRS Support Node — a lieu avec le protocole NS — Network Service — en couche 2 et le protocole BSSGP — Base Station Subsystem GPRS Protocol — en couche 3.

La connexion entre le SGSN — Serving GPRS Support Node — et le GGSN — Gateway GPRS Support Node — utilise le protocole IP. Les connexions en couche 4 se font avec le protocole LLC — Logical Link Control — entre la MS et le SGSN, et avec l'UDP entre le SGSN et le GGSN.

Finalement, au-dessus des couches 4 se trouvent deux autres protocoles : SNDCP — Sub Network Dependent Converge Protocol — entre la MS et le SGSN, et GTP — GPRS Tunnelling Protocol — entre le SGSN et le GGSN.

3.2.6 les réseaux Ethernet

Le réseau local Ethernet date de la fin des années 70, il découle des études de DEC, Intel et Xerox, avant la normalisation. Ceci explique que les couches supérieures du modèle OSI

ne sont pas spécifiées. Tous les PC peuvent communiquer sur le câble réseau informatique simultanément.

Les réseaux Ethernet ont évolué au fil du temps en normes :

Ethernet, IEEE 802.3 10 Base 5 et 802.3 10 Base 2

10 Base 5 est la version d'origine d'Ethernet. Elle permet une vitesse de 10 Mb/s sur un câble coaxial de 500 mètres par segments.

Chaque équipement utilise une carte Ethernet qui effectue l'adaptation physique des données et gère l'algorithme CSMA/CD. Comme pour toutes connexions utilisant un câblage coaxial, les 2 extrémités sont terminées par un bouchon — une résistance de terminaison — qui atténue les réverbérations des données et réduit les collisions.

Ethernet, IEEE 802.3 10 Base 2

Cette version est plus connue. Elle utilise un câble coaxial fin — Thin Ethernet — avec une longueur maximum de 185 mètres pour 30 stations maximum. Chaque station est raccordée au coaxial via un T de type BNC, 50 centimètres minimum doivent séparer deux stations. De nouveau, on utilise une résistance de terminaison — bouchon — de 50 ohms.

Réseau Ethernet, IEEE 802.3 10 Base T

Une première version — 802.3 1 base 5 ou Starlan — utilise le câblage précablé pour le téléphone dans les immeubles. Elle se base sur une topologie en étoile via des hub avec une distance maximum de 250 mètres. La version suivante — Ethernet 802.3 10 base T — se base sur cette norme mais avec une vitesse supérieure — 10 Mb/s — sur une distance maximum de 100 mètres.

Ethernet 100 Base TX et 100 Base T4, Fast Ethernet

Sorti en 1992, la norme 100 base T — Fast Eternet — a un débit théorique est de 100 Mb/s — toujours 100 mètres maximum —. Elle utilise la même topologie.

On retrouve 2 normes de 100 Base T : le 100 Base T4 — obsolète — et le 100 Base TX. Le 100 Base TX utilise le même câblage que le 10 Base T, le 100 Base T4 utilisait les 4 paires — découpage de la bande passante en 4 pour l'émission comme pour la réception —il n'est donc pas Full Duplex, incompatible avec une communication bidirectionnelle simultanée. Le 100 T4 utilise du câble de catégorie 3, 4 ou 5.

3.2.7 Le bluetooth

Le Bluetooth est un standard de transmission de données sans fil développé en 1994 par Ericsson est normalisé par l'IEEE 802.15.1. Son débit théorique est de 1 Mb/s mais en pratique il atteint 720 Kb/s. Le bluetooth a une portée de 10 à 20 mètres et permet l'interconnexion de huit terminaux simultanément. Son point fort réside dans sa faible consommation d'énergie. On le voit apparaître de plus en plus dans de nombreux matériels, comme les téléphones mobiles et les PDA.

Principe de fonctionnement

Le standard Bluetooth, à la manière du WiFi utilise la technique FHSS — Frequency Hopping Spread Spectrum, en français étalement de spectre par saut de fréquence ou étalement de spectre par évasion de fréquence —, consistant à découper la bande de fréquence 2.402 – 2.480 GHz en 79 canaux — appelés hops ou sauts — d'une largeur de 1 MHz, puis de transmettre en utilisant une combinaison de canaux connue des stations de la cellule.

Ainsi, en changeant de canal jusqu'à 1600 fois par seconde, le standard Bluetooth permet d'éviter les interférences avec les signaux d'autres modules radio.

Principe de communication

Le standard Bluetooth est basé sur un mode de fonctionnement maître/esclave. Ainsi, on appelle « picoréseau » — en anglais piconet — le réseau formé par un périphérique et tous les périphériques présents dans son rayon de portée. Il peut coexister jusqu'à 10 picoréseaux dans une même zone de couverture. Un maître peut être connecté simultanément à un maximum de 7 périphériques esclaves actifs — 255 en mode parked —. En effet, les périphériques d'un picoréseau possèdent une adresse logique de 3 bits, ce qui permet un maximum de 8 appareils. Les appareils dits en mode parked sont synchronisés mais ne possèdent pas d'adresse physique dans le picoréseau.

En réalité, à un instant donné, le périphérique maître ne peut se connecter qu'à un seul esclave à la fois. Il commute donc très rapidement d'un esclave à un autre afin de donner l'illusion d'une connexion simultanée à l'ensemble des périphériques esclaves.

Le standard Bluetooth prévoit la possibilité de relier deux piconets entre eux afin de former un réseau élargi, appelé « réseau chaîné », grâce à certains périphériques faisant office de pont entre les deux piconets.

3.2.8 ZigBee

ZigBee est un protocole de haut niveau permettant la communication de petites radios, à consommation réduite, basée sur la norme IEEE 802.15.4 pour les réseaux à dimension

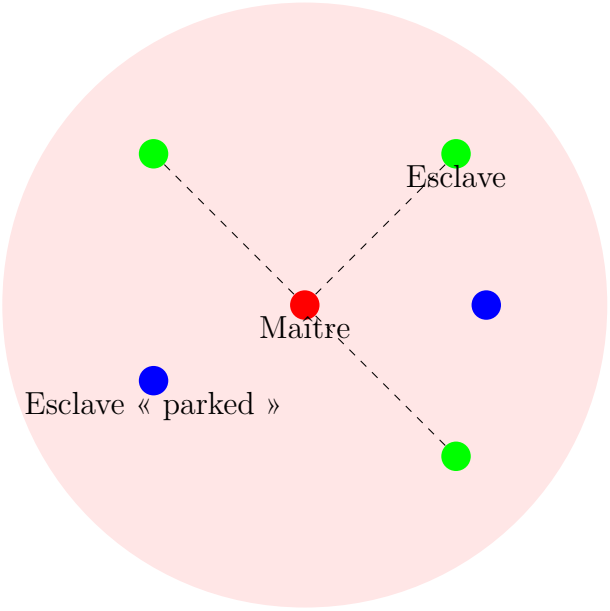


FIGURE 3.4 – Un piconet Bluetooth.

personnelle — Wireless Personal Area Networks.

La spécification initiale de ZigBee propose un protocole lent dont le rayon d'action est relativement faible, mais dont la fiabilité est assez élevée, le prix de revient faible et la consommation considérablement réduite.

On retrouve donc ce protocole dans des « environnements embarqués » où la consommation est un critère de sélection. Ainsi, la domotique et les nombreux capteurs qu'elle implémente apprécie particulièrement ce protocole en plein essor et dont la configuration du réseau maillée se fait automatiquement en fonction de l'ajout ou de la suppression de nœuds. On retrouve aussi ZigBee dans les contrôles industriels, les applications médicales, les détecteurs de fumée et d'intrusion.

Les nœuds sont conçus pour fonctionner plusieurs mois — jusqu'à dix ans pour les moins consommant — en autonomie complète grâce à une simple pile alcaline de 1,5 V.

3.2.9 WI-FI

Le Wi-Fi est une technologie qui permet de relier sans fil plusieurs appareils informatiques — ordinateur, routeur, décodeur Internet, etc. — au sein d'un réseau informatique.

Grâce au Wi-Fi, il est possible de créer des réseaux locaux sans fil à haut débit. Dans la pratique, le Wi-Fi permet de relier des ordinateurs portables, des machines de bureau, des assistants personnels — PDA, des objets communicants ou même des périphériques à une liaison haut débit — de 11 Mbit/s théoriques ou 6 Mbit/s réels en 802.11b — sur un rayon de plusieurs dizaines de mètres en intérieur — généralement entre une vingtaine et une cinquantaine de mètres —. Dans un environnement ouvert, la portée peut atteindre plusieurs centaines de mètres voire dans des conditions optimales plusieurs dizaines de

kilomètres — pour la variante WiMAX ou avec des antennes directionnelles.

Il existe bien évidemment d'autres technologies qu'on a pas listé, soit parce qu'elles sont obsolètes ou bien qu'elles ne sont pas assez utilisées dans l'industrie.

3.3 Protocoles industriels

3.3.1 Protocole ModBus

Le protocole MODBUS consiste en la définition de trames d'échange.

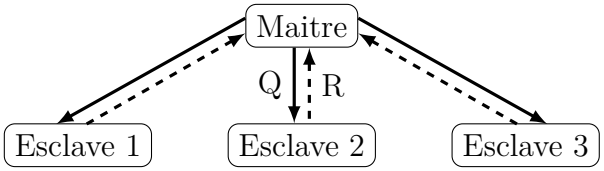


FIGURE 3.5 – Principe de communication ModBus.

Le maître envoie une demande et attend une réponse. Deux esclaves ne peuvent dialoguer ensemble. Le dialogue maître – esclave peut être schématisé sous une forme successive de liaisons point à point.

Adressage

Les abonnés du bus sont identifiés par des adresses attribuées par l'utilisateur. L'adresse de chaque abonné est indépendante de son emplacement physique. Les adresses vont de 1 à 64 et ne doivent pas obligatoirement être attribuées de manière séquentielle. Deux abonnés ne peuvent avoir la même adresse.

Echange maître vers un esclave

Le maître interroge un esclave de numéro unique sur le réseau et attend de la part de cet esclave une réponse.

Echange Maître vers tous les esclaves

Le maître diffuse un message à tous les esclaves présents sur le réseau, ceux-ci exécutent l'ordre du message sans émettre une réponse.

Trame d'échange question/réponse

La question : Elle contient un code fonction indiquant à l'esclave adressé quel type d'action est demandé. Les données contiennent des informations complémentaires dont

l'esclave a besoin pour exécuter cette fonction. Le champ octets de contrôle permet à l'esclave de s'assurer de l'intégralité du contenu de la question.

N° d'esclave	Code fonction	Information spécifique concernant la commande	Mot de contrôle
1 octet	1 octet	n octets	2 octets

FIGURE 3.6 – Trame maître → esclave

La réponse : Si une erreur apparaît, le code fonction est modifié pour indiquer que la réponse est une réponse d'erreur.

N° d'esclave	Code fonction	Données reçues	Mot de contrôle
1 octet	1 octet	n octets	2 octets

FIGURE 3.7 – Trame esclave → maître

Les données contiennent alors un code (code d'exception) permettant de connaître le type d'erreur.

Le champ de contrôle permet au maître de confirmer que le message est valide.

N° d'esclave	Code fonction	Code d'exception	Mot de contrôle
1 octet	1 octet	1 octets	2 octets

FIGURE 3.8 – Trame rapport d'erreur.

Format général d'une trame

Deux types de codage peuvent être utilisés pour communiquer sur un réseau Modbus. Tous les équipements présents sur le réseau doivent être configurés selon le même type.

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

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

FIGURE 3.9 – Trame type ASCII.

LRC : C'est la somme en hexadécimal modulo 256 du contenu de la trame hors délimiteurs, complémentée à 2 et transmise en ASCII.

Type RTU (Unité terminale distante) : chaque octet composant une trame est codé sur 2 caractères hexadécimaux (2 fois 4 bits).

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

FIGURE 3.10 – Trame type RTU.

La taille maximale des données est de 256 octets.

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

3.3.2 Ethernet

Ethernet est un protocole de réseau local à commutation de paquets. Bien qu'il implémente la couche physique — PHY — et la sous-couche Media Access Control — MAC — du modèle OSI, le protocole Ethernet est classé dans la couche de liaison, car les formats de trames que le standard définit sont normalisés et peuvent être encapsulés dans des protocoles autres que ses propres couches physiques MAC et PHY. Ces couches physiques font l'objet de normes séparées en fonction des débits, du support de transmission, de la longueur des liaisons et des conditions environnementales.

Ethernet a été standardisé sous le nom IEEE 802.3. C'est maintenant une norme internationale : ISO/IEC 8802-3. Depuis les années 1990, on utilise très fréquemment Ethernet sur paires torsadées ou câble coaxial pour la connexion des postes clients, et des versions sur fibre optique pour le cœur du réseau.

Dans un réseau Ethernet, le câble diffuse les données à toutes les machines connectées, de la même façon que les ondes radiofréquences parviennent à tous les récepteurs. Le nom Ethernet dérive de cette analogie : avant le XXe siècle on imaginait que les ondes se propageaient dans l'éther, milieu hypothétique censé baigner l'Univers. Quant au suffixe net, il s'agit de l'abréviation du mot network — réseau — en anglais.

Principe de fonctionnement

L'Ethernet est basé sur le principe de membres — pairs — sur le réseau, envoyant des messages dans ce qui était essentiellement un système radio, captif à l'intérieur d'un fil ou d'un canal commun, parfois appelé l'éther. Chaque pair est identifié par une clé globalement unique, appelée adresse MAC, pour s'assurer que tous les postes sur un réseau Ethernet aient des adresses distinctes.

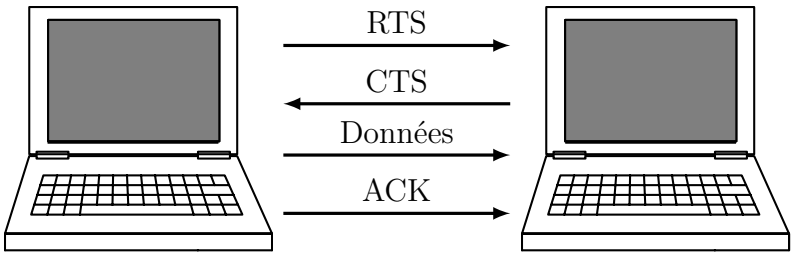


FIGURE 3.11 – Représentation schématique de la méthode d'accès CSMA/CA.

CSMA/CD

Une technologie connue sous le nom de Carrier Sense Multiple Access with Collision Detection — Écoute de porteuse avec accès multiples et détection de collision — ou CSMA/CD régit la façon dont les postes accèdent au média. Au départ développée durant les années 1960 pour ALOHAnet à Hawaii en utilisant la radio, la technologie est relativement simple comparée à Token Ring ou aux réseaux contrôlés par un maître. Lorsqu'un ordinateur veut envoyer de l'information, il obéit à l'algorithme suivant :

Explications

1. Si le média n'est pas utilisé, commencer la transmission, sinon aller à l'étape 4.
2. **Transmission de l'information** si une collision est détectée, continuer à transmettre jusqu'à ce que le temps minimal pour un paquet soit dépassé — pour s'assurer que tous les postes détectent la collision —, puis aller à l'étape 4.
3. **Fin d'une transmission réussie** indiquer la réussite au protocole du niveau supérieur et sortir du mode de transfert.
4. **Câble occupé** attendre jusqu'à ce que le fil soit inutilisé.
5. **Le câble est redevenu libre** attendre pendant un temps aléatoire, puis retourner à l'étape 1, sauf si le nombre maximal d'essais de transmission a été dépassé.
6. **Nombre maximal d'essais de transmission dépassé** annoncer l'échec au protocole de niveau supérieur et sortir du mode de transmission.

En pratique, ceci fonctionne comme une discussion ordinaire, où les gens utilisent tous un médium commun — l'air — pour parler à quelqu'un d'autre. Avant de parler, chaque personne attend poliment que plus personne ne parle. Si deux personnes commencent à parler en même temps, les deux s'arrêtent et attendent un court temps aléatoire. Il y a de bonnes chances que les deux personnes attendent un délai différent, évitant donc une autre collision. Des temps d'attente exponentiels sont utilisés lorsque plusieurs collisions surviennent à la suite.

Types de trames Ethernet et champ EtherType

Il y a quatre types de trame Ethernet :
- Ethernet originale version I — n'est plus utilisée —
- Ethernet Version 2 ou Ethernet II — appelée trame DIX, toujours utilisée —
- IEEE 802.x LLC
- IEEE 802.x LLC/SNAP

3.3.3 ControlNet

ControlNet est un système de communication industriel qui échange des données de façon déterminée et prévisible dans le temps. Les dispositifs utilisant ControlNet sont des entrées-sorties simples, comme des détecteurs/déclencheurs, aussi bien que des dispositifs de contrôle complexes comme des robots, des PLC, des contrôleurs de procédé, des interfaces opérateur, etc.

fonctionnement

À la différence des systèmes de communication usuels qui fonctionnent sur le modèle Source/Destination, ce réseau utilise un modèle Producteur-Consommateur. Le modèle Producteur/Consommateur permet l'échange d'information critique dans le temps entre un dispositif d'envoi — le Producteur — et plusieurs dispositifs de réception — les Consommateurs — sans avoir à générer de multiples transferts aux destinataires. Ceci est accompli en attachant un identificateur unique à chaque pièce d'information qui est produite sur le réseau. N'importe quel dispositif ayant besoin d'une pièce d'information spécifique, filtre l'information pour détecter l'identificateur approprié. Plusieurs dispositifs peuvent consommer la même pièce d'information produite par un dispositif producteur unique.

ControlNet fournit un haut degré d'efficacité en utilisant un mécanisme de « Token »implicite dans le temps. Ce mécanisme permet à tous les dispositifs d'avoir un accès égal au réseau pour transmettre leurs données, sans toutefois occasionner de latence due aux transferts des paquets « Token ». Le protocole utilise un mécanisme de cédule, avec une base de temps prédéfinie, qui assure aux dispositifs un accès déterminé et prévisible, et qui élimine les collisions sur le réseau. Ce mécanisme de cédule permet aux données critiques dans le temps, qui sont exigées sur une base périodique, répétable et prévisible, d'être produites avec une périodicité prédéterminée sans perte d'efficacité généralement associée aux requêtes multiples des autres types de réseaux. Lorsqu'un échange est cédulé périodiquement — ex : à tous les 10 milli-secondes —, ControlNet garantie ce temps de livraison — ou mieux —.

Le protocole soutient aussi un mécanisme complémentaire qui permet aux données qui ne sont pas critiques dans le temps ou qui sont transigées occasionnellement, d'uti-

liser le temps restant sur le réseau. Ces données non critiques sont transmises entre les productions de données critiques .

Token Ring

Token Ring, est un protocole de réseau local qui fonctionne sur les couches Physique et Liaison du modèle OSI. Il utilise une trame spéciale de trois octets, appelée jeton, qui circule dans une seule direction autour d'un anneau. Les trames Token Ring parcourent l'anneau dans un sens qui est toujours le même.

Le paradigme est celui du rond-point, qui se montre généralement capable d'écouler un débit plus grand qu'un carrefour, toutes choses égales par ailleurs. De plus le fait d'éviter le temps perdu en collisions. En contrepartie, on se créait des contraintes topologiques : l'Ethernet est concevable sur n'importe quel support, y compris en théorie par infrarouge sur un plafond blanc par contre le token-ring ne peut fonctionner que sur une boucle

Conclusion

Nous remarquons qu'en matière de communication ce n'est pas les techniques qui manquent. Mais il faut savoir choisir la plus appropriée au système selon l'environnement où elles vont être déployées et les contraintes du système. En plus, nous nous focalisons sur la notion de réseau local et réseau large qui joue un rôle primordial dans le choix de la technologie à utiliser.

Chapitre 4

Solutions technologiques retenues

Introduction

L'expansion fulgurante réalisée par l'informatique ces vingt dernières années a conduit à l'émergence d'un nombre très important de technologies, logiciels, langages et techniques de programmation. Le choix est laissé à l'utilisateur selon ses besoins, budget, aptitudes, etc. Dans le cadre de notre étude, nous aurons besoin de plusieurs éléments présentant une variété impressionnantes. Ceci nous oblige, dès à présent, à choisir. Pour cela, nous devons définir des critères de sélection que ces éléments doivent satisfaire.

4.1 Liberté, efficacité, simplicité ! !

Les technologies choisies doivent satisfaire ces trois contraintes, à savoir :

Liberté : cette notion est relative à la licence des logiciels utilisés, qui doit être *libre*, c.-à.d. qu'elle permet l'accès, la modification, la copie et la redistribution du dit logiciel sous forme binaire ou de sources.

Efficacité : le choix des logiciels libre — donc, l'accès aux sources et la possibilité de les modifier — ne doit pas nous mener à la réutilisation systématique de codes source non adaptés. Le choix de l'écriture ou de la réutilisation de codes sera dicté par l'efficacité, ce qui veut dire que si un logiciel correspond *exactement* à nos besoin, nous le réutiliserons, sinon on écrira le notre depuis le début.

Simplicité : à vrai dire, ce n'est pas une contrainte mais une ligne de conduite qu'on devrait adopter au sein de chaque projet informatique ou autre. Cela peut se résumer par la philosophie *KISS — Keep it simple, stupid.*

4.2 Le modèle internet

Nous avons remarqué que la plupart des études réalisées dans le domaine des réseaux possèdent une partie présentant le modèle OSI — qui est un modèle de communications proposé par l'ISO — pour expliquer le fonctionnement des couches internet. Or, cette approche est incorrecte puisqu'internet s'appuie plutôt sur le modèle TCP/IP — ou plus précisément, le modèle internet — et tenter de faire correspondre les deux est une utopie. Déjà, le modèle internet ne comporte que quatre couches alors que le modèle OSI en compte sept. On serait, alors, tenter de faire l'analogie entre le modèle TCP/IP et un sous ensemble du modèle OSI, mais le fait est que le modèle OSI n'offre pas une richesse suffisante au niveau des couches basses pour représenter la réalité. Il serait donc plus judicieux de présenter le modèle internet, ce qu'on va faire dans la suite.

4.2.1 Définition

La suite de protocoles internet est l'ensemble des protocoles de communication utilisés dans internet et les réseaux similaires. Elle est communément appelée modèle TCP/IP, d'après ses deux protocoles les plus importants TCP — Transmission Control Protocol — et IP — Internet Protocol — qui sont les deux premiers protocoles à être définis dans ce standard. C'est la synthèse de plusieurs années de recherche qui a commencé dans les années soixante.

On peut le voir comme une suite de couches. Chaque couche résolvant un certain nombre de problèmes et fournissant des services à la couche supérieure. Les couches supérieures sont plus proches de l'utilisateur final. Elles manipulent des données plus abstraites et confient la conversion de données sous forme physiquement transmissible au couches inférieures.

4.2.2 Les couches

Ce modèle se compose de quatre couches, en l'occurrence : la couche application, la couche transport, la couche réseau et la couche de liaison de données.

La couche de liaison de données : elle spécifie comment les paquets sont transportés sur la couche physique, et en particulier le tramage — les séquences de bits particulières qui marquent le début et la fin des paquets. Les en-têtes des trames Ethernet, par exemple, contiennent des champs qui indiquent à quelle machine du réseau un paquet est destiné.

Exemples de protocoles de la couche de liaison de données : Ethernet, Wireless Ethernet, SLIP, Token Ring et ATM.

La couche réseau : cette couche résout le problème de l'acheminement de données depuis un réseau source vers un réseau destinataire. Ceci implique généralement le

routage des paquets à travers internet, IP assure l'acheminement des paquets depuis une source vers une destination, et supporte aussi d'autres protocoles, comme ICMP — utilisé pour transférer des messages de diagnostic liés aux transmissions IP — et IGMP — utilisé pour gérer les données multicast. ICMP et IGMP sont situés au-dessus d'IP, mais assurent des fonctions de la couche réseau, ce qui illustre l'incompatibilité entre les modèles Internet et OSI.

Exemples de protocoles de la couche réseau : IP.

La couche transport : Les protocoles de la couche de transport peuvent résoudre des problèmes comme la fiabilité des échanges et assurer que les données arrivent dans l'ordre correct. Dans la suite de protocoles TCP/IP, les protocoles de transport déterminent aussi à quelle application chaque paquet de données doit être délivré.

TCP est un protocole de transport « fiable », orienté connexion, qui fournit un flux d'octets fiable assurant l'arrivée des données sans altérations et dans l'ordre, avec retransmission en cas de perte, et élimination des données dupliquées. TCP essaie de délivrer toutes les données correctement et en séquence — c'est son but et son principal avantage sur UDP, même si ça peut être un désavantage pour des applications de transfert ou de routage de flux en temps-réel, avec des taux de perte élevées au niveau de la couche réseau.

Exemples de protocoles de la couche transport : TCP, UDP.

La couche application : c'est la couche de niveau le plus haut, utilisée par la plupart des applications pour la communication réseau.

Comme la suite de protocoles internet ne définis pas de couche supplémentaire entre la couche transport et la couche application, cette dernière se doit d'inclure un protocole qui agirait comme les couches présentation et session du modèle OSI. Ceci est souvent implémenter par des librairies.

Exemples de protocoles de la couche application : HTTP, FTP, SMTP, DNS, etc.

4.2.3 L'encapsulation

Le modèle TCP/IP repose sur le principe de l'encapsulation pour fournir l'abstraction nécessaire des protocoles et services. Cela ce traduit par le fait qu'une application voulant communiquer à travers le réseau se doit d'utiliser un ensemble de protocoles pour envoyer ses données aux couches inférieurs. Ces même données seront encapsuler à chaque niveau. La figure 4.1 représente une vulgarisation de ce concept.

4.2.4 Notre choix

En réalité, c'est plus une obligation imposée par la nature de notre projet qu'un véritable choix de notre part.

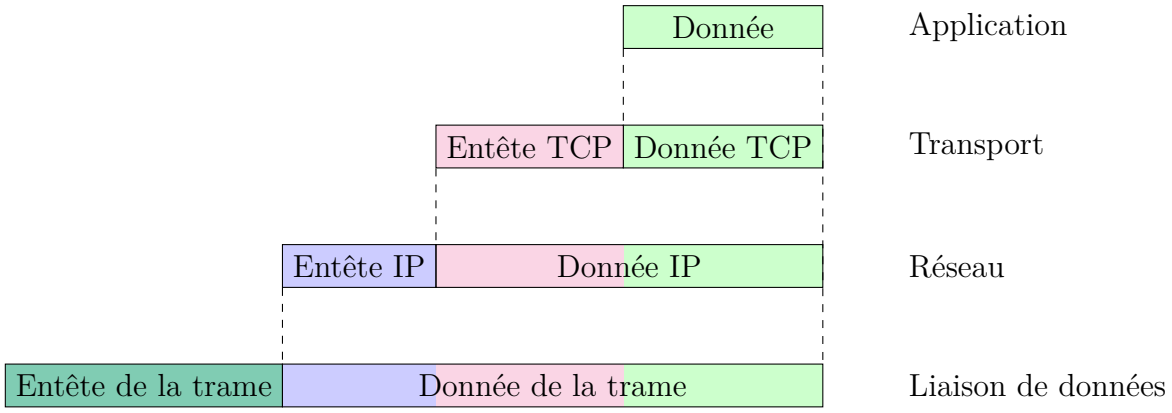


FIGURE 4.1 – Encapsulation des données à travers la pile des protocoles.

Étant donné qu'on veut réaliser une application qui *utilise* internet — en opposition à une application qui fournis des services de bas niveau. On se doit de coder au niveau de la couche application. Cela signifie qu'on s'appuiera sur la couche inférieure — la couche transport.

4.3 L'architecture réseau

Il existe plusieurs façons de mettre deux ordinateurs en réseau. Cela dépend essentiellement du rôle que chaque organe a à jouer.

Les principales architectures utilisées de nos jours sont l'architecture client/serveur et l'architecture poste à poste — peer to peer.

4.3.1 L'architecture client/serveur

L'architecture client/serveur désigne un mode de communication entre plusieurs ordinateurs d'un réseau qui distingue un ou plusieurs clients du serveur : chaque client peut envoyer des requêtes à un serveur.

Le serveur est initialement passif, il est en attente d'une requête — provenant du client — qu'il traite puis renvoi une réponse.

Le client est le maitre, il envoi des requête et attend la réponse du serveur.

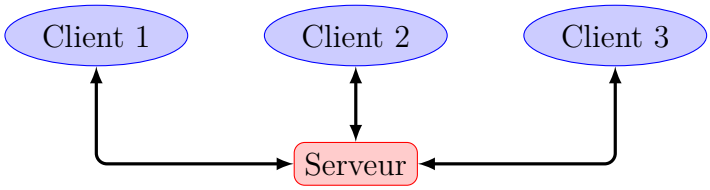


FIGURE 4.2 – L'architecture client/serveur.

Avantages

- Une administration au niveau serveur, les clients ayant peu d'importance dans ce modèle, ils ont moins besoin d'être administrés.

Inconvénients

- Si le serveur n'est plus disponible, tout le système est hors service.

4.3.2 L'architecture poste à poste

C'est un modèle proche du modèle client/serveur mais où tous les composants du réseau jouent au même temps le rôle de client et de serveur.

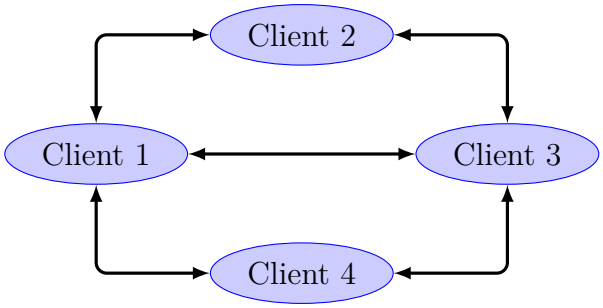


FIGURE 4.3 – L'architecture poste à poste.

On note — d'après la figure 4.3 que les clients ne sont pas nécessairement *tous* interconnectés.

Avantages

- Même si un poste n'est pas joignable le réseau continue de fonctionner.

Inconvénients

- Difficulté de maintenance due au nombre important de postes à contrôler.

4.3.3 Notre choix

Dans le cas présent, le choix est vite fait puisque l'architecture poste à poste n'aura aucun sens. Elle a été présenté juste pour savoir qu'il existe d'autres configurations possibles dans l'absolu.

On utilisera donc l'architecture client/serveur, un client *distant* envoi des requêtes — qui sont en faite des commandes — et le serveur s'occupe de les transmettre au système. En réponse le serveur envoie l'état du système après modification.

4.4 Protocole de communication

Les différentes entités composant un réseau informatique doivent pouvoir communiquer entre elles, pour cela elles doivent « parler la même langue », en termes plus techniques on dira que les logiciels se trouvant sur les différents nœuds du réseau utilisent le même protocole de communication.

4.4.1 Définition

Un protocole de communication est, dans un cadre informatique, l'ensemble des règles syntaxiques et sémantiques qui donnent un sens précis aux suites de bits échangés entre deux stations connectées. Il peut aussi s'occuper de l'authentification, la détection et la correction d'erreurs de transmission.

4.4.2 Notre choix

Notre choix en ce qui concerne le protocole de communication à utiliser s'est porté sur le HTTP — HyperText Transport Protocol — pour ça simplicité d'implémentation et facilité d'utilisation. En plus, les technologies du coté client — exposées ultérieurement — impliquent son utilisation.

4.4.3 Le protocole HTTP

HTTP est probablement le protocole de communication le plus utilisé sur le réseau internet. La fonction de base de ce protocole est le transfert de fichier de type texte depuis un serveur web vers un client — un navigateur web, un robot d'indexation, un aspirateur de sites web. Son adoption est due à ça simplicité puisqu'une requête HTTP est une chaine de caractères contenant l'adresse du fichier qu'on veut récupérer. En retour le serveur envoie une réponse HTTP contenant le fichier demandé.

La requête HTTP

La requête HTTP est créée à partir de l'adresse HTTP entrée dans le navigateur web, appelée aussi URL — Unified Ressource Locator — et est constituée de :

l'adresse du serveur : un serveur n'est ni plus ni moins qu'un ordinateur connecté à internet, il dispose donc d'une adresse IP. En pratique, on utilise un nom de domaine, il s'agit d'une chaine de caractères plus facile à mémoriser.

le numéro de port : c'est un numéro associé à un service permettant au serveur de savoir quel type de ressource est demandé. Le port associé par défaut à HTTP est le port n° 80, mais on peut évidemment le modifier.

chemin d'accès au fichier : c'est tout simplement l'emplacement du fichier dans la mémoire du serveur.

Exemple de requête HTTP :

– **GET** http ://www.debian.org :80/index.html

La réponse HTTP

Le serveur web identifié par l'adresse IP va retourner au client le fichier demandé sous forme d'une réponse HTTP. Si la requête contient des variables, celles-ci sont récupérées par le serveur. On peut alors imaginer un programme annexe qui traiterait ces informations et, suivant leur contenu, agirait sur notre système. Il existe même un standard industriel pour ce genre de communication qui est le CGI – Common Gateway Interface.

4.5 L'interface client

Pour que le client puisse envoyer ses commandes au serveur — qui, à son tour, les enverra au système, il doit disposer d'une interface de commande.

Il existe deux orientations possibles pour réaliser une interface de commande :

– utilisation d'un navigateur web ;

– création de l'interface par programmation.

Un navigateur web est déjà présent sur presque la totalité des stations de travail. En plus, cela nous évitera le travail fastidieux qu'est l'écriture d'une interface graphique. Seulement, dès que le projet prendra de l'envergure il ne sera plus possible d'assurer l'ergonomie avec un simple navigateur web, et on serait alors contraint de passer à une interface personnalisée.

L'écriture d'une interface est, certes, un travail supplémentaire, mais avec les bibliothèques modernes, il est de plus en plus facile d'arriver à nos fins. En plus, cela nous permettra de réaliser une interface plus claire puisque les différents éléments de notre système y seront représentés par des diagrammes plus parlant à l'opérateur humain.

4.6 Le langage de programmation

Le choix du langage de programmation dans un projet informatique est une étape délicate et importante, elle repose sur plusieurs considérations techniques et pratiques. Il vaut mieux se donner le temps de choisir le langage de programmation le plus approprié aux tâches qu'on veut réaliser. Mais pour cela, il faudrait connaitre ces langages, leurs caractéristiques, avantages et inconvénients.

4.6.1 La classification des langages de programmation

On peut classer les langages de programmation selon plusieurs critères, que voila :

Compilé ou interprété ?

Ceci est en relation avec la manière dont le code source écrit dans un certain langage de programmation est transformé en langage machine. Il existe deux grandes familles, les langages de programmation compilés, et les langages interprétés. Pour la première famille, le code source *entier* est soumis à un *compilateur* qui se charge de le convertir en fichier exécutable — donc, en langage machine. Tandis que dans le deuxième cas, *l'interpréteur* « lis », convertit en code machine et exécute le code source instruction par instruction. Évidemment chaque famille a ces avantages et inconvénients.

Langages compilés

Avantages
- Autonomie : une fois le programme compilé, nous n'avons plus besoin du compilateur.
- Rapidité d'exécution : puisque le programme est sous forme de code machine.
- Optimisation logicielle — c.-à.d par le compilateur : puisque le code source est « vu » comme un *tout* par le compilateur.
- Pouvoir cacher le code source — Ce qui peut être considéré comme un inconvénient.

inconvénients
- Chaque petite modification apportée au programme implique une recompilation pour pouvoir être prise en compte.

Exemples : C, C++, Fortran, Pascal, etc.

Langages interprétés

Avantages
- Facilité d'apprentissage.
- Facilité de debugage.

inconvénients
- Lenteur : puisque le code est à chaque exécution réinterprété, puis exécuté.
- Difficulté d'optimisation : puisque chaque instruction est interprétée et exécutée indépendamment du reste du code.

Exemples : Python, Ruby, Perl, etc.

En réalité il existe d'autres types qu'on a délibérément omis de mentionner dans un soucis de brièveté.

Les langages hybride que sont le java et le C# ont de plus en plus de succès. Ils conjuguent les avantages des deux techniques, puisqu'ils sont dans un premier temps compilé en pseudo-code machine — machine virtuelle — puis interprétés instruction par instruction par cette machine virtuelle.

Le paradigme de programmation

C'est le niveau d'abstraction mis à notre disposition par le langage de programmation, pour nous permettre de résoudre les problèmes auxquels nous sommes confrontés.

Langages impératifs C'est la méthode la plus naturelle qui vient à l'esprit quand on pense à la programmation. Le problème est résolu par une succession d'instructions basiques — affectation, branchement, boucles, etc.

Exemples : C, C++, Pascal, Fortran, etc.

Langages orientés objet C'est un nouveau — relativement — paradigme apparu dans les années 80. Dans ce type de langages, la solution est écrite directement dans l'espace *problème*, puisque chaque élément de l'espace problème a son pendant dans l'espace solution qui un *objet* possédant ses caractéristiques et ses actions — c'est là, le véritable plus de cette technique.

Exemples : C++, Java, C#, Python, Ruby, etc.

Là aussi, il existe d'autre paradigmes.

Nous remarquons aussi que quelques langages de programmation apparaissent dans la liste des exemples des deux types, c'est le cas du C++. On les appelle des langages *multiparadigme*.

Il y va sans dire qu'il existe d'autres types de classification, mais les précédents plus les contraintes que nous nous sommes imposées pour les choix des technologies utilisées dans ce projet suffiront à dégager un seul langage.

4.6.2 Notre choix

Parmi les contraintes que nous nous sommes imposées, il y a l'efficacité dont la rapidité fait partie, nous préfèrerons donc, un langage compilé — bien que la différence de vitesse se fait de moins en moins ressentir, et qu'à long terme cet argument ne sera plus valable.

Ce projet n'est, certes, qu'un projet de fin d'étude, mais sa finalité est industrielle, ce qui signifie que tôt ou tard — dans un développement futur — les choses seront amenées à ce compliquer. Nous prenons donc les devants en choisissant un langage de programmation orienté objet, qui nous assurera une modularité, une facilité d'implémentation et la possibilité de réutilisation de code. En plus, cela s'accorde avec la contrainte de simplicité.

De ce qui précède, il nous reste que peu de langages : C++, C#, Java — Pour ce dernier, rien n'est sûr, puisque nous pouvons l'éliminer au vue de sa lourdeur. C'est là qu'entre en jeu la *liberté*. Le C#, et le Java, bien qu'innovants, sont des langages de programmation produits par des sociétés commerciales — Microsoft, pour le premier et Oracle pour le second — ce qui n'apporte aucune garantie quant à leur développement futur. Tandis que le C++, est libre de tous droits et standardisé par un organisme international, en l'occurrence l'ISO.

Par conséquent, notre choix final s'est porté sur le C++, un langage multiparadigme, fortement typé, et compilé. C'est l'héritier du C, ce qui veut dire qu'il est bien adapté à la programmation système et réseau. C'est le meilleur compromis que nous puissions faire entre langage de bas niveau et simplicité d'implémentation.

Conclusion

Les technologies à utiliser étant définies, on peut maintenant entrer dans le vif du sujet en concevant une application capable de superviser et de contrôler un système à distance.

Chapitre 5

Implémentation logicielle

Introduction

Après avoir vu les différents aspects d'un système de commande et de supervision à distance et exposé ces différents composants, il est temps de matérialiser les connaissances acquises par une réalisation pratique pour comprendre le fonctionnement et combler les éventuelles lacunes qui apparaitront.

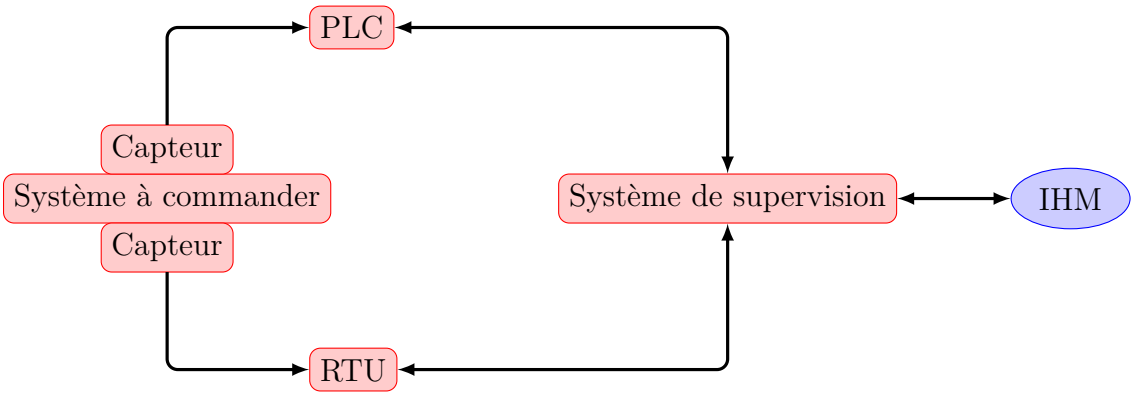


FIGURE 5.1 – Représentation schématique d'un système SCADA.

Pour rappelle, la figure 5.1 représente l'agencement des différents composants d'un système SCADA.

On a pas représenté les actionneurs pour des raisons de clarté du schéma.

5.1 Le système

Pour réaliser un système de commande et de supervision à distance, il faut — évidemment — choisir un système.

Dans ce chapitre, on a délibérément choisi un système simpliste qui n'a pas vraiment d'intérêt pratique — tout du moins son intérêt est minime — mais son but est plus

pédagogique.

On va commander trois LEDs — ou lampes — à l'allumage — et à l'extinction — en utilisant un navigateur web.

5.2 Réel ou simulé ?

Comme précisé dans la section précédente le système est plus pédagogique que pratique, et son intérêt réside dans le fait d'expliquer comment réaliser une connexion entre deux postes, puis envoyer de l'information en exploitant un certain protocole.

Donc, la majeure partie du système sera simulée comme le présente la figure 5.2.

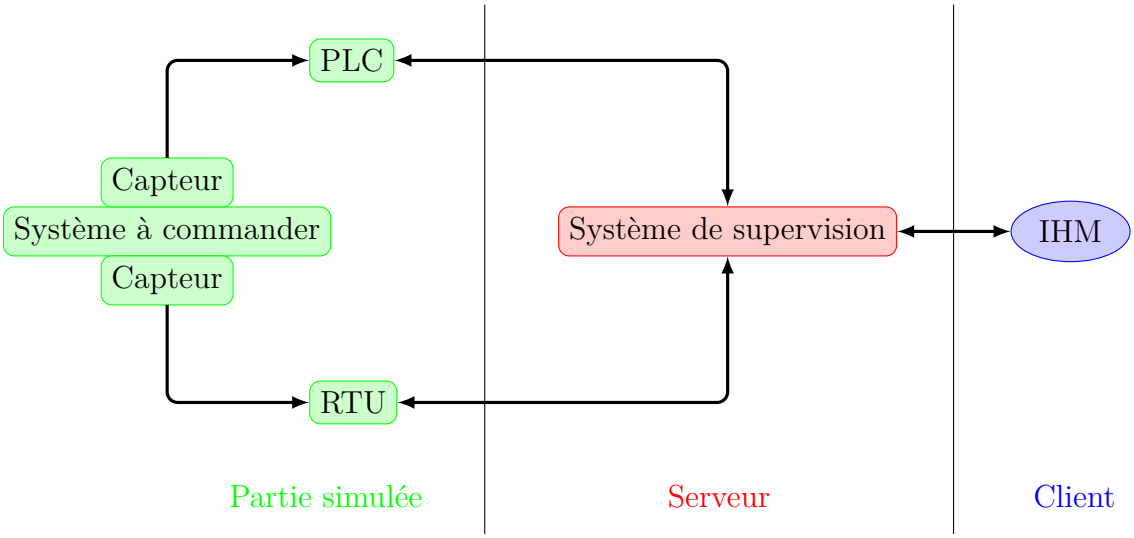


FIGURE 5.2 – La partie simulée du système.

En plus, cela aidera le lecteur puisque il n'aura pas à se préoccuper des interface d'entrées/sorties, etc.

5.3 Fonctionnement du système

Le système est composé de deux parties : la partie client et la partie serveur — puisque le système est simulé par le serveur, sinon on aurait trois parties.

5.3.1 Coté serveur

Tous le travail de programmation se trouve de ce coté. Le fonctionnement est résumé par l'organigramme de la figure 5.3.

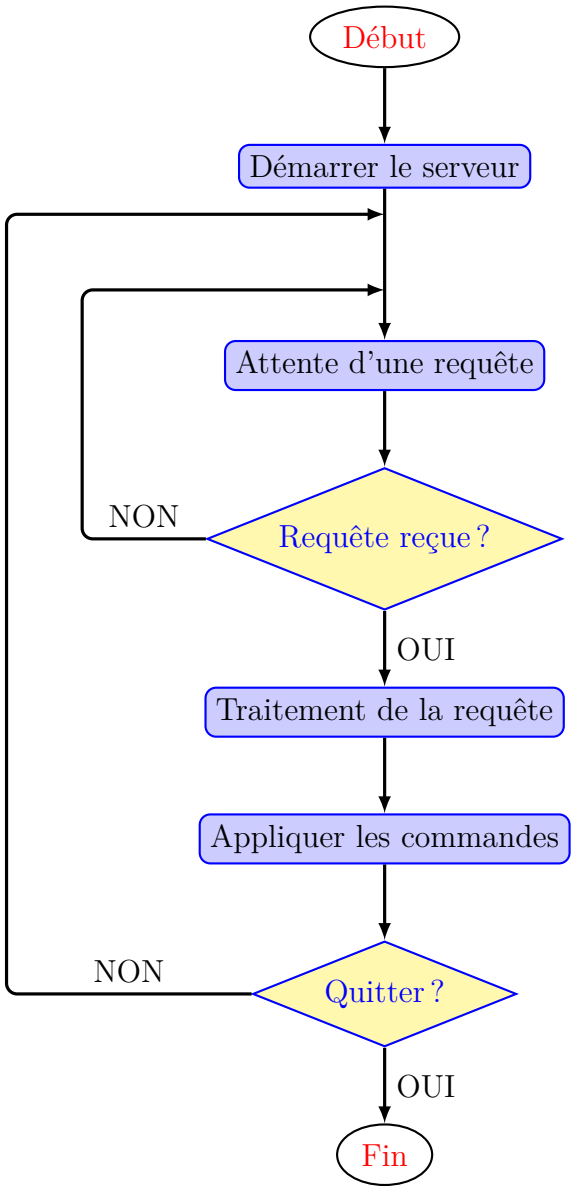


FIGURE 5.3 – Les actions du serveur.

Démarrer le serveur et attendre une requête

C'est le plus gros morceau du travail, réaliser un serveur auquel on peut se connecter et qui répond aux requêtes envoyées dans un certain protocole.

Dès qu'on veut exploiter les réseaux en programmation, on doit passer par les *socket*. Il s'agit d'une interface logicielle avec les services du système d'exploitation, grâce à laquelle un développeur exploitera facilement et de manière uniforme les services d'un protocole réseau.

Il lui sera ainsi par exemple aisé d'établir une session TCP, puis de recevoir et d'expédier des données grâce à elles. Cela simplifie sa tâche car cette couche logicielle, de laquelle il requiert des services en appelant des fonctions, masque le nécessaire travail de gestion du réseau, pris en charge par le système.

On a, donc, entrepris la réalisation d'un wrap — enveloppe — des facilités du système d'exploitation qui consiste en un objet *Socket* qui sera plus en adéquation avec le modèle orienté objet du C++. La figure 5.4 représente une schématisation en UML simplifié de cet objet.

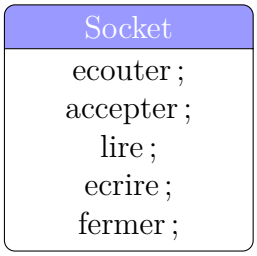


FIGURE 5.4 – UML simplifié d'un objet Socket.

Pour utiliser une socket, on doit renseigner deux champs : le protocole de communication et le protocole de transport. Le premier se fait en choisissant le nom du protocole — http, ftp, imap, etc. ou en précisant le numéro de port associé à ce protocole — http : 80, ftp : 21, imap : 143, etc. Quant au second il y a deux choix possible : une socket connectée — TCP — ou une socket déconnectée — UDP.

Traitement de la requête

Le traitement de la requête, dans notre cas, consiste en l'envoi du fichier demandé par le client par le biais de la requête HTTP envoyée par son navigateur web s'il existe ou le code d'erreur approprié le cas échéant.

En plus du nom du fichier, la requête peut contenir des variables représentant la commande désirée par le client. C'est un ensemble de pairs de valeurs — variable=valeur — envoyées dans l'URL dans le cas d'une requête *GET*.

Le fait d'extraire les variables à partir de l'URL s'appelle *parser*.

Appliquer les commandes

Après avoir récupérer les variables de commande, il suffit d'afficher la phrase « LEDx allumée » si la variable LEDx=ON, sinon on affichera « LEDx éteinte », avec x allant de 1 à 3.

5.3.2 Coté client

Il n'y a rien a programmer du coté client — et c'est là tout l'intérêt d'utiliser un navigateur web au lieu d'une interface graphique dédiée — Il lui suffit de suivre les étapes de l'organigramme de la figure 5.5.

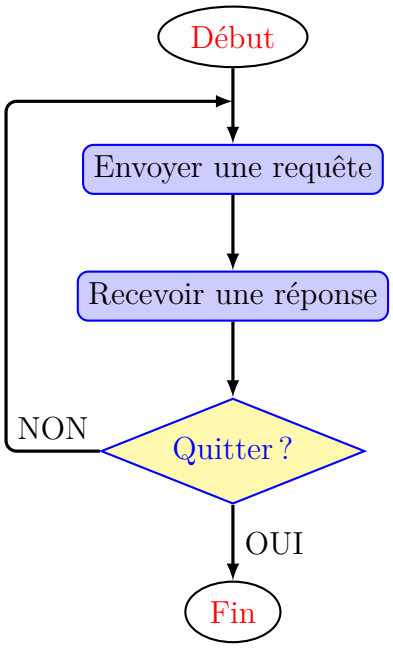


FIGURE 5.5 – Les actions du client.

Envoyer une requête

Le client peut contacter le serveur en entrant son adresse IP suivi du numéro de port si ce n'est pas celui par défaut — qui est 80.

Exemple : http ://196.32.0.1 :3425

Ou bien — si le serveur à un nom de domaine associé — utiliser le nom du serveur qui est plus facile à mémoriser pour un humain.

Il existe une adresse IP *spéciale* appelée adresse *loopback*. Son intérêt réside dans la facilité des testes qu'elle offre. On lance le serveur et le client sur le même poste et on se connecte au serveur à partir du navigateur web grâce à l'adresse loopback.

Cette adresse est : 127.0.0.1 ou localhost.

5.4 Essai de l'application

La figure 5.6 représente un essai.

Le mode d'emploi est simple et intuitif. Si on veut allumer une LED il suffit de cocher la case correspondante. Une fois notre choix terminé, on clique sur le bouton « send » pour valider la commande.

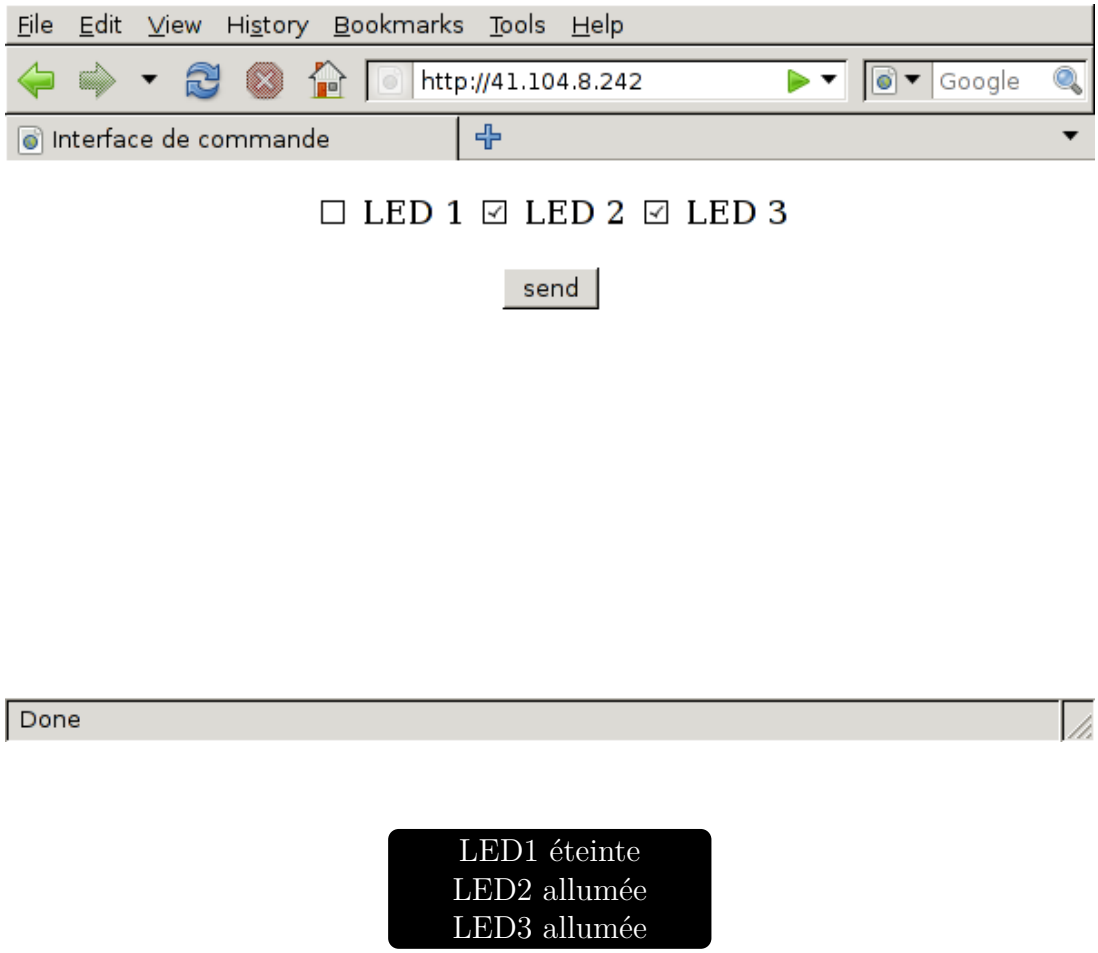


FIGURE 5.6 – Captures d'écran.

5.5 Amélioration

Au milieu de notre projet nous avons réalisé que la simulation précédente n'était pas très « parlante », on aurait préféré un affichage graphique plutôt que de simple phrases. Qu'à cela ne tienne, il nous suffit de trouver une bibliothèques C++ permettant d'afficher des images. Il en existe beaucoup, mais celle qui a retenu notre attention est la sfml.

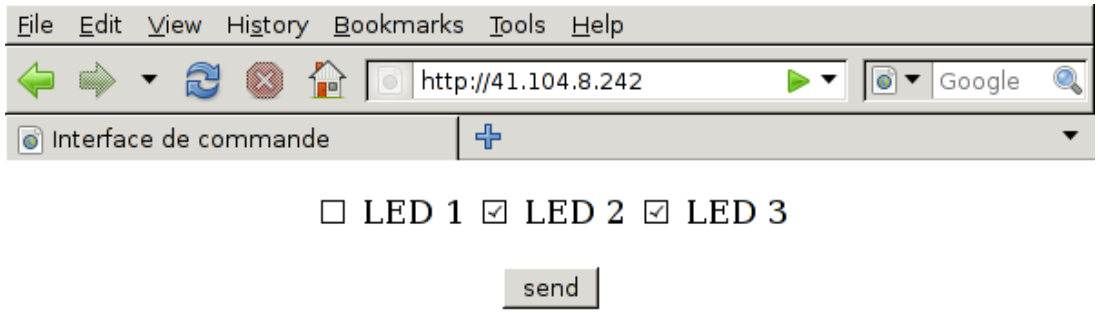
5.5.1 La bibliothèque sfml

sfml est un bibliothèque multimédia fortement orienté objet, elle permet de gérer les images, les sons, les contexte OpenGL, etc. C'est une bibliothèque libre et multi-plateforme.

5.5.2 Nouvelle version

On comprend ici l'intérêt d'avoir un code modulaire, puisque pour effectuer les modifications nécessaire, il nous a suffit de changer la ligne du code « afficher la phrase... » par « afficher l'image... » et ça sera aussi le cas si on voulais faire une réalisation réelle.

La figure 5.7 présente un capture d'écran du même cas précédent, mais avec un affichage graphique coté serveur.



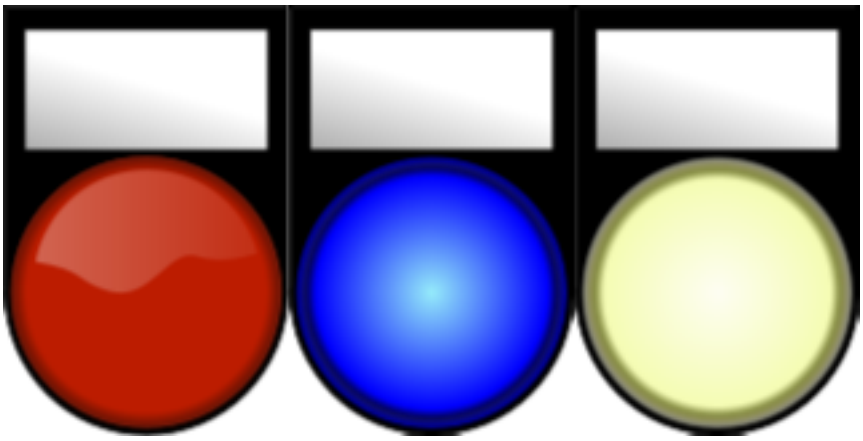


FIGURE 5.7 – Captures d'écran avec sfml.

Conclusion

Même si l'application réalisée est très simple, elle permet d'avoir une idée sur le fonctionnement des réseaux en programmation. En plus tous les systèmes aussi complexes soient-ils suivront la même méthodologie.

Chapitre 6

système de télé-relève

Introduction

Dans ce chapitre nous proposerons une implémentation pratique d'un système SCADA à même de résoudre le problème exposé dans la section suivante. Ceci, en incluant au système de distribution électrique un système de télé-relève capable de rassembler un nombre d'informations suffisant pour des calculs statistiques assez fiables.

6.1 Positionnement du problème

Nous avons posé le problème suivant : L'énergie est elle gérer d'une manière optimale ? L'usager de certains services de première nécessité est-il satisfait ? Aux deux questions, nous répondons NON. Alors un autre problème se pose : Existe-il des moyens pour optimiser la gestion de ces ressources ? Si oui proposer une solution.

6.2 Cahier de charges

Le problème dans la gestion électrique est le suivant : « L'électricité est une énergie qui ne se stock pas !, ce qui est produit doit être consommé instantanément ».

Partant de ce principe nous devons réaliser un système qui fera en sorte que la production et la consommation soient suffisamment équilibrés pour éviter des désagréments aux usagers.

Un système intelligent capable d'équilibrer la production et la consommation, qui sera composé d'un ensemble de sous systèmes décentralisés, c'est-à-dire que chaque secteur — lotissement , cité, quartier, etc. — doit en être doté et il doit être capable de prévoir sa consommation à l'avance et la transmettre au système de supervision, de plus ce sous système doit être capable d'appliquer une tarification en temps réel à ces abonnés, cette tarification est préalablement définie par le superviseur et mise à disposition du client afin

qu'il puisse s'organiser et ainsi choisir de consommer lorsque l'énergie est la moins cher mais aussi par la même occasion lorsque celle-ci est le plus disponible.

6.3 Architecture globale

De par l'étude technologique faite précédemment nous avons confectionné une architecture dédié à la gestion d'un réseau électrique.

Le réseau en question sera structuré en plusieurs niveaux :

premier niveau ce niveau sera une couche de terrain c'est-à-dire la partie du réseau qui fournira les données relatives aux clients.

deuxième niveau le second niveau correspond aux équipements qui s'occuperont de la collecte des données terrains.

troisième niveau concentration et archivage des données, la configuration adoptée étant la topologie étoile il fera office de Hub, mais il sera charger en outre d'exécuter des applications informatiques pour les calculs et aussi se chargera de toutes les tâches liées à la gestion du réseau.

6.4 Choix matériel

6.4.1 Ensemble de comptage

Nous les avons ainsi nommer « ensemble de comptage » car ça sera des boitier incluant un compteur et un modem ainsi qu'un coupleur.

Il y a là deux ensembles de comptage distincts que nous allons devoir former, l'un pour des usagés résidentiels et l'autre pour des usagés industriels.

les compteurs

Notre choix s'est porté sur deux types de compteurs selon le type de client

Le compteur industriel Voici les caractéristiques qu'il doit présenter :

- une précision de classe 0,2.
- mesure les énergies active et réactive triphasées sur 4 quadrants (2 sens de direction pour chaque type d'énergie).
- enregistrement de 32 valeurs sélectionnées au minimum à des instants préprogrammés ou sur commande externe. Les minimas et maximas des valeurs instantanées doivent être également enregistrés.
- possibilité de compenser par calcul les pertes du système : pertes du transformateur de puissance, pertes en ligne et pertes des équipements auxiliaires.

- correction des erreurs de rapport et de déphasage des transformateurs de mesure par phase.
- réalisation d'autodiagnostics pour pouvoir s'assurer du bon fonctionnement du compteur, avec classement en erreurs non-fatales et fatales et aussi être en mesure d'envoyer un rapport.
- communications sur port RS 232 : 2 ports fonctionnant jusqu'à 115 200 bauds avec possibilité de raccord de modems externes.
- 500 Ko de mémoire RAM minimum à partager entre toutes les données mémorisées , la sauvegarde en cas de coupure d'alimentation est assurée par une pile.
- qualité de tension : mesure des creux de tension, des surtensions, des coupures et des déséquilibres de phases, enregistrés en fonction de la programmation.
- mesure des harmoniques sur l'onde de tension.
- programmation : par l'intermédiaire d'un port RS 232 .

Le compteur individuel Voici les caractéristiques qu'il doit présenter :

- une précision de classe 2.
- mesure dans les 4 quadrants.
- horloge de tarification intégrée.
- préservation de la stabilité de l'heure par le biais d'une pile.
- fonctions d'anti fraude.
- fichier journal pour stocker toutes les évènements avec horodatage.
- mesure des grandeurs instantanées.
- mesure de la qualimétrie.
- deux sorties électroniques RS232.

Modem

Différents types de modulations existent avec pour chacune une application particulière. Avec les contraintes que nous avons

- le réseau sur lequel vont être injecté les signaux se trouve en outdoor et donc suppose des problèmes d'atténuation et de perturbation du signal.
- une grande densité de communications— plusieurs milliers de compteurs à interroger à plusieurs reprises par jour dans un réseau isolé galvaniquement —.
- communication en broadcasting.

La plus appropriée des modulations sera l'OFDM (Orthogonal Frequency Division Multiplexing). Sachant que plusieurs firmes proposent des puces OFDM parmi lesquelles nous citerons FREESCALE, MAXIM, etc.

Terminal distant

Le terminal distant est l'organe qui s'occupe de la récolte des données depuis les compteurs mais aussi la commande de la tarification à appliquer pour ceux-ci.

Centre de traitement de données

Définition : Un centre de traitement des données se présente comme un lieu où se trouvent différents équipements électroniques, surtout des ordinateurs et des équipements de télécommunications. Comme son nom l'indique, il sert surtout à traiter les informations nécessaires aux activités d'une entreprise.

Les bases de données étant souvent cruciales au fonctionnement des entreprises, celles-ci sont très sensibles à leur protection. Pour cette raison, ces centres maintiennent de hauts niveaux de sécurité et de service dans le but d'assurer l'intégrité et le fonctionnement des appareils sur place.

Les équipements présents dans un centre de traitement de données Les équipements principaux sont :

- routeurs.
- commutateurs.
- pare-feu.
- disques durs.
- serveurs.
- câblage d'interconnexion.

Tous ces équipements sont pluguer (inséré) dans des armoires RACK.

Des équipements périphériques :

- climatisation précise et stable.
- contrôle précis de la poussière environnante.
- unité de distribution de l'énergie.
- bloc d'alimentation d'urgence.
- système d'alerte d'incendie.
- extinction automatique des incendies par microgouttelettes ou gaz inerte.
- surveillance par caméras en circuit fermé.
- contrôle des accès, ainsi que la sécurité physique.

6.5 Choix et développement logiciel

Dans cette section nous avons développé des interfaces homme-machine avec différents types de publics — client, superviseur —.

Les deux interfaces web sont faciles d'accès et surtout ne demandent pas d'installations particulières au préalable. En effet, elles ont été codées en HTML (hyper text market langage) qui est interprété par n'importe quel navigateur web et affiché sous forme de widgets exploitables par l'utilisateur.

6.5.1 Interface utilisateur

Cette interface doit être publique et accessible depuis n'importe quel point d'accès internet. Cependant, pour des raisons de sécurité cette page est verrouillée par un nom d'utilisateur et un mot de passe propre à chaque client.

Cette identification de l'utilisateur lui permettra d'aller directement dans son espace personnel et d'avoir accès aux informations qui lui seront utiles.

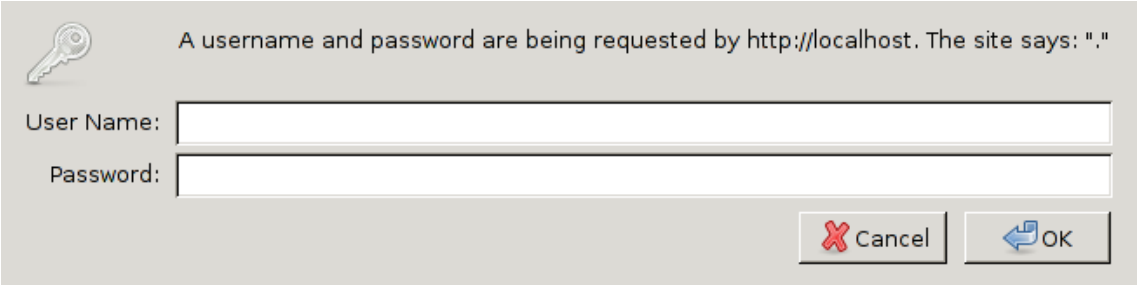


FIGURE 6.1 – Saisie du login et du mot de passe.

Il faut savoir que le verrouillage a été réalisé de manière standard — par le serveur http(thttpd) — qui n'est pas propre a nôtre travail.

Une fois l'authentification réussie, le serveur fournit les informations relatives au compte du client. La figure 6.2 représente cette page.

La page en question lui affiche la zone à laquelle il appartient car la société de distribution peut affecter des tarifs différents à des endroits différents pendant la même heure. Seulement, un calendrier est mis à disposition du client afin qu'il puisse prendre connaissance de la tarification envisagée à l'avance.

Le client peut aussi avoir connaissance de sa consommation quotidienne d'énergie, même si cela n'est pas du temps réel au sens électronique ou même informatique du terme. Cela est considéré comme étant du temps réel dans le domaine, car avant la technique de la télé-relève les relevé d'index se faisait une fois tout les deux mois.

Un troisième champ concerne la facturation. Celle-ci est calculée selon des paliers de consommation qui sont préalablement définis.

6.5.2 Interface superviseur

L'accessibilité de cette page par internet, permettra, par exemple, à l'ingénieur de s'y connecter depuis une station mobile — PDA ou autres.

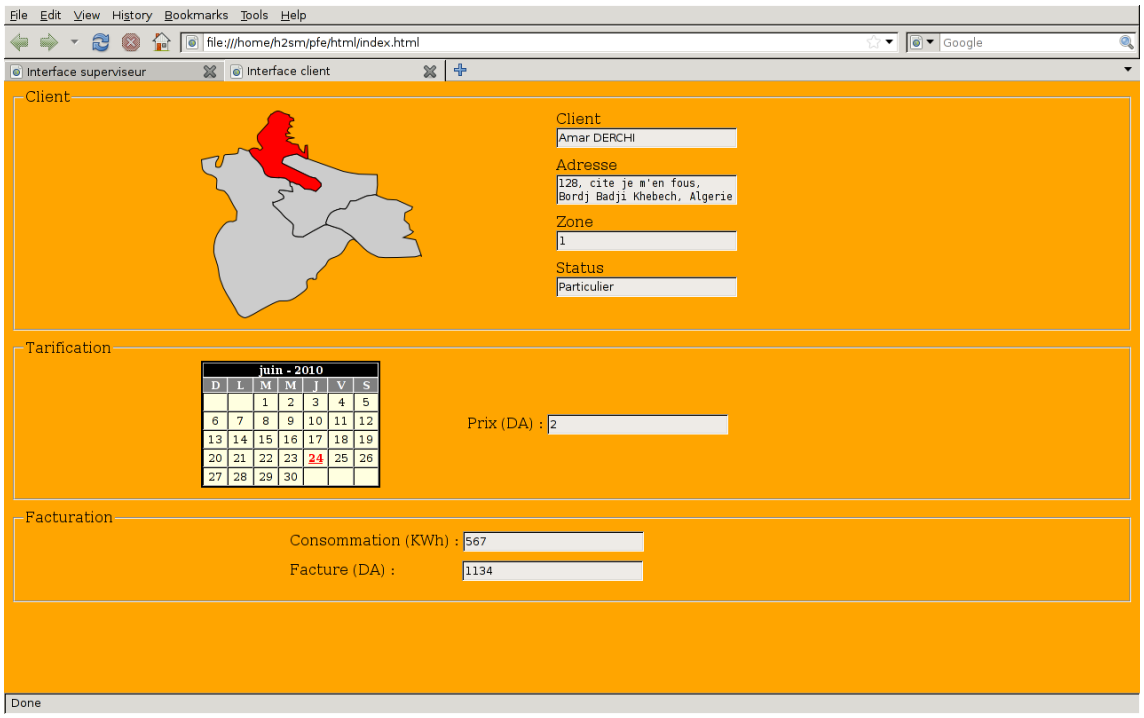


FIGURE 6.2 – Capture d'écran de l'interface coté client.

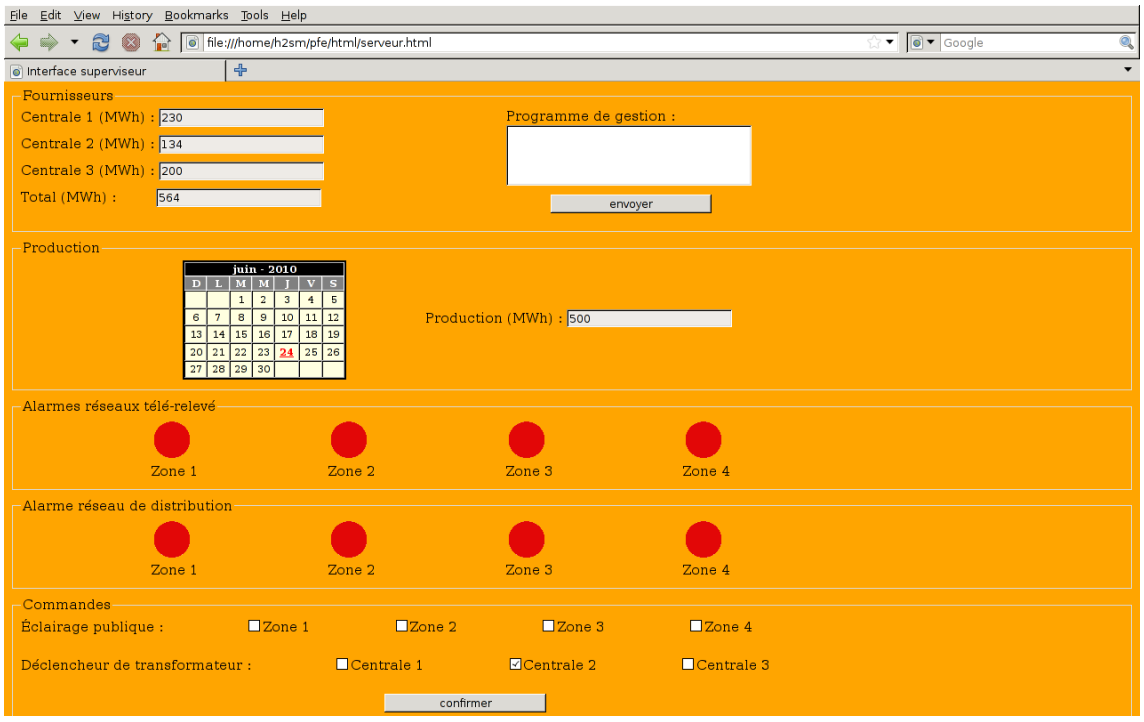


FIGURE 6.3 – Capture d'écran de l'interface coté superviseur.

Celle-ci contient un nombre d'alarmes qui peut varier d'une IHM à l'autre mais aussi des données relatives à la production, à la consommation globale.

Là encore la puissance de cette solution réside dans sa portabilité car ces alarmes doivent être constamment surveillées.

En plus, d'autres outils de supervisions sont mis à disposition dans cette interface à savoir des interrupteurs de régulation forcé telle la diminution de charge via le délestage (déclencheurs de transformateurs) ou bien avec l'alternative bilatérale (allumage et extinction)de l'éclairage publique.

On pourrait aussi imaginer une case permettant le dialogue avec les centrales via un langage prédéfini — représenté dans notre interface par la partie « programme de gestion » —.

D'un point de vue transmission notre travail n'a pas était achevé mais notons que dans ce projet Deux types de transmissions sont envisagés La première entre le serveur et les clients qui n'est rien d'autre que le réseau internet ADSL d'un FAI. La seconde vient de l'idée d'exploiter le réseau existant — celui du distributeur d'énergie —pour transmettre les données relatives à la télé-relève Pour ce faire la technologie CPL nous a paru comme étant un bon support (relativement fiable et avec une compatibilité protocolaire assez intéressante)

Conclusion

L'architecture proposée est une architecture très répondue en industrie. Cependant, nous avons pris deux aspects essentiels que sont l'électronique industrielle et l'informatique avec des applications haut niveau pour en faire un projet de télésurveillance et d'acquisition de données. Malheureusement, ce travail reste pour le moment au stade d'étude.

Conclusion et perspectives

L'objectif de notre travail a été l'étude d'un système de supervision et de commande à distance, pour fournir un document introductif aux techniques utilisées actuellement dans l'industrie.

Étant donné que le sujet n'a jamais été abordé au sein de l'École Nationale Polytechnique, nous avons essayer de fournir un début de réponse à quiconque voudrait approfondir ces connaissances du sujet, tout en essayant d'être le plus général possible.

Les technologies de communication disponibles pour ce genre d'installation présentent une variété impressionnante ce qui est un plus considérable à l'adoption de la technique.

Nous avons opter pour une introduction progressive de la programmation réseau orientée objet, avec un exemple didactique qui est l'allumage et l'extinction de LEDs à distance, cette application aussi simpliste soit elle montre bien les principes de base d'un système SCADA.

Finalement, une application plus conséquente a été présentée. La télé-relève est un domaine prometteur pour la simple raison que dans notre modèle social actuel, l'énergie joue un rôle centrale, ce qui oblige les industriels à optimiser la gestion de cette ressource.

Il reste, bien évidemment, des possibilités d'approfondissement de l'étude en ce qui concerne la partie pratique. Par exemple, dans la première application l'aspect authentification a été totalement ignoré pour des raison de clarté.

Ce qui a fait défaut à notre travail a été l'absence d'une réalisation pratique, mais ce choix peut être justifié. Dans la première application, une réalisation pratique n'aurait pas une valeur ajoutée réelle puisque le but été de présenter la programmation réseau. Tandis que le deuxième exemple nécessite un matériel qu'on a pas pu fournir pour des raisons financières.

Bibliographie

[1] A. Alhereish. *Design and implementation of home automation system.* IEEE, 2009.

[2] Laurent Bacon and Cédric Bellec. *Making the most of hydropower through a cascade center.* 2006.

[3] R. Capobianchi, A. Coen-Porisini, D. Mandrioli, and A. Morzenti. *A framework architecture for supervision and control systems. ACM Comput. Surv.* 2000.

[4] CEI. *Digital data communication for measurement and control - FieldBus for use in industrial control systems.* 2003.

[5] Tuan Dang and Chéramy Robert. *Impacts of electricity market liberalization on centralized generation and telecontrol infrastructure.* 2006.

[6] Thomas Hadlich. *Providing device integration with OPC-UA.* 2006.

[7] Brian Hall. *Beej's Guide to Network Programming.* Jorgensen Publishing, third edition, 2009.

[8] Modicon inc. *Modicon ModBus protocol reference guide.* 1996.

[9] Object Management Group Inc. *Data Acquisition from Industrial Systems.* 2005.

[10] Object Management Group Inc. *Data Distribution Service for Real-time Systems.* 2007.

[11] OPC inc. *OPC Unified Architecture, Specification – Part 1 : Concepts.* 2006.

[12] W. Kang, H. Kim, and H. S. Park. *Design and performance analysis of middleware-based distributed control systems.* 2001.

[13] Bill Kennedy and Chuck Musciano. *HTML : the definitive guide.* O'Reilly, 1998.

[14] P. Marti, J. Aguado, F. Rolando, M. Velasco, J. Colomar, and J. Fuertes. *A java-based framework for distributed supervision and control of industrial processes.* 1999.

[15] Mahmoud shaker Nasr. *Friendly Home Automation System Using Cell Phone and J2ME with Feedback Instant Voice Messages.* University of Garyounis, 2009.

[16] O.M.G. *Data Acquisition from Industrial Systems (DAIS).* 1999.

[17] David Rey. *Contrôle, commande et mesure via Internet.* Dunod, 2008.

[18] J. Tisal. *GSM, réseaux et services.* Masson, 2006.