

Rapport d'entreprise



IUT DE CACHAN

Maître d'apprentissage : Yann Brossard
Administrateur automatismes et systèmes
informatiques industriels

Tuteur pédagogique : Benoit Obled

Rédigé par : Valère Gay-Heuzey

Le 20 août 2020

Introduction

Je m'appelle Valère Gay-Heuzey, apprenti en licence au sein de L'Université Cachan Paris-Saclay. Je suis en Licence MECSE option SESAM (Métiers de l'Electronique, Communication, Systèmes Embarqués, option Systèmes Embarqués et Solution par Application Mobile).

Depuis mon baccalauréat, je suis une formation en alternance afin de multiplier les expériences de travail. J'ai ainsi réalisé 22 semaines de stages en bac professionnel et 2 ans d'apprentissage en BTS. C'est dans ce même esprit, que j'ai entrepris d'accomplir une licence en alternance.

Je suis actuellement en contrat avec la société RATP Dev Angers, filiale de RATP Group, en qualité de développeur. Ce contrat dure 1 an et s'arrête le 4 septembre 2020.

Du fait de sa taille et de son positionnement sur le marché français et international, j'ai eu l'occasion de découvrir la complexité très intéressante des rouages de cette entreprise.

De plus j'ai eu à travailler sur un projet riche, abordant de multiples aspects et domaines technique.

Remerciement

Je remercie M. Brossard et M. Prunier pour leur implication et leur disponibilité. Leurs conseils ont été précieux dans le cadre de ma formation professionnelle. Je les remercie également pour m'avoir permis de travailler en grande autonomie, en étant responsable de ma charge de travail et des résultats à fournir.

Abstract

I'm currently a student of the Cachan's University (Paris Sud). I do a professional, apprenticeship, licence year, studying especially embedded electronics and IT. I've been hired by RATP Dev as an IT Developer of application.

My main project is to develop a file's sharing system, from an automaton to a database.

It is divides in 2 main parts:

- Install a connexion between an industrial automaton and an embedded computer.
- Transfer file from an embedded computer to a distant database.

This project also may have 2 extended objectives:

- Develop an GUI (Graphic User Interface) to use the collected data
- Find a way to analyse those information

These extended objectives may being accomplished by myself, but in a next year.

Table des matières

Introduction	2
Remerciement.....	3
Abstract.....	4
Présentation de l'entreprise	6
Description.....	6
Contrat de délégation publique entre Angers Métropole et RATP Dev	7
L'offre de service de RATP Dev à Angers.	8
Organigramme RATP Dev à Angers.....	9
Présentation RATP Dev & Groupe RATP	10
RATP Dev en France & à l'international : concurrence de RATP Group.	11
Contraintes stratégiques.....	12
Projet Professionnel	14
Contexte	14
Enjeux	14
Objectifs.....	15
Diagramme de Gantt	17
Description projet - Partie acquisition	18
Serveur TCP/IP	19
Serveur Modbus	21
Enregistrement des valeurs	22
WatchDog - Descriptif	25
WatchDog – Fonctionnement	26
Description projet - Partie communication	27
Réseau 4G & GPS.....	27
Serveur Specinov & Batch	29
Développement futur	31
IHM	31
Développement futur – Analyse.....	34
Panne et Incident	34
Développement futur – Analyse.....	35
Maintenance Prédictive.....	35
Conclusion.....	36
Annexe	37
Tables des matières	37
Lien GitHub	38

Présentation de l'entreprise

Description

La RATP, régie autonome des transports parisiens, existe depuis plus d'un siècle et a créé la première ligne de métro parisienne achevée le 19 juillet 1900 à l'occasion de l'exposition universelle.

Depuis, cette société n'a cessé de développer de multiples technologies dans le domaine des transports, comme les premiers métros automatique, tramways électriques et navettes autonomes, faisant d'elle aujourd'hui le leader mondial du métro automatique et du tramway.

RATP Dev est une filiale de RATP Group : créée en 2002, elle a pour objectif de développer, exploiter et maintenir de nouveaux réseaux en France, en Europe et dans le monde, en s'appuyant et en déployant l'expertise du groupe RATP. Son rôle consiste à assurer le fonctionnement des services de transport de personnes, et la maintenance du réseau et du matériel. Pour cela, RATP Dev innove afin de proposer des solutions de transports adaptées au contexte urbain dans lequel elles s'inscrivent.



Figure 1 : photo du métro en 1957

Nombre de salariés : 20 000 salariés

Capital Social : 369M€

Chiffres d'affaires : 1.3Md€

Contrat de délégation publique entre Angers Métropole et RATP Dev

RATP Dev est lié par un contrat de délégation publique à la métropole de Angers Maine et Loire pour une durée de 6 ans (jusqu'en juin 2025). Ce type de contrat implique que la collectivité délègue et confie à une entreprise, la responsabilité et la gestion d'un service public, en l'occurrence le transport public des biens et des personnes.

Dans cette configuration, l'ensemble des services, fournitures et matériels appartiennent à la collectivité. Cela comprend les bus, tramways et taxis utilisés pour le transport de personnes, mais aussi la location des locaux ainsi que les fournitures et pièces de rechanges nécessaires à la maintenance.

Tous les bénéfices reviennent à la métropole qui paie à la société titulaire une redevance de 39M€ par an, pour la gestion du réseau.

Il existe ainsi 3 entités :

La métropole (collectivité) Angers Maine-et-Loire : le Déléguant

L'entité commerciale à travers laquelle la métropole vend des services et obtient des bénéfices : Irigo

La société titulaire du contrat : le Délégataire (ou opérateur)

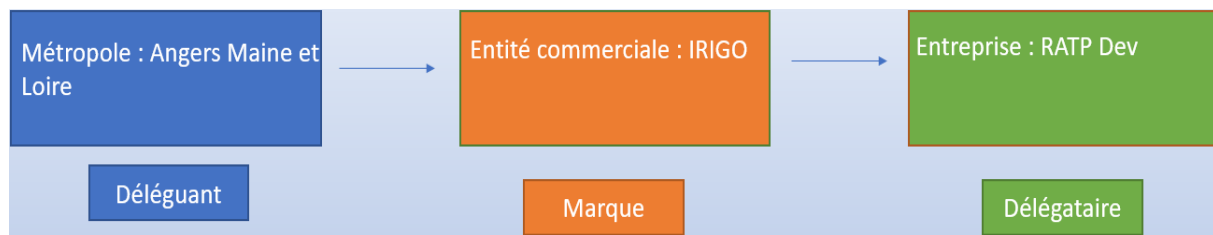


Figure 2 : schéma des relations contractuelles

La société RATP Dev Angers prend en charge un « bassin de vie » de près de 400 000 habitants au total dont 290 000 sont situés sur la ville de Angers. Le réseau de la métropole angevine couvre près 51 000 hectares et 29 communes.

Le précédent opérateur de transports urbains de la métropole d'Angers Maine-et-Loire était la société KEOLIS pendant près de 30 ans. Néanmoins, lors de l'appel d'offres, la société RATP Dev a été « mieux disante » fait une offre moins chère que celle de KEOLIS (45M€ pour ces derniers contre 39M€ pour RATP Dev).

L'offre de service de RATP Dev à Angers.

La société RATP Dev fournit un panel de services visant à satisfaire l'ensemble de la population présente sur son territoire. Sont notés ci-dessous les services précédemment proposés par le précédent opérateur KEOLIS, puis repris par RATP Dev.

Service de transport classique :



Services spéciaux :



Figure 3 : offre de services RATP Dev

Depuis, RATP Dev a également mis en place plusieurs services propres sur la métropole d'Angers :

- ✚ Une ligne de bus circulaire, visant le transport d'étudiants faisant la fête (Angers étant une ville très étudiante notamment) ;
- ✚ Le M ticket, un ticket dématérialisé à télécharger sur son téléphone. Cela permet d'éviter les gâchis de papier, la perte de son ticket, et l'achat en bornes ;
- ✚ La mise en place d'un parc automobile roulant à 100% au gaz naturel. En effet, tous les nouveaux bus achetés ne rouleront plus au gasoil dans un souci écologique.

Organigramme RATP Dev à Angers.

Je travaille au sein du service SI & CFA : Service Informatique et Courant Faible (les locaux étant situés au même endroit).

Mon responsable de service est M. Thierry Prunier et mon tuteur M. Yann Brossard.

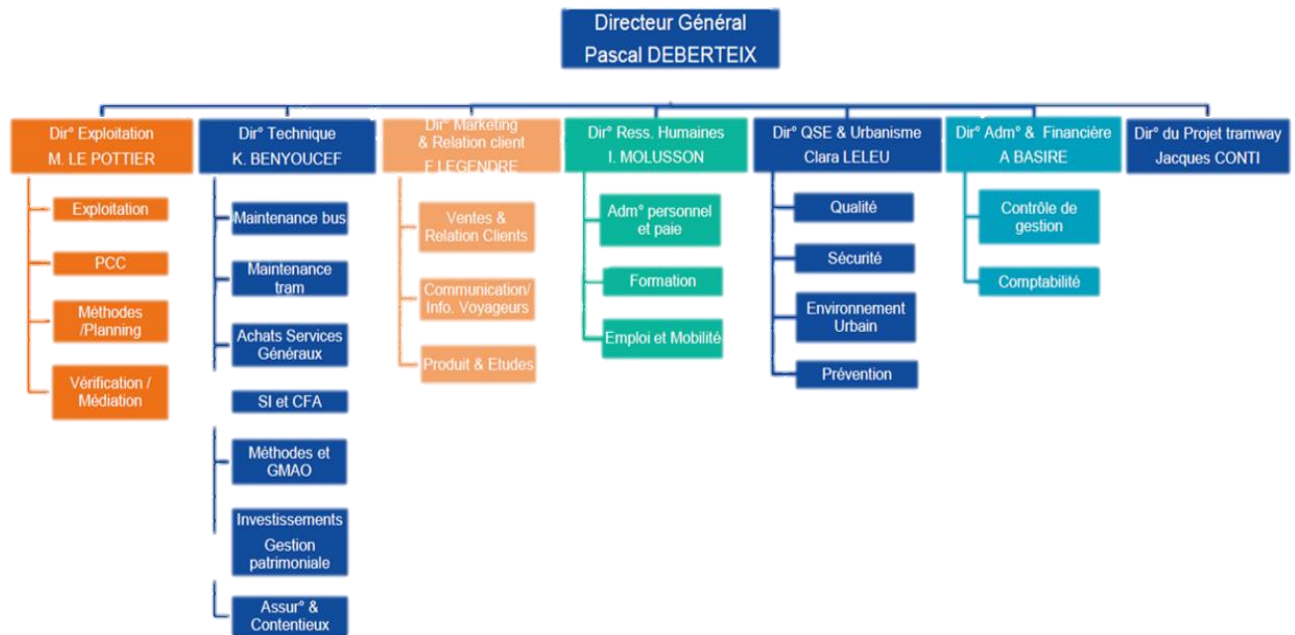


Figure 4 : organigramme RATP Dev

Présentation RATP Dev & Groupe RATP

Le groupe RATP comporte des activités réparties dans plusieurs secteurs principalement dans la « mobilité urbaine » mais aussi dans l'immobilier, et le mécénat.

Son chiffre d'affaires en 2019 était de 5.56 Milliards d'€. L'effectif du groupe s'élève à plus de 58 000 personnes à travers le monde.

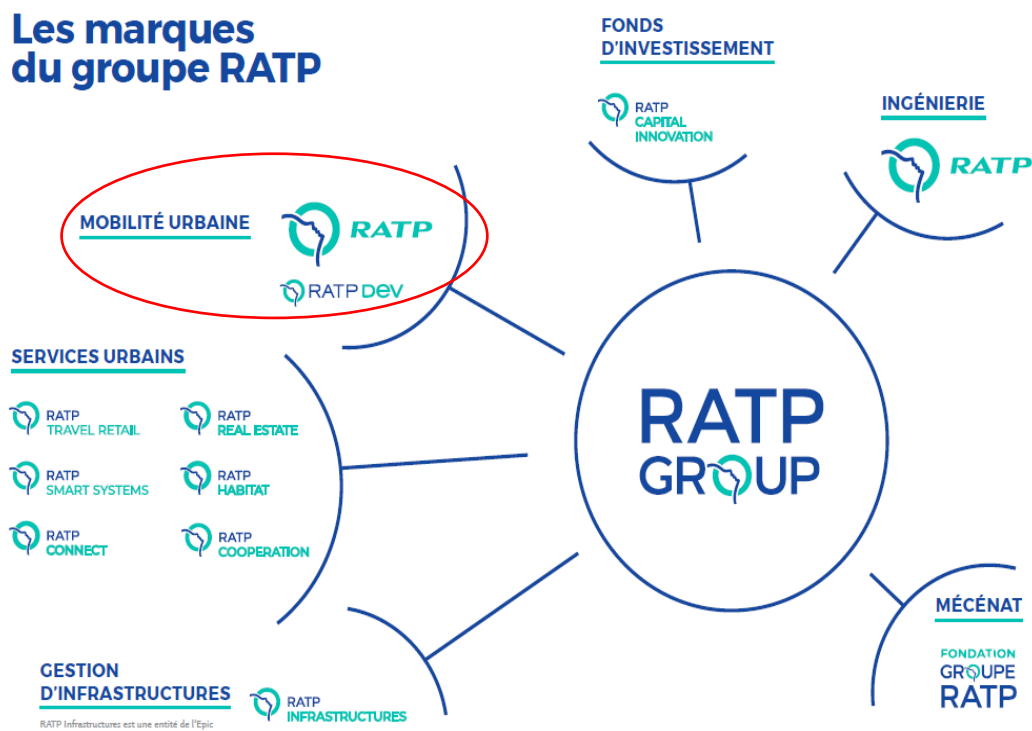


Figure 5: les marques du groupe RATP

La société RATP Dev est une filiale récente (création en 2002) du groupe RATP. Sa mission consiste à opérer l'intermodalité des moyens de transports, c'est-à-dire à utiliser tous les moyens à sa disposition pour construire ou restructurer un réseau de transport.

L'ensemble des filiales a fait une progression de plus de 142M€ en un an. RATP Dev est en pleine croissance (+10.4% sur l'année 2019).

RATP Dev en France & à l'international : concurrence de RATP Group.

La RATP a été en situation de monopole pendant plus de 40 ans. Cependant, la loi orientation mobilité (l'OM) promulguée fin 2019 ouvre à la concurrence le marché des transports publics de la région Île-de-France.

Avant cette ouverture à la concurrence, le groupe RATP qui n'a pas été soumis à la pression de ses concurrents a pu se consacrer à développer un panel de produits de qualité avec des offres commerciales proposées selon les publics. Au sein de la métropole d'Angers par exemple, Mouv'Irigo a mis en place un service d'aide destiné aux personnes à mobilité réduite.

Depuis l'ouverture à la concurrence en 2019, la RATP se retrouve face à plusieurs géants du transport de personnes, notamment Transdev et Keolis qui interviennent sur le même segment et qui se concurrencent. Keolis est positionné pour la construction de la future ligne de métro 15 à Paris qui entrera en service en 2025. De son côté, la filiale RATP Dev a obtenu fin 2019 un important contrat de gestion des transports publics dans l'ouest du pays (dont Angers, Lorient, Brest...). La RATP demeure un groupe important dans le domaine des transports de personnes.

En somme, le groupe RATP n'est plus en situation dite « d'océan bleu », elle rentre en état « d'océan rouge ». Il est important de noter que cette mise en concurrence a permis à la RATP de ne pas stagner et de se confronter à la concurrence afin de permettre l'essor de nouvelles technologies.



Figure 6: présence de RATP dans le monde

Contraintes stratégiques.

La RATP rencontre plusieurs contraintes liées à ses missions :

- La sécurité : elle représente la principale contrainte. On estime qu'environ 3.5 Milliards (2016) de personnes circulent chaque année ; aussi la sécurité est-elle un enjeu majeur. L'ensemble des dispositifs fournis par le groupe sont soumis à de multiples vérifications et normes, tout autant européennes que françaises. A titre d'exemple, mon projet au sein de RATP Dev concerne un système d'alimentation électrique de secours. Ainsi, avant que le prototype lié à mon projet ne soit implémenté dans les tramways sur lesquels je travaille, une commission gouvernementale sera chargée de valider l'emplacement du prototype au sein du matériel ainsi que d'évaluer sa dangerosité.
- Les valeurs éthiques : il existe un guide de 16 pages relatif à l'éthique au sein de l'ensemble du groupe RATP, qui traite par exemple de la communication de la société avec les médias ou encore de la conduite des employés vis-à-vis des discriminations et du racisme.
- L'écologie : elle a une place importante au sein du groupe. Les initiatives en ce sens sont multiples. Par exemple, l'ensemble du parc autobus de la métropole d'Angers Maine-et-Loire et de la région Île-de-France est dorénavant complété par des bus plus écologiques, roulant au gaz naturel. Citons également la ville de Londres, qui est équipée depuis 2018 par des bus 100% électriques fournis par RATP Dev.



Figure 7: un bus électrique avec l'inscription « ligne 100% électrique »

Le groupe RATP a des contraintes de service public, qui déterminent la tarification des billets imposée par les différentes métropoles et groupes représentatifs.

Il revient à la collectivité délégante de fixer les tarifs et les modalités de leur évolution, le délégataire n'étant pas compétent en la matière (Cf. Article 32 de l'ordonnance n° 2016-65 du 29 janvier 2016 relative aux contrats de concession ».

Il existe aujourd'hui une controverse entre les différents opérateurs des transports publics (tout opérateur confondu : Keolis, Transdev, RATP etc...) et les collectivités. Les opérateurs se plaignent d'une hausse de la quantité et de qualité des services proposés alors que les prix eux n'ont pas suivis, ce qui se traduit par une baisse de rentabilité pour tous ces groupes. En outre, RATP Dev est lié par une délégation des services publics pour une durée de 6 ans, ce qui fait qu'elle doit aussi rendre des comptes au délégataire, la métropole.

Site RATP Dev en France :

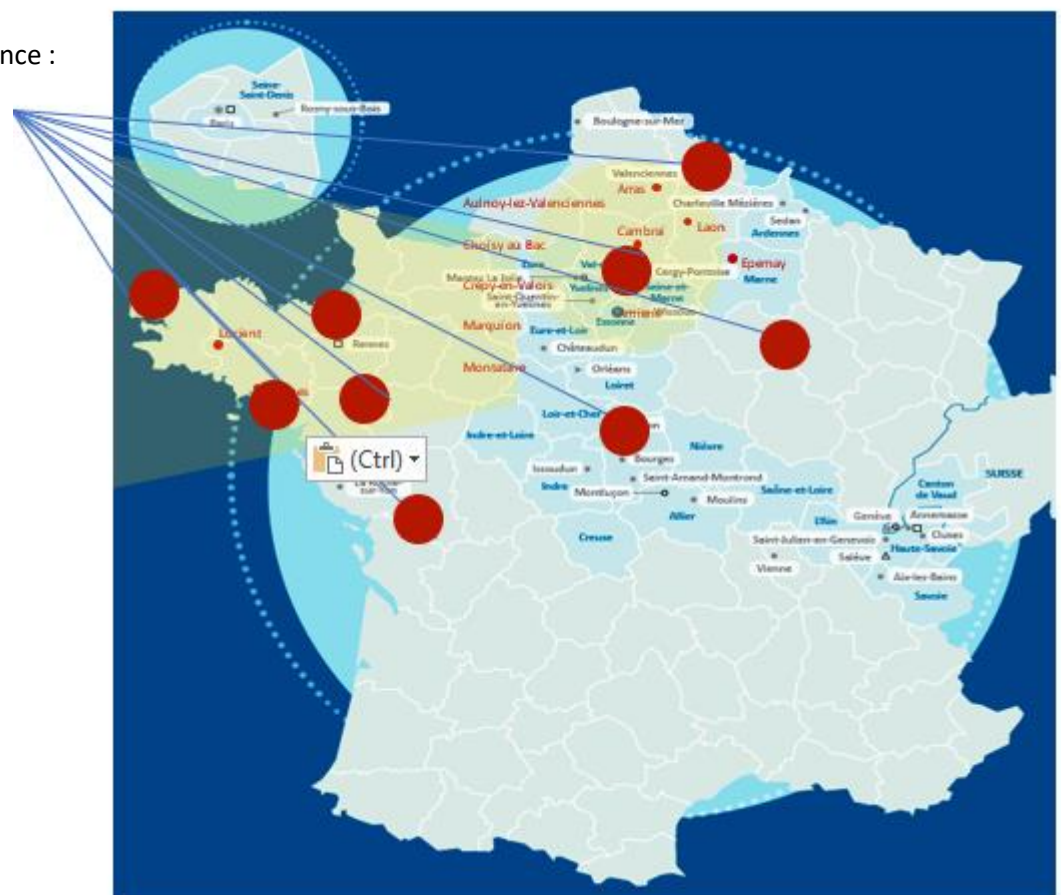


Figure 8: les implantations de RATP Dev dans l'Ouest

Projet Professionnel

Contexte

Un incendie est survenu il y a 18 mois, au sein d'un tramway de la métropole d'Angers, suite à une panne du système d'alimentation de secours : les batteries de secours qui alimentent le tramway, ont pris feu lors et ont brûlé le toit en provoquant uniquement des dégâts matériels. L'entreprise a décidé de remédier à ce grave dysfonctionnement en analysant les données en provenance des batteries de secours.

Emplacement batterie de secours



Figure 9: tramway angevin

Enjeux

Ma mission au sein de RAPT Dev consiste à réaliser un système de récupération de données, et à contribuer à la mise en place d'une interface IHM (Interface Homme-Machine) ¹ pour exploiter les valeurs recueillies en provenance des batteries de secours.

De plus le cahier des charges impose une récupération rapide des données : système en semi temps réel (voir partie sur acquisition) pour lequel les informations sont mises à jour toutes les minutes ainsi qu'une IHM qui permettra une bonne analyse des données.

Ce projet a été mené sous forme de POC (proof of concept) afin de déterminer et valider la faisabilité de cette nouvelle fonctionnalité de recueil et d'analyse des données. L'équipe dédiée était constituée de mon tuteur et de moi-même.

J'ai travaillé sur la base d'un cahier des charges qui a évolué au fil du temps selon l'avancée des travaux et les ressources disponibles.

Si la cause de ce départ de feu n'est pas connue aujourd'hui, son analyse toujours en cours, est située en dehors du projet décrit ci-après.

¹ Définition : les interfaces homme-machine ou IHM sont les moyens et outils mis en œuvre afin qu'un humain puisse contrôler et communiquer avec une machine. Les ingénieurs en ce domaine étudient leurs interactions avec les ordinateurs ainsi que la façon de concevoir des systèmes qui soient ergonomiques, efficaces, faciles à utiliser ou plus généralement adaptés à leur contexte d'utilisation.

Objectifs

Les objectifs du projet consistent d'une part à établir un pont de communication entre un automate présent dans chaque tramway et la base de données distante, et d'autre part à analyser les informations obtenues afin de prévenir les avaries et réaliser de la maintenance prédictive.

L'automate industriel a pour rôle de recueillir des valeurs, transmettre des informations, assurer le bon fonctionnement du matériel.



Figure 10 : automate industriel

Il a été décidé de raccorder à cet automate, des capteurs disposés sur les batteries de secours afin d'obtenir plus d'informations et permettre de déterminer l'origine de l'incendie.



Figure 11: batteries de secours

Ce raccordement a été effectué par le service MA (matériel roulant, cf. organigramme) à la demande de mes supérieurs. De plus, mon maître d'apprentissage a fait les modifications nécessaires sur l'automate afin d'enregistrer les nouvelles valeurs de tensions dans l'automate.

Le projet est découpé par mon tuteur en quatre étapes successives :



- 1- Acquisition, consistant à récupérer des variables depuis un automate vers un PC embarqué.
- 2- Communication : établir une communication distante depuis le PC embarqué vers une base de données externe.
- 3- IHM : réaliser une interface d'exploitation depuis une base de données.
- 4- Analyse : permettre l'exploitation des données.

Chaque étape doit être terminée avant de passer à la réalisation de la suivante. Un journal de bord détaille le travail accompli et m'a servi à poser les évolutions du projet.

La mission menée en alternance durant la licence SESAM porte sur les deux premières parties « acquisition » et communication ». Elle a vocation à se poursuivre sur une année supplémentaire pour achever les étapes 3 et 4.

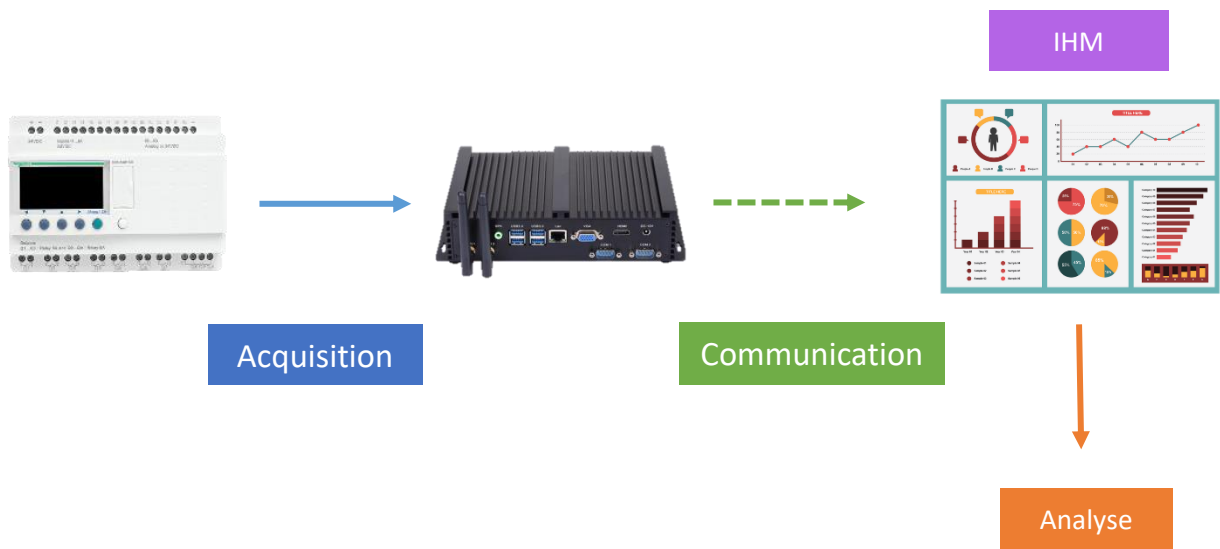


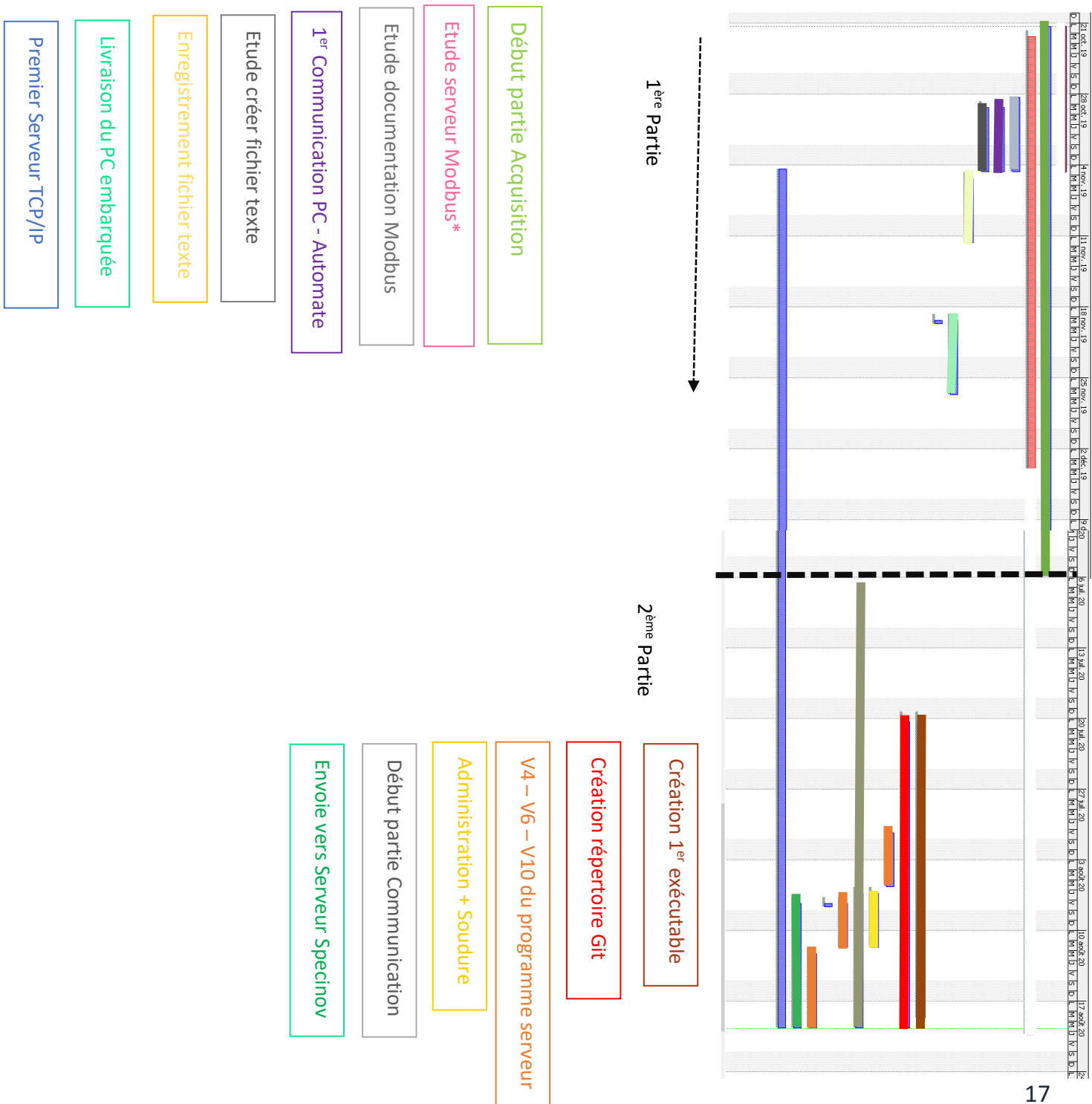
Figure 12: Schématisation étapes du projet

Diagramme de Gantt

Durant cette année alternance j'ai écrit un journal de bord, me permettant de retracer les évolutions du projet, et réaliser ce diagramme de Gantt. Celui-ci décrit essentiellement les évolutions de la première partie (Acquisition), ainsi que la seconde partie (Communication).

Ce diagramme se divise en 2 parties :

- A gauche : Partie Acquisition (du 21/10/19 au 18/11/19)
- A droite : Partie Communication (du 19/07/20 au 19/08/20)



Description projet - Partie acquisition

Pour permettre une connexion entre 2 appareils, il est nécessaire de mettre en place un serveur.

Celui-ci héberge et gère les communications entre plusieurs appareils comme un automate industriel et un PC embarquée

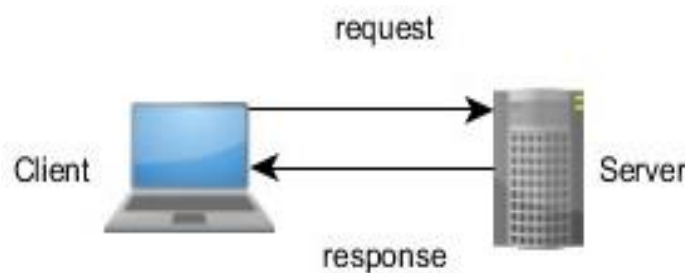


Figure 13: Communication entre serveur et client

Pour héberger la communication entre 2 appareils, il est nécessaire d'utiliser un protocole de communication identique. Un protocole de communication établit des règles communes afin d'organiser les échanges entre différents appareils. Par défaut, l'automate communique grâce au protocole Modbus, néanmoins j'ai décidé de changer son protocole (voir partie Serveur Modbus).

Le transfert des informations de l'automate vers une base de données a été effectué en s'appuyant sur un PC embarqué dans le tramway. Ce PC est qualifié de PC « industriel » car il répond aux normes ferroviaires de sécurité.

De plus, le PC dispose d'un Windows 10 dit « Windows embarqué », correspondant à un système Windows professionnel. Il se différencie d'un système classique par l'absence de fonction utilisateur tel que l'Intelligence Artificielle d'aide à la personne Cortana. Il sert deux objectifs :

- Communiquer avec l'automate afin de récupérer ses variables ;
- Transférer vers l'extérieur, les informations reçues.



Figure 14 : PC embarquée

Serveur TCP/IP

Le fonctionnement d'un serveur est de mettre à disposition des informations que des clients viendront récupérer en fonctions de leurs usages. On dit que le/les client(s) interrogent le serveur.

Il s'agit, le plus couramment, d'un système reposant sur le principe de maître – esclave.

Le client y est maître : il interroge le serveur, envoie ou récupère des informations, il est actif

Le serveur y est esclave : il attend des connexions, renvoie les informations demandées, ou stocke des valeurs. Dans le cadre du protocole TCP/IP, le serveur renvoie une unique réponse, confirmant la connexion au client (à l'instar du ModBus qui renvoient une trame à chaque réception d'informations).

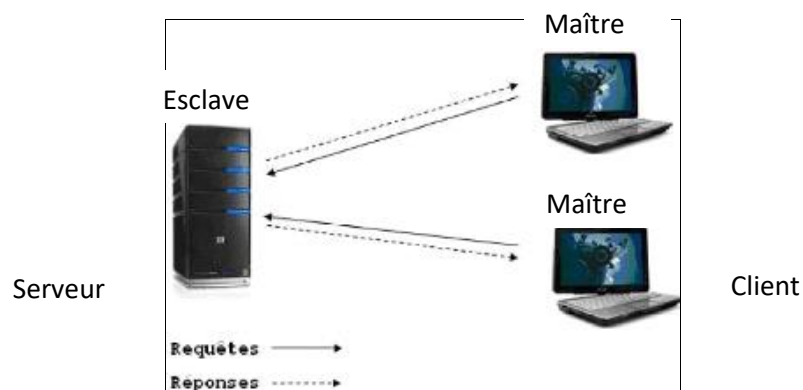


Figure 15: Schéma Maître - Esclave

Dans le cadre de ce projet, le PC est serveur et attend les requêtes de l'automate.

L'automate est client. La requête que celui-ci transmet contient les informations nécessaires à son identification ainsi que les variables à récupérer.

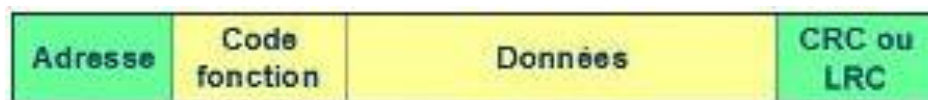


Figure 16: Exemple de trame requête

En l'absence de réponse qui accuse de la réception d'une requête, l'automate se met en défaut (conséquence ModBus - voir partie Serveur Modbus), bloquant le transfert de toute informations. La requête envoyée par l'automate contient les informations permettant d'identifier la machine vers qui elle doit communiquer (son adresse IP), ainsi que les variables à récupérer. Mon tuteur a modifié le code de l'automate faisant en sorte que dès que l'automate a fini d'envoyer la trame, celui-ci

stoppe la communication, compte une seconde, et renvoie directement les informations suivantes ; Ainsi l'automate n'a pas besoin de recevoir une trame réponse : cette méthode fonctionne sans générer d'erreur.

Pour permettre à l'ordinateur embarqué de générer un serveur, il est nécessaire de lui attribuer ce qui s'appelle un socket.

Un socket sert d'interface entre les ports physiques de l'ordinateur (USB, RJ45, port audio, etc.), et les ports virtuels de l'application qui vient traiter les informations reçues.

Un socket est composé d'une adresse IP et d'un numéro de port virtuel. L'adresse IP est unique sur un réseau et permet de définir une machine en particulier.

Le numéro de port virtuel est le point d'accès de l'application. En fonction du rôle de l'application, un port précis peut lui être attribué (par exemple les applications qui utilisent le protocole HTTP ont le port 80, pour le Modbus il s'agit du port 502). Néanmoins, toutes les applications n'ont pas de port précis et il leur en sera attribué un parmi ceux qui sont disponibles.

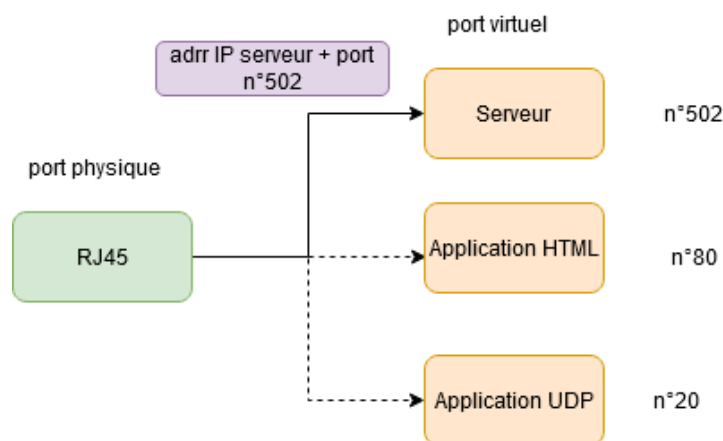


Figure 17: Définition socket

L'adresse IP des/de la machine vers qui l'automate doit se connecter est précisé au socket. Ici, l'adresse IP est celle du PC. Le port utilisé est celui du Modbus.

```
QString addressIp = "::ffff:10.32.105.129";  
QHostAddress addressIpUtf8 (addressIp);  
m_iServer = new QTcpServer(this);  
m_iServer->listen(addressIpUtf8, 502);
```

Figure 18: Exemple de création d'un serveur

La figure 13 montre la création d'un serveur TCP/IP. Le serveur attend les informations à l'adresse *adressIpUtf8* (adresse IP de l'automate) et au port 502.

A noter : le protocole TCP/IP, comme le Modbus, possède un numéro de port précis (le n°21) mais les trames de l'automate sont de type Modbus ce qui nécessite d'utiliser le port 502.

Serveur Modbus

L'automate présent dans les tramways utilise le protocole ModBus de communication largement présente dans le monde de l'industrie. Il est basé sur une structure hiérarchisée entre un maître et plusieurs esclaves. Ce protocole est celui utilisé par l'automate, néanmoins j'ai décidé d'utiliser le protocole TCP/IP.

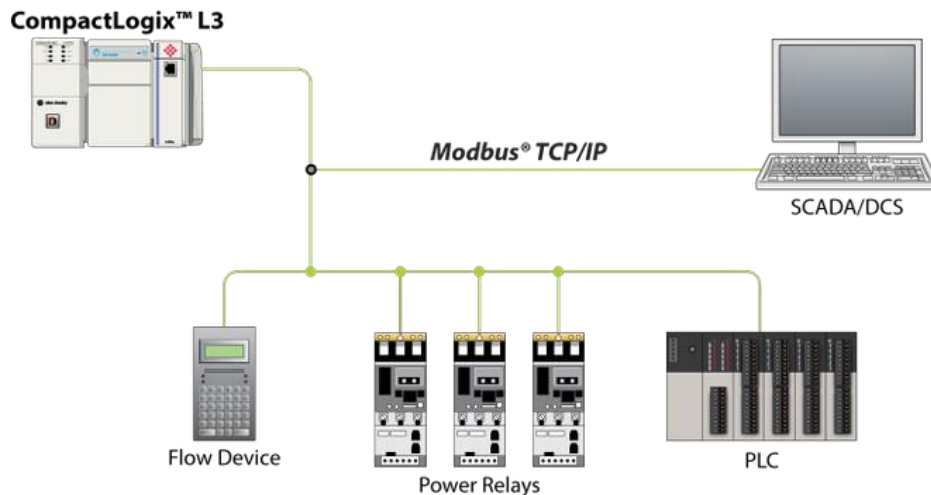


Figure 19: Exemple de communication Modbus entre automates

Le protocole Modbus répond à des exigences de fiabilité car il autorise beaucoup de communications. , et permet de nombreux échanges entre les dispositifs équipements. Il présente cependant plusieurs inconvénients :


- Ce protocole est légèrement inadapté car il contraint le PC à envoyer une réponse à la réception de chaque nouvelle information (le PC confirme l'état de la connexion est la bonne réception des informations). En l'absence de réponse, le protocole place l'équipement en défaut. Or, l'automate envoie ses informations en continu vers le PC (environ un envoi par seconde). La connexion est « d'office » confirmée chaque seconde, et renvoyer une réponse par le PC lui ferait notamment perdre du temps de traitement.
- De plus le protocole Modbus n'est pas aussi courant que le TCP/IP (les industriels fournissant peu de documentation), et il est compliqué d'obtenir des informations, notamment sur la composition de la trame réponse.
- Enfin, j'ai passé plusieurs semaines à tenter de mettre en place un serveur Modbus, mais en l'absence de résultats concluants, mon maître d'apprentissage m'a incité à passer sur une autre solution : en conséquence, j'ai choisi d'utiliser un serveur TCP/IP « classique ».

Enregistrement des valeurs

La connexion serveur établie permet d'enregistrer les valeurs de l'automate. Pour cela, il convient « d'analyser » la trame reçue de l'automate qui fait plus de 180 caractères, comprenant :

- Les informations « adresse » et « code fonction » (comme indiqué dans la figure 11)
- Les variables de l'automate
- Le CRC, qui sert à fiabiliser les communications, mais n'est pas activé ni utilisé ici. Il implique un échange d'informations entre l'automate et le serveur ce qui ralentirait trop la connexion.

Les variables de l'automate commencent à partir du 24^{ième} caractère : tout ce qui se trouve avant le 24^{ième} caractère est supprimé pour ne conserver que le reste.



```
Trame_recu
Trame_complète036500000063ff10000a002e5c00672500154407212020000b000c000d000e000f0010001100120
01300150016001700180019001a001b001c001d001e001f0020002100010001000000010000000100000010001
000000000000000100010001000100010000
Trame_séparé
00672500154407212020000b000c000d000e000f001000110012001300150016001700180019001a001b001c001
d001e001f002000210001000100000001000000010000000100010000000000000001000100010001000100
00
```

Figure 20: Exemple de trame complète et séparée

Une fois cette suppression faite, je convertis en hexadécimal les variables. Cette conversion est nécessaire car les valeurs que je reçois sont issues de plusieurs « bases » mathématique (les « bases » mathématiques correspondent au nombre de valeurs qu'une unité est capable de prendre).

Exemple :

- Base 2 (binaire) = 2 valeurs possibles : 0 & 1
- Base 10 (décimal) = 10 valeurs possibles : 0 1 2 3 4 5 6 7 8 9
- Base 16 (hexadécimal) = 16 valeurs possibles : 0 1 2 3 4 5 6 7 8 9 A B C D E F

Ainsi les variables reçues contiennent à la fois des valeurs décimales et binaires. Une conversion de toutes ces valeurs en base 16 me permet d'avoir des valeurs sous la même forme. De plus, afin de faciliter l'exploitation de ces valeurs, la conversion vient rajouter des 0 (tout à gauche de la valeur, là où elle n'a pas d'impact sur la valeur) afin d'avoir 4 unités.

Une valeur décimale qui vaut 456 deviendra 01C8

Une valeur binaire qui vaut 1 deviendra 0001

Je me charge ensuite de séparer en 4 par 4 les variables.

Une fois cette séparation faite, je peux mettre en place une mise en forme des valeurs afin de les enregistrer plus facilement dans un fichier texte. La mise en forme de ces valeurs implique 3 choses :

- Ajout des noms de chacune des valeurs ;
- Ajout de l'heure que fournit l'automate à chaque envoi ;
- Ajout des données GPS.

L'ajout des noms se fait en fonction de l'ordre reçu. En effet, je connais au préalable l'ordre de réceptions des valeurs, de sorte qu'il me suffit de placer les noms à la suite.

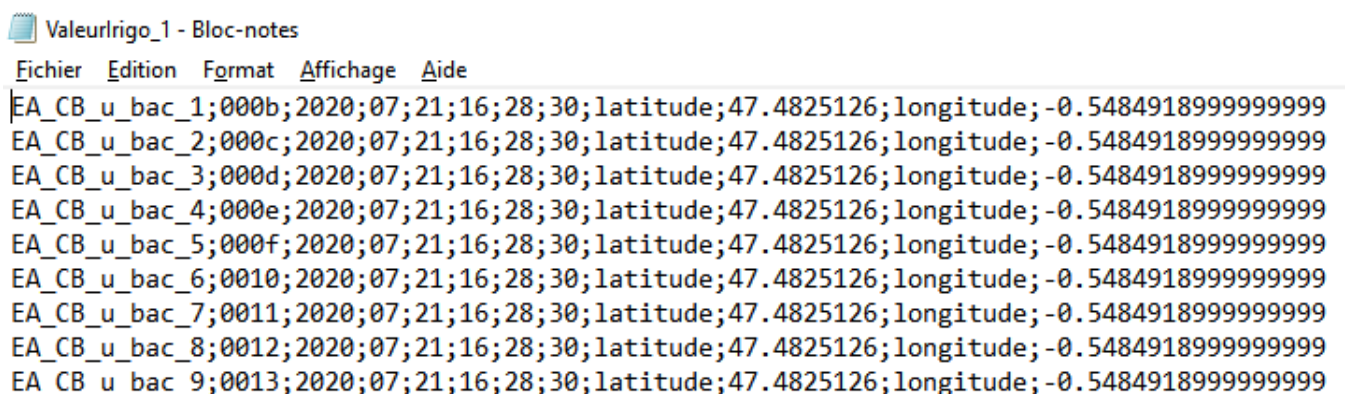
Chaque nom est placé devant la valeur qui lui est attribué.

L'heure de l'automate est la première valeur qu'envoie l'automate. Une fois cette valeur reçue, elle est ajoutée à la fin de chaque ligne (avant les données GPS).

```
listNomVal.append("EA_CB_u_bac_1");  
listNomVal.append("EA_CB_u_bac_2");  
listNomVal.append("EA_CB_u_bac_3");  
listNomVal.append("EA_CB_u_bac_4");  
listNomVal.append("EA_CB_u_bac_5");  
listNomVal.append("EA_CB_u_bac_6");  
listNomVal.append("EA_CB_u_bac_7");  
listNomVal.append("EA_CB_u_bac_8");  
listNomVal.append("EA_CB_u_bac_9");  
listNomVal.append("EA_CB_u_bac_10");
```

Figure 21: Liste de noms de variables

Enfin, les données GPS doivent être ajoutées à la fin de chaque ligne. N'ayant pas à l'heure actuelle le matériel nécessaire me permettant d'obtenir la position exacte de l'automate, j'ai ajouté moi-même les données GPS de l'entreprise.



Valeurlrigo_1 - Bloc-notes

Fichier Edition Format Affichage Aide

EA_CB_u_bac_1;000b;2020;07;21;16;28;30;latitude;47.4825126;longitude;-0.5484918999999999
EA_CB_u_bac_2;000c;2020;07;21;16;28;30;latitude;47.4825126;longitude;-0.5484918999999999
EA_CB_u_bac_3;000d;2020;07;21;16;28;30;latitude;47.4825126;longitude;-0.5484918999999999
EA_CB_u_bac_4;000e;2020;07;21;16;28;30;latitude;47.4825126;longitude;-0.5484918999999999
EA_CB_u_bac_5;000f;2020;07;21;16;28;30;latitude;47.4825126;longitude;-0.5484918999999999
EA_CB_u_bac_6;0010;2020;07;21;16;28;30;latitude;47.4825126;longitude;-0.5484918999999999
EA_CB_u_bac_7;0011;2020;07;21;16;28;30;latitude;47.4825126;longitude;-0.5484918999999999
EA_CB_u_bac_8;0012;2020;07;21;16;28;30;latitude;47.4825126;longitude;-0.5484918999999999
EA_CB_u_bac_9;0013;2020;07;21;16;28;30;latitude;47.4825126;longitude;-0.5484918999999999

Figure 22: Exemple de mise en forme des valeurs dans un fichier texte

La mise en forme accomplie ; il reste à placer les variables dans un fichier texte. Cela permettra de les transférer vers la base de données.

Afin de placer ces valeurs dans un fichier texte, il faut garantir 2 choses :

- S'assurer que le chemin vers le fichier texte, existe (que le dossier où seront stockés les fichiers texte, existe).
- Générer un fichier texte avec le bon chemin, et écrire dans celui-ci.

Tout d'abord, je crée un dossier avec un chemin: « C://FTP ».

Une fois que ce dossier est créé et qu'il n'y a pas d'erreur de création, je génère un fichier texte dans lequel je vais écrire mes valeurs « mis-en-forme ». Les fichiers textes sont tous créés avec le même nom : ValeurIrigo (numéro du fichier). Un fichier texte est créé toutes les 10 secondes. J'ajoute à la fin de chaque nom, un numéro qui s'incrémente à chaque nouveau fichier (soit une incrémentation toutes les 10s).





	ValeurIrigo_3	21/07/2020 16:28	Document texte	4 Ko
	ValeurIrigo_2	21/07/2020 16:28	Document texte	18 Ko
	ValeurIrigo_1	21/07/2020 16:28	Document texte	18 Ko
	ValeurIrigo_0	21/07/2020 16:28	Document texte	18 Ko

Figure 23: Exemple de génération de fichier texte

Le fichier texte ValeurIrigo_3 présent à la figure 17 pèse moins lourd que les autres. En effet, il s'agit du fichier qui est en cours d'écriture par le serveur. Le serveur transfère les valeurs reçues vers le fichier texte, qui les ajoute à son contenu. Ainsi, ce dernier pèsera 18 ko lorsqu'il aura fini d'écrire à l'intérieur du fichier texte. Les fichiers sont prêts à être transférés vers la base de données.

```
C:\Windows\SYSTEM32\cmd.exe
ValeurIrigo_13.txt
ValeurIrigo_14.txt
ValeurIrigo_15.txt
ValeurIrigo_16.txt
ValeurIrigo_17.txt
ValeurIrigo_8.txt
ValeurIrigo_9.txt
1 fichier(s) copié(s).

C:\FTP>ftp -i -s:C:\FTP\FTP_POUR_SPECINOV\commande_ftp.cmd
ftp> rem open ftp.specinov.fr
Non connecté.
ftp> open ftp.specinov.fr
Connecté à ftp.specinov.fr.
220 Bienvenue sur le site FTP de SPECINOV
202 UTF8 mode is always enabled. No need to send this command.
Utilisateur (ftp.specinov.fr:(none)) :
331 Password required for irigo_pc_indus

230 Logged on
ftp> binary
200 Type set to I
ftp> cd /Time_Paris
550 CWD failed. "/Time_Paris": directory not found.
ftp> lcd C:\FTP
Dossier local maintenant C:\FTP.
ftp> mput *.csv
200 Port command successful
150 Opening data channel for file upload to server of "/tempo.csv"
```

Figure 24: Envoie des fichiers vers base de données

WatchDog - Descriptif

Afin de garantir le fonctionnement du serveur, il est nécessaire de mettre en place un Watchdog.

Un watchdog est un programme qui surveille d'autres applications, et vérifie qu'elles n'ont pas dysfonctionné. En effet, si pour une raison quelconque, le serveur ne fonctionne plus correctement (crash de l'application, bug à la conception qui bloque l'app, etc...) il est important d'en avoir connaissance afin de le relancer car son rôle est d'assurer le bon fonctionnement du serveur.

Le Watchdog consiste en un compteur qui s'incrémente toutes les secondes. Si ce compteur atteint une valeur définie (ici 30), alors le serveur a planté et doit être redémarré.

Pour ne pas atteindre cette valeur, le serveur la met à 0 toutes les 10 secondes. En revanche, si le serveur a crash ou est bloqué quelque part dans son programme, le compteur ne sera pas remis à 0.

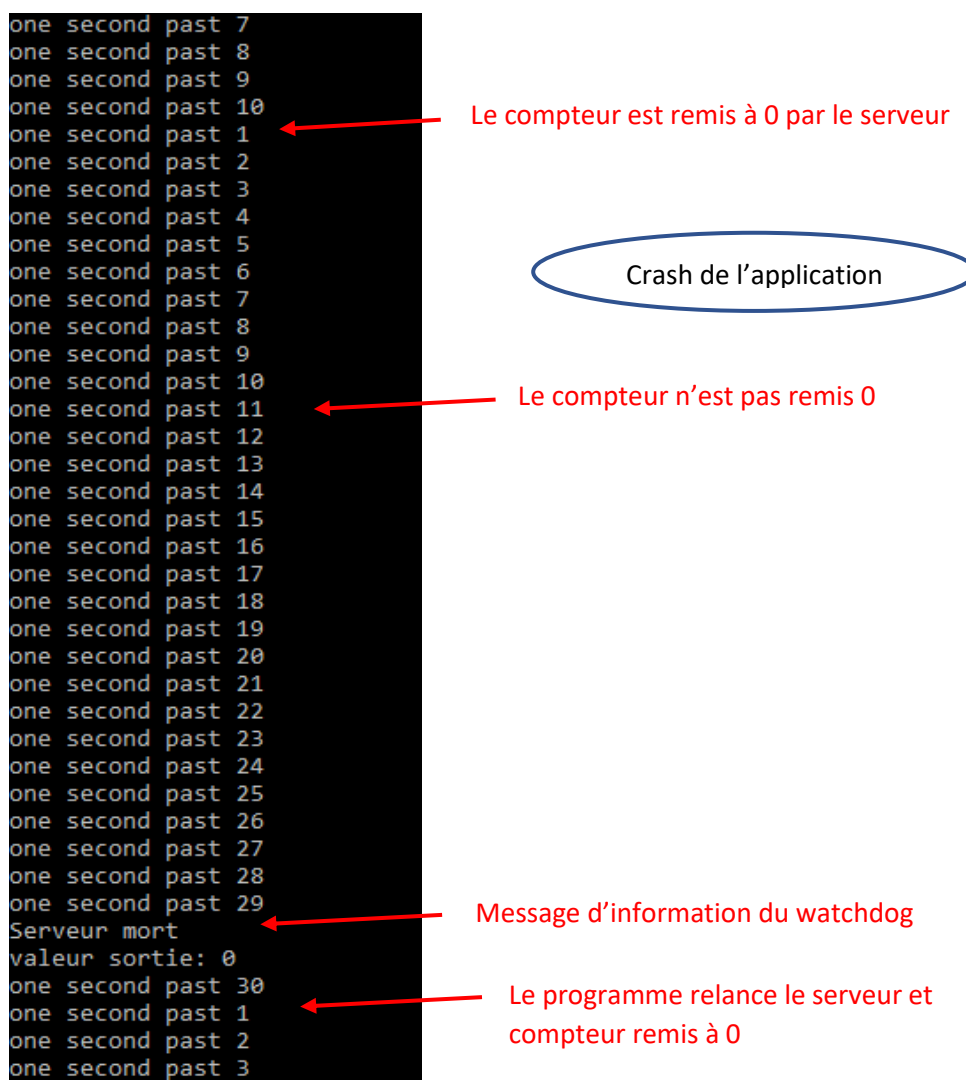


Figure 25: Démonstration du programme Watchdog

Le WatchDog est un programme indépendant du serveur. Il est lancé de la même manière que lui, grâce au Planificateur de tâches Windows. Les deux programmes sont déclenchés simultanément, 1 min après le lancement de Windows, afin de laisser à l'OS le temps d'initialiser les ports de communication.

WatchDog – Fonctionnement

Le programme écrit, chaque seconde, la valeur incrémentée du compteur dans un fichier texte.

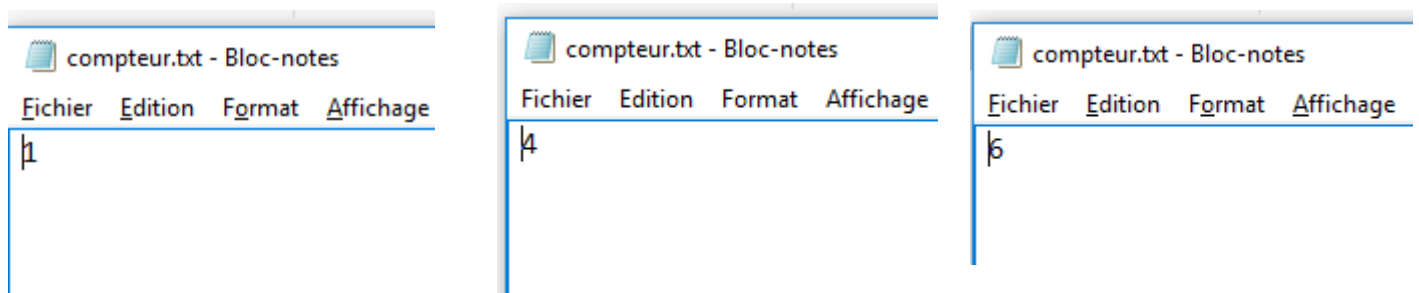


Figure 26: Le fichier texte qui s'incrémente chaque seconde en changeant la valeur

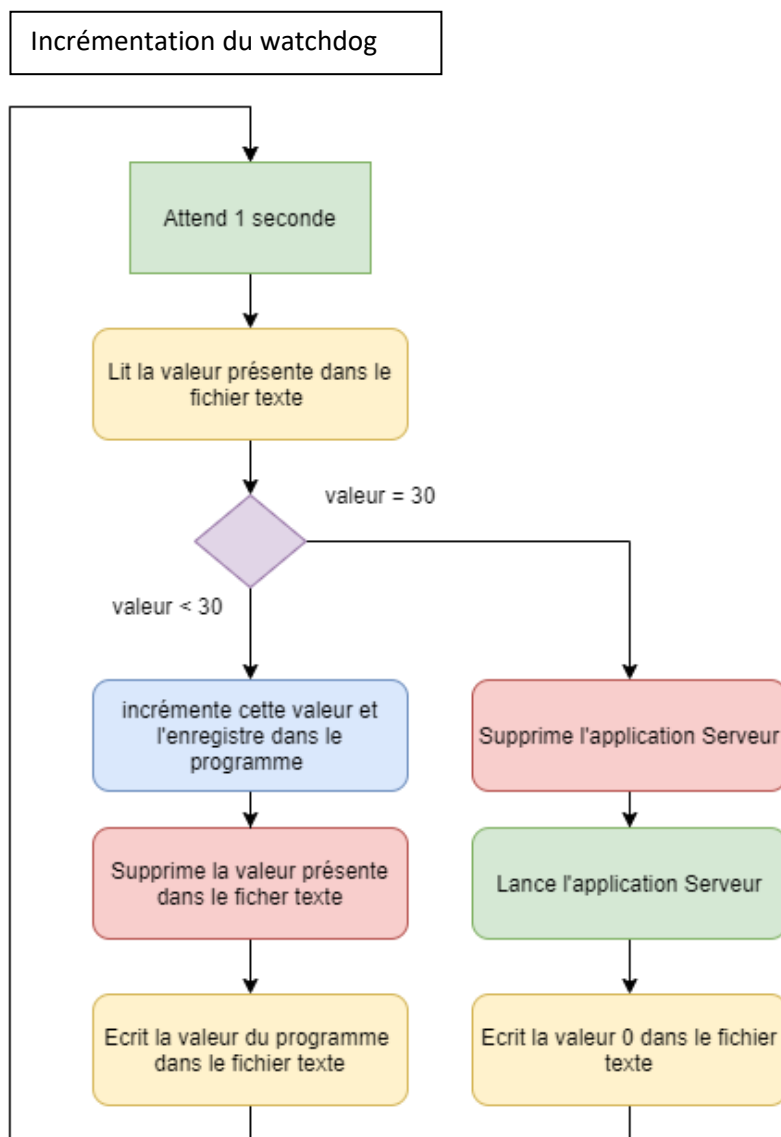


Figure 27: Fonctionnement de l'incrémentation

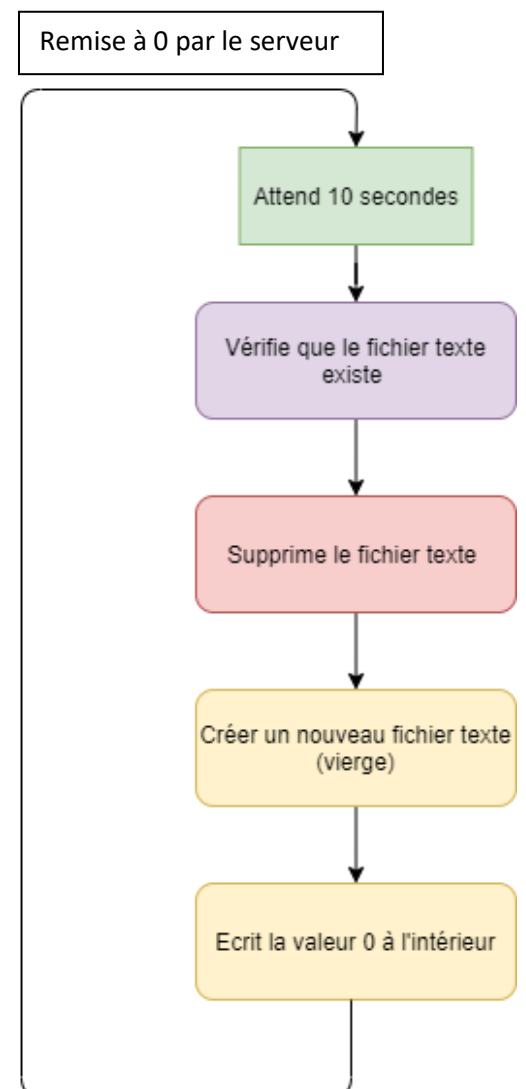


Figure 28: Fonctionnement de la remise à 0 du compteur par le serveur

Description projet - Partie communication

Réseau 4G & GPS

L'exploitation des données recueillies nécessite de les transférer vers une base distante afin de les conserver et analyser. Ces analyses permettront à terme d'éviter que de nouveaux incendies n'apparaissent, en prévenant les avaries avant qu'elles ne surviennent.

Pour ce transfert, la solution retenue utilise le réseau 4G, couramment utilisé par les téléphones mobiles : cette technologie a l'avantage d'être très simple à mettre en place et permet une connexion internet fiable. Elle s'appuie sur une carte SIM et une antenne capable de capter les ondes 4G.



Figure 29: Ports de connexions du PC, en vert : 2 emplacements pour carte SIM, en rouge : emplacement pour antenne

Ainsi, le PC dans le tramway, possède des emplacements pour carte SIM et pour accueillir des antennes. De plus, le système d'exploitation Windows professionnel dont nous disposons, fournit un paramétrage simple de cette technologie.

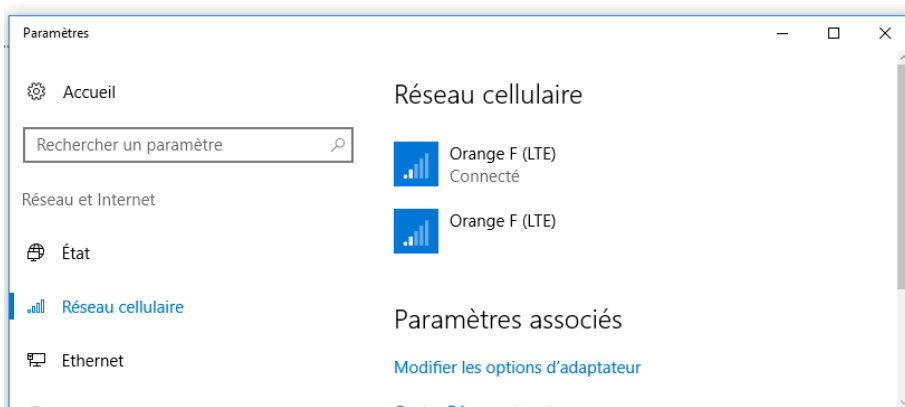


Figure 30 : Exemple d'outils 4G Windows

Par ailleurs, le cahier des charges impose la récupération de données GPS. Ces informations ne proviennent pas de l'automate mais d'une antenne GPS qui (à venir) indiquera la position exacte du tramway.

Le PC permet les communications 4G, mais il est également possible de brancher une antenne dédiée au GPS. Il dès lors nécessaire d'ouvrir le PC, et de modifier le câblage de la carte réseau car, il faut ajouter un câble entre la nouvelle antenne, et l'emplacement GPS de la carte réseau.



Figure 31 : Carte réseau du PC, en rouge : l'emplacement du slot pour GPS

A terme, cette antenne GPS me permettra de récupérer des trames au protocole NMEA (National Marine Electronics Association). Il s'agit du protocole standard de communication traitant des informations de position.

\$GPGGA,123519,4807.038,N,01131.324,E,1,08,0.9,545.4,M,46.9,M, , *42

Figure 32: Exemple de trame au protocole NMEA 0183

L'exemple présenté sur la figure numéro 20 met en évidence une trame NMEA. (Cf. Annexe 1, détail de cette trame).

Une fois ces informations obtenues, je pourrai les ajouter à la fin de chaque ligne de mes fichiers texte afin de spécifier la position de l'automate, et par conséquent, du tramway. Elles serviront ensuite à préciser la position du tramway sur la ligne.

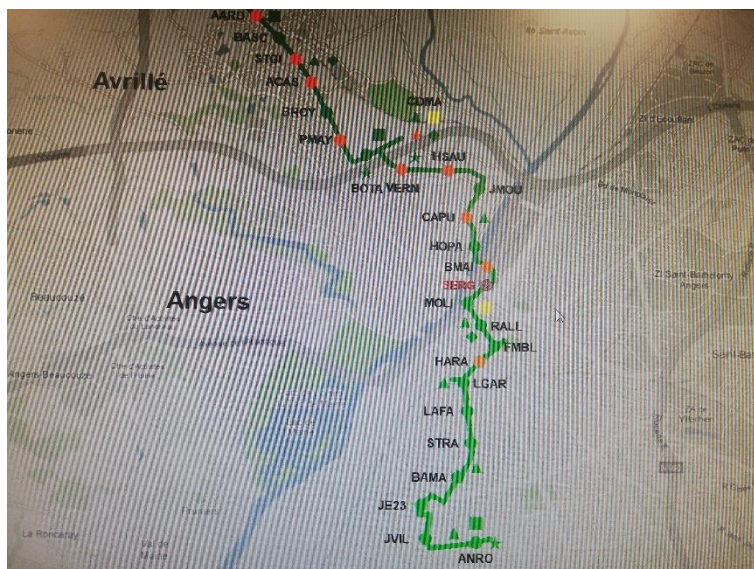


Figure 33: Ligne du tramway avec, à terme, la position du tramway d'indiqué

Serveur Specinov & Batch

Une fois les informations mises à disposition grâce à une connexion 4G, il est possible de les transférer vers un serveur externe géré par la société Specinov.

Specinov est une société (angevine) prestataire de l'entreprise. Elle est en charge du développement de certains logiciels de l'entreprise, mais également de deux serveurs. Cette volonté de faire appel à des prestataires, garantit à RATP Dev la sécurité des informations enregistrées en faisant appel à une équipe extérieure, et responsabilise une société tierce en cas de fuite d'informations.



Figure 34: Logo Specinov

Afin de transmettre les informations, un collègue me fournit un code « batch » me permettant de transférer efficacement des fichiers vers le serveur.

Le Batch est un langage de script. Il permet d'exécuter une suite d'instructions ayant un but précis et pouvant servir à l'automatisation de tâches (par exemple, renommer tous les fichiers d'un dossier d'un coup). Cette suite d'instructions est exécutée par l'interpréteur de commande Windows : la console (cmd.exe).

```
batch exemple 2 - Bloc-...
Fichier Edition Format Affichage Aide
@echo off
for /l %%b in (0, 1, 200) do (
set r=%%b+1
)
pause > nul
exit
```

Figure 35: Exemple de script Batch, compte de 1 à 200 et affiche la valeur à chaque incrémentation

Le script batch fournit, vient « récupérer » les fichiers texte générés, et les envoie sur le serveur grâce au protocole FTP. Il s'agit d'un protocole de communication dédié au partage de fichiers, il est donc tout à fait adapté au projet.

Son script est exécuté de manière récurrente toutes les minutes. Cela est rendu possible grâce aux planificateurs de tâches Windows. Cet outil développé par l'OS (Système d'exploitation) du même nom permet de déclencher des logiciels, des exécutables et même envoyer des mails. Ici il sert à lancer le script batch, qui est un exécutable.

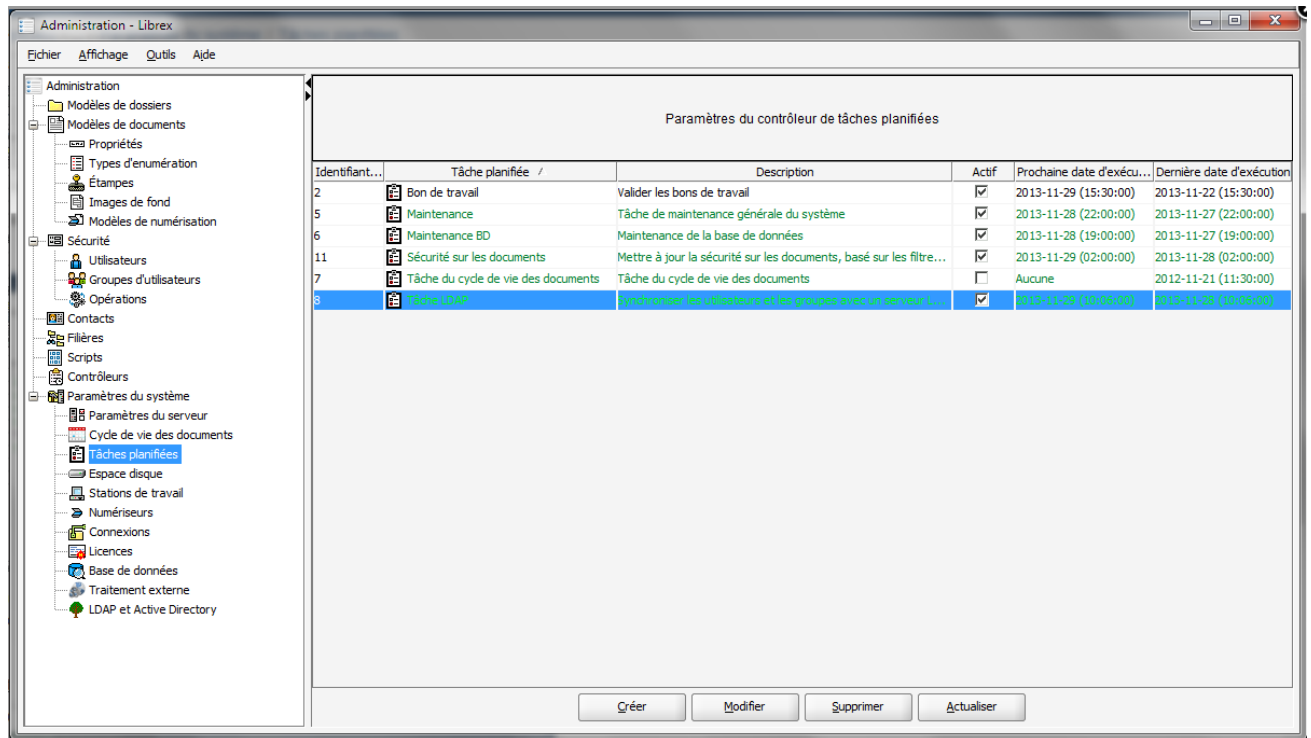


Figure 36 : Exemple de tâche planifié

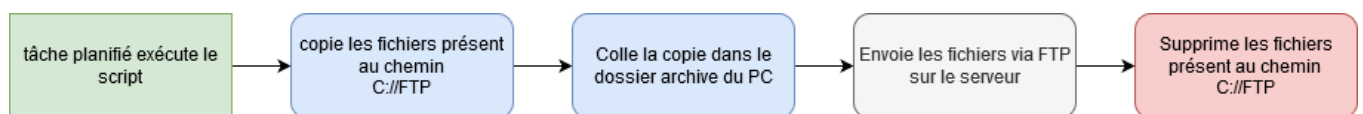


Figure 37: Exécution du script

A noter : Les tâches planifiées sont une composante essentielle du projet. En effet, elles permettent le transfert d'informations sur le serveur Specinov, mais elles servent également à initialiser le serveur TCP/IP. Pour cela, il suffit d'une seconde tâche qui lance l'exécutable du serveur au démarrage du PC.

Développement futur

IHM

L'IHM (ou interface-homme-machine), représente le support à travers lequel un utilisateur va interagir et contrôler une machine.

Il existe plusieurs catégorie d'IHM, qui se définissent par leur type d'interaction avec l'utilisateur :

- *Mode parlé* : commandes vocales, guides vocaux...
- *Mode écrit* : entrées par le clavier et la tablette graphique, affichage du texte sur l'écran...
- *Mode gestuel* : désignation 2D ou 3D (souris, gants de données , écran tactile), retour d'effort...
- *Mode visuel* : graphiques, images, animations...

Il s'agit ici d'un logiciel graphique.

Ce logiciel devra à terme fournir un rendu graphique complet et cohérent des valeurs, afin de les rendre accessibles aux opérateurs. De plus l'affichage devra être dynamique et permettre des interactions entre les différentes informations afin de les faire évoluer.



Figure 38 : Exemple d'IHM

Afin de réaliser ce logiciel, je compte utiliser le langage graphique QML (Qt Markup Language) de Qt creator (mon logiciel de programmation).

Les langages informatiques sont divisés en deux catégories :

- La programmation impérative, qui décrit *comment* le code s'exécute (le serveur et le WatchDog sont fait grâce au C++ notamment)
- La programmation déclarative, qui *déclare* les éléments qui vont composer l'interface. Le QML en est un.

Qt propose des outils tel quel les Charts (graphique en anglais) qui me permettront de générer des courbes, affichant les valeurs de l'automate en fonction du temps.



Figure 39: Exemple d'utilisation des Charts

Ces outils ont l'avantage d'être complet et de permettre différentes formes d'affichage, ainsi que des animations. De plus, ils ont l'avantage de facilement se lier avec le C++. Cela autorise une très grande liberté de création dans la mesure où chaque élément d'un graphique peut être modifié (par exemple la taille de ces graphs, leurs couleurs, les polices utilisées pour les valeurs, etc....)

Le C++ est un langage de bas niveau, c'est-à-dire un langage avec peu d'outils à disposition, mais avec la possibilité d'en créer et de les modifier facilement. Il s'oppose aux langages de haut niveau, comme le Python par exemple. Ces langages de haut niveau, fournissent des outils informatiques déjà existants qui facilitent le travail du développeur. En contrepartie, le développeur a moins de contrôle sur le code car les outils fournis ne peuvent être modifiés.

A noter : Habituellement, les langages de bas niveau sont impératifs. Ils permettent d'écrire un code précisément, et autorisent beaucoup de modifications. Les langages de plus haut niveau décrivent l'environnement dans lequel le code va être exécuté.

Ce logiciel graphique aura pour rôle principal d'afficher et de rendre disponible les informations, mais aura également un second rôle important : celui de communiquer des alertes.

En effet, parmi les valeurs rapportées de l'automate se trouveront des valeurs « prioritaires ». Ces valeurs pourront correspondre à des situations dangereuses ou bien des anomalies qu'il faut prévenir en priorité. Elles seront reçues de l'automate chaque seconde, et seront traitées en premier par le serveur.

Ces valeurs seront binaires (0 ou 1). Si l'une d'entre elles est à 1, alors l'alarme est active et doit être transmise à opérateur. Ainsi, lors d'une détection, le serveur déclenchera une tâche Windows. Celle-ci enverra un mail, contrairement au WatchDog qui exécute un programme.

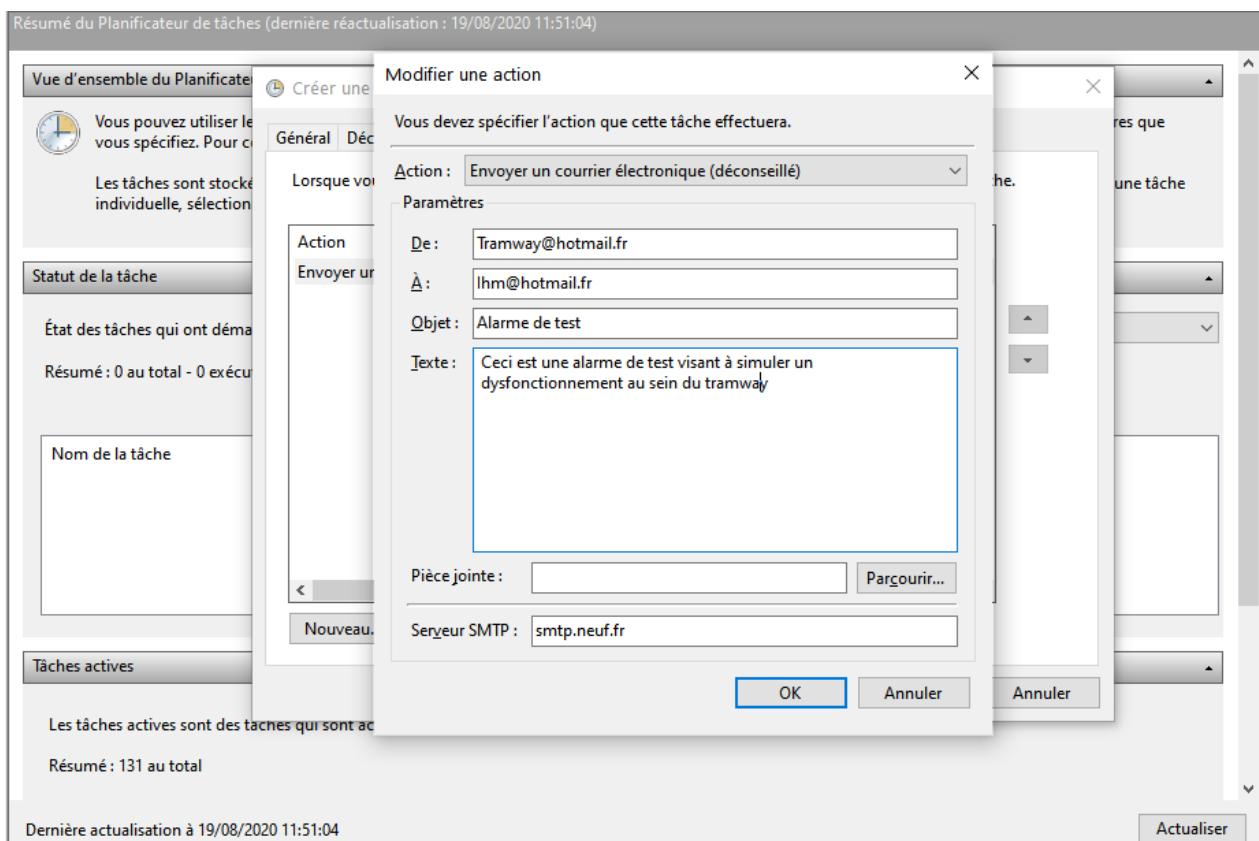


Figure 40: Tâche planifiée permettant l'envoi de mail

En plus d'un mail envoyé à un responsable, l'IHM sera chargée d'afficher l'alarme de manière ostensible afin de prévenir un maximum de personnes.

En parallèle d'un logiciel, le développement d'une application mobile est possible. Celle-ci serait en mesure de fournir des informations simples ainsi que de prévenir plus efficacement les alarmes à travers des notifications ou des sonneries

Développement futur – Analyse

Panne et Incident

Ce projet a pour objectif de récupérer différentes informations issues de l'automate. La nature de ces informations vise en priorité les valeurs de tensions placées sur les batteries de secours, mais cela concerne également toutes les informations que l'automate reçoit. Par exemple, l'automate est en mesure de savoir si les portes du tramway sont ouvertes ou fermées, car des capteurs sont placés sur celles-ci. Ainsi l'analyse des données va permettre 2 choses :

- Prévenir les pannes et incidents
- Réaliser de la maintenance prédictive

Pour prévenir des pannes, je vais devoir créer un programme vérifiant en permanence les valeurs reçues sur l'IHM. De sorte que si l'une des valeurs chute brutalement, une alarme se déclenche afin de prévenir les opérateurs. Ainsi seront détectées par l'automate sous forme de valeurs prioritaires les alarmes « classiques » et les variations de données trop importantes.

En plus de ce mécanisme de sécurité, il est envisageable de faire de l'analyse plus poussée, notamment en étudiant les évolutions de courbes sur de longues période.

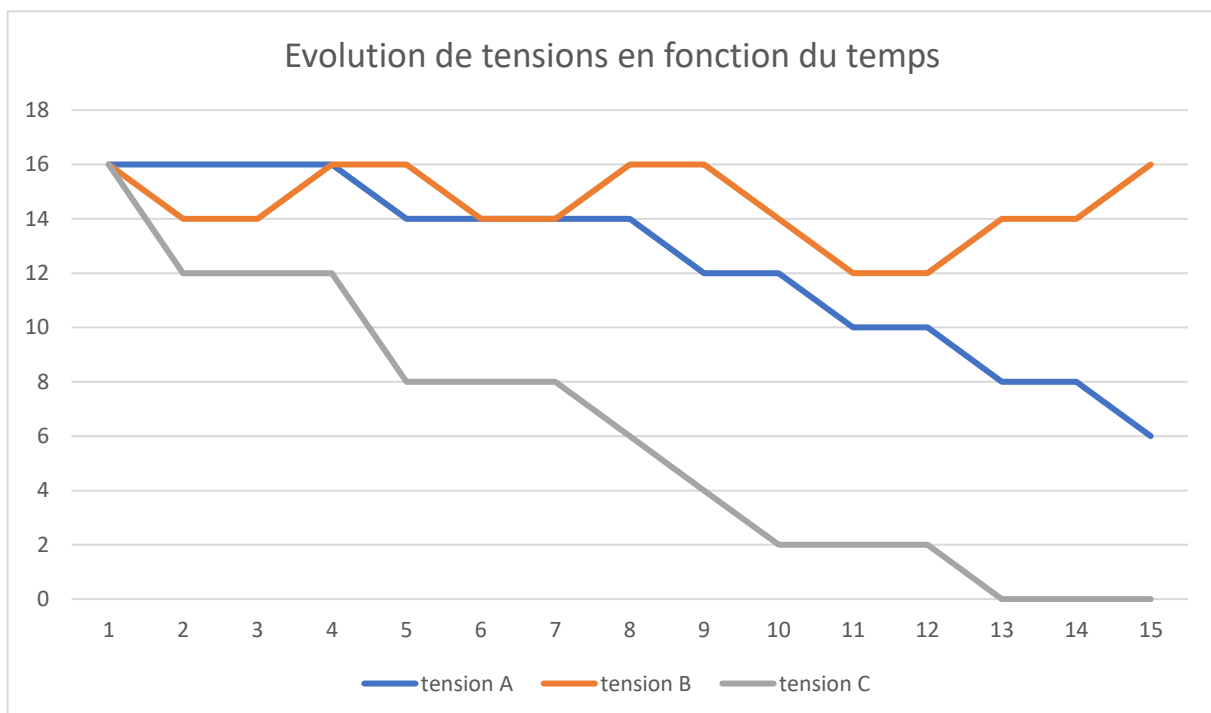


Figure 41 : Exemple de courbes

Sur la figure X, la courbe « tension B » est à une tension régulière. La « tension C » est en chute brutale de tension. Celle-ci doit déclencher une alarme. La « tension C » est en chute modérée, sur une plus longue période de temps. Ce genre de chute est plus complexe à identifier car ne doit pas être confondue avec une tension régulière qui varie.

Développement futur – Analyse

Maintenance Prédictive

L'analyse de données sur le long terme va permettre l'élaboration d'un stock d'informations conséquent, ouvrant la voie vers de la maintenance prédictive.

La maintenance prédictive (ou maintenance prévisionnelle) consiste pour un équipement à anticiper le franchissement d'un seuil jugé critique. L'objectif d'une telle maintenance est de remplacer un élément avant qu'il n'atteigne une période dysfonctionnement trop importante.

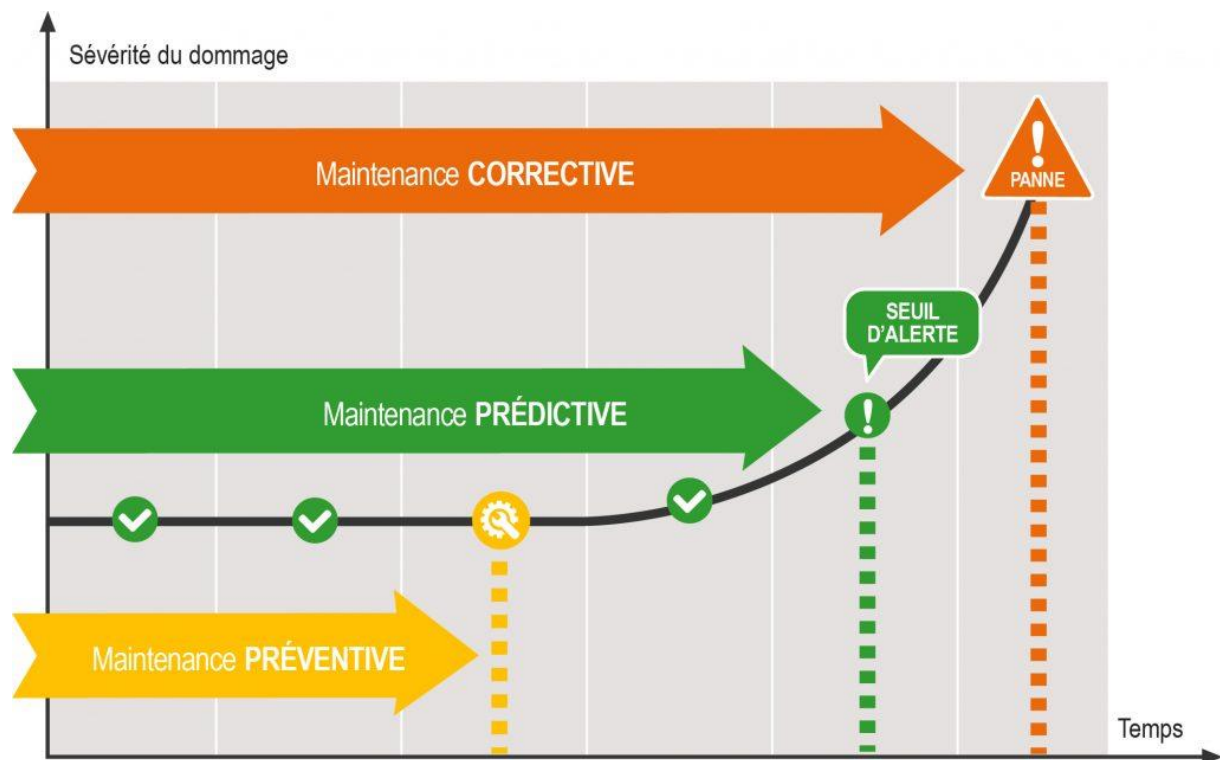


Figure 42: Types de maintenance possible

Il existe 3 types de maintenance :

- Préventive : maintenance faite selon les statistiques de durée de vie ou selon l'équipementier. Elle impose au matériel des vérifications régulières, car il n'a pas de vérification en continu de l'état des composants. De plus, l'équipement est régulièrement mis à l'arrêt afin de prévenir les pannes.
- Prédictive : maintenance réalisée grâce aux capteurs positionnés sur le matériel. Cela permet une surveillance en continu et l'établissement d'un calendrier des pannes les plus probables. De fait les opérations de maintenances sont moins nombreuses et sont précisément planifiées.
- Corrective : consiste à remplacer un équipement défectueux.

Conclusion

Cette année d'alternance, riche de nouvelles connaissances, m'a permis de mettre en pratique des compétences acquises durant ma formation. La partie Acquisition, la plus complexe et longue à mettre en œuvre ; m'a permis d'aborder beaucoup de connaissances sur les serveurs, sur leurs mises en place et leurs fonctionnements. Travailler en au sein d'une équipe à taille humaine, m'a laissé une grande autonomie dans mes choix et solution retenus pour le projet. J'ai ainsi pu réaliser les étapes de mon projet dans l'ordre que je souhaitais tout en répondant aux problématiques soulevées par RATP dev. (J'ai par exemple décidé laisser en plan l'étude de la documentation Modbus, au profit de l'enregistrement de valeur, voir Gantt).

Par ailleurs cette alternance m'a permis de toucher à de nombreux domaines du monde informatique, en abordant des thématiques de communication longue distance, l'enregistrement de valeurs en temps réel ou encore sur les méthodes pour rendre une information accessible à un opérateur grâce à un IHM. De plus, j'ai eu à de multiples reprises, l'occasion de fournir mon expertise sur un problème ainsi que sur les manières de résoudre (ou de contourner) ce problème, ce qui m'a permis de me sentir valorisé, ainsi que de voir mon travail être utilisé et évolué.

En somme, je souhaite continuer ce projet afin d'en achever la seconde partie et conclure complètement le système de récupération de données, dans l'optique de le rendre opérationnel, efficace et sécurisé.

Annexe

Tables des matières

Figure 1 : photo du métro en 1957	6
Figure 2 : schéma des relations contractuelles	7
Figure 3 : offre de services RATP Dev	8
Figure 4 : organigramme RATP Dev	9
Figure 5: les marques du groupe RATP	10
Figure 6: présence de RATP dans le monde	11
Figure 7: un bus électrique avec l'inscription « ligne 100% électrique »	12
Figure 8: les implantations de RATP Dev dans l'Ouest.....	13
Figure 9: tramway angevin	14
Figure 10 : automate industriel	15
Figure 11: batteries de secours	15
Figure 12: Schématisation étapes du projet	16
Figure 13: Communication entre serveur et client	18
Figure 14 : PC embarquée	18
Figure 15: Schéma Maître - Esclave.....	19
Figure 16: Exemple de trame requête.....	19
Figure 17: Définition socket	20
Figure 18: Exemple de création d'un serveur	20
Figure 19: Exemple de communication Modbus entres automates	21
Figure 20: Exemple de trame complète et séparée	22
Figure 21: Liste de noms de variables	23
Figure 22: Exemple de mise en forme des valeurs dans un fichier texte.....	23
Figure 23: Exemple de génération de fichier texte	24
Figure 24: Envoie des fichiers vers base de données.....	24
Figure 25: Démonstration du programme Watchdog.....	25
Figure 26: Le fichier texte qui s'incrémente chaque seconde en changeant la valeur	26
Figure 27: Fonctionnement de l'incrémentation	26
Figure 28: Fonctionnement de la remise à 0 du compteur par le serveur.....	26
Figure 29: Ports de connexions du PC, en vert : 2 emplacements pour carte SIM, en rouge : emplacement pour antenne.....	27
Figure 30 : Exemple d'outils 4G Windows.....	27
Figure 31 : Carte réseau du PC, en rouge : l'emplacement du slot pour GPS.....	28
Figure 32: Exemple de trame au protocole NMEA 0183.....	28
Figure 33: Ligne du tramway avec, à terme, la position du tramway d'indiqué.....	28
Figure 34: Logo Specinov.....	29
Figure 35: Exemple de script Batch compte de 1 à 200, affiche la valeur à chaque incrémentation....	29
Figure 36 : Exemple de tâche planifié	30
Figure 37: Exécution du script	30
Figure 38 : Exemple d'IHM	31
Figure 39: Exemple d'utilisation des Charts	32
Figure 40: Tâche planifiée permettant l'envoi de mail	33
Figure 41 : Exemple de courbes	34
Figure 42: Types de maintenance possible	35

Lien GitHub

L'ensemble de mes annexes se trouvent à ce lien : <https://github.com/Valere21/ServeurIrigo.git>

Il s'agit d'un répertoire Git permettant d'héberger des fichiers (principalement texte). C'est un outil pensé pour les développeurs car il permet des transferts rapidement sur internet en passant par des lignes de commandes plutôt que par un site internet.

Ce répertoire est divisé en « branche », chacune d'entre elles contient une partie des annexes.

Branche Serveur Irigo : Ensemble des différents codes ayant servi à la création du serveur. (Le fichier mainwindow.h est commenté et apporte des explications claires) :

Branche WatchDog : : Ensemble des différents codes ayant servi à la création du programme WatchDog

Branche Journal de bord : Contient mon journal de bord, m'ayant principalement aidé à la rédaction du diagramme de Gantt, il n'en reste pas moins complet (environs 10 pages).

Branche Ressource et Documentations : Contient la plupart des manuel et sources de documentation qui m'ont servi.

Aide sur les différents type de maintenance : <https://www.mobility-work.com/fr/blog/maintenance-predictive-vs-maintenance-preventive-strategie-entreprise>.

StackOverflow: <https://stackoverflow.com/> - Site d'informatique mondialement connu, m'ayant apporté une aide constante sur l'ensemble de l'année.

Developez-net : <https://www.developez.net/forums/> Autre site d'informatique/forum m'ayant beaucoup aidé

Open Classroom : <https://openclassrooms.com/> Site et forum d'informatique – Référence française en matière d'informatique

La liste serait très longue mais il me semble important de citer ces 3 forums d'informatiques, qui ont été mes principales sources de solutions.