



KENENS Dorian
Promotion 2017
Année universitaire 2016-2017

Diplôme d'ingénieur Telecom Physique Strasbourg

Mémoire de stage de 3^{ème} année

« Etude et essais d'une solution de rétro-vision avec des caméras IP »



ALSTOM Transport SA
6 Route de Strasbourg
67110 Reichshoffen

SECHET Jean Philippe
jean-philippe.sechet@transport.alstom.com
27/02/2017-31/08/2017

Résumé

« Etude et essais d'une solution de rétro-vision avec des caméras IP »

Travaillant au sein de la gamme « Coradia Export » dont sont issus, entre autres, les projets « Algérie » et « Dakar », j'ai contribué à l'amélioration du système de rétrovision actuellement mis en place. Ce dernier nécessite la présence d'un parc matériel conséquent et spécifique à cette utilisation, de plus, les caméras utilisées sont analogiques et leurs données doivent être converties pour être émises sur le réseau embarqué du train. L'objectif de mon projet était donc d'étudier la faisabilité d'une solution entièrement numérique basée sur des caméras IP qui émettraient directement sur le réseau ethernet embarqué du train en étant connectées sur des routeurs. Il s'agissait aussi de proposer une solution concrète qui pouvait être directement expérimentable dans un laboratoire avec équipements réels.

Après avoir élaboré le cahier des charges du système en se basant sur celui existant, j'ai procédé à l'étude des différentes solutions envisageables. Cela incluait la création de plusieurs applications de test afin de retenir une solution optimisée en termes d'ergonomie et de cout d'intégration. Une fois cette dernière obtenue, qui a été l'intégration dans un écran déjà existant sur le poste de conduite, j'ai créé une sous-application complète, prête à être intégrée dans la gamme existante. Puis, de nombreuses adaptations et corrections ont été effectuées après concertation avec mon maitre de stage, lors des phases de simulation sur ordinateur. Enfin, la dernière étape de ce projet fût les phases de test en conditions semblables au réel sur un laboratoire contenant les systèmes informatiques du train. S'en suivi alors une validation de la solution par mon maitre de stage.

Abstract

“Studies and tests of a rearview solution using IP cameras”

Working on the product line “Coradia Export” which includes, among others, the “Algerie” and “Dakar” projects, I contributed to the improvement of the rearview system which is currently set up. It needs a significant amount of equipment specially dedicated to this task; moreover, the cameras currently used are analog and their data have to be converted for the transmission on the train embedded network. The aim of my project was the study of the feasibility for a completely numerical solution based on IP cameras which would directly transmit on the embedded Ethernet network of the train by being connected on routers. It also consisted of giving a concrete solution which could be directly tested inside of a laboratory with real equipment.

After formulating my system requirement specifications based on the current one, I have begun with studies about all the different feasible solutions. It included the creation of several test applications in order to choose a solution optimized in terms of integration costs and ergonomics. Once this choice has been made, which was the integration inside of a screen already in use on the cabin, I have created a complete sub-application ready to be put on the screen. Then, several modifications and arrangements suggested by my tutor have been implemented during the simulations on computer. Finally, the last step of my project has been the huge amount of tests on a laboratory containing all the embedded systems on conditions which are close to real train. Then, my tutor has validated my solution.

Remerciements

Je tiens à remercier particulièrement mon tuteur de stage Jean-Philippe Sechet qui a su se rendre disponible pour répondre à mes nombreuses questions lors de mon projet et pour m'avoir laissé une certaine autonomie.

Je tiens aussi à remercier toutes les personnes du Laborame qui ont répondu avec patience à mes questions ainsi que m'avoir aidé à mettre en place le matériel nécessaire lors de mes phases de tests.

Enfin, je remercie aussi les équipes en charge du développement sur la console ACE qui m'ont aidé lors de la configuration et la mise en place de mon code.

Sommaire

RESUME.....	I
ABSTRACT.....	II
REMERCIEMENTS.....	III
SOMMAIRE	IV
TERMINOLOGIE	V
LISTE DES FIGURES & TABLEAUX.....	VI
INTRODUCTION	1
I. ALSTOM	2
1. HISTORIQUE DU GROUPE	2
2. SITUATION ACTUELLE DU GROUPE	3
i. <i>Activité du groupe</i>	3
ii. <i>Solutions proposées</i>	4
iii. <i>Le site de Reichshoffen</i>	8
II. MON PROJET	11
1. CONTEXTE ET MATERIEL/LOGICIEL UTILISES.....	11
2. DEFINITION DU CAHIER DES CHARGES (CDC).....	12
i. <i>Grandes étapes du projet</i>	12
ii. <i>Contraintes sur l'affichage des images de rétrovision</i>	13
iii. <i>Gestion des configurations train</i>	14
iv. <i>Autres fonctionnalités demandées</i>	14
3. POINT DE VUE SYSTEME DU TRAIN	15
i. <i>Présentation du système actuel</i>	15
ii. <i>Améliorations potentielles communes aux deux écrans</i>	16
iii. <i>Améliorations propres à la solution petit écran type ACE</i>	16
iv. <i>Conclusion – système</i>	17
4. SELECTION DE L'ÉCRAN DE DEVELOPPEMENT DE L'APPLICATION	18
i. <i>Le Laborame PRASA</i>	18
ii. <i>Développement application grand écran</i>	19
iii. <i>Développement application petit écran</i>	20
iv. <i>Tests sur caméras réelles</i>	21
v. <i>Tests sur caméras « simulées »</i>	23
vi. <i>Conclusion – choix de l'écran</i>	24
5. APPLICATION CONSOLE ACE.....	25
i. <i>Présentation & règles de codage de l'application</i>	25
ii. <i>Face arrière de la partie surveillance</i>	27
iii. <i>Face avant de la partie surveillance</i>	29
iv. <i>Face arrière de la partie rétrovision</i>	31
v. <i>Face avant de la partie de rétrovision</i>	34
vi. <i>Phase d'intégration Laborame</i>	35
6. COMPARAISON AVEC LE SYSTEME ACTUEL	38
CONCLUSION	40
BIBLIOGRAPHIE & WEBOGRAPHIE	41
ANNEXES	42

Terminologie

Cabine :	poste de conduite, généralement 2 par élément.
CRS :	Constant Ring Switch (switch du réseau ethernet embarqué)
DDU :	Driver Display Unit, écran de visualisation des différents états des éléments du train (climatisation, caméras de surveillance, etc).
Elément :	composition de plusieurs véhicules couplés entre eux.
IHM :	Interface Homme Machine.
Laborame :	laboratoire composé du matériel informatique embarqué d'un train.
Miniature :	référence aux flux provenant des caméras lorsqu'ils sont tous affichés au même moment. Peut aussi faire référence à la miniature train.
MPU :	Main Process Unit (calculateur principal du train).
MTBF :	Mean Time Between Failure (temps moyen entre deux pannes), moyen statistique de caractériser la fiabilité d'un système (peut atteindre facilement des centaines de milliers d'heures).
PPI :	Pixel Per Inch (unité communément utilisée pour comparer des résolutions d'écran).
PPx :	Porteur Polyvalent (type PPG : 6 véhicules, PPM : 4 véhicules & PPP : 3 véhicules).
RTSP :	Real Time Stream Protocol (protocole de communication sur un réseau).
SSH :	Secure SHell (protocole permettant la connexion à un ordinateur à distance via un terminal et ce de manière sécurisée).
TCMS :	Train Control Monitoring System (Système de commande et de contrôle du train).
TCP :	Transmission Control Protocol (protocole de transmission de données avec garantie de réception).
TRS :	Train Router Switch (routeur du réseau ethernet embarqué).
UDP :	User Datagram Protocol (protocole de transmission de données sans garantie de réception).
Unité multiple (UM) :	composition de plusieurs éléments (UM2 et UM3 : 2 et 3 éléments).
Unité simple (US) :	composition d'un seul élément.
Véhicule (Veh) :	communément appelé wagon, lieu où se trouvent les voyageurs.
Visualisation :	terme utilisé pour le codage de l'application, il s'agit des différents écrans de l'IHM qui sont associés à des fonctionnalités propres (par ex : visualisation DOOR qui correspond à la fonctionnalité de supervision de l'état des portes).

Liste des figures & tableaux

Figure 1: Train grande vitesse commandé par Amtrak	2
Figure 2: Directives « Alstom 2020 »	3
Figure 3: Gamme Citadis (gauche) et Metropolis (droite)	4
Figure 4: Bus Aptis (gauche) et diminution encombrement arrêt (droite)	5
Figure 5: Coradia Nordic en Suède (gauche) et Coradia Polyvalent en France (droite)	6
Figure 6: Répartition du personnel Reichshoffen	8
Figure 7: Vue aérienne site Reichshoffen	8
Figure 8: Organigramme ingénieur Reichshoffen	9
Figure 9: Hiérarchie directe	10
Figure 10: Exemple de disposition des caméras de rétrovision	12
Figure 11: Exemple de configuration train	14
Figure 12: Schéma intégration système rétrovision actuel	15
Figure 13: Schéma intégration nouveau système rétrovision	16
Figure 14: Modification structurelle cabine	17
Figure 15: Pupitre cabine conducteur Algérie & Dakar	18
Figure 16: Application basique grand écran	19
Figure 17: Application basique petit écran simulé	20
Figure 18: Caméra Moxa (gauche) et page web de configuration (droite)	21
Figure 19: Charge CPU, application grand écran	22
Figure 20: Visualisation petit écran (coupée), caméras "simulées"	23
Figure 21: Comparaison grand et petit écran	23
Figure 22: Structure application "Coradia Export" (gauche) & extrait dossier Screens (droite)	25
Figure 23: Langage texte structuré (gauche) et langage par blocs (droite)	26
Figure 24: Etapes lors de l'appui sur la touche "Manu"	28
Figure 25: Face avant IHM - partie surveillance	29
Figure 26: Sortie de la fonction CONF_TRAIN	32
Figure 27: Face avant IHM - partie rétrovision	34
Figure 28: Charge CPU - Rétrovision	36
Figure 29: Charge CPU - Rétrovision automatique	36
Figure 30: Charge réseau entrante sur ACE	37
Figure 31: Comparaison zoom système actuel (gauche) et système ACE (droite)	39
Figure 32: Comparaison miniatures système actuel (gauche) et système ACE (droite)	39
Tableau 1: Cahier des charges	13
Tableau 2: Comparaison des deux écrans	24
Tableau 3: Comparaison performance application & cahier des charges	35
Tableau 4: Comparaison CDC - système existant - nouveau système	38

Introduction

Dans le cadre de la validation de mon diplôme d'ingénieur spécialisé dans les systèmes, automatique et vision, j'ai choisi d'effectuer mon stage en entreprise chez ALSTOM sur le site de Reichshoffen en Alsace par intérêt des systèmes embarqués dans les trains. J'ai eu l'opportunité de travailler sur les projets rattachés à la gamme Cordia Export (qui sera présenté dans ce rapport) et plus précisément sur l'amélioration du système de rétrovision actuellement mis en place. Ces dernières sont l'équivalent plus évolué des rétroviseurs classiques à miroir et servent principalement à surveiller les portes d'accès au train.

Dans un premier temps, vous trouverez la présentation du groupe ALSTOM ainsi que du site de Reichshoffen, en passant en revue les produits proposés par l'entreprise. Ensuite, la seconde partie concernera mon projet en lui-même, avec une présentation du contexte du projet suivi de la définition du cahier des charges à suivre. S'en suivra les changements possibles, d'un point de vue système, avec le nouveau mode de fonctionnement des caméras de rétrovision. Puis les sous-parties suivantes porteront sur la création de la nouvelle application de visualisation des caméras ainsi que sur le choix de l'écran et des phases de tests dans un environnement proche d'un train réel. Cette partie se terminera par la comparaison portant sur différents critères avec le système actuel.

I. Alstom

1. Historique du groupe

Entreprise mondialement connue pour son matériel ferroviaire en opération dans de nombreux pays, Alstom a connu une histoire mouvementée.

Sa création débute par la naissance de l'entreprise SACM - Société Alsacienne de Constructions Mécaniques - en 1839, puis, de la fusion avec la société d'électricité américaine Thomson Houston en 1928, naît alors le groupe Alsthom. L'entreprise est alors spécialisée dans le domaine électrique avec la construction de transformateurs et diverses machines électriques.

Identité confirmée par la fusion avec l'atelier Constructions électriques de France en 1932 puis par le rapprochement avec la branche Power Systems du groupe General Electric Compagny en 1989 et avec la Compagnie Générale d'électricité pour former ainsi GEC-Alsthom en 1991. S'en suit alors la fusion avec une autre filiale de la Compagnie Générale d'électricité, Alcatel, le groupe se renomme alors en Alcatel-Alsthom.

Cependant, sept ans plus tard, les deux groupes deviennent indépendants et se renomment en Alcatel et Alstom. Cette même année, Alstom rachète au groupe De Dietrich le site de Reichshoffen, lieu où s'est déroulé mon projet de fin d'études.

Puis, à partir de 2005, le groupe commence petit à petit à céder ses parts dans le marché de l'énergie, avec la vente de son activité Power Conversion à Barclays Private Equity par exemple. Cessions qui se terminent en 2014 par la vente, fortement médiatisée, des parties énergies restantes à la firme américaine General Electric. Le groupe se concentre alors sur ses activités de transport, avec, par exemple l'imposant contrat passé aux Etats Unis avec Amtrak portant sur la construction de 28 trains à grande vitesse pour un montant de deux milliards de dollars.



Figure 1: Train grande vitesse commandé par Amtrak

2. Situation actuelle du groupe

i. Activité du groupe

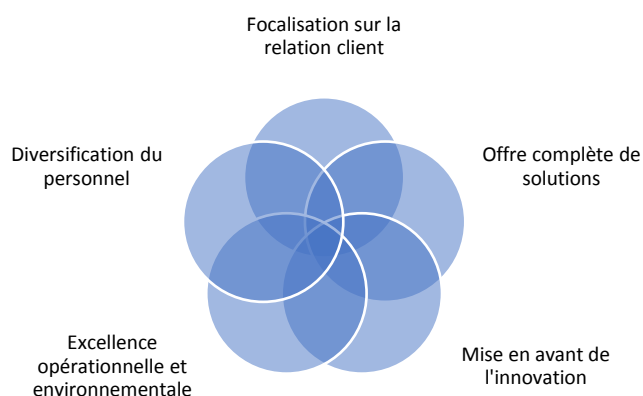
Sur l'année fiscale 2016/2017, Alstom a dégagé un chiffre d'affaire de 7.3 milliards d'euros contre une dette de 208 millions d'euros, deux données en évolution positive par rapport à l'année fiscale précédente.

Sans surprise, 43% des ventes du groupe sont concentrées sur le matériel ferroviaire roulant, c'est-à-dire les trains, trams, locomotives, etc. Le reste de l'activité est subdivisé en trois parties ; 20% concerne les services – maintenances, modernisations, réparations, etc. –, 19% concerne la signalétique – postes de contrôle, solutions de signalisation sur les voies, etc. – et enfin 18% concerne les systèmes intégrés et infrastructures.

Sur cette année fiscale, 10 milliards d'euros de commandes ont été placées, parmi elles se trouvent la commande de 17 trains Coradia (cf. **ii.Solutions proposées par le groupe**) par la SNCF, la commande de 8 trains pendulaires à grande vitesse pour NTV, compagnie ferroviaire italienne, l'achat de 800 locomotives électriques en Inde ou encore la modernisation de trains pour la compagnie canadienne Rocky Mountaineer. Au total, l'Europe représente à elle seule 51% chacun des commandes passées, 29% pour les Amériques, 14% pour le Moyen Orient et l'Afrique et 6% pour l'Asie. Cela nous démontre alors clairement l'ouverture internationale du groupe.

En effet, sur les 32800 employés que compte la firme, 20700 d'entre eux se trouvent en Europe, 5200 aux Amériques, 4000 en Asie et 2900 en Afrique. Cela représente 105 sites répartis dans 60 pays dans le monde. Remarquons, par exemple, la construction du site de 60000m² en Afrique du Sud, destiné à la production des 580 X'Trapolis Mega (cf. **ii.Solutions proposées par le groupe**) commandés par le pays. Mais encore le projet liant Alstom et l'entreprise ferroviaire russe TMH sur la construction de locomotives de fret et de passagers pour le Kazakhstan ainsi que l'entreprise KEP créée conjointement par Alstom et Kamkor, firme kazakhe, ayant pour objectif la production d'équipement ferroviaire de signalisation dans le pays. Citons encore Madhepura Electric Locomotive Private Limited, entreprise née d'un partenariat avec la compagnie ferroviaire indienne Indian Railways, qui est en charge de la production de 800 locomotives électriques ainsi que de la maintenance associée avec des livraisons s'étalant sur 10 ans, de 2018 à 2028.

Alstom possède ainsi un réseau très important de partenaires, entreprises et projets à travers le monde ce qui permet alors à la firme d'être présente sur tous les continents et de répondre à une demande quelle qu'en soit sa provenance.



L'entreprise propose aussi un ensemble d'idées directrices pour le futur proche du groupe. Nommé « Alstom 2020 », le document pose cinq piliers de la stratégie future d'Alstom, présentés dans le digramme ci-contre.

Figure 2: Directives « Alstom 2020 »

ii. Solutions proposées

Dans la continuité de la partie précédente, la plupart des solutions offertes par Alstom concernent le matériel ferroviaire roulant. Il est possible de scinder ce matériel en deux parties ; la partie urbaine et la partie grandes lignes. Puis une dernière partie concernant les autres solutions proposées par le groupe (signalétique, gestion, etc) sera présentée.

Concernant cette première partie, les tramways Citadis sont certainement les plus connus du grand public. En effet, ces derniers équipent de nombreuses grandes villes françaises (Paris, Bordeaux, Nice, Strasbourg, etc) mais aussi mondiales (Nottingham, Casablanca, etc) portant ainsi le nombre total de Citadis en circulation à 1600 (2016). Pour relativiser ce nombre, un tramway sur quatre, à plancher bas, dans le monde, provient de la gamme.

Ce qui fait donc la force de cette solution c'est sa disponibilité, rendue possible par la standardisation des systèmes, impliquant alors des coûts de production et de maintenance réduits. Cependant, la gamme Citadis n'en reste pas moins personnalisable ; en effet, il est possible par exemple de varier la taille d'une rame de 24m à 45m, faisant donc passer le nombre de passagers transportables de 142 à 341 (à raison de 4 passagers / m² + places assises). Il est aussi possible d'en varier l'aspect extérieur afin de s'adapter au matériel roulant existant dans la ville en personnalisant le nez du train et sa peinture, mais aussi l'intérieur en y intégrant différentes options à destination des voyageurs.

La motorisation est assurée par un moteur à aimant permanent, assurant à la rame une vitesse maximale de 70 à 80km/h selon options ainsi qu'une accélération maximale de 1.3m/s².

Toujours dans la catégorie urbaine, mais cependant moins visible, la gamme Metropolis reste très présente dans le monde pour les solutions de métro. En effet, 17000 rames sont actuellement en service pour plus de 50 clients, par exemple Amsterdam, Buenos Aires, Singapour, Sydney, etc. Parmi ces rames se trouvent 360 rames automatiques – sans conducteurs -, avec comme ville pionnière Lille qui a intégré ce système la première dans les années 80.

Dans le même ordre d'idée que les tramways de la marque, Metropolis est une gamme éprouvée et dont les options sont nombreuses afin de s'adapter aux différents marchés. On peut alors y retrouver des services voyageurs à jour des technologies sur le marché comme des ports de charge USB ou encore le WIFI à bord. Concernant la configuration matériel, chaque métro peut aller de 18 à 24m de long, avec la possibilité d'en coupler jusqu'à 6, pour maximiser le nombre de passagers transportables (jusqu'à 1800 à 6 passagers / m²).



Figure 3: Gamme Citadis (gauche) et Metropolis (droite)

Les tramways sur pneus Translohr d'Alstom et NTL (filiale d'Alstom) sont encore une preuve de la grande capacité d'adaptation de la firme. En effet, ces derniers sont à mi-chemin entre le tramway et le bus ; ils sont guidés par un rail unique mais restent supportés par des roues. Cette solution intermédiaire permet de limiter les potentielles lourdes infrastructures liées au tramway « classique » tout en ayant une plus grande capacité et une meilleure manœuvrabilité que le bus (10.5m de rayon de braquage et 6.7m de gabarit contre 11m et 12m pour des bus de ville). Les pneus de ce véhicule lui autorise à gravir des pentes jusqu'à 13% en plus d'être plus silencieux.

D'une taille pouvant aller de 25 à 46m et d'une capacité de 178 à 358 passagers, il se distingue principalement par une largeur de 2.2m seulement. A ce jour (2016), plus de 120 tramways Translohr sont en activité dans le monde, en allant de Paris à Padua (Italie) jusqu'en Chine à Shanghai par exemple.

De manière beaucoup plus récente (2017), le dernier né de la collaboration avec NTL est un bus entièrement électrique, nommé Aptis. Ce dernier se distingue des autres bus en circulation par ses 4 roues directrices afin de limiter son encombrement en courbe mais aussi de limiter l'espace nécessaire entourant un arrêt (cf. Figure 4). D'une capacité de 95 passagers pour 12m de long et 2.55m de large, cette solution peut être mise en place dans des villes souhaitant s'affranchir des infrastructures du tramway tout en ayant une partie de ses avantages.

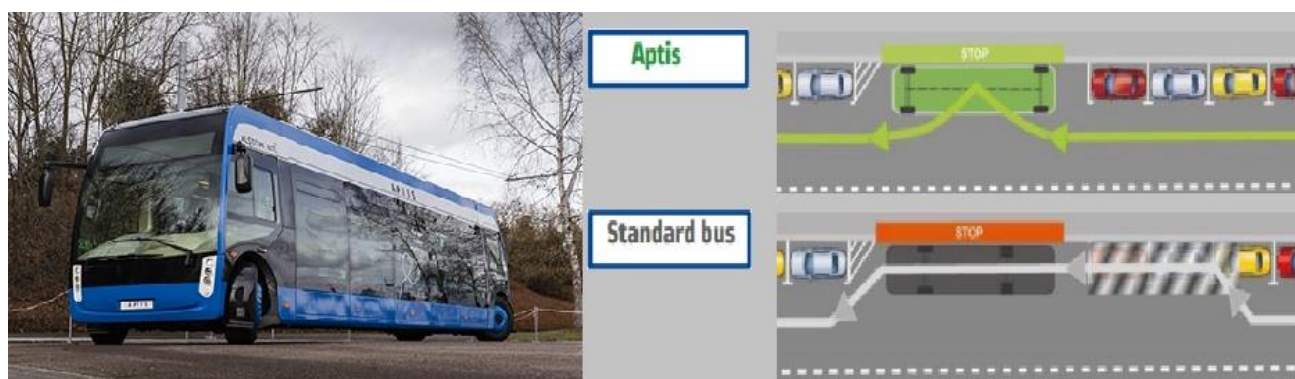


Figure 4: Bus Aptis (gauche) et diminution encombrement arrêt (droite)

En s'éloignant du domaine purement urbain, se trouve plusieurs solutions hybrides qui permettent au matériel de rouler à la fois en milieu urbain et périurbain. Parmi celles-ci se trouve la gamme Citadis Dualis, qui est une extension de la gamme présentée précédemment. Comme son nom peut le laisser supposer, ce train allie les avantages du train urbain du même nom, tout en étant capable de voyager à la vitesse d'un train régional une fois sorti de la ville ; cela permet aux voyageurs habitant en périphérie de joindre le centre-ville sans rupture de charge. Il est à nouveau hautement modulaire, permettant de s'adapter à la plupart des projets.

Dans le même ordre d'idée se trouve la gamme X'Trapolis, d'une capacité pouvant atteindre 30000 passagers par heure et par direction. Cette solution se veut plus économe en consommant 31% de moins que la flotte existante tout en affichant une forte disponibilité (97%).

La deuxième grande catégorie de matériel roulant ferroviaire proposé par Alstom concerne donc les grandes lignes, véhicules voués à parcourir de plus grandes distances à des vitesses bien plus importantes qu'en ville.

Parmi ses solutions se trouve la gamme Coradia, qui sera ensuite déclinée selon les projets et les régions. La version « Polyvalent » est, comme son nom le laisse supposer, une version qui peut opérer sur plusieurs terrains différents ; périurbain (qui peut rejoindre les paragraphes précédents), régional et interville. Disponible en trois dimensions de base (56 à 110m), il est possible de combiner plusieurs éléments pour accroître la capacité du train en fonctionnement et passer alors en unité multiple. A noter de plus que la gamme est dotée de deux motorisations, électrique et bi-mode (diesel + électrique) sous trois tensions possibles 25kV en 50Hz, 1.5kV en DC puis 15kV en 16.7Hz pour les trains transfrontaliers (comme pour le projet « CEVA », ligne reliant France et Suisse). S'ajoute à cela une grande variété d'options, tant pour le confort des passagers que pour l'utilisation à laquelle le train est vouée. Il existe ainsi plusieurs déclinaisons du même train :

La version « Continental », qui a pour but d'opérer en Europe en respectant les contraintes d'interopérabilité, roulant par exemple en Allemagne.

La version « Nordic », adaptée aux températures extrêmes (-35° en fonctionnement et -40° en dépôt) destiné à la Scandinavie.

La version « Meridian » adaptée au marché italien, qui sont plus de 130 à fonctionner dans le pays actuellement (2017).

La version « Lint » désignée à l'origine pour le marché allemand, c'est environ 900 trains de la gamme qui ont été commandé dont 280 pour la DB, opérateur ferroviaire allemand.

La version « Liner V200 », dernier né de la gamme Coradia a été conçu pour les longs trajets. Ce train dispose ainsi d'un confort accru pour les passagers, qui peuvent être jusqu'à 1000 dans la configuration maximale. Le nom du train provient de sa vitesse maximale, 200km/h, adaptée aux longues distances.



Figure 5: Coradia Nordic en Suède (gauche) et Coradia Polyvalent en France (droite)

Puis, en augmentant encore la vitesse maximale, se trouve la catégorie des trains à très haute vitesse comme la gamme Avelia roulant jusqu'à 350km/h en version commerciale. Cette gamme a rencontré et rencontre un franc succès car 1/3 des trains à haute vitesse dans le monde proviennent de cette série. Par ailleurs, c'est cette gamme, modifiée pour l'occasion, qui battu le record du monde de vitesse sur rail (574.8km/h) en avril 2007 en France.

La version « Euroduplex », outre sa capacité maximale importante de 1268 passagers sur deux niveaux, est interopérable dans plusieurs pays européens, ce qui permet de traverser les frontières sans désagréments. Sa taille maximale est alors de 400m pour une vitesse pouvant atteindre 320km/h.

La version « Liberty » est plus sobre en terme du nombre de passagers (jusqu'à 450) mais peut atteindre 350 km/h et existe en version « Tilt » qui permet de franchir des courbes à plus haute vitesse (gain de 30% par rapport à un train conventionnel). Il existe ainsi plus de 1000 rames de cette version en circulation dans le monde.

La dernière catégorie de solutions ferroviaires proposées par Alstom est basée sur des services et du matériel d'infrastructure.

Dans le domaine urbain, se trouve le système de sécurité Pegasus 101 qui assure la sécurité des lignes de Tramway en contrôlant sa vitesse ou son arrêt automatique par exemple. Système éprouvé sur 500 tramways, cela reste une solution simple et facile à installer sur un parc déjà existant. Dans le même domaine se trouve Urbalis 400, qui assure le contrôle des trains via communication réseau. Cela permet une optimisation du trafic (moins de retards, moins de conflits, etc) et peut donc améliorer le rendement d'une ligne. Ce système est en place dans plusieurs villes comme Mexico, Milan ou encore Toronto. Puis, dans une perspective d'avenir, Alstom développe aussi un système haute technologie d'optimisation de flux passagers, appelé Optimet. Il permet, grâce à différents systèmes embarqués dans le train (capteur, caméras, etc) d'estimer son taux d'occupation par véhicule, et ainsi de diriger le flux de passagers entrant vers un véhicule peu encombré grâce à plusieurs signalétiques en amont mais aussi à même le train via des LEDs de couleur.

Dans le domaine des grandes lignes, il est possible de retrouver le système Atlas, qui gère la norme ERTMS (European Rail Traffic Management System) visant à uniformiser la circulation et la signalétique des trains en Europe. Il s'agit d'un système complet avec des infrastructures au sol ainsi que des systèmes embarqués dans le train. L'entreprise est ainsi première sur ce marché, avec 1/3 des systèmes ERTMS mis en place.

Puis il existe des solutions mixtes, pouvant s'adapter aux deux précédents domaines. Par exemple les centres de contrôle Iconis et RailEdge qui visent à surveiller et automatiser les infrastructures d'une ville ou d'une région, ou encore les signalisations Smartway qui sont la base d'une infrastructure ferroviaire. Et enfin, le système de gestion multimodale (plusieurs moyens de transport) Mastria permet de centraliser la gestion de tout un réseau de transport.

iii. Le site de Reichshoffen

a. Historique

Le lieu où s'est déroulé mon stage a donc été le site Alstom Reichshoffen en Alsace. Historiquement parlant, le site appartenait à De Dietrich et a été à l'origine des premiers matériels ferroviaires en France en 1848. C'est en 1998 qu'Alstom rachète le site et continue à y produire le même type de matériel. Actuellement, le pôle reichshoffenois est principalement utilisé pour la gamme des Coradia (cf **ii.Solutions proposées**).

b. Personnel et compétences

Le site comporte 876 personnes (+68 intérimaires + 119 sous-traitants) réparties comme sur le graphique ci-contre. Ce personnel se partage les 190000 m² du site qui comprend tous les éléments nécessaires à l'assemblage des trains et leurs gestions ; du pôle de formation avec l'école des fondamentaux – visant à apprendre les techniques spécifiques du ferroviaire -, en passant par les chaudronneries pour la structure du train, mais encore les lignes d'assemblages des différents éléments, puis les bancs de tests et de validations avec une voie ferrée spéciale de test sur site, mais aussi les pôles humains et informatiques pour finir sur le bureau d'étude – le lieu de mon stage-.

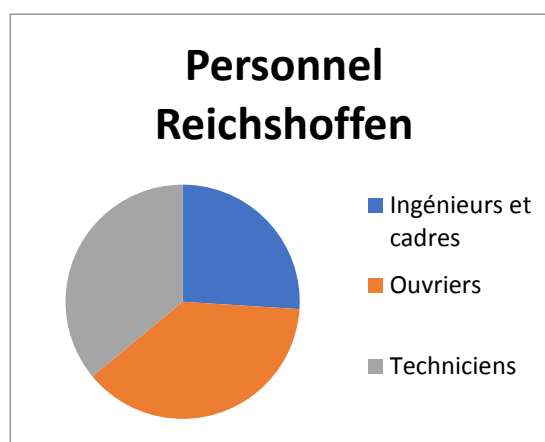


Figure 6: Répartition du personnel Reichshoffen

c. Commandes & projets en cours

Quatre principaux contrats sont actuellement (en avril 2017) en cours sur Reichshoffen : 285 trains pour la SNCF (type Coradia Polyvalent & Intercité), 17 trains pour la SNTF - Société Nationale des Transports Ferroviaires, équivalent algérien de la SNCF – (Coradia), 15 trains pour le Sénégal et enfin 31 trains MI84 (rénovation et modernisation).



Figure 7: Vue aérienne site Reichshoffen

d. Outils utilisés

La grande diversité des opérations du site impose l'utilisation d'outils pour centraliser et régulariser les informations. Selon le domaine, il est possible de retrouver différents software, à noter que les principaux logiciels utilisés pour mon projet seront plus détaillés dans la partie **II. Mon projet** :

SAP : gestion et mise en relation de tous les sous-logiciels des différents domaines, utilisation management.

ASCOT : mise en relation entre le service achat d'Alstom et les fournisseurs pour les besoins en temps réel.

Auto RAL : analyse et traitement des défauts pour le service qualité.

Auto FIE : gestion et instructions pour le service des essais.

MES : visualisation en temps réel des activités de production.

TrainTracer : maintenance des trains, interfaçage avec les systèmes embarqués du train pour en faciliter la maintenance et la supervision.

ClearCase : outil de gestion projet, permet de mettre en ligne différents documents et leurs mises à jour pour consultation et modification par d'autres personnes, sous réserve d'autorisation.

CodeSys : logiciel utilisé entre autre pour coder les différents écrans du DDU.

Alchemist : gestion de projet au niveau des systèmes embarqués ; liaisons entre le code provenant de CodeSys et le hardware.

e. Le bureau d'études (BE)

Le BE est composé de 150 ingénieurs regroupés principalement sur deux étages, puis agencés selon leurs fonctions dans le développement des différents projets. Ils forment un ensemble complet de connaissances afin de mener à bien tous les projets en cours. Ils sont organisés comme selon l'organigramme ci-dessous :

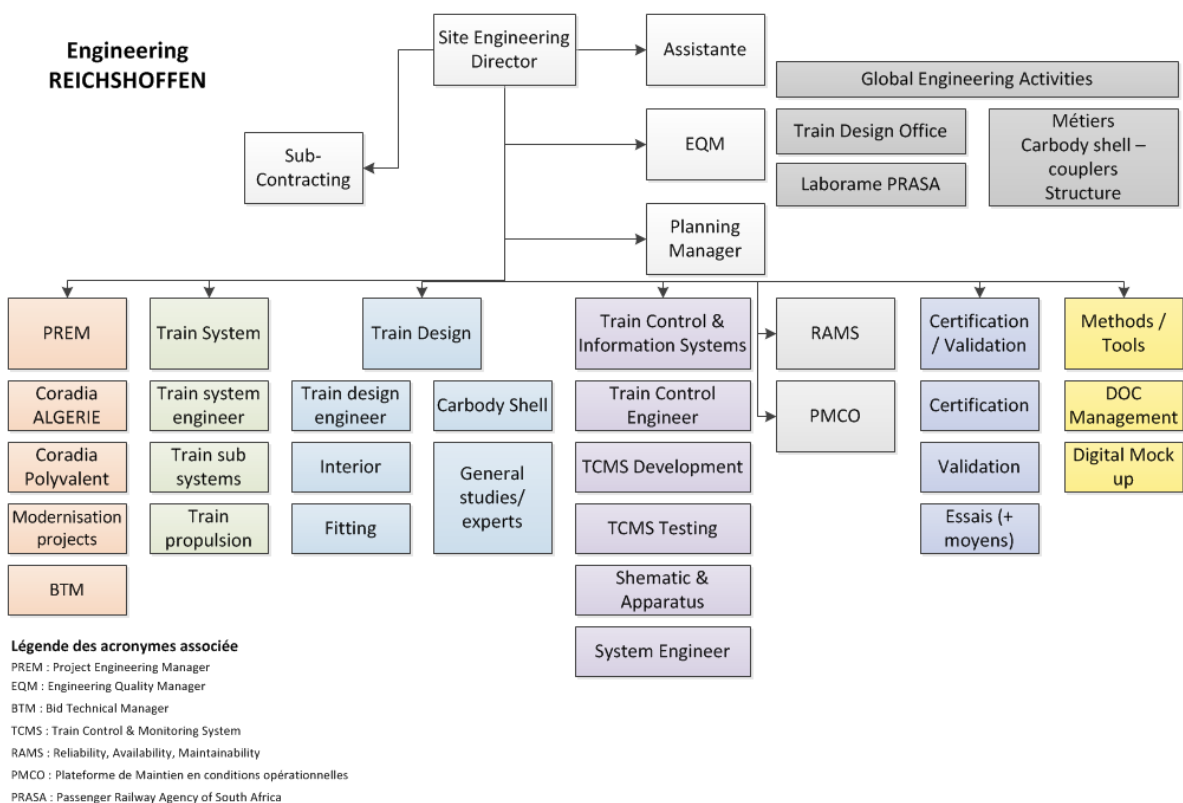
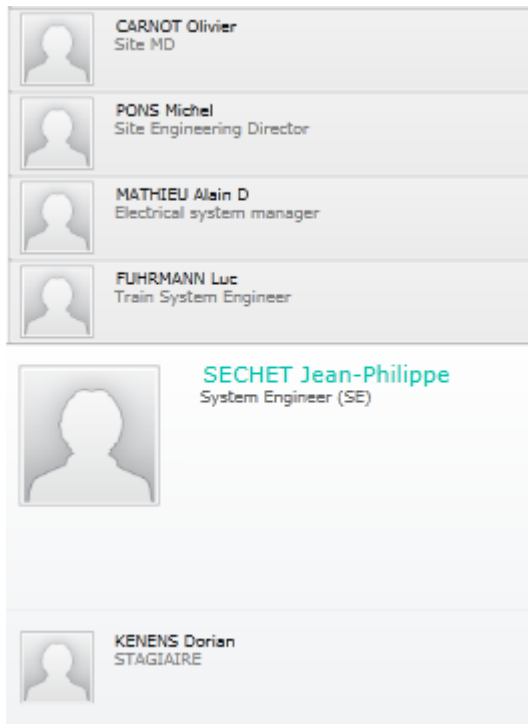


Figure 8: Organigramme ingénieur Reichshoffen



En se recentrant sur la branche du personnel dont je dépends, on retrouve donc mon maître de stage M. Sechet, puis son supérieur M. Fuhrmann, puis M. Mathieu et enfin M. Pons, directeur du pôle d'ingénierie. La dernière personne sur l'organigramme ci-contre est M. Carnot, directeur du site de Reichshoffen.

Figure 9: Hiérarchie directe

II. Mon projet

1. Contexte et matériel/logiciel utilisés

Comme abordé dans l'introduction, mon projet s'est axé sur l'amélioration des systèmes de caméras actuellement mis en place dans les trains. En effet, pour la partie de rétrovision, les caméras utilisées sont analogiques et nécessitent un convertisseur afin d'émettre sur le réseau ethernet embarqué du train (le système sera explicité dans la partie point de vue système). Cela implique alors un coût supplémentaire tant à l'achat que pour la maintenance, ainsi qu'une latence potentiellement plus élevée. L'objectif visé était donc, de maintenir les performances du système analogique en réduisant ainsi les coûts et au mieux d'avoir une amélioration significative des performances en latence comme en qualité de l'image (le détail sera précisé dans **le cahier des charges**). Pour cela, j'ai travaillé sur les projets des trains de type Coradia Export, dont dépendent les projets « Algérie » et « Dakar ». Mon stage s'est donc focalisé sur le remplacement du système actuel dans sa globalité (partie software et hardware) avec pour finalité la création d'une IHM présente sur la DDU dans la cabine conducteur. De plus, j'ai eu l'opportunité de travailler aussi sur les caméras de surveillance des projets.

Le choix majeur du projet fût le choix de l'écran de visualisation des caméras ; il fallait en effet trouver un compromis entre la qualité de visualisation des caméras, les coûts nécessaires à la mise en place de ce système et la minimisation des modifications structurelles de la cabine. Ce choix important sera entièrement détaillé dans la partie de **Sélection de l'écran de développement**.

L'un des matériels clefs de ce projet fût le Laborame. Il s'agit de tout le système informatique d'un train présent dans une salle prévue à cet effet ; à Reichshoffen il en existe actuellement deux versions, une pour le projet en Afrique du Sud (PRASA) et une autre pour la SNCF (Regiolis), qui peut aussi être configurée pour le projet « Algérie » avec la SNTF. Ces « trains » sont utilisés pour tester les systèmes embarqués au niveau software et hardware et sont donc une étape indispensable au développement d'une nouvelle application et c'est donc naturellement que je m'en suis servi à de nombreuses reprises durant mon stage.

Les écrans de visualisation (comme la DDU) sont plus précisément des ordinateurs embarqués avec écrans fonctionnant sous Unix. Ils sont conçus pour être très robustes et aux normes du ferroviaire (en particulier la norme EN 50155). J'ai eu l'occasion de travailler avec différentes versions, ACE et Evol2. A noter que les deux modèles possèdent des dalles tactiles.

Les caméras numériques Moxa, déjà utilisées comme caméras de surveillance dans d'autres projets, ont été la solution choisie comme caméra de remplacement potentielle pour la rétrovision. Un grand avantage de ces dernières est leur présence directe dans le Laborame PRASA ainsi qu'une interface de configuration complète et intuitive via un simple navigateur web.

Dans une moindre mesure, certains éléments du système embarqué du train ont aussi été utilisés ; le MPU pour la gestion de certaines variables réseaux mais aussi les TRS et CRS sur lesquels ont été branchés les caméras. Ces appareils sont configurables par interface web ou en s'y connectant directement en SSH.

Pour la partie logiciel, toute l'IHM a entièrement été conçue sur CodeSys. Il s'agit d'un environnement de développement principalement utilisé dans les domaines de l'automatisation. Parmi ses nombreuses fonctionnalités, seuls les langages de texte structuré (proche du BASIC) et les blocs fonctionnels ont été utilisés ainsi que toute la partie de création de visualisation qui est la « face avant » du code, visible par l'utilisateur. De plus, ce logiciel permet la simulation sur ordinateur et gère le téléchargement du code ainsi créé sur une console connectée au même réseau local.

2. Définition du cahier des charges (CDC)

Bien que je me sois basé sur la gamme « Coradia Export », le cahier des charges a été construit à l'aide de différents projets. Parmi ces derniers, « Regiolis » a été le plus utilisé car il présente les contraintes les plus précises et clairement définies. A noter cependant que pour ces trains, ce n'est pas directement Alstom qui a développé le système ; il n'est pas intégré aux autres applications existantes et nécessite deux écrans séparés de 15".

La figure ci-dessous est à caractère uniquement informatif et représente une disposition possible des caméras de rétrovision sur le train ainsi qu'une possible configuration d'un écran de visualisation afin de faciliter la compréhension du cahier des charges.

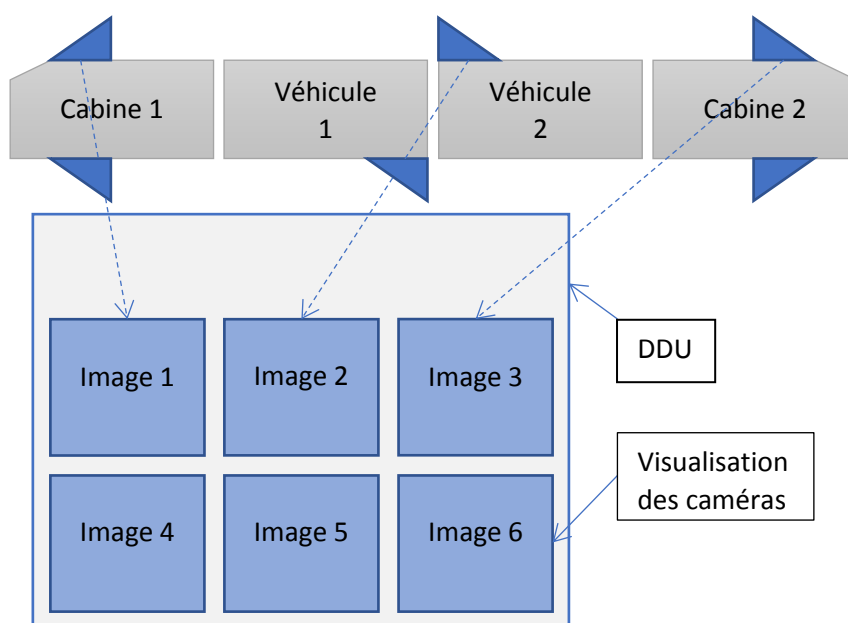


Figure 10: Exemple de disposition des caméras de rétrovision

i. Grandes étapes du projet

Les étapes principales de mon stage sont :

- Documentation et étude des systèmes de rétrovision des projets « Regiolis » et « Algérie ».
- Définition du cahier des charges.
- Elaboration du choix de l'écran de visualisation et validation par le tuteur.
- Intégration dans le projet existant et développement de la sous application.
- Phase de test en simulation sur ordinateur.
- Phase de test sur console hors Laborame.
- Comparaison finale sur Laborame avec le système déjà existant.

Ses étapes ont été ponctuées de réunions et mises au point fréquentes avec mon tuteur afin de communiquer l'avancement du projet et corriger les éventuelles erreurs de conception de l'application. De plus, j'ai été, durant les phases de test, en échange fréquent avec les personnes en charge du Laborame mais aussi avec les développeurs de l'application globale.

ii. Contraintes sur l'affichage des images de rétrovision

Les différentes contraintes chiffrées relatives à cette partie seront exposées dans un tableau afin d'en faciliter la lecture et la compréhension, elles seront reprises ensuite plus en détail.

Temps maximal d'affichage lors de l'accès à l'écran de rétrovision	3 secondes
Temps de basculement maximal entre les côtés du train	3 secondes
Temps de latence maximal	500 ms
Temps maximal de basculement entre le zoom sur une image et son affichage par défaut et inversement.	1.5 secondes
Nombre minimal d'images par seconde	12
Définition minimale – si possible -	CIF – 352x288 pixels
Nombre minimal de flux à afficher simultanément	6 flux
Rapport minimal entre la taille de la visualisation d'une personne d'1m60 et la taille totale de l'image	5%

Tableau 1: Cahier des charges

- Temps de basculement entre les côtés du train : dans la majeure partie des cas, l'entrée et la sortie des voyageurs se fera d'un seul côté – côté quai -, la fonctionnalité de basculement n'étant alors pas forcément utilisée. Cependant, il existe des cas de doubles quais ou simplement si le conducteur souhaite vérifier l'autre côté du train.
- Temps de latence : il s'agit d'une variable clef qui se retrouvera dans n'importe quel système vidéo et qui sera presque toujours à minimiser pour des raisons évidentes. En effet, le système de rétrovision doit être le plus réactif possible afin de remplacer efficacement un système classique de miroirs.
- Temps de basculement zoom : cette fonctionnalité permet au conducteur de zoomer sur une image pour avoir plus de détails et de lever/confirmer un doute qui ne serait pas forcément possible sur une miniature. En considérant que ce zoom occupe l'intégralité de l'écran, il est alors nécessaire de revenir au mode nominal rapidement.
- Nombre minimal d'images par seconde : cette variable se place dans le même ordre d'idée que la latence, sauf qu'il s'agit de la maximiser afin d'être le plus proche possible d'une vision directe.
- Définition minimale : cette valeur est directement en lien avec la dernière variable du tableau. Il s'agit en effet, de voir le plus de détails possibles pour le conducteur. Cependant, ce paramètre est contraint par plusieurs données – occupation réseau, définition de l'écran de visualisation, etc -, il n'est donc pas possible de choisir simplement la meilleure définition du marché. Il s'agira alors de trouver le meilleur compromis.
- Nombre minimal de flux simultanés : variable majeure du projet, il est nécessaire d'afficher au moins un côté de manière complète sur l'écran – soit trois caméras par élément - donc neufs au total (UM3, configuration maximale). Cette valeur est impossible à tenir sur des écrans de cette taille, c'est pourquoi il faudra les afficher en deux temps, avec six images au maximum en simultané.
- Rapport taille d'une personne/taille de la miniature : ce rapport fixe la limite de distinction d'une personne dans les images perçues par le conducteur. La valeur de 5% a été extraite d'une exigence du cahier des charges de la SNCF dans le cadre du projet « Regiolis ».

Ces contraintes chiffrées ont donc été une base de travail pour la conception de l'application et le choix du matériel. Elles ont été établies suite à la lecture des documents techniques du projet « Regiolis » puis confirmées par mon tuteur de stage.

iii. Gestion des configurations train

L'une des difficultés majeures du système de rétrovision est l'adaptation aux différentes configurations du train (jusqu'en UM3 dans le cadre de mon projet). Cela peut être plus simplement imaginé en considérant un arbre, présenté dans la figure ci-dessous. A noter que ce dernier n'est pas complet dans un souci de lisibilité, de plus, il faut comprendre qu'un train n'est pas forcément symétrique par rapport à un axe central ; il faut donc considérer son orientation.

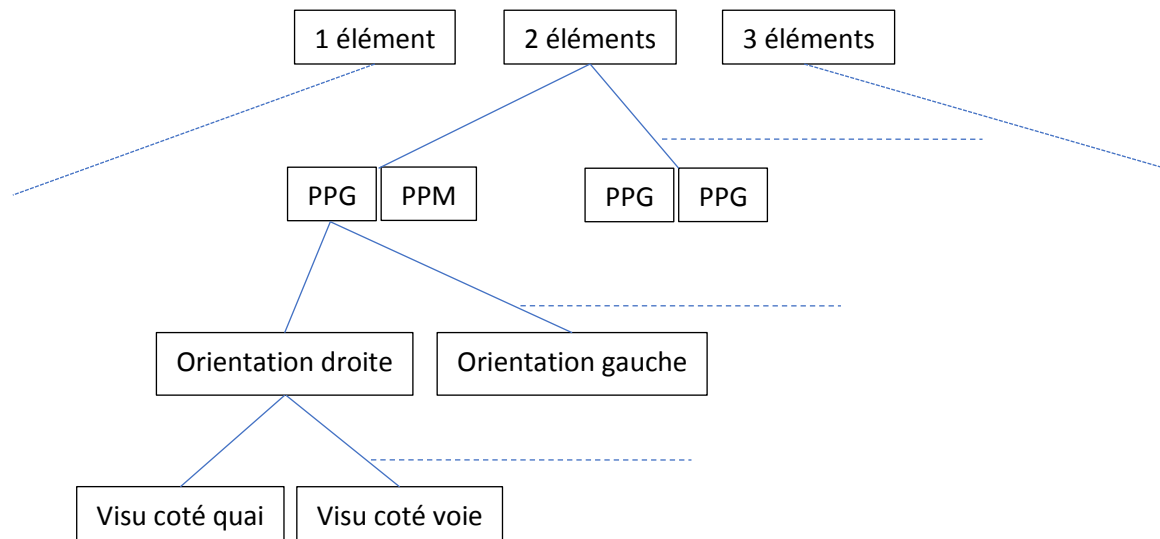


Figure 11: Exemple de configuration train

Ces configurations peuvent être détectées en lisant les variables réseaux envoyées par le MPU (qui seront détaillées dans la suite de ce rapport). Il restera alors à interpréter ces dernières afin d'assurer au conducteur que les caméras visibles à l'écran correspondent bien aux caméras concernées sur le train.

Par défaut, l'application doit sélectionner la visualisation des caméras du côté du quai, sans que le conducteur ne doive faire de choix. Il pourra ensuite, s'il le souhaite, changer de côté manuellement.

Dans le cahier des charges du projet « Regiolis », il est stipulé que les caméras doivent permettre la visualisation de deux véhicules de suite au maximum ainsi que toutes les portes d'accès au train. Cependant, dans le cadre de mon projet, le placement des caméras sur le train étant fixe, il faudra adapter éventuellement les différentes contraintes évoquées dans la partie précédente afin de satisfaire à cette exigence (**Contraintes sur l'affichage des images de rétrovision**).

Une fonctionnalité inédite (non présente dans les autres versions de rétrovision que j'ai pu observer) et demandée par mon maître de stage pour mon projet est l'ajout d'une visualisation de la configuration train actuelle. Elle a pour objectif d'indiquer au conducteur où se trouve sa cabine par rapport au train mais aussi d'indiquer quels véhicules et donc caméras sont actuellement visualisées. Cette fonctionnalité devra ainsi être dynamique et intuitive.

iv. Autres fonctionnalités demandées

Dans le cas où l'écran devait être tactile, il a été demandé de tirer pleinement partie de cette particularité afin de démarquer la solution.

3. Point de vue système du train

i. Présentation du système actuel

Le système actuellement en place pour la gamme « Coradia Export » est donc un système analogique-numérique (similaire au système présent dans le projet « Regiolis »), développé par la société Faiveley. Il nécessite d'importants besoins matériels (qui seront représentés sur la figure 12.) :

- 6 caméras de rétrovision analogiques – Camx sur fig.12.
- 4 écrans de visualisation (du même type que les écrans EVOL2) – Ecrx sur fig.12.
- 2 serveurs de conversion analogique – numérique – Serveur sur fig. 12.
- 2 claviers (semblable pavé numérique classique) – Clv sur fig. 12.
- Câblage des écrans, du serveur, des caméras, etc.

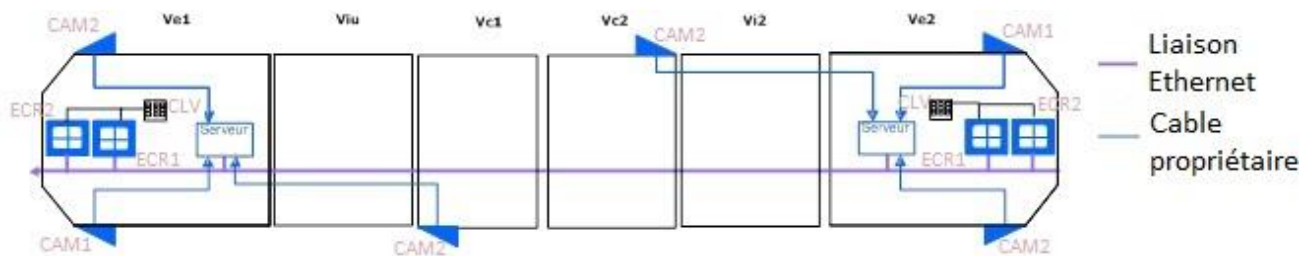


Figure 12: Schéma intégration système rétrovision actuel

Dans chaque cabine se trouve donc la majorité du matériel, à l'exception des caméras présentes aux véhicules « VC1 » et « VC2 », qu'il faut alors obligatoirement câbler jusqu'en cabine ou se trouve les serveurs de conversion.

Les claviers de rétrovision sont utilisés pour permettre au conducteur d'interagir avec les écrans (zooms, défilement des caméras, etc) qui sont déportés. Ces claviers doivent être alors câblés sur les deux écrans.

Enfin, la partie logicielle est elle aussi implémentée par Faiveley, mais doit être capable de lire les variables présentes sur le réseau.

ii. Améliorations potentielles communes aux deux écrans

Cette sous-partie a pour but de préciser les améliorations indépendantes du choix de l'écran de rétrovision (qui sera abordé dans la partie suivante) qui seraient conséquentes à un changement du système.

- Les caméras de rétrovision : les caméras Moxa sont du matériel de haute qualité adapté aux normes du ferroviaires. Elles sont munies d'une interface web qui permet un accès simple et rapide à une multitude de paramètres de correction d'image, de définition du flux, d'adressage des flux, etc. De plus, le « MTBF » théorique annoncé de 1 944 687 heures confirme la grande robustesse du matériel.
- Les serveurs de conversion : ils peuvent simplement être supprimés, les caméras émettent directement un flux numérique sur le réseau ethernet. Cela libère alors un emplacement dans les baies informatiques.
- Câblage : les caméras sont directement câblées sur le CRS de l'élément local, il n'est alors nécessaire de traverser plusieurs véhicules pour venir connecter les caméras sur les serveurs de conversion (gain pouvant atteindre 30m par élément).
- Logiciel : développé en interne avec des outils utilisés par des ingénieurs Alstom, cela implique une réduction du nombre de logiciels différents en fonctionnement dans un train.
- De plus, la maintenance se trouve facilitée, il n'est plus nécessaire de faire appel à une entreprise extérieure.
- Les écrans : à minima un écran (par cabine) pourrait être supprimé, selon la solution choisie.

iii. Améliorations propres à la solution petit écran type ACE

En plus des améliorations précisées dans la sous partie précédente, on peut y ajouter les points suivants :

- Les écrans : suppressions de tous les écrans spécifiques à la rétrovision, la solution serait intégrée directement dans un écran déjà utilisé sur le pupitre.
- Les claviers : suppressions des claviers, la console sera utilisée dans son mode tactile.
- Câblage : suppression du câblage des écrans et des claviers associés.
- La structure : il n'est alors plus nécessaire d'intégrer des emplacements pour les écrans de rétrovision déportés, ni pour les claviers de rétrovision.

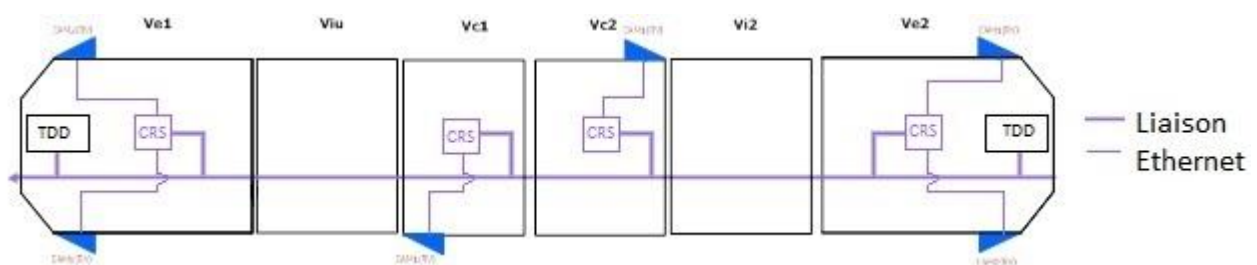


Figure 13: Schéma intégration nouveau système rétrovision

iv. Conclusion – système

D'un point de vue système train, les deux solutions envisagées peuvent apporter des améliorations significatives au niveau financier (suppression de plusieurs éléments). De plus, Alstom se retrouverait avec un système conçu en interne qui peut alors être ajusté selon les besoins clients de manière bien plus rapide, sans passer par des process d'une entreprise extérieure. Alstom gagnerait alors en adaptabilité tout en réduisant les coûts. Mais encore, la présence d'un grand nombre de caméras IP sur le marché, permet d'adapter éventuellement le système à certaines contraintes spécifiques à différents projets sans changer l'intégralité du système.

Il est cependant nécessaire de noter que la solution du petit écran est bien plus avantageuse par la suppression des modifications structurelles de la cabine (claviers & écrans), ce qui implique un gain de temps qui peut être conséquent dans le processus de fabrication. Ces modifications ont été mises en valeur sur la figure ci-dessous.

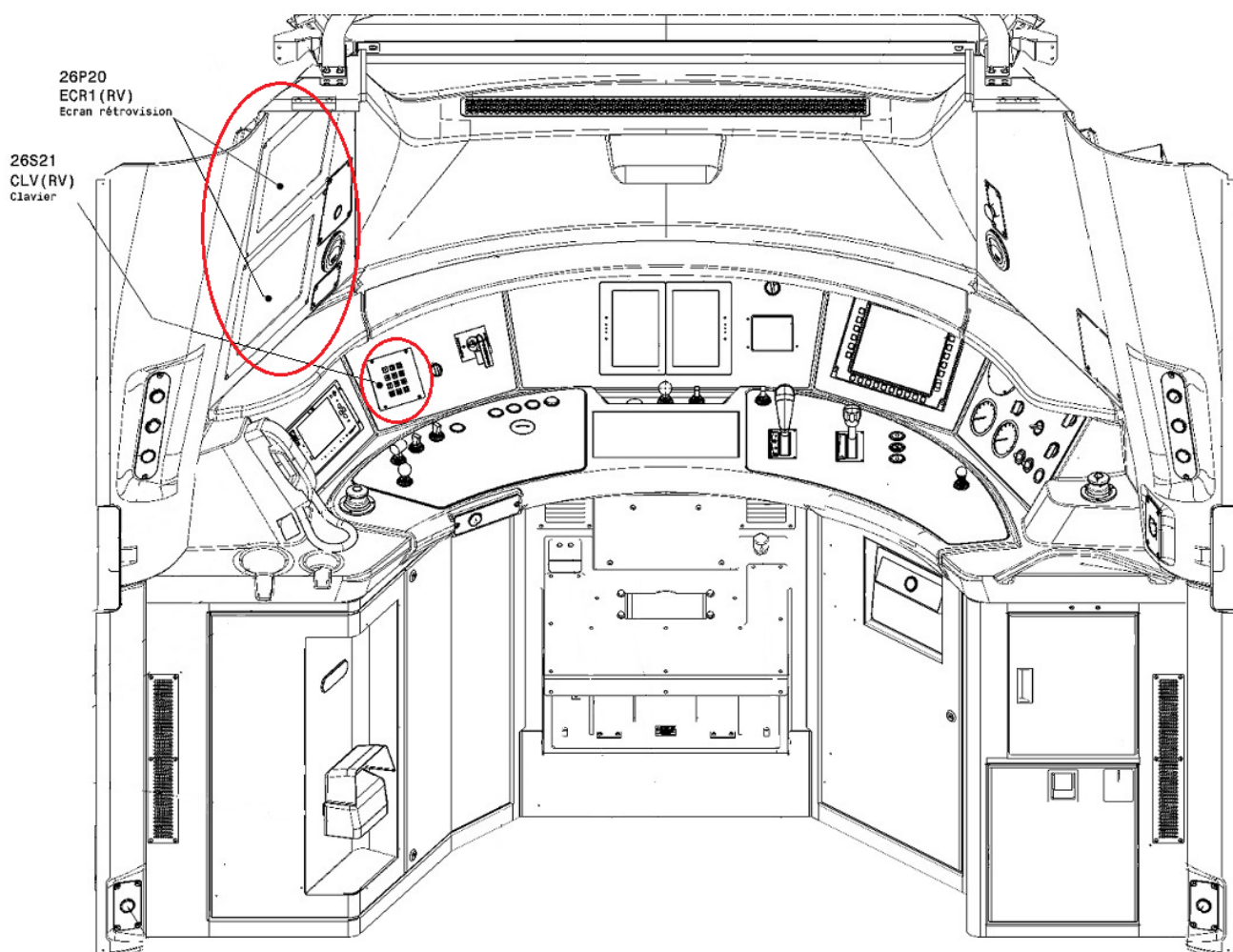


Figure 14: Modification structurelle cabine

Ainsi, d'un point de vue système, la solution du petit écran apporte le plus d'améliorations. Il reste alors à considérer et étudier l'ergonomie ainsi que l'intégration au système existant.

4. Sélection de l'écran de développement de l'application

Cette partie présente la phase de sélection de l'écran de développement pour la nouvelle sous-application de rétrovision. En effet, il était nécessaire de déterminer si un écran de petite taille (type ACE – 640x480 – 10''4) qui est actuellement utilisé pour les projets Algérie et Dakar était suffisant pour satisfaire à la plupart des exigences du cahier des charges présenté précédemment (**définition du cahier des charges**) ou si un écran indépendant plus grand (type EVOL2 – 1024x768 – 15'') était nécessaire.

Certaines règles de codage n'ont pas été nécessairement suivies car le code créé dans le cadre de la sélection de l'écran de développement n'a pas vocation à être réutilisé ni intégré à un autre projet. Elles seront par contre détaillées et suivies dans l'application finale (**application finale**).

i. Le Laborame PRASA

Durant la première phase de mon stage, je me suis intéressé au Laborame PRASA pour des raisons de disponibilité et d'accessibilité. En effet ce dernier n'était pas utilisé à 100%, le matériel nécessaire à la visualisation des caméras de rétrovision était la majeure partie du temps disponible.

Ce dernier est composé d'un pupitre de commande qui tend vers une copie de la cabine conducteur, à quelques différences près (agencement, matériel en remplacement, etc) mais aussi et surtout d'une grande partie du parc informatique embarqué d'un train. Son avantage principal est l'accès direct en face arrière aux CRS et TRS en prise ethernet RJ45, qui permet alors de faciliter la connexion au réseau embarqué avec des ordinateurs classiques. L'un de ces derniers, qui y est à demeure, est configuré pour un accès au réseau train et possède le logiciel CodeSys qui permet de coder l'application puis de la télécharger directement sur les écrans de visualisation présent sur le réseau train. De plus, le projet comporte déjà une application qui gère, entre autre, la partie de caméras de surveillance, je m'en suis donc inspiré pour créer les applications de tests qui seront présentées dans la suite de cette partie.

Concernant les caméras, le Laborame en possède sept, toutes de la même marque (Moxa) et seule une diffère légèrement de modèle (caméra sensiblement plus performante). Leurs adresses IP étant déterminées par leurs ports de connexion aux différents CRS, l'utilisation du plan d'adressage du projet est alors nécessaire.

Enfin, l'écran de visualisation utilisé par l'application CodeSys du projet en cours (et donc de mes applications de test) est de type Evol2. A noter que l'écran de petite taille a pu être simulé afin de comparer entre les deux solutions.

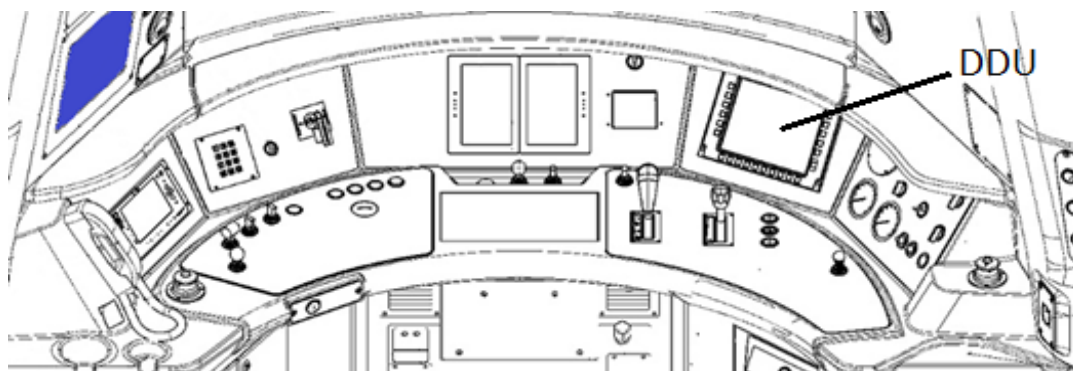


Figure 15: Pupitre cabine conducteur Algérie & Dakar

ii. Développement application grand écran

La première application qui a été mise en œuvre est une simple application indépendante de visualisation des flux. Elle a été construite en prenant exemple sur le code de la partie CCTV du projet PRASA. Son but est de comparer différentes tailles des miniatures, leurs différentes positions ainsi que de me permettre, dans le même temps, de saisir l'utilisation des fonctions de récupérations des flux du réseau qui sont fournies sous forme compilées par Alstom Villeurbanne.

Comme précisé dans le début de cette partie, dans l'idée où ce serait l'écran type EVOL2 qui était choisi, ce dernier serait indépendant et donc ne serait pas rattaché à une application déjà existante. Cette précision est nécessaire dans la mesure où cette indépendance se traduit par l'absence de réservation de certaines zones d'affichage pour diverses fonctionnalités comme des barres de navigation, pour l'affichage de l'heure, etc.

Ses fonctionnalités sont donc relativement basiques ;

- Affichage des flux aux zones Cx.
- Bouton cliquable de démarrage et d'arrêt de tous les flux (1) & bouton cliquable d'arrêt du flux haut gauche uniquement (2).
- Etat du flux haut gauche (entier lié à l'état de la fonction de visualisation) (3).
- Zoom en cliquant sur la miniature de l'image.
- Espace réservé à une éventuelle visualisation de l'état du train (non implémenté pour les phases de sélection de l'écran).

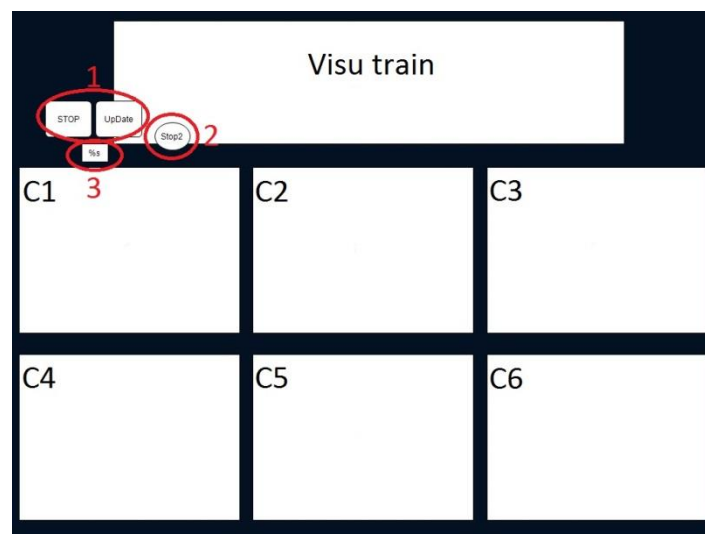


Figure 16: Application basique grand écran

La taille des miniatures n'a pas été choisie au hasard ; il s'agit de la définition QVGA (320x240, ¼ de VGA), la plus haute définition possible en sortie de caméra pour un affichage de 6 flux simultanés sur EVOL2. Ce point sera cependant sujet à discussion dans la partie de tests sur caméras réelles (**Tests sur caméras réelles**).

Il s'en est suivi une phase rapide de simulation sur ordinateur pour supprimer les éventuelles erreurs de conception et vérifier l'intégrité du code. A noter que les fonctions fournies par Alstom Villeurbanne concernant la visualisation des flux vidéos ne permettent pas d'avoir des images lors des phases de simulation ; il est nécessaire d'utiliser un écran de visualisation intégré à un réseau pour visualiser les caméras. Le bon fonctionnement des boutons cliquables et du zoom n'ont donc pas pu être testés en phase de simulation.

iii. Développement application petit écran

Afin de comparer les différents moniteurs il était nécessaire alors de coder une application semblable sur petit écran. Cependant, le projet PRASA n'inclut pas d'écran 10''4 utilisable directement ; après validation par mon maître de stage, nous avons fait le choix de le « simuler » sur EVOL2. Il fallait designer une application qui entre sur 640x480 pixels et qui contient les espaces réservés par l'application « Dakar » en cours afin d'être au plus proche d'un résultat final. Notons cependant que la résolution sera alors différente entre l'écran simulé et l'écran réel ; 85.33 PPI pour l'écran simulé contre 76.92 PPI pour ACE. Cela impliquera une différence sensible de la taille des miniatures en cm entre la simulation et la réalité.

Ces fonctionnalités sont semblables à celles du grand écran :

- Affichage des flux aux zones Cx.
- Bouton cliquable d'arrêt et de démarrage de tous les flux (1).
- Bouton cliquable d'arrêt et de démarrage pour chaque flux (2).
- Présence de zones « espace réservé ».
- Présence d'une ébauche de visualisation train.
- Présence d'indicateur de dimension en pixel (3).

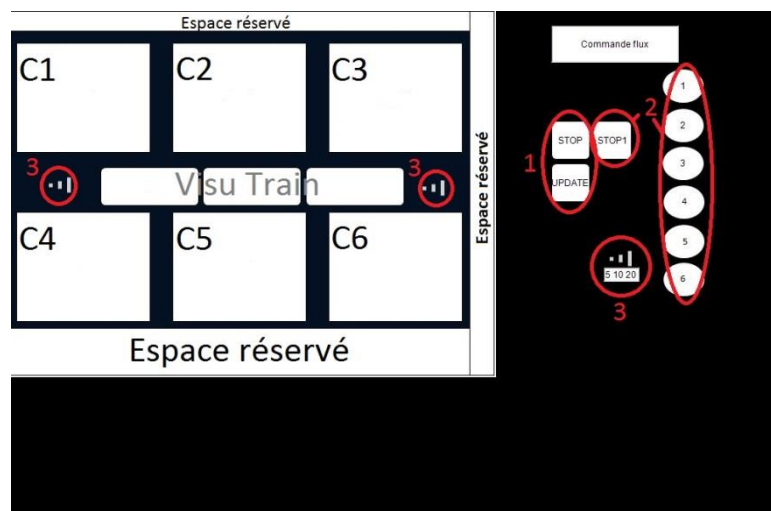


Figure 17: Application basique petit écran simulé

Les zones d'espace réservé sont des parties de l'écran utilisées comme barres de navigation, barre de titre et affichage de l'heure système. Elles n'ont pas été implémentées dans cette application car inutiles vis-à-vis de l'objectif fixé. Elles seront cependant détaillées dans l'application finale qui a été retenue.

Les indicateurs de dimension en pixel (5 – 10 – 20 px) ont été codés dans le but d'estimer grossièrement la taille d'une personne sur les miniatures afin de comparer avec les 5% du cahier des charges (rapport personne de 1m60/taille de l'image).

L'ébauche de visualisation train a pour but de représenter un train en UM3, configuration la plus gourmande en terme de taille à l'écran. Il s'agit de trouver un compromis entre visibilité et espace réservé ; il m'a été demandé en effet de prendre en compte la possibilité d'interaction tactile avec cet élément.

La taille des miniatures a été choisie de la même manière que pour l'application précédente ; définition la plus grande possible pour disposer de six flux simultanés. De cette façon, c'est une définition clairement en deçà du cahier des charges, QCIF (176x144, ¼ de CIF), qui s'est imposée. Bien qu'à nouveau sujet à discussion, il n'est cependant clairement pas possible de maintenir à la fois la condition des six flux et la définition CIF du cahier des charges sur un écran de cette taille. Nous verrons alors dans la partie de tests avec caméras réelles comment cela peut être perçu avec un flux véritable et détermineront alors quelle clause du cahier des charges peut être considérée comme prioritaire.

Enfin, la présence de boutons hors du cadre de l'écran simulé est cohérente avec leurs utilités ; autrement dit, ces derniers sont utilisés pour compenser certaines fonctionnalités qui seront présentes dans l'application complète. Ainsi, ils ne sont nécessaires que dans le cadre de cette partie.

iv. Tests sur caméras réelles

Suite au design des applications, l'étape suivante a été la phase de tests avec les caméras du Laborame PRASA (MOXA VPort P06-1MP-M12). Il s'agissait de vérifier le bon fonctionnement des fonctions de visualisation ainsi que des différents boutons qui n'ont pas pu être testés en simulation.

Cette partie s'est déroulée en deux étapes ; la première consistait à configurer les caméras et la seconde à modifier le code des applications qui dépend alors de la première étape.

Concernant alors les caméras, elles sont fournies avec plusieurs documents détaillant leurs configurations ainsi qu'avec un logiciel constructeur qui va analyser toutes les caméras du réseau en émettant une annonce en broadcast. Cela permet alors de récupérer toutes les adresses IP ainsi que les adresses MAC correspondantes. Une fois ces adresses récupérées, il a fallu simplement se connecter à la page web associée à la caméra (qui est son adresse IP) et de la paramétrer en mode dynamique si nécessaire (adresse IP imposée par le réseau – selon ports de branchement au CRS -). Ensuite, paramétrer les flux vidéo pour qu'ils soient émis selon les protocoles RTSP & UDP et configurer leur aspect (définition, correction des couleurs, anti-scintillement, etc). Une fois ces différentes étapes validées pour toutes les caméras, le logiciel VLC de lecture de vidéos a été utilisé pour vérifier la présence des flux sur le réseau selon l'adresse suivante : 'rtsp://[IP de la caméra]:554/udpstream_ch1_stream1'. A noter l'utilisation du protocole UDP à la place du TCP pour des contraintes de temps réel ; le premier peut fournir une image de qualité potentiellement moindre (dues aux pertes de paquets éventuelles) mais aura une latence plus faible que le second.



Figure 18: Caméra Moxa (gauche) et page web de configuration (droite)

Puis, dans un second temps, il aura fallu modifier les adresses URL du code des deux applications et comprendre le fonctionnement des fonctions de visualisations. En effet, ces dernières redimensionnent automatiquement le flux entrant ; il faut configurer au préalable une aire définie dans l'application puis, peu importe le flux vidéo récupéré, ce dernier sera redimensionné pour convenir exactement à l'espace réservé. C'est pourquoi il n'est pas forcément nécessaire d'avoir des miniatures qui correspondent aux tailles des sorties vidéos des caméras. Ainsi, l'application sur grand écran est passée sur des miniatures 300x300 pour les phases de tests mais l'application sur petit écran est restée sur la même définition par manque de place disponible (seuls quelques pixels pouvaient être gagnés).

Une fois l'étape de configuration effectuée, les premiers tests avec caméras intégrées au Laborame ont été effectués. La première remarque est l'absence de latence visible à l'œil nu, à nuancer par le très peu de charge réseau pesant alors sur les systèmes du train. La qualité des visualisations est bonne, malgré un redimensionnement opéré par les fonctions de visualisation ; en effet, le flux d'origine était du VGA (640x480). De plus, la qualité ne semble pas être fortement modifiée par le changement de taille du flux d'origine, du moment que celui-ci ne soit pas en dessous de la taille de la vignette, ainsi pour des raisons de charge réseau, il ne sera pas nécessaire d'émettre un flux d'une définition très supérieure à la taille des vignettes. Enfin, les boutons ont pu être testés et leurs fonctionnements validés. Ces remarques sont valables pour les deux applications créées et les détails seront résumés dans un tableau à la fin de cette partie (**conclusion**).

Pour donner un ordre de grandeur de la charge CPU allouée aux flux vidéos, j'ai utilisé la commande « top » que j'ai modifiée puis envoyée dans un fichier texte pour y construire le graphique ci-dessous (commande complète : `top -b -d 0.5 > top.txt` puis `cat top.txt | grep X | cut -c 41-50`, avec X représentant l'application sur la console puis 1^{ère} colonne = charge CPU et 2^{ème} colonne = charge mémoire). A noter que ce test a été effectué uniquement pour l'application grand écran, cela n'avait pas de sens pour l'application petit écran considérant que cette dernière sera implémentée sur une autre console.

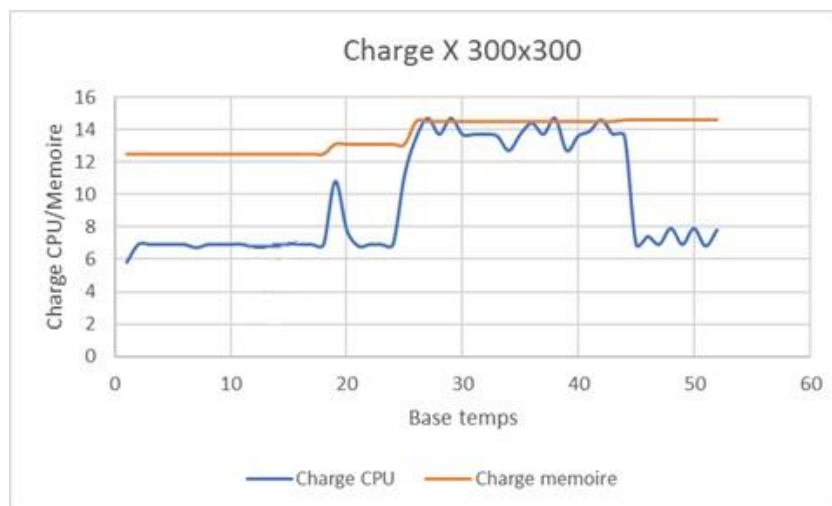


Figure 19: Charge CPU, application grand écran

Le premier pic de charge correspond au démarrage puis l'arrêt d'un seul flux (vers 20s), le second pic, qui se stabilise à vers sa valeur maximale, correspond à la lecture de tous les flux. On remarque donc une augmentation assez importante de la charge CPU, environ x2, cependant la charge globale (env. 14%) reste relativement faible. On rappelle toutefois que l'application a uniquement pour but d'afficher les flux, ce test de charge mémoire permet donc seulement de dégager une tendance sur l'utilisation de la mémoire lors de l'affichage des flux.

v. Tests sur caméras « simulées »

Une donnée majeure manquante sur les caméras du Laborame est la taille des passagers par rapport à la taille des miniatures. En effet, la composition de la pièce contenant ce train simulé, ne permet pas de filmer une personne convenablement, à la manière d'un train classique. La solution retenue avec mon tuteur de stage fût de filmer avec un smartphone à la place des caméras de rétrovision sur un train Coradia en gare. Puis, j'y ai opéré un post-traitement pour réduire sa taille (640x480) et enfin la diffuser sur le réseau embarqué du train à l'aide de VLC. Cela permettra d'avoir une idée des miniatures en conditions réelles.

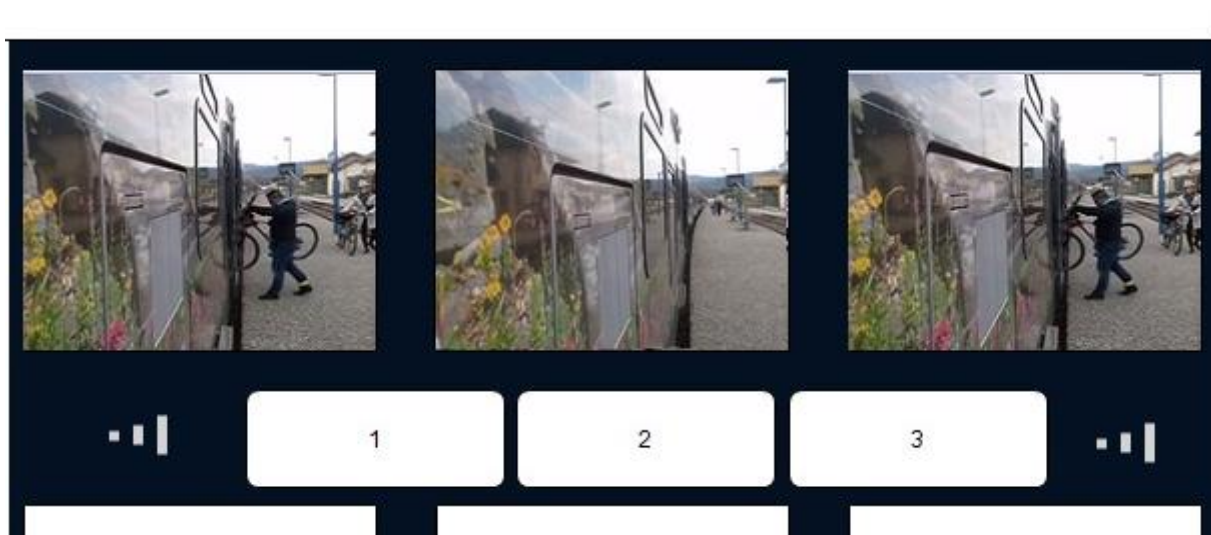


Figure 20: Visualisation petit écran (coupée), caméras "simulées"

La personne en noir, qui entre dans le véhicule, se situe au niveau de la première porte (la caméra est censé couvrir deux véhicules consécutifs) et est clairement visible, son ratio est bien supérieur aux 5% demandés par le cahier des charges (env. 60px/144px, soit 41%). La seconde personne, se situe environ au niveau de la seconde porte, son ratio est encore supérieur aux 5% (env. 35px/144px, soit 24%). Puis, dans la seconde image, cette même personne habillée en noir, est située dans le cas le plus extrême de visualisation, soit au niveau de la porte la plus éloignée sur deuxième véhicule. On l'observe sous la taille de 10 px, soit un ratio d'environ 7%. On se trouve donc à la limite du cahier des charges, il est clair que le conducteur ne peut que difficilement distinguer une personne dans ce cas, d'où la nécessité d'un zoom sur l'image. On note cependant que ces images sont des images filmées « à la main » puis modifiées, elles ne donnent qu'une idée de ce que pourrait donner une miniature de cette taille.

Dans le cadre de la visualisation grand écran, lors d'une miniature à 300x300px, on trouve donc les mêmes ratios, les personnes sont cependant plus visibles pour des raisons évidentes de taille en pixels.



Figure 21: Comparaison grand et petit écran

vi. Conclusion – choix de l'écran

Le tableau ci-dessous présente de manière synthétique les différents points relevés lors de phases de test sur la console au Laborame PRASA.

Type d'écran Caractéristiques	Grand écran – Type EVOL2	Petit écran – Type ACE
Définition de l'écran (et donc définition du zoom maximale)	1024x768	640x480
Taille de l'écran	15'' (38.1cm)	10''4 (26.4cm)
Aspect visuel (définition du flux)	Bon (300x300)	Moyen (176x144)
Aspect visuel du zoom (plein écran)	Bon	Bon
Ratio du CDC – cas extrême	>5%	>5%
Taille pixel d'une personne – cas extrême	Env. 20px	Env. 10px
Placement	Déporté ou modification du pupitre	Intégration directe dans la DDU existante ou création d'un emplacement dédié
Utilisation tactile	Impossible (écran trop éloigné) si déporté – Utilisable si intégration au pupitre	Oui
Evolution du cout global par rapport à la solution précédente	Suppression du tiroir de rétrovision (convertisseur analogique-numérique) & suppression d'un écran de visualisation pour cas déporté et si intégration au pupitre : cout de modification de la structure du pupitre rajouté	Suppression du tiroir de rétrovision & suppression des deux écrans (intégration dans DDU déjà existante)

Tableau 2: Comparaison des deux écrans

Notes :

- un placement « déporté » correspond à celui représenté par un carré bleu en **figure 15**.
- les paramètres de latence et de temps changement de coté de train n'ont pas été testé car le code utilisé n'était pas représentatif du code final, ces variables pouvaient alors changer significativement.

Après consultation avec mon maitre de stage ainsi qu'avec M. Fuhrmann (**organigramme réduit**), la solution retenue fût celle du petit écran pour deux raisons majeures : la première est la facilité d'implémentation dans le pupitre actuel (pas de modifications structurelles) couplé à une réduction importante des couts. On retrouve alors les améliorations mentionnées dans la partie du point de vue système. De plus, bien que sensiblement moins performante au premier abord que la solution avec écran EVOL2, cette solution reste cohérente avec le cahier des charges et peut apporter une plus-value non négligeable en terme d'ergonomie grâce à l'utilisation du mode tactile.

Ainsi, la partie suivante de ce rapport va présenter le développement de cette application intégrée à l'application déjà existante sur la DDU.

5. Application console ACE

i. Présentation & règles de codage de l'application

Suite à la validation de la solution « écran type ACE », j'ai donc commencé le développement de l'application destinée à cet écran. Pour cela, il aura fallu passer par une étape de compréhension du code sur lequel doit se greffer mon application de rétrovision. En effet, il était nécessaire de suivre la même structure afin de ne pas venir perturber le reste du code et faciliter sa relecture.

L'application de base déjà conçue pour la gamme « Coradia Export » ne contenait ni une partie de rétrovision (développement par Faiveley) ni de partie de vidéosurveillance. Il aura fallu importer les librairies concernées dans un premier temps.

La structure de cette application est clairement définie sous forme de dossiers et sous-dossiers dont les fonctionnalités sont définies par des règles. On y retrouve à la racine, le nom du projet puis l'appareil utilisé (en l'occurrence DMI ACE), puis s'en suit l'application en elle-même qui est contenue dans « App_main » (cf. figure 22). On y retrouve ensuite un premier dossier (Common) de fonctions et structures essentielles au fonctionnement du code qui seront utilisées à de nombreuses reprises (comme la fonction d'appel de visualisations). Puis on distingue le dossier (Global Variables) de variables globales, sous-classées par leurs fonctionnalités (configuration, état, temporaire, etc) qui permet l'échange de données entre les différentes fonctions. Ensuite se situe les dossiers (Models & Picto) d'images intégrées au projet (comme les boutons sur l'écran tactile) et les visualisations de base qui peuvent être utilisées dans différentes sous parties (par exemple l'affichage de l'heure). Enfin, on trouve le dossier (Screens) des visualisations en lui-même ; il s'agit là de toutes les fonctionnalités que l'on veut ajouter à l'application, c'est donc à ce niveau que je me suis intégré dans le projet.

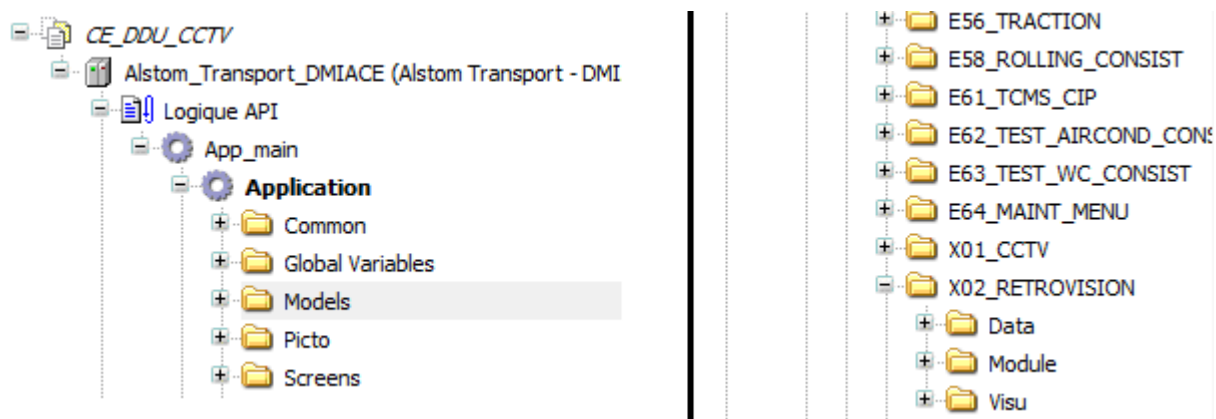


Figure 22: Structure application "Coradia Export" (gauche) & extrait dossier Screens (droite)

Une fois dans ce sous-dossier, on va observer trois parties :

- Data : contient éventuellement les structures (similaire langage C) nécessaires au code du sous-dossier.
- Module : contient le code de la fonctionnalité (face arrière ou cachée de l'IHM).
- Visu : contient les visualisations propres à la fonctionnalité (face avant de l'IHM).

En revenant sur le code en lui-même déjà présent, deux langages parmi les sept proposés par l'environnement CodeSys ont été utilisés : le langage texte structuré ainsi qu'un langage par blocs (cf figure 23). C'est donc de cette manière que j'ai dû construire mes propres visualisations.

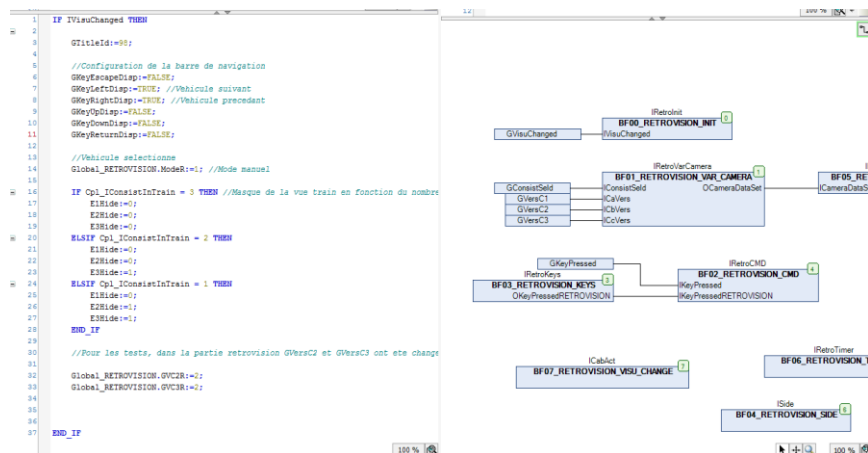


Figure 23: Langage texte structuré (gauche) et langage par blocs (droite)

Le langage texte structuré (ST dans la suite de ce rapport) est un classique de programmation, comparable au C, avec compilateur intégré dans l'environnement de programmation. Le langage par bloc lui à l'avantage d'être bien plus visuel, et permet de saisir rapidement l'intégralité du fonctionnement des différentes visualisations ainsi que l'interaction entre les différents blocs. Il ne peut cependant pas se suffire à lui-même, il est nécessaire de construire au préalable ces différents blocs (que l'on peut observer sur la figure 23) codés en langage ST.

Dans les différents documents internes, liés à la programmation d'application, que j'ai eu l'occasion de consulter, l'accent a été fait sur l'importance des commentaires afin de faciliter une relecture par une tierce personne. Il est stipulé qu'un commentaire global présentant la fonctionnalité d'un bloc doit être obligatoirement placé dès le début du code ST.

On notera aussi la présence de différentes variables réseau envoyées par le MPU et récupérées par le code de l'application déjà existante. Il est toutefois nécessaire de développer plus en détail la variable indiquant la cabine en service. En effet, le système embarqué du train va ordonner les différents éléments présents en leur donnant les noms C1, C2 et C3 (dans le cas d'une UM3) qui peuvent ne pas être toujours les mêmes éléments physiquement ; il s'agit d'un ordre logique. Dans chacun de ces éléments va se trouver deux cabines en extrémité qui seront nommées Ve1 et Ve2, ses noms restent physiquement au même point dans le train, peu importe sa configuration. Ainsi, la cabine en service sera une combinaison de ces deux informations, par exemple C1_Ve1 = vrai indiquera que la cabine en question est en service.

Enfin, les couleurs utilisées dans la face avant n'ont pas été choisies au hasard, elles sont imposées par des normes prédéfinies.

Ainsi, c'est en se basant sur ses règles et sur le code déjà existant que j'ai créé les visualisations des caméras de rétrovision ainsi que des caméras de surveillance. Le reste de cette partie sera présentée en deux temps pour chaque visualisations ; dans un premier temps la partie cachée au conducteur (ST, blocs) puis dans un second temps la partie visible avec laquelle le conducteur peut interagir.

ii. Code de la partie surveillance.

Bien que non directement liée à la rétrovision, cette partie de gestion des caméras de surveillance fût cependant un bon moyen d'appréhender la manière dont devait être designé une visualisation de caméras. Il s'agissait en effet de faire défiler les unes après les autres, toutes les caméras situées à l'intérieur du train, en prenant garde de mettre côte à côte les caméras situées dans une même zone (salle, plateforme d'accès), afin de permettre au conducteur de contrôler tous les véhicules ainsi que ses entrées.

Plusieurs modes de défilement des caméras sont alors possibles : un mode automatique qui va passer en revue toutes les caméras en suivant une temporisation prédéfinie, un mode manuel qui va suivre la même logique que le mode précédant, avec temporisation choisie par le conducteur et enfin un mode de sélection direct des caméras ou le conducteur va choisir simplement les caméras à partir d'une miniature train (présentée en face avant).

a. Data

On va donc retrouver dans ce dossier, les différentes structures nécessaires au fonctionnement de la visualisation ainsi que les différentes variables globales.

La structure clef de cette partie est la caméra. En effet, il s'agit d'associer un nom et une adresse URL à une même variable, puis, j'ai choisi d'associer toutes les caméras d'un élément dans une structure nommé « camera data » pour faciliter la construction des différents blocs.

Dans un second temps, suite aux différentes observations faites sur le code déjà existant, j'ai décidé de créer une structure « keys » qui a été complétée au fur et à mesure de la construction du reste du code. Elle va de pair avec deux blocs qui seront présentés ultérieurement dans ce rapport (BF02_CCTV_CMD et BF03_CCTV_KEYS) et regroupe toutes les touches sur lesquelles le conducteur peut appuyer.

Enfin, la liste des variables globales a été placée dans ce dossier. Cette dernière a aussi été complétée au fur et à mesure de l'avancement du reste du code ; elle permet un échange simplifié entre plusieurs fonctions et blocs ainsi qu'un regroupement de certaines variables clefs de la visualisation. Le nom des variables présentes dans cette partie a été choisi avec soin et commenté afin de comprendre rapidement leur utilité.

b. Module

C'est dans cette partie que se trouve tous les blocs fonctionnels (remarquables par leurs noms commençant par BFXX).

On retrouve en premier lieu le bloc d'initialisation qui s'exécutera une seule fois, au moment où le conducteur choisi de passer sur l'écran de visualisation des caméras. Son but est simplement d'initialiser les différentes variables graphiques et fonctionnelles qui vont être utilisées dans la suite comme par exemple l'ajustement du nombre d'élément dans la miniature train (qui sera présentée dans la partie face avant).

Puis on trouve le bloc d'affectation des caméras (VAR_CAMERA) : en entrée on a l'élément sélectionné par le conducteur (cas manuel) ou par la séquence (cas automatique) ainsi que les versions des différents éléments (PPM ou PPG – envoyé par le MPU) et en sortie on retrouve la structure « camera data » qui est alors affectée à l'élément d'entrée en fonction de sa configuration.

Les deux blocs suivants sont les blocs CMD et KEYS mentionnés dans la partie a. Ils font donc le lien entre les touches pressées par le conducteur et une action à effectuer dans les visualisations. Pour mieux comprendre leur utilité, nous allons considérer un exemple : lorsque le conducteur souhaite passer en mode de défilement manuel, c'est-à-dire lorsque ce dernier appuie sur le bouton affecté à cette tâche (voir **visualisation surveillance**). L'enchaînement des événements est alors décrit sur la figure suivante.

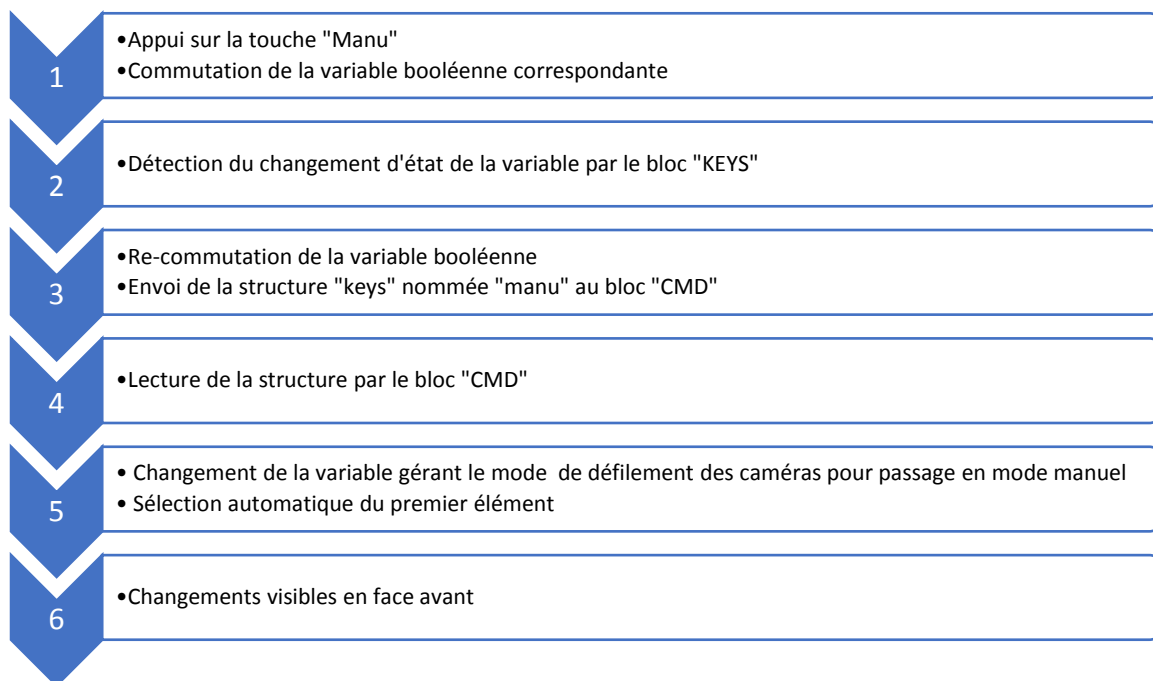


Figure 24: Etapes lors de l'appui sur la touche "Manu"

Bien que cette méthode semble lourde à mettre en place, elle permet de centraliser toutes les actions possibles pour le conducteur. Cela permet aussi de simplifier les propriétés des objets qui seront créés en face avant ; en effet, il suffira uniquement d'indiquer la commutation d'une variable lors d'un clic.

Le bloc suivant (DISPLAY) est le cœur de cette visualisation car c'est celui-ci qui va afficher le flux provenant des caméras. Il fonctionne de deux manières différentes selon le mode de défilement sélectionné. Si le mode manuel ou automatique est utilisé, alors une variable de séquence sera incrémentée et balayera alors toutes les caméras en service en partant d'une cabine d'extrémité du train. Le placement de ces caméras sera étudié dans la partie suivante (face avant). Ce mode tient compte de l'orientation des éléments (cf **schéma simplifié de configuration**) ainsi que du type de véhicule sélectionné (configuration des caméras différente s'il s'agit d'un véhicule avec cabine ou non). De plus, à chaque séquence, un groupe de variable est configuré pour la mise en valeur des caméras sur la miniature train (cf **face avant**). On remarquera aussi pour ce mode, la présence de conditions empêchant la divergence de la variable de gestion de séquence. Si le mode de sélection est utilisé, alors le conducteur devra sélectionner les caméras qu'il souhaite afficher en appuyant sur ces dernières dans la miniature train. Il doit, de plus, choisir un côté d'affichage (droite/gauche) via des boutons situés au-dessus des visualisations caméra. A noter que si une caméra est déjà en train d'être visualisée, le choix d'une nouvelle va écraser la précédente ; pour comparer le conducteur peut utiliser l'autre emplacement disponible.

Le bloc suivant (TIMER) est uniquement utilisé pour le mode automatique. En effet, c'est cette partie qui est en charge de gérer la temporisation entre les différentes séquences. On y retrouve une gestion des dépassements de la variable de séquence.

Enfin, le dernier bloc (VISU_TRAIN) est en charge de mettre en valeur la position du conducteur par rapport à la miniature train. Ce bloc se base sur une variable MPU qui indique à quel emplacement se trouve la cabine en service et à fortiori où se trouve le conducteur.

Une fois l'intégralité de ces blocs codés et définis dans l'application, il faut créer un programme les appelant ; P01_CCTV. Ce dernier va simplement permettre la connexion présentée dans la figure 23 (à droite) en reliant toutes les différentes entrées et sorties des blocs créés. A noter que si un bloc n'est pas utilisé dans P01_CCTV, celui-ci n'aura aucuns effets sur l'application ; bien que compilé, il sera ignoré.

iii. Visualisation de la partie surveillance

La face avant de la fonctionnalité de caméras de surveillance a été découpée en différentes zones qui seront développées à la suite de la figure.

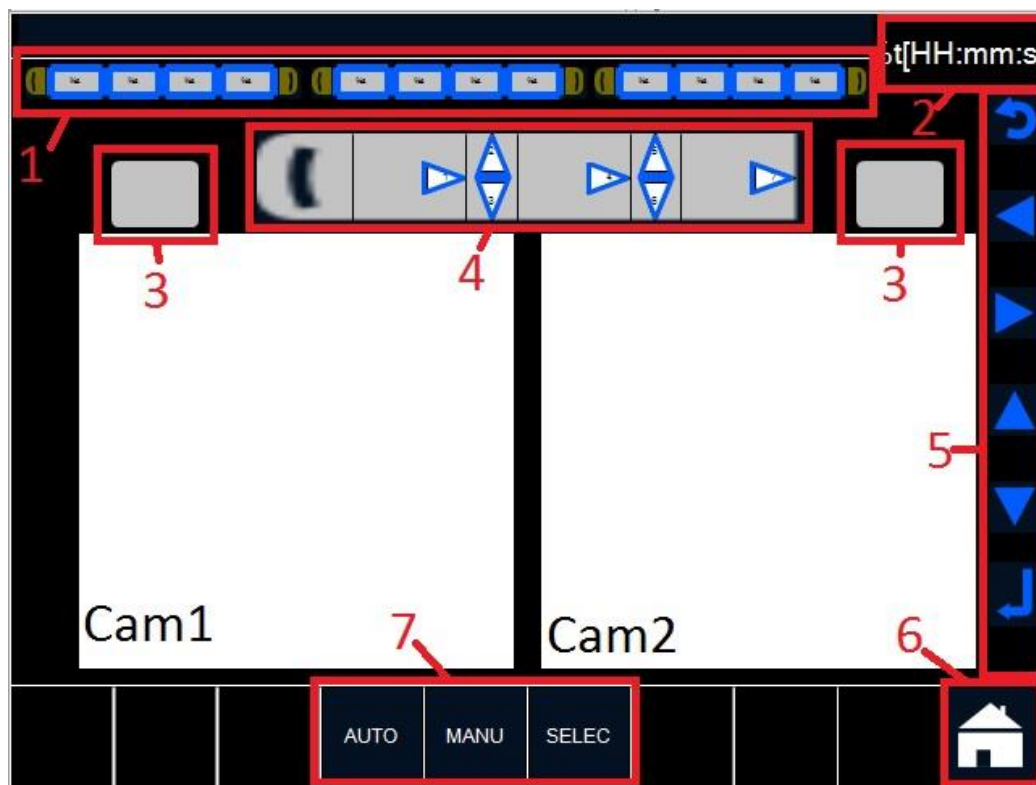


Figure 25: Face avant IHM - partie surveillance

On remarque directement la présence des zones réservées qui ont été évoqués dans la partie de tests de construction d'une application petit écran (**zones réservées**), de plus, certaines parties de ces zones sont obligatoirement affichées. En effet, la zone 2 qui indique l'heure système est présente sur toutes les visualisations de l'application et sert aussi d'indicateur de gel de l'écran. La zone 6 est elle aussi toujours réservée par la même fonctionnalité : le retour à l'écran principal, bouton « home ». Dans le cadre du développement de notre application, ce dernier va dans le même temps couper tous les flux vidéo ainsi que les potentiels modes de zoom.

Toujours dans le même registre se trouve la zone 5. Cette dernière est aussi une zone réservée mais qui offre plus de libertés, il est en effet possible de n'afficher que les boutons qui seront utiles parmi les 6 possibles (retour, gauche, droite, haut, bas, entrée). Dans le cadre du développement de l'application, seuls les boutons gauche/droite ont été utilisés ; ils permettent au conducteur de passer à la séquence de caméras précédente/suivante en mode manuel.

Enfin, dans la même grande zone du bas de l'application (qui contient la zone 6) se trouve la zone 7. Comme les textes présents peuvent le laisser supposer, il s'agit des boutons de changement du mode d'affichage. A noter que ces derniers stoppent aussi le flux en cours.

Le point commun entre toutes ces zones d'interaction (zones 5-6-7) est qu'elles sont doublées par des boutons physiques présents sur les côtés de l'écran (voir image en annexe). Un appui sur l'écran ou sur ces boutons aura le même effet dans l'application, cela peut palier à une défaillance du mode tactile éventuel et permet au conducteur de visualiser tout de même les caméras du train, avec un mode de sélection qui n'est cependant plus opérationnel car il dépend du mode tactile.

Parmi les zones « libres » se trouve alors la zone 1 qui est la miniature globale du train. Elle permet simplement au conducteur de situer les caméras ainsi que sa position par rapport à l'ensemble du train. Ses fonctionnalités en détail sont les suivantes :

- Miniature dynamique : selon la configuration du train, la miniature va être adaptée. En effet, elle a été désignée en tenant compte du nombre d'éléments. Sur la figure 25, le train est en UM3 ; dans le cadre d'une US ou UM2, certaines parties de la miniature vont être cachées. Le type d'élément est aussi pris en compte ; s'il s'agit d'un PPM ou d'un PPG, le nombre de véhicules ne sera pas le même, et la miniature suivra cette modification.
- Affichage de la cabine : la cabine en service sera mise en valeur par un changement de couleur, passage en couleur bleue (actuellement couleur brune aux extrémités des éléments sur figure 25). A noter que cette fonctionnalité tient compte du sens de l'élément pour situer correctement la cabine en service afin d'être représentatif de la réalité train.
- Mise en valeur du véhicule sélectionné : sur la figure 25, tous les éléments sont considérés comme sélectionnés (cela au dû au fait que l'application ne communiquait pas avec le réseau lors de la capture d'écran) mais dans le cas d'un fonctionnement normal, seul le véhicule dont les caméras sont actuellement visualisées sera de couleur bleue.
- Placement des éléments : dans le cadre des caméras de surveillance, il a été choisi de placer l'élément C1 toujours à gauche de la miniature (on notera la différence avec le mode de rétrovision dans les parties suivantes).
- Interaction : tous les véhicules sont cliquables dans le cadre du mode de sélection.

La deuxième zone liée à cette miniature est la zone 4 qui peut être considérée comme un zoom de la zone 1. En effet, il s'agit de montrer précisément à partir du véhicule mis en valeur en zone 1 ou se trouve physiquement les caméras dans ce dernier. Cet affichage se fait de manière dynamique selon le véhicule. C'est à partir du plan d'intégration des caméras que cette miniature a été conçue mais c'est aussi ces plans qui ont dictés la manière dont les séquences de caméras ont été organisées. Par exemple, la caméra à gauche est visualisée seule en zone « Cam1 » lorsque la séquence est initialisée, puis lors de l'incrémentation de cette variable, ce sont les deux caméras suivantes (vers la droite) qui sont visualisées aux zones « Cam1 » et « Cam2 » et de la même manière jusqu'à la fin du véhicule. Ainsi, deux caméras qui se font face seront toujours visualisées côte à côte. De plus, ces caméras sont mises en valeurs en les entourant de couleur bleue et sont cliquables dans le mode de sélection. La zone 3 est alors liée à cette fonctionnalité et va dicter dans quelle zone va être affichée la caméra sélectionnée.

iv. Code de la partie rétrovision

La partie rétrovision fonctionne de la même manière globalement que la partie de surveillance, on y retrouvera une grande partie de points de communs avec cependant quelques nuances de conception étant donné un cahier des charges plus précis.

Les modes de défilement des caméras sont semblables à la partie précédente, on notera néanmoins que les séquences seront agencées de manière à se mettre en accord avec le cahier des charges.

Une précision de départ importante est la dynamique complexe de la miniature train. En effet, il a été demandé de concevoir une miniature train qui suit une logique spécifique et non la logique de numérotation MPU (cf. **variable MPU**). Il faut en effet que la cabine en service se trouve toujours la plus à gauche possible, quel que soit la configuration du train mais en gardant la réalité de ce dernier. Par exemple, il n'est pas possible d'avoir deux véhicules devant soit en miniature train alors que dans la réalité se trouve seulement un véhicule devant la cabine. De plus, il a été demandé que les caméras soient toujours visualisées selon la même logique ; du point dans le train le plus en avant de la cabine en service vers le point le plus en arrière.

a. Data

On retrouve dans cette partie, la même structure « camera » que précédent, ainsi que la structure « camera data » qui est néanmoins adaptée au mode de rétrovision (moins de caméra, différente dénomination). On notera que pour la rétrovision, la structure « camera data » sera uniquement utilisée pour le mode de sélection, les deux autres modes seront gérés par deux autres structures.

La différence majeure avec la partie de surveillance porte sur les deux nouvelles structures « Element » et « Train ». La première est composée des six caméras possibles par élément (cf. **figure 13**) et le second n'est autre qu'une composition de ces éléments (jusqu'à trois). Il s'agissait en effet de simplifier la gestion de toutes les configurations trains possibles, point qui était très difficilement faisable avec la même approche que pour la partie surveillance. C'est donc une approche programmation « objet » qui a été pensée (on note qu'il s'agit uniquement d'un mode de pensée et non une implémentation directe en langage objet, impossible en ST).

On retrouve aussi la structure « keys » qui regroupe les touches d'interaction avec le conducteur du train et qui sera en lien avec les blocs « RETROVISION_KEYS » et « RETROVISION_CMD ».

Enfin, on distingue la liste des variables globales liées à la rétrovision. Pour éviter certaines confusions avec des variables liées à la surveillance (comme par exemple pour les modes automatique et manuel), la lettre R a été rajoutée à la fin des variables potentiellement problématiques afin de les différencier aisément.

b. Module

Une particularité de cette partie est l'utilisation de fonctions qui seront elle-même utilisées dans les blocs. Elles permettent un allègement du code présent dans ces blocs (sans quoi la partie d'affichage serait bien trop importante, une dizaine de milliers de lignes de code contre seulement 1300 avec les fonctions). Ainsi, trois fonctions ont été développées :

- **F_SELEC_CAMERA** : possède globalement la même fonctionnalité que le bloc **VAR_CAMERA**, elle va adresser les caméras avec leurs noms et leurs adresses IP dans la structure « camera data ». Cette fonction a été principalement créée pour une meilleure synergie avec les fonctions suivantes.
- **F_CONF_TRAIN** : cette fonction possède la lourde tâche d'ajustement de la visualisation train. A partir du nombre de véhicules dans le train, l'emplacement de la cabine en service et le sens de chacun des éléments, la fonction va créer un byte en sortie qui va imposer l'ordre de la miniature train en face avant. Ce byte est représenté en figure 26 ; la partie 1 est la configuration MPU (qui est un argument de **CONF_TRAIN**), avec cabine en service en C2, à droite. La sortie de la fonction sera alors la configuration 2, ou les éléments sont inversés du point de vue MPU, mais la réalité train est respectée avec une cabine à gauche. De plus, le programme va aussi être en charge de la mise en valeur de l'emplacement de la caméra du poste de conduite actuel (cf. face avant).
- **F_CONF_CAM_TRAIN** : cette fonction est tout aussi critique que la précédente car elle va assigner les caméras (récupérées par la fonction **SELEC_CAMERA**) aux bons emplacements de visualisation en face avant à partir de la configuration de sortie de la fonction **CONF_TRAIN**. En effet, même pour un élément donné, selon son orientation, les caméras visualisées à un même emplacement ne seront pas forcément les mêmes.

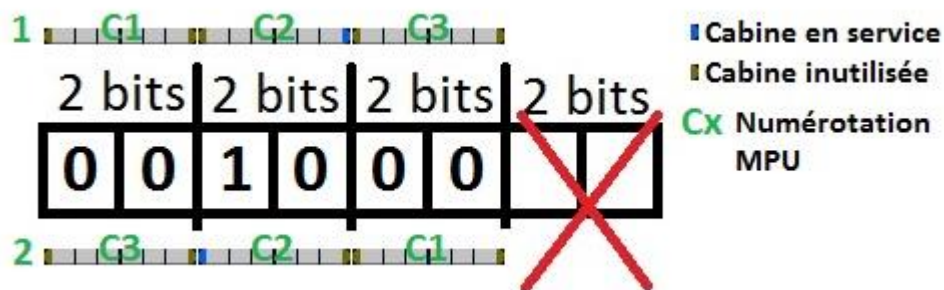


Figure 26: Sortie de la fonction **CONF_TRAIN**

On retrouve dans un premier temps le bloc d'initialisation (**INIT**), en charge des mêmes fonctionnalités que son équivalent pour les caméras de surveillance. Il initialise aussi le mode par défaut qui est le mode manuel.

Puis, se trouve encore une fois le même bloc d'affectation des caméras (**VAR_CAMERA**) adapté au mode de rétrovision. On note cependant que ce dernier ne sera utile que pour le mode de sélection, les deux autres modes utiliseront les fonctions spécifiques présentées au début de cette partie.

On distingue ensuite les blocs **CMD** et **KEYS**, qui fonctionnent de la même manière que pour la surveillance.

On trouve ensuite un bloc spécifique à la rétrovision (SIDE) qui va détecter le côté où se trouve le quai. En effet, les données des capteurs des portes (qui sont des variables MPU) vont être lues et interprétées pour permettre un affichage automatique des caméras de rétrovision du bon côté ou vont donc monter les passagers. Ce bloc gère aussi la mise en valeur du signallement de côté droit/gauche (voir **face avant**). On notera que seuls les modes manuel et automatique vont utiliser cette fonctionnalité ; en effet, dans le mode de sélection, le côté est choisi directement par le conducteur.

Le cœur de cette partie est donc le bloc d'affichage des caméras (DISPLAY). Il fonctionne en deux temps, le premier est le mode automatique et manuel. Cette partie va utiliser les fonctions définies précédemment ; en effet, il s'agira uniquement d'afficher le bon côté des caméras selon la détection du quai (bloc SIDE), les caméras sont effectivement déjà préconfigurées par la fonction CONF_CAM_TRAIN. Le défilement des éléments se fera alors par séquence, de la même manière que pour la partie de surveillance. On aura donc les caméras de deux éléments successifs à chaque affichage (sur deux lignes, cf. face avant), ce qui implique donc un maximum de deux séquences (UM3 étant la configuration maximale). Pour le deuxième mode de fonctionnement, le conducteur va cliquer sur les éléments ainsi que sur un côté pour pouvoir afficher les caméras. Pour pouvoir arrêter l'affichage de ces dernières, il suffit alors de recliquer sur l'élément en question et la zone sera libérée. On notera que ce bloc gère aussi la mise en valeur des éléments sélectionnés dans les deux modes.

Le bloc suivant (TIMER) est uniquement en charge de la temporisation entre deux séquences en mode automatique ainsi que de la gestion du dépassement de la variable de séquence.

Le dernier bloc (VISU_CHANGE) va interpréter la sortie de la fonction CONF_TRAIN pour adapter la visualisation de la miniature train en face avant.

v. Visualisation de la partie de rétrovision

La face avant de la fonctionnalité de caméras de rétrovision a été découpée en différentes zones qui seront développées à la suite de la figure.

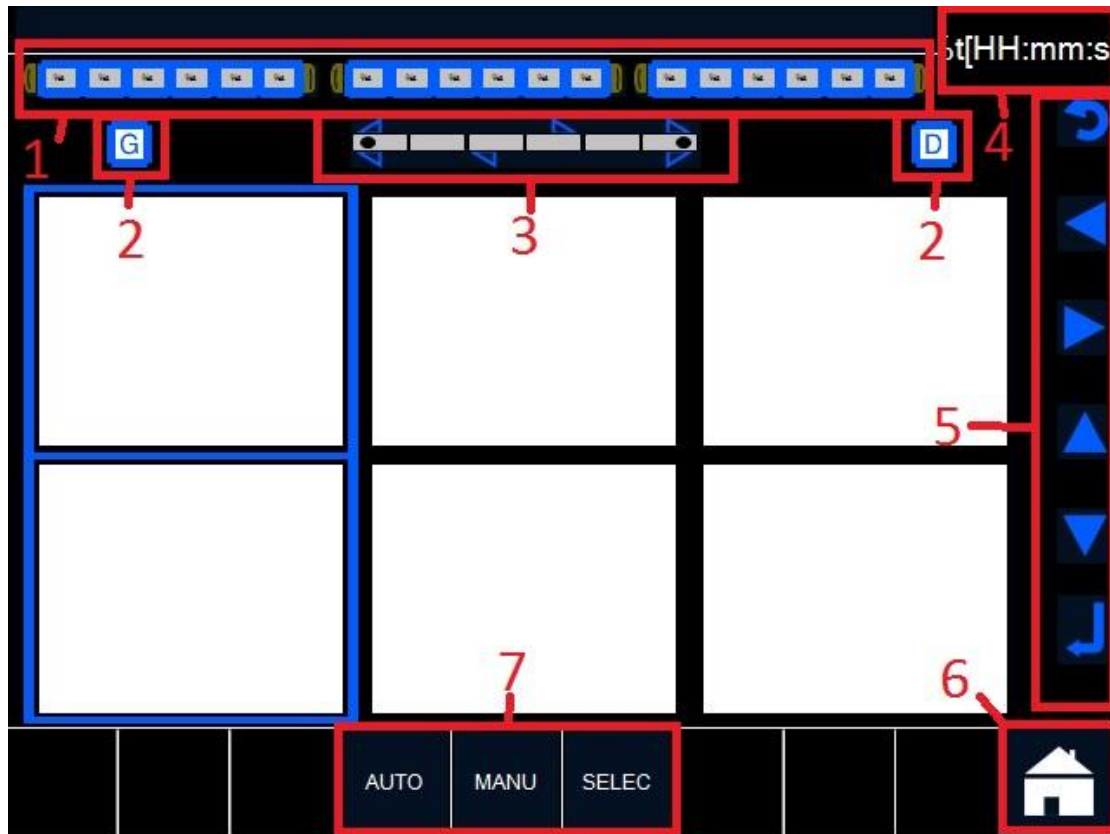


Figure 27: Face avant IHM - partie rétrovision

Les zones 4, 5, 6 et 7 sont communes à la partie de surveillance et ne seront pas reprises dans cette partie. Les touches droite/gauche de la zone 5 servent aussi au défilement des séquences en mode automatique et manuel. Elles sont aussi doublées par des touches physiques.

La zone 1 est la miniature du train, mentionnée à de nombreuses reprises dans la partie précédente. Elle permet au conducteur de se situer et de situer les véhicules dont les caméras sont en train d'être visualisées. Ses fonctionnalités sont les suivantes :

- Miniature dynamique : à partir des différents blocs et fonctions précédents, la miniature va être ajustée selon la configuration du train.
- Affichage de la cabine : lorsqu'une cabine est en service, elle passera en couleur bleu (cf. figure 26). Elle est située de manière dynamique à partir des fonctions précédentes.
- Mise en valeur de l'élément sélectionné : sur la figure 27 tous les éléments sont sélectionnés, cependant en fonctionnement nominal, deux éléments sont mis en valeur (en UM2 & UM3) au maximum et correspondent aux deux lignes des visualisations de caméras.
- Interaction : tous les véhicules sont cliquables dans le cadre du mode de sélection.

La zone 2 indique simplement le coté visualisé (mode auto et manuel) ainsi que le coté choisi en mode de sélection (les boutons sont alors cliquables). La zone 3 est en complément et sert simplement à rappeler au conducteur où se trouve les caméras.

vi. Phase d'intégration Laborame

Suite au design de l'application, il était alors nécessaire passer par la phase d'intégration au Laborame. Seule la rétrovision sera présentée dans cette partie, il sera cependant possible de retrouver les mêmes informations en annexes pour la partie de surveillance (graphique des différents tests).

Cette partie ne précisera pas toutes les corrections logiciels diverses qui sont apparues au fur et à mesure des tests. Cependant, elle présentera certaines pistes d'améliorations potentielles qui n'ont pas pu être mises en œuvre lors de mon stage.

a. Cahier des charges

La première vérification qui a été opérée sur le Laborame, une fois l'application entièrement opérationnelle fût la cohérence de l'application avec le cahier des charges. Le tableau ci-dessous regroupe les différentes valeurs mesurées. La méthode de mesure de la latence sera expliquée en annexe.

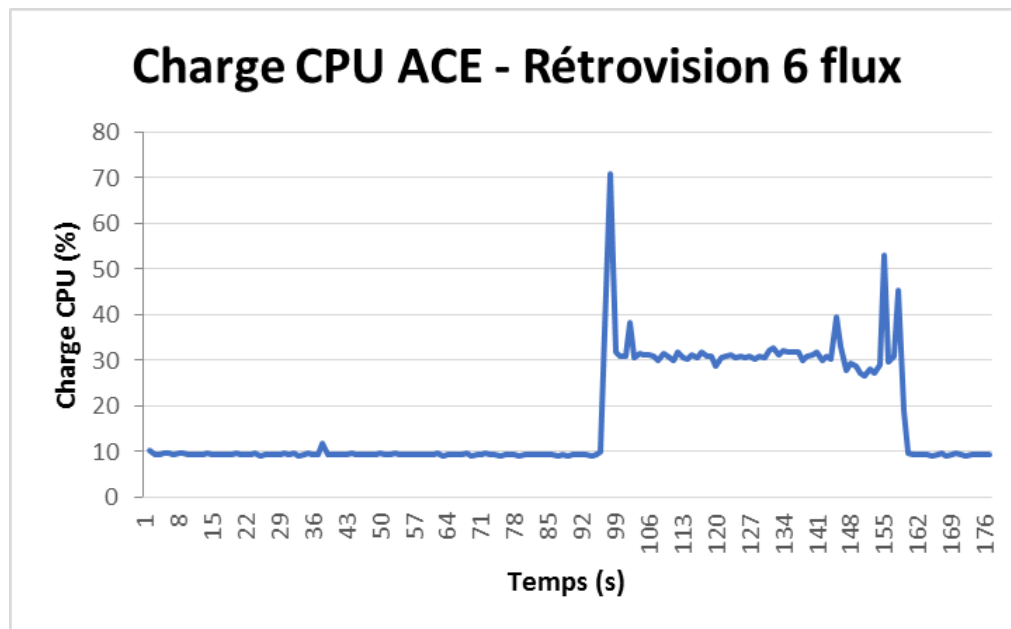
Valeur mesurée	Application en condition Laborame	CDC
Temps accès à l'écran	2 s si caméras déjà chargées et 3-4 s sinon	3 s
Temps changement de côté	3-4 s	3 s
Temps basculement zoom et inversement	< 1 s	1.5 s
Temps de latence maximal	230 ms	500 ms
Nombre minimal d'images par seconde	30 (source, non garanti à l'écran)	12
Définition	176x144 – 640x480 en zoom	352x288
Nombre de flux simultanés	6	6
Rapport taille personne/taille de la vignette	> 5%	5%

Tableau 3: Comparaison performance application & cahier des charges

On remarque donc que l'application créée est cohérente avec le CDC et apporte même certaines améliorations. Cependant, une valeur devrait être améliorée ; le temps d'accès aux six flux lors d'un accès à l'application ou lors d'un changement de côté. En effet, ce temps s'est établi à quelque reprises au niveau de 4 secondes, ce qui dépasse la valeur préconisée par le CDC. Deux pistes sont à explorer : une optimisation de mon code (qui a été sensiblement opérée par mes soins) et une optimisation des fonctions de visualisations des flux vidéos qui sont actuellement synchrones (le programme attend que le flux soit récupéré pour continuer l'exécution).

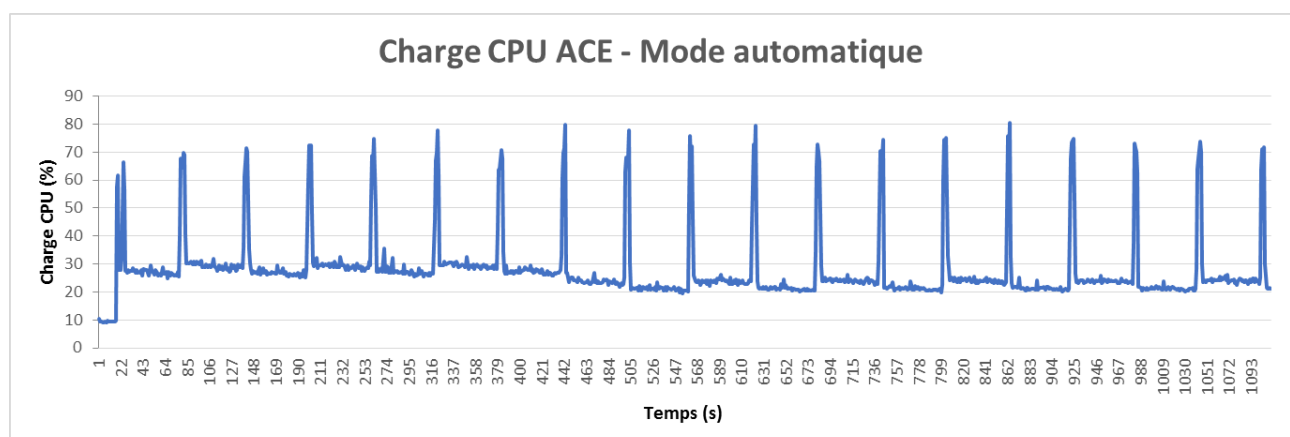
b. Charge CPU de la console

Dans le but de ne pas dégrader les autres fonctionnalités de la console ACE, il m'a été demandé d'étudier la charge CPU à divers instants et lors de certaines interactions avec l'application. La méthode pour obtenir ces données sera présentée en annexe.

*Figure 28: Charge CPU - Rétrovision*

Le plateau initial (jusqu'à 98 s) peut être considéré comme l'offset de l'application globale fonctionnant sur la console. Puis, on retrouve un pic (environ 70%) qui correspond à la connexion avec les six flux des caméras de rétrovision, ce pic ne dépassait pas les 80% de charge suite aux différents tests et est toujours très court (1 s maximum). On distingue ensuite à nouveau un plateau, qui est la charge stabilisée (30%) lorsque les flux sont en cours de lecture. Cette charge reste raisonnable en considérant le nombre de flux visualisés, le zoom sur l'un des vignettes n'a que peu d'importance. Il s'agit en effet du pic final (50%) qui est directement suivi de la déconnexion des flux et du retour à l'écran principal.

La figure 29 montre l'utilisation CPU sur la durée (18 minutes) en mode automatique sans aucune intervention. A remarquer la légère baisse d'utilisation à partir de 430 s qui est dû au plantage d'une caméra.

*Figure 29: Charge CPU - Rétrovision automatique*

c. Charge réseau entrante sur la console

Le cahier des charges « Regiolis » indique que la rétrovision ne devait pas occuper plus de 2000 trames/seconde sur le réseau embarqué du train. Etant donné la taille des flux vidéos, il était fort peu probable de dépasser cette valeur. J'ai cependant mesuré la charge réseau entrante (RX) sur la console à « vide », sans aucuns flux, puis avec les six flux afin de pouvoir affirmer que l'application ne dépassait pas la valeur seuil.

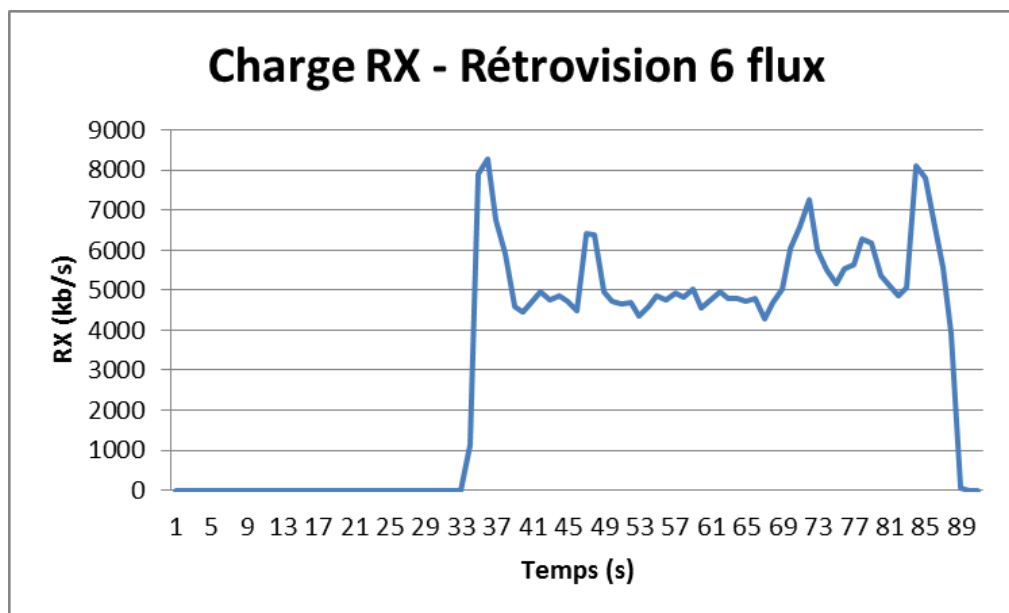


Figure 30: Charge réseau entrante sur ACE

On remarque directement que l'offset à vide (environ 7-8 kb/s) a été soustrait à la charge entrante (ce qui explique le plateau initial à 0 kb/s). On se retrouve donc à un flux entrant culminant à 8280 kb/s et ayant pour moyenne 5325 kb/s. Ainsi, en considérant la longueur maximale d'une trame (1500 octets), on se retrouve donc à 22% du maximum des 2000 trames/seconde (soit 5.3mb/s sur 36 mb/s).

6. Comparaison avec le système actuel

Le tableau suivant regroupe différentes caractéristiques du cahier des charges, comparées avec le système existant et le système que j'ai développé.

Valeur mesurée	Application Laborame	Application Faiveley	Cahier des charges
Temps d'accès à l'écran	2 s si caméras pré-chargées et 3-4 s sinon	2 s	3 s
Temps changement coté	3-4 s	1 s	3 s
Temps basculement zoom (et inverse)	< 500 ms	~ 500 ms	1,5 s
Temps de latence maxi	230 ms	500 ms	500 ms
Nombre d'images par seconde	30 (source)	12	12
Définition	176x144 – 640x480 en zoom (vrai zoom)	352x288 – 704x576 en zoom (agrandissement image)	Non explicité (352x288 comme base)
Nombre de flux simultanés	6	6 / écran	6
Rapport taille personne / taille de l'image	> 5%	> 5%	> 5%

Tableau 4: Comparaison CDC - système existant - nouveau système

On peut remarquer que le temps d'accès à l'écran et le temps de changement de côté sont supérieurs aux temps de l'application actuelle. En plus des points soulevés dans le paragraphe vi.a qui peuvent expliquer cet écart, on note que le système actuel est indépendant et ne fait fonctionner que les flux vidéo ; l'écran de visualisation est dédié à cette tâche, contrairement au système que j'ai construit.

Cependant, le temps de basculement en mode zoom (et inversement) est bien meilleur que l'application actuelle, ainsi que la latence qui est grandement réduite (le signal ne passe plus par un serveur de conversion).

On obtient aussi une nette amélioration du nombre d'images par seconde, qui n'est néanmoins pas garanti à l'écran (uniquement en source).

Enfin, l'amélioration la plus remarquable est le zoom qui n'est plus un agrandissement de l'image comme le faisait l'application actuelle, mais directement un flux de la taille de l'écran. En effet, comme précisé dans ce rapport, les caméras envoient un flux d'une définition de 640x480 sur le réseau, qui est redimensionné pour les miniatures, mais qui est alors en natif lorsqu'un zoom est demandé.

Les figures suivantes comparent visuellement les deux solutions. On remarque que la solution actuelle possède un meilleur rendu en miniature, mais le zoom est de mauvaise qualité et n'est pas en plein écran. De plus, la solution Faiveley est en niveau de gris non pas pour des raisons technologiques mais pour des raisons de contraste et de cohérence avec le CDC client.

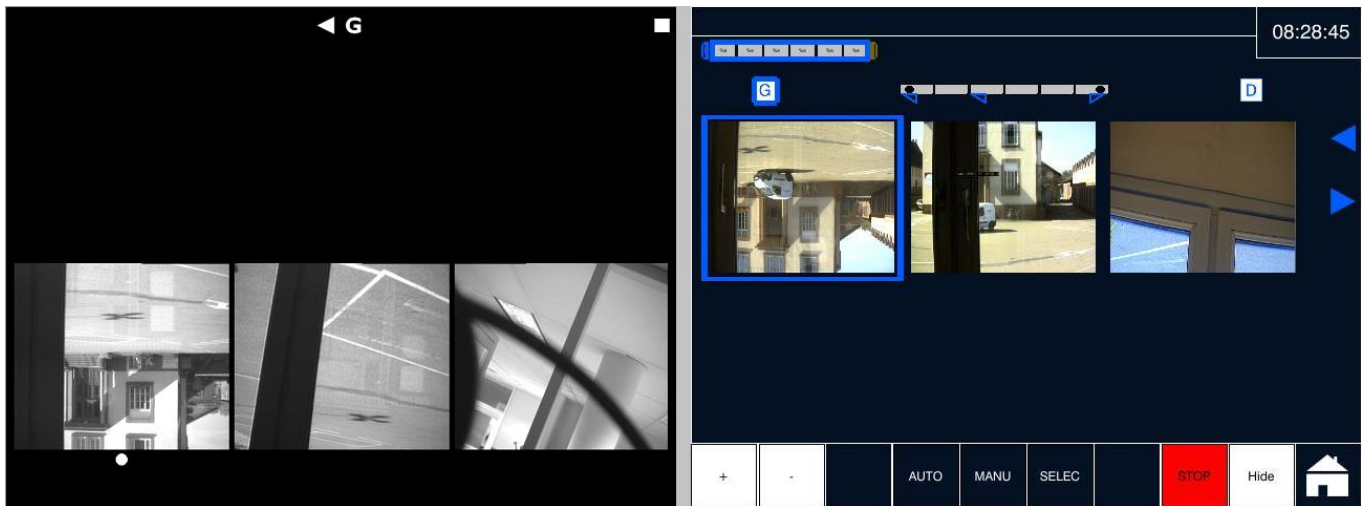


Figure 31: Comparaison zoom système actuel (gauche) et système ACE (droite)

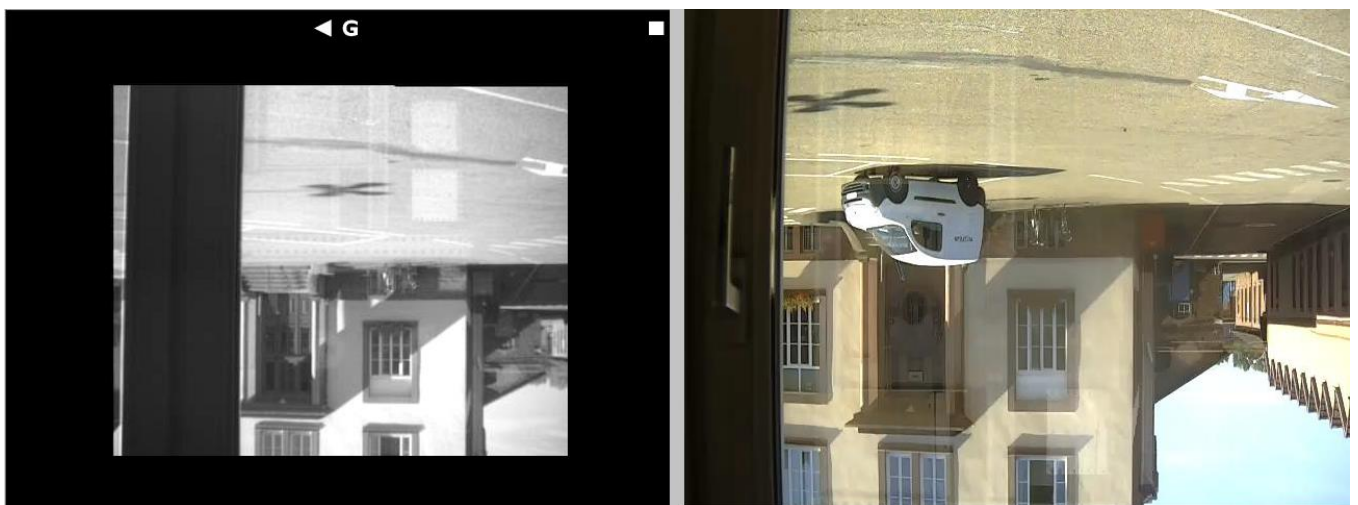


Figure 32: Comparaison miniatures système actuel (gauche) et système ACE (droite)

Conclusion

Dans ce projet, j'ai donc étudié la mise en place d'un nouveau système de rétrovision sur la gamme « Coradia Export », ses différentes possibilités d'implantation dans le train, ses nombreuses implications au niveau système mais aussi au niveau logiciel puis j'ai procédé aux phases de test de sélection d'une solution. S'en suit ainsi, le développement de la dite solution ainsi que de nombreux tests en conditions réelles en laboratoire. Dans la continuité du projet, il sera nécessaire de procéder à des tests directement sur train afin d'inclure le facteur environnement, qui était globalement maîtrisé en laboratoire (humidité, chaleur, luminosité, etc).

Durant cette période de stage, j'ai su mettre en œuvre et renforcer mes capacités d'organisation et d'adaptation ainsi que mon autonomie. Afin de mener à bien mon projet, je n'ai pas hésité à me renseigner auprès des équipes qui m'entouraient et j'ai su être à l'écoute des éventuelles remarques quant à mon avancement. La mise en place du nouveau système de rétrovision a aussi renforcé mes compétences et connaissances techniques, acquises lors de mon cursus scolaire, particulièrement en informatique et systèmes embarqués. Cette expérience au sein d'Alstom m'a conforté dans mon projet d'élaborer une carrière dans le monde industriel.

Bibliographie & webographie

Alstom – Activity report 2016/2017 - <http://www.activity-report.alstom.com/index.php/en/>

Alstom – Product catalogue - <http://www.alstom.com/fr/products-services/product-catalogue/>

Alstom – Reichshoffen Panorama

Alstom – RSAD Rétrovision Algérie & Regiolis

<https://www.computerhope.com/unix.htm>

<https://www.codesys.com/support-training/self-help/codesys-manual.html>

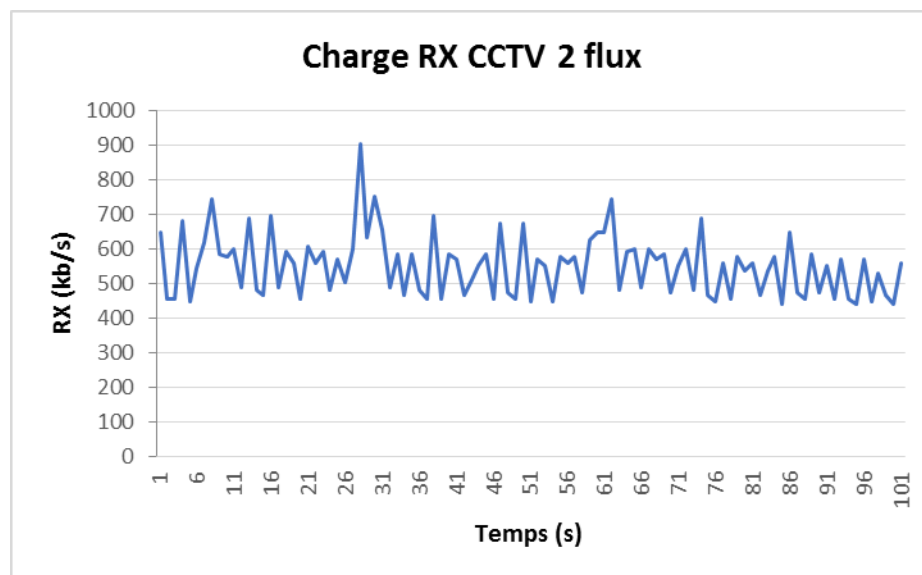
https://www.moxa.com/product/VPort_P06-1MP-M12.htm

<https://www.axis.com/us/en/learning/web-articles/technical-guide-to-network-video/compression-formats>

Annexes

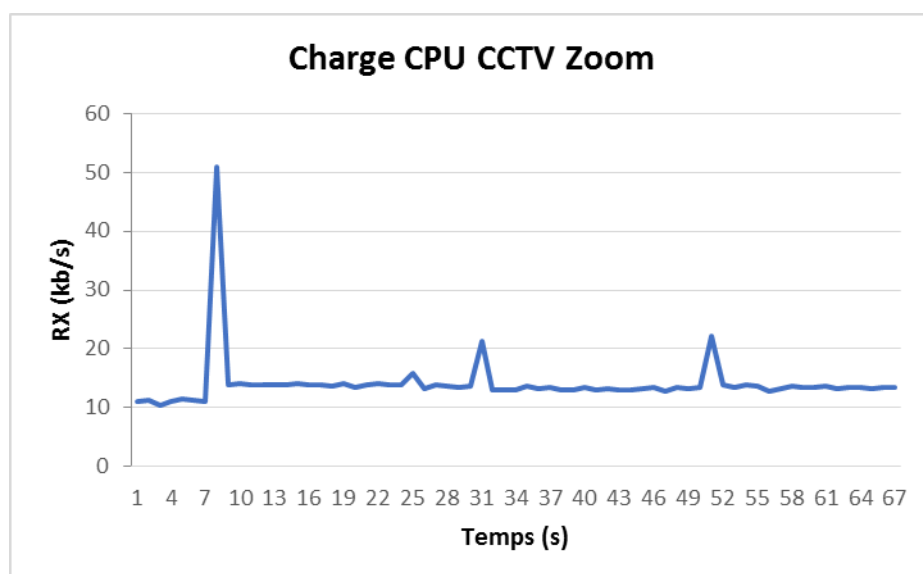
Performances application surveillance

De la même manière que pour la partie de rétrovision, des tests ont été opérés pour analyser les performances de l'application.

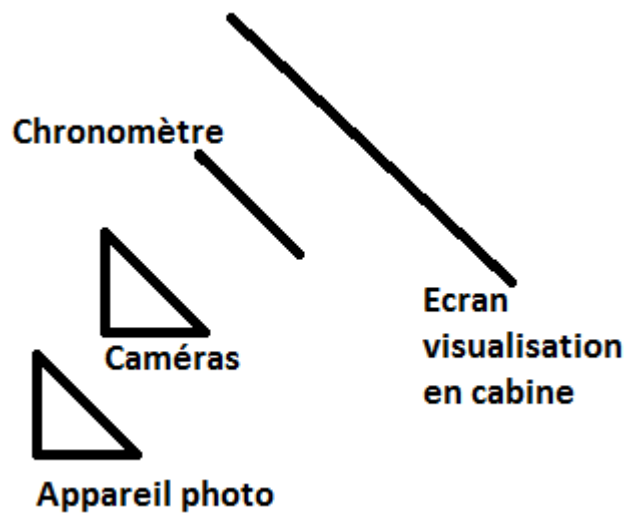


Charge bien plus faible que pour la rétrovision, explicable simplement par la présence de 2 flux seulement.

De la même manière, le nombre faible de flux se répercute aussi sur la charge CPU. Les deux pics sont dus au zoom, et le pic initial est dû à l'acquisition des flux.



Mesure de la latence



Ce schéma ci-contre présente la méthode qui a été utilisée pour mesurer la latence induite par le système.

Les caméras filment un chronomètre, l'appareil photo doit avoir dans son cadre à la fois le chronomètre réel et le chronomètre filmé, visible sur l'écran. Il suffit alors de faire la différence entre les deux valeurs sur la photo.

Cette technique est limitée en précision par la fréquence de rafraîchissement des différents appareils (60Hz pour l'écran de visualisation).

Mesure de la charge CPU

La méthode utilisée pour mesurer la charge CPU a été relativement simple à mettre en place. En effet, les écrans de visualisation fonctionnent sous Unix, la commande « cyclesoak » a donc été utilisée. Elle permet d'afficher le % de CPU en cours d'utilisation et de préciser le nombre de cœurs sur la machine. J'ai ensuite envoyé le résultat de cette commande dans un fichier texte pour pouvoir ensuite tracer le graphique de la charge CPU.

Boutons physiques sur DMI ACE (semblable EVOL2)