

Laboratoire 1

Document de Vision

Dash Display

Dash Display

Vision

Version 1.0

Historique

Table des matières

1. Introduction 5

1.1 Objectif 5

1.2 Portée 5

1.3 Définitions, acronymes et abréviations 6

1.4 Références 8

2. Positionnement 8

2.1 Énoncé du problème 8

2.2 Positionnement du produit 9

3. Descriptions des intervenants et des utilisateurs 10

3.1 Résumé des intervenants 10

3.2 Résumé des utilisateurs 11

3.3 Environnement utilisateur 12

3.4 Principaux besoins des intervenants et utilisateurs 12

4. Vue d’ensemble du produit 14

4.1 Perspective du produit 14

4.2 Principaux avantages 14

4.3 Hypothèses et dépendances 15

4.4 Licences et installation 15

5. Caractéristiques du produit 15

5.1 CAR01 - Afficher les données demandées au pilote 15

5.2 CAR02 - Afficher les données demandées aux ingénieurs 15

5.3 CAR03 - Offrir une interface modifiable en mode ingénieurs 16

5.4 CAR04 - Afficher l’interface en mode paysage pour le pilote 16

5.5 CAR05 - Recevoir les données en tout temps 16

5.6 CAR06 - Permettre la transition entre les deux modes avec un bouton 16

5.7 CAR07 - Recevoir les données directement de la voiture 16

5.8 CAR08 - Afficher les textes de l’application en anglais 16

5.9 CAR09 - Afficher l’application du pilote dès l’ouverture 17

5.10 CAR10 - L’application doit utiliser des technologies légales 17

5.11 CAR11 - Afficher les dernières données lors de la perte du signal 17

5.12 CAR12 - Mettre en veille le mode pilote après 3 secondes sans données 17

5.13 CAR13 - Permettre au pilote de changer les interfaces 17

5.14 CAR14 - Empêcher la mise en veille de l’application 17

5.15 CAR15 - Utiliser les configurations du fichier XML 18

6. Contraintes 18

7. Gammes de qualité 18

8. Attributs des caractéristiques 18

9. Autres exigences du produit 20

9.1 Standards applicables 20

9.2 Exigences du système 20

9.3 Exigences de performance 20

9.4 Exigences environnementales 21

10. Exigences de documentation 21

10.1 Manuel de l’utilisateur 21

10.2 Aide en ligne 21

10.3 Guides d’installation, de configuration et fichier à lire 21

Annexes 22

A Attributs des caractéristiques 22

État 22

Bénéfice 22

Effort 22

Risque 23

Stabilité 23

Priorité 23

# 1. Introduction

## 1.1 Objectif

Le document de vision a pour but de recueillir, d’analyser et de définir les besoins et les fonctionnalités de haut niveau pour une application mobile destinée à un club étudiant de l’École de technologie supérieure.

Le produit traité dans ce document consiste en un tableau de bord destiné au pilote et aux ingénieurs de piste pour la voiture de la Formule ÉTS. Les modules installés sur le véhicule et l’application doivent communiquer ensemble pour que les différents intervenants reçoivent rapidement les données récoltées par les capteurs.

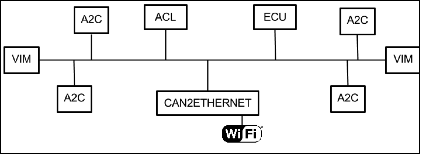
Dans les sections suivantes, le positionnement et les descriptions des intervenants sont présentés. Par la suite, une vue d’ensemble du projet est établie pour déterminer si les besoins formulés sont comblés. Finalement, les caractéristiques, les contraintes et les exigences sont abordées pour fournir de plus amples détails sur le produit.

## 1.2 Portée

Ce document porte spécifiquement sur les deux premières phases du projet « Dash Display », soit l’interface pilote et l’interface ingénieur de piste. Les autres phases vont être complétées ultérieurement par la Formule ÉTS.

Présentement, l’acquisition de données est effectuée par un système nommé MoTeC ACL. La seule méthode de visualisation des données en temps réel est l’utilisation du mode « debug » de l’ACL Manager. De plus, l’acquisition des données détaillées s’effectue une fois la voiture arrêtée à l’aide du logiciel MoTeC i2 Pro. Cependant, le pilote n’a pas accès aux données durant les tests et les ingénieurs de piste requièrent un ordinateur pour les visualiser.

La solution proposée est divisée en deux parties distinctes. Une visualisation des données importantes en temps réel pour le pilote et une présentation des données plus complète pour les ingénieurs supervisant les tests de roulage. La portée du projet porte sur les données provenant du module fourni par le club Formule ÉTS jusqu’à leurs affichages sur l’application du pilote et des ingénieurs de piste.



**Figure 1 : Diagramme des modules de communication**

Le diagramme précédent représente les différentes unités connectées au véhicule qui envoient les informations des capteurs en signaux CAN. Par la suite, le module Can2Ethernet envoie les données reçues par le réseau Wi-Fi afin que l’application puisse en faire l’interprétation.

## 1.3 Définitions, acronymes et abréviations

|  |  |
| --- | --- |
| **Terme** | **Définition** |
| ACL | *Acronyme :* Advanced Central Logger |
| ACL Manager | Un logiciel de la compagnie MoTeC utilisé pour configurer et contrôler le module MoTeC ACL. |
| Android | Un système d’opération de la compagnie Google destiné aux produits tablettes et mobiles qui supportent ce système. |
| A2C | *Acronyme :* Analog To CAN  Une unité similaire au VIM, mais fabriquée par la Formule ÉTS. Il y a un module A2C par roue sur le véhicule. |
| CAN | *Acronyme :* Controller Area Network  Un bus série très populaire dans le domaine de l’automobile. Il est normalisé par la norme ISO 11898 et utilise une approche populaire, soit le multiplexage, pour effectuer la communication entre les composants d’un ordinateur. |
| Can2Ethernet | Un module développé par la Formule ÉTS qui effectue le transfert du bus CAN vers l’Ethernet. |
| ECU | *Acronyme :* Engine Control Unit  Une unité qui gère le moteur et les capteurs reliés à son fonctionnement. |
| Ethernet | Un système servant à connecter un certain nombre de systèmes informatisés pour former un réseau local. Des protocoles sont utilisés pour contrôler les informations partagées et pour empêcher la transmission simultanée de données par deux systèmes ou plus. |
| ÉTS | *Acronyme :* École de technologie supérieure |
| Formule ÉTS | Un club étudiant scientifique de l’École de technologie supérieure qui fabrique une voiture de course monoplace de type formule performante. |
| iPod Touch | Un produit de la compagnie Apple qui sert principalement de lecteur de musique. Il possède aussi la capacité de naviguer sur Internet via le sans-fil et d’utiliser des applications logicielles. |
| iOS | Un système d’opération de la compagnie Apple destiné aux produits tablettes et mobiles qui supportent ce système. |
| iOS jailbreaking | Un processus permettant aux produits Apple qui fonctionnent sous le système d’exploitation iOS d'obtenir un accès complet pour déverrouiller toutes les fonctionnalités du système d'exploitation, éliminant ainsi les restrictions posées par Apple. |
| MoTeC ACL | Un outil d’acquisition de données et de communication hautement configurable de la compagnie MoTeC. |
| MoTeC i2 Pro | Un logiciel de la compagnie MoTeC utilisé pour l’analyse des données provenant du MoTeC ACL |
| Protocole de communication | Un protocole de communication est une définition de plusieurs règles utilisées par un moyen de communication spécifique. |
| UDP | *Acronyme :* User Datagram Protocol  Le protocole UDP est l’un des principaux protocoles employés par la communication via Internet. |
| USB | *Acronyme :* Universal Serial Bus |
| VIM | *Acronyme :* Versatile input Module  Une unité qui convertit les signaux des capteurs en signal CAN. |
| Wi-Fi | Un ensemble de protocoles de communication sans fil permettant de relier plusieurs systèmes informatiques entre eux pour permettre la transmission d’informations. Le Wi-Fi est régi par la norme IEEE 802.11 (ISO/CEI 8802-11). |
| Xcode | Un environnement de développement pour Mac OS X ainsi que iOS. |

## 1.4 Références

* Le site Web du club Formule ÉTS : <http://www.formuleets.ca/> (consulté le 4 février 2014);
* Le site Web du cours LOG410 : <https://cours.etsmtl.ca/log410/private/> (consulté du 24 janvier au 22 février 2014);
* Définition d’un bus : <http://fr.wikipedia.org/wiki/Bus_informatique> (consulté le 7 février 2014);
* Définition de CAN : <http://fr.wikipedia.org/wiki/Controller_area_network> (consulté le 7 février 2014);
* Définition de UDP: <http://fr.wikipedia.org/wiki/User_Datagram_Protocol> (consulté le 7 février 2014);
* Définition d’un protocole de communication : <http://fr.wikipedia.org/wiki/Protocole_de_communication> (consulté le 7 février 2014);
* Définition de USB: <http://fr.wikipedia.org/wiki/Universal_Serial_Bus> (consulté le 7 février 2014);
* Définition de Wi-Fi: <http://fr.wikipedia.org/wiki/Wi-Fi> (consulté le 9 février 2014);
* Définition d’Ethernet : <http://fr.wikipedia.org/wiki/Ethernet> (consulté le 9 février 2014);
* Le site Web de MoTeC : <http://www.motec.com/> (consulté le 9 février 2014);
* Définition de iOS jailbreaking : <http://en.wikipedia.org/wiki/IOS_jailbreaking> (consulté le 11 février 2014);
* Définition de Xcode : <http://fr.wikipedia.org/wiki/Xcode> (consulté le 17 février 2014);
* Présentation : *Projet Dash Display Formule SAE ÉTS* : par Benjamin Comeau et Xavier Abram (consulté le 9 février 2014);
* Document PDF : *Application mobile : Tableau de bord* : par Benjamin Comeau et Xavier Abram (consulté le 9 février 2014);
* Document Excel : *Table des messages CAN* : par La Formule ÉTS (consulté le 9 février 2014).

# 2. Positionnement

## 2.1 Énoncé du problème

|  |  |
| --- | --- |
| **Le problème :** | Actuellement, la visualisation des données en temps réel des différents capteurs du véhicule est effectuée au travers d’une interface faite pour le « debug ». De plus, les données détaillées sont obtenues seulement lorsque le véhicule est arrêté et que les informations sont téléchargées.  Les données ne sont pas accessibles en temps réel au pilote et les ingénieurs de piste doivent utiliser un ordinateur lors des essais sur la piste. Aussi, la Formule ÉTS souhaite se départir de l’utilisation du mode « debug » de l’ACL Manager. |
| **affecte** | les ingénieurs de piste et les pilotes de la Formule ÉTS |
| **dont l’impact est** | la difficulté d’obtention des données du véhicule durant les tests de piste, l’impossibilité de visionner les données en temps réel pour le pilote et la perte de temps lors du développement sont des impacts directs.  Étant donné que les tests prennent plus de temps que prévu, l’équipe doit donc effectuer plusieurs journées de roulage ce qui augmente les coûts de location, de déplacement, d’essence et d’utilisation des pneus. De plus, l’utilisation des outils et des logiciels de MoTeC coûte cher. Ces coûts résultent en un impact supplémentaire dans la planification et la logistique du développement du véhicule. |
| **Une bonne solution serait** | de permettre au pilote et aux ingénieurs de visualiser les données des capteurs sur une application de manière simple, directe et personnalisable. L’application pourrait offrir deux modes soit le mode pilote et le mode ingénieur de piste qui reçoit les données directement du véhicule sans passer par un intermédiaire développé par MoTeC. |

## 2.2 Positionnement du produit

|  |  |
| --- | --- |
| **Pour** | les membres du club étudiant Formule ÉTS. |
| **Qui** | ont besoin d’un outil avec un mode destiné au pilote pour lui permettre de visualiser les données du véhicule en temps réel. Aussi, l’outil doit comporter un mode pour les ingénieurs de piste qui montrent les données détaillées en temps réel sans avoir recourt à l’ACL Manager. |
| **Le Dash Display** | est une application avec un mode pilote et un mode ingénieur de piste qui permet de visualiser les données des capteurs du véhicule. |
| **Qui** | permet au pilote de visualiser rapidement et facilement les informations de différents capteurs du véhicule en temps réel afin de l’aider dans l’amélioration de sa conduite.  L’application permet aussi aux ingénieurs de piste de visualiser des données plus détaillées sans exiger que la voiture s’arrête pour télécharger les informations. |
| **Contrairement aux** | outils et aux logiciels MoTeC actuellement utilisés par le club. |
| **Notre produit** | permet de diminuer le temps requis pour effectuer des tests sur la piste et, ainsi, diminuer les coûts encourus par les tests supplémentaires. |

# 3. Descriptions des intervenants et des utilisateurs

## 3.1 Résumé des intervenants

|  |  |  |
| --- | --- | --- |
| **Nom** | **Description** | **Responsabilités** |
| Jean-Philippe Breton | - Capitaine;  - Étudiant en génie mécanique. | - Gestion de l’équipe au complet;  - Prend les décisions finales. |
| Nikolas Parent | - Trésorier;  - Étudiant en génie mécanique. | - Gestion des ressources financières. |
| Xavier Abram | - Responsable de la dynamique du véhicule;  - Étudiant en génie mécanique. | - Gestion des diverses influences physiques sur le véhicule. |
| Benjamin Comeau | - Responsable de l’électronique et du logiciel;  - Développeur iOS;  - Étudiant en génie logiciel. | - Gestion des projets se rattachant au logiciel ou à l’électronique;  - Assure le développement en Objective-C. |
| Martin Grenier | - Architecte et concepteur logiciel;  - Développeur Android;  - Étudiant en génie logiciel. | - Assure d’avoir un système optimal pour les spécifications données;  - Assure le développement Android. |
| Louis-Philippe Trottier | - Développeur et concepteur Objective-C;  - Étudiant en génie logiciel. | - Assure le développement en Objective-C. |
| Étienne Saint-Laurent | - Pilote du véhicule;  - Chef du département Moteur;  - Étudiant en génie mécanique. | - Conduire le véhicule lors des essais sur piste et dans les compétitions;  - Gestion de l’équipe qui s’occupe du moteur dans le véhicule. |
| Jean-François Lafontaine | - Pilote du véhicule. | - Conduire le véhicule lors des essais sur piste et dans les compétitions. |
| Charles Ratelle | - Responsable de la dynamique du véhicule;  - Étudiant en génie mécanique. | - Gestion des diverses influences physiques sur le véhicule, mais plus précisément les pneus et les freins. |
| Charlie Brault | - Responsable de la dynamique du véhicule;  - Étudiant en génie mécanique. | - Gestion des diverses influences physiques sur le véhicule, mais plus précisément les pneus et les freins. |

## 3.2 Résumé des utilisateurs

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Description** | **Responsabilités** | **Intervenant** |
| Pilotes | - Utilisateur de l’application pour les pilotes;  - Pilote du véhicule. | - Conduire le véhicule lors des essais sur piste et dans les compétitions. | - Les pilotes du véhicule sont : Étienne Saint-Laurent et Jean-François Lafontaine |
| Ingénieurs de piste | - Utilisateur de l’application pour les ingénieurs de piste;  - Les autres membres du club, à l’exception des pilotes, vont agir en tant qu’ingénieur de piste. | - Analyser les données reçues par l’application;  - Présent sur le bord de la piste pour assister le pilote. | - Les intervenants présentés dans le tableau de la section 3.1 sont les principaux utilisateurs de l’application pour les ingénieurs de piste. |

## 3.3 Environnement utilisateur

Les deux modes de l’application seront utilisés respectivement par les pilotes et les ingénieurs de piste. Actuellement, une dizaine de personnes travaille sur le projet et ce nombre peut changer au fil du temps étant donné qu’un club étudiant, en général, connaît un bon roulement de membres.

En moyenne, un test de roulage a une durée de 10 à 30 minutes et est effectué à l’extérieur sous une météo généralement ensoleillée. L’application est utilisée lors des journées de test qui dure environ de 4 à 8 heures.

La plateforme visée pour l’application est, tout d’abord, le iPod Touch 5 avec écran de 4 pouces. Dans une prochaine phase, l’application va être adaptée à tout appareil iOS et Android.

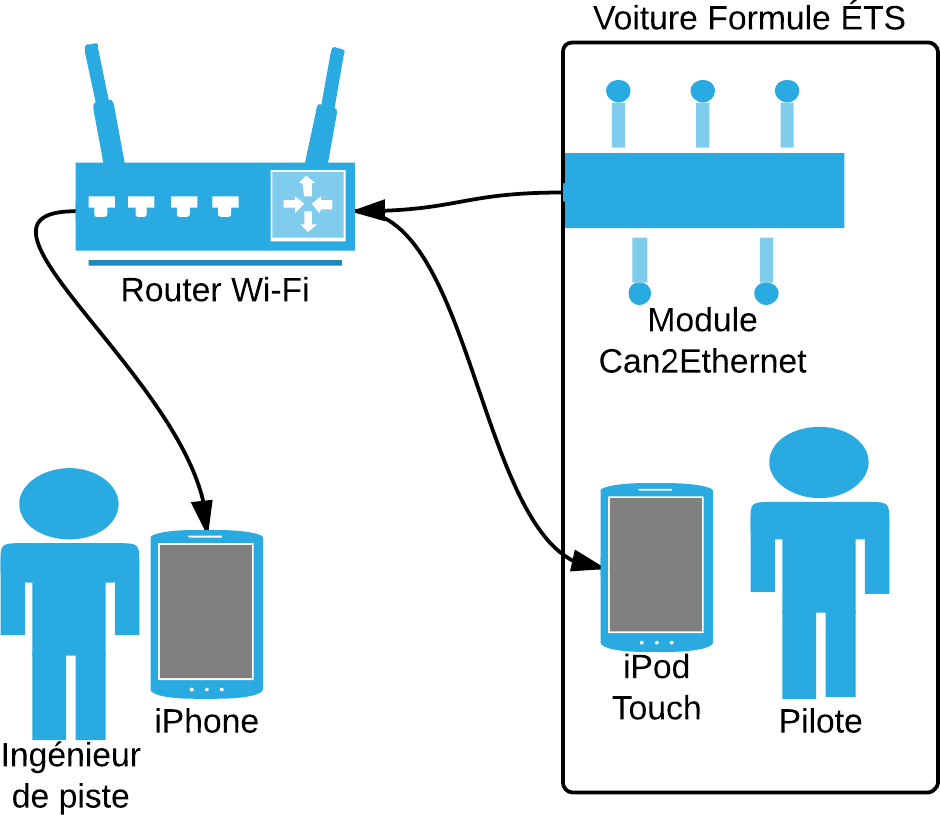
Présentement, les programmes MoTeC i2 Pro et ACL Manager sont respectivement utilisés pour l’analyse pendant et après les roulages sur piste. L’application développée dans le cadre de ce projet vise à remplacer l’ACL Manager pour la visualisation des données en temps réel et l’affichage des données détaillées. Le remplacement du logiciel MoTeC i2 Pro pour l’analyse des informations va s’effectuer dans une phase ultérieure.

## 3.4 Principaux besoins des intervenants et utilisateurs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Besoin** | **Priorité** | **Préoccupations** | **Solution actuelle** | **Solution proposée** |
| *B01*  Un mode destiné au tableau de bord du pilote | Critique | L’interface doit afficher les données des différents capteurs. Le pilote ne doit pas manipuler le tableau de bord lorsque le véhicule est en mouvement sauf avec les boutons sur son volant. | Les données du véhicule en temps réel sont seulement accessibles avec le mode « debug » de l’ACL Manager. De plus, seuls les ingénieurs de piste peuvent les voir avec un ordinateur. | Un logiciel implanté sous forme d’application dans un appareil mobile situé en face du pilote. L’interface agit comme tableau de bord et peut être personnalisée pour obtenir diverses données. L’application doit servir pour l’affichage seulement. |
| *B02*  Un mode destiné aux ingénieurs de piste | Critique | L’interface doit permettre aux ingénieurs de piste d’accéder à des informations plus détaillées grâce aux données récoltées par les différents capteurs. | Les données du véhicule doivent être téléchargées à chaque fois que le véhicule est arrêté pour visionner les données détaillées. Cette manoeuvre prend beaucoup de temps à effectuer. Elles sont visualisées dans le logiciel MoTeC i2 Pro à l’aide d’un ordinateur. | Un logiciel implanté sous forme d’application utilisable par les téléphones des ingénieurs de piste. L’interface possède plus de fonctionnalités que le tableau de bord puisque l’interface doit servir à visionner les données plus détaillées. |
| *B03*  L’application doit fonctionner avec la plateforme iOS dans le format iPod Touch et iPhone | Critique | L’application doit être supportée par la plateforme iOS et l’interface doit être ajustée pour afficher toutes les informations dans un écran d’environ 4 pouces. | Les données sont actuellement visualisées dans un logiciel appelé MOTEC I2 Pro à l’aide d’un ordinateur. Un ordinateur est plus encombrant et la batterie dure moins longtemps qu’un appareil portable. La majorité des membres possèdent un appareil avec le système d’opération iOS et l’appareil pour le véhicule utilise aussi ce système. | L’application est développée pour le système d’opération iOS avant les autres plateformes pour satisfaire le plus grand nombre de membres de la Formule ÉTS et pour être compatible avec le iPod Touch à bord du véhicule. Les interfaces vont être adaptées pour être optimales sur un écran de 4 pouces. |

# 4. Vue d’ensemble du produit

## 4.1 Perspective du produit



**Figure 2 : Diagramme de la communication entre la voiture et les appareils mobiles**

## 4.2 Principaux avantages

**Table 4.1 - Système Dash Display**

|  |  |
| --- | --- |
| **Bénéfices pour le client** | **Caractéristiques correspondantes** |
| B01 - Un mode destiné au tableau de bord du pilote | CAR01, CAR04, CAR06, CAR07, CAR08, CAR09, CAR12, CAR13, CAR15 |
| B02 - Un mode destiné aux ingénieurs de piste | CAR02, CAR03, CAR06, CAR07, CAR08, CAR11, CAR15 |
| B04 - L’application doit fonctionner avec la plateforme iOS dans le format iPod Touch et iPhone | CAR04, CAR05, CAR10, CAR14 |

## 4.3 Hypothèses et dépendances

Les hypothèses suivantes sont essentielles pour le développement du produit et sont supposées comme valides. Dans le cas de l’invalidité d’un élément de la liste, le projet présenté dans ce document s’en retrouve altéré et son fonctionnement compromit.

* Le module CAN à Ethernet (Can2Ethernet) conçu par la Formule ÉTS est fonctionnel;
* Les données sont envoyées par Wi-Fi via le protocole UPD et le bus CAN;
* La voiture est située dans la portée de l’émetteur Wi-Fi;
* Le port UDP est le port par défaut soit 1337;
* Les « id » et les « offset » qui sont fournis par la table CAN ne changent pas.

## 4.4 Licences et installation

Pour l’application développée, aucune licence commerciale n’est nécessaire puisqu’elle est utilisée seulement par le club étudiant Formule ÉTS. L’École de technologie supérieure fournit les licences de développeurs Apple afin d’utiliser l’application légalement sur les appareils mobiles iOS.

Pour l’installation du produit, un projet compatible avec Xcode et iOS est fourni sans guide d’installation particulier. Étant donné que le club possède des développeurs expérimentés, ces informations ne sont pas nécessaires.

# 5. Caractéristiques du produit

## 5.1 CAR01 - Afficher les données demandées au pilote

Le pilote peut facilement avoir accès aux données provenant des différents capteurs du véhicule. Les informations sont affichées de manière claire pour que le pilote puisse les lire rapidement durant le roulage sur piste. Le pilote peut ainsi maximiser sa performance et mieux comprendre les impacts des paramètres du véhicule.

## 5.2 CAR02 - Afficher les données demandées aux ingénieurs

Contrairement au pilote, les ingénieurs ont accès à l’ensemble des données provenant des capteurs de la voiture durant le roulage sur piste. L’efficacité des tests est donc grandement améliorée en permettant de comprendre en temps réel les comportements de la voiture. De plus, un affichage plus détaillé est possible sur l’ensemble des données choisies.

## 5.3 CAR03 - Offrir une interface modifiable en mode ingénieurs

Les informations disponibles pour l’affichage, dans le mode ingénieur, sont celles des capteurs répertoriés dans la table CAN fournie par la Formule ÉTS. Étant donné qu’il y a plusieurs dizaines de capteurs, l’interface doit être personnalisable afin d’afficher seulement les données désirées. De plus, les différents capteurs possèdent des codes de couleur et des affichages différents qui pourront être intégrés sous forme de composant d’interface graphique.

## 5.4 CAR04 - Afficher l’interface en mode paysage pour le pilote

L'appareil est fixé en face du pilote et l'application doit toujours rester en mode paysage de façon à faciliter la visualisation des données. Contrairement au mode ingénieur de piste, les données sont montrées de façon graphique plutôt que sous le format d’une liste.

## 5.5 CAR05 - Recevoir les données en tout temps

Dans le mode ingénieur de piste, il faut permettre à l’application de continuer à recevoir les données en arrière-plan tant qu’elle n’est pas complètement fermée ou que la communication est interrompue. Les données doivent continuer à être enregistrées même si l’ingénieur de piste change d’application pendant les tests sur piste.

## 5.6 CAR06 - Permettre la transition entre les deux modes avec un bouton

L’application peut basculer entre le mode pilote et le mode ingénieur de piste avec un bouton toujours visible situé dans l'interface de l'application. De cette façon, une seule application est nécessaire.

## 5.7 CAR07 - Recevoir les données directement de la voiture

Les données sont envoyées directement de la voiture grâce aux capteurs installés directement sur les roues et les différentes parties du véhicule. Les différentes unités ACL, ECU, A2C et VIM installées transmettent les données sous format CAN que le module Can2Ethernet envoie à l’application par Wi-Fi. L’application va ensuite afficher les informations grâce à la table CAN qui définit la signification des messages reçus. De cette façon, la Formule ÉTS ne va plus dépendre des outils MoTeC.

## 5.8 CAR08 - Afficher les textes de l’application en anglais

Les deux modes de l’application vont être en anglais seulement. En effet, l’anglais est privilégié pour permettre aux compagnies, aux juges, etc. lors des différentes compétitions de comprendre l’application utilisée pour gérer les données du véhicule.

## 5.9 CAR09 - Afficher l’application du pilote dès l’ouverture

Dans le but de permettre au pilote d'utiliser le plus rapidement possible l'application, elle s'ouvre par défaut en mode pilote. Toutefois, le mode par défaut peut être configuré via le fichier de configuration de l’application.

## 5.10 CAR10 - L’application doit utiliser des technologies légales

Toutes les technologies utilisées pour le développement de l’application doivent pouvoir être utilisées sur un iPod Touch de façon légale. En effet, les membres ne veulent pas effectuer un « iOS jailbreaking » sur leurs appareils pour pouvoir utiliser l’application puisque leur garantie s’en retrouverait révoqués.

## 5.11 CAR11 - Afficher les dernières données lors de la perte du signal

Pour le mode ingénieur de piste, les dernières données reçues doivent rester affichées même si l’application perd le signal avec la voiture ou avec le Wi-Fi. Contrairement au mode pilote, l’application ne doit pas s’éteindre afin de ne pas perdre les données qui peuvent être exportées ou synchronisées.

## 5.12 CAR12 - Mettre en veille le mode pilote après 3 secondes sans données

Si l'application en mode pilote ne reçoit plus de données, elle doit se mettre automatiquement en veille après 3 secondes. Ce cas peut se produire lorsqu’il y a une perte de communication par exemple.

## 5.13 CAR13 - Permettre au pilote de changer les interfaces

L'application peut recevoir un message du module CAN indiquant que le pilote a appuyé sur le bouton situé sur son volant. Cette action changera l'interface présentement affichée par l'application. Le bouton sur le volant est important pour le pilote puisque ses gants l’empêchent d’appuyer sur l’écran.

## 5.14 CAR14 - Empêcher la mise en veille de l’application

Afin de s’assurer que le pilote et les ingénieurs aient toujours accès aux données fournies par l’application, la mise en veille automatique est désactivée pendant l’utilisation.

## 5.15 CAR15 - Utiliser les configurations du fichier XML

L’application doit utiliser les configurations établies dans un fichier XML facilement accessible et modifiable par les membres de la Formule ÉTS. Les configurations principales de l’application vont s’y retrouver.

# 6. Contraintes

Le club Formule ÉTS a exigé que les contraintes suivantes soient respectées dans le cadre du projet :

* l’application ne doit pas envoyer de données; elle peut seulement en recevoir;
* les couleurs demandées pour l’affichage des capteurs doivent être respectées;
* les messages sont reçus en texte brut à partir du module Can2Ethernet via Wi-Fi;
* les capteurs disponibles sont limités à ceux offerts par la table CAN;
* les capteurs sont déterminés par l’« id » et l’« offset » reçus et déterminés dans la table CAN fournie.

# 7. Gammes de qualité

Selon la norme ISO 9126, plusieurs modèles de qualité se rattachent au projet « Dash Display ». Les principales caractéristiques qui définissent l’application sont les suivantes :

* **Fiabilité :** comme l’application est utilisée lors des tests de piste qui sont assez coûteux, elle doit être fiable pour fonctionner durant une durée déterminée sans défaillance. De plus, cette caractéristique est encore plus importante lorsque l’application est utilisée durant les compétitions;
* **Maintenabilité :** la capacité d’évolution de l’application est primordiale puisque les membres du club veulent effectuer trois autres phases de développement après les phases couvertes par ce document. La possibilité d’ajouter des fonctionnalités est donc très importante et le code source doit être bien commenté pour une meilleure compréhension;
* **Portabilité :** suite au développement des deux phases couvertes par ce document, les membres du club souhaitent transposer l’application sur une plateforme Android et sur les tablettes. La facilité d’adaptation est donc importante pour faciliter la tâche de changement de plateformes.

# 8. Attributs des caractéristiques

Le tableau suivant utilise les attributs de caractéristiques présentés dans l’annexe A du présent document.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Caractéristiques** | **État** | **Bénéfice** | **Effort** | **Risque** | **Stabilité** | **Priorité** |
| CAR01 - Afficher les données demandées au pilote | Proposé | Élevé | Élevé | Faible | Élevé | Critique |
| CAR02 - Afficher les données demandées aux ingénieurs | Proposé | Élevé | Élevé | Faible | Élevé | Critique |
| CAR03 - Offrir une interface modifiable en mode ingénieur | Proposé | Élevé | Élevé | Faible | Élevé | Critique |
| CAR04 - Afficher l’interface en mode paysage pour le pilote | Proposé | Élevé | Faible | Faible | Élevé | Critique |
| CAR05 - Recevoir les données en tout temps | Proposé | Élevé | Moyen | Faible | Élevé | Utile |
| CAR06 - Permettre la transition entre les deux modes avec un bouton | Proposé | Élevé | Faible | Faible | Élevé | Critique |
| CAR07 - Recevoir les données directement de la voiture | Proposé | Élevé | Faible | Faible | Élevé | Critique |
| CAR08 - Afficher les textes de l’application en anglais | Proposé | Moyen | Faible | Faible | Élevé | Important |
| CAR09 - Afficher l’application du pilote dès l’ouverture | Proposé | Faible | Faible | Faible | Élevé | Utile |
| CAR10 - L’application doit utiliser des technologies légales | Proposé | Élevé | Faible | Faible | Élevé | Critique |
| CAR11 - Afficher les dernières données lors de la perte du signal | Proposé | Élevé | Faible | Faible | Élevé | Critique |
| CAR12 - Mettre en veille le mode pilote après 3 secondes sans données | Proposé | Élevé | Faible | Faible | Élevé | Utile |
| CAR13 - Permettre au pilote de changer les interfaces | Proposé | Moyen | Moyen | Faible | Élevé | Critique |
| CAR14 - Empêcher la mise en veille de l’application | Proposé | Élevé | Moyen | Faible | Élevé | Critique |
| CAR15 - Utiliser les configurations du fichier XML | Proposé | Élevé | Faible | Faible | Élevé | Important |

# 9. Autres exigences du produit

## 9.1 Standards applicables

Le code source de l’application doit respecter les standards d’Objective-C et du développement d’application sous le système d’opération iOS. Le fichier de configuration de l’application doit suivre la norme XML selon un gabarit défini par le club.

Les données sont transmises des capteurs vers le module Can2Ethernet par des signaux empruntant le bus CAN et le protocole UDP. Le module émet les données par des signaux Wi-Fi vers l’application qui reçoit les messages en texte brut qui suivent le format de la table CAN fournie par la Formule ÉTS.

## 9.2 Exigences du système

L’application va être installée sur un appareil mobile de type iPod Touch ou iPhone avec la dernière version du système d’exploitation iOS. L’écran utilisé pour le pilote possède une diagonale de 4 pouces et les écrans des ingénieurs de piste peuvent varier légèrement. Grâce au logiciel de développement Xcode, les variations des formats d’écran sont prises en charge pour une intégration plus facile.

## 9.3 Exigences de performance

La transmission des données est quasi instantanée alors la performance dépend surtout du transfert par Wi-Fi. L’application doit maximalement rafraîchir les données deux fois chaque seconde pour conserver une fluidité dans l’affichage des capteurs et des alarmes.

La transition entre les interfaces et l’apparition des données en temps réel doivent être instantanées. Cependant, pour le mode ingénieur, l’affichage des données détaillées peut avoir un léger délai étant donné la quantité d’informations à afficher et la nécessité moins immédiate de ces données.

## 9.4 Exigences environnementales

Le nombre d’utilisateurs concurrents sur l’application est illimité puisqu’une seule instance du logiciel roule par appareil. Les erreurs doivent être capturées et conservées en mémoire pour un futur débogage. De cette façon, l’application peut continuer de fonctionner sans interruption et permettre une meilleure maintenance.

# 10. Exigences de documentation

## 10.1 Manuel de l’utilisateur

Étant donné que l’application est destinée aux membres du club seulement, le manuel d’utilisateur n’est pas nécessaire. En effet, les utilisateurs vont utiliser le logiciel en présence des développeurs du projet qui peuvent configurer le logiciel et répondre à leurs questions.

## 10.2 Aide en ligne

Le club Formule ÉTS souhaite ne pas avoir de documentation en ligne pour le Dash Display. Plusieurs membres sont des utilisateurs expérimentés des technologies informatiques utilisées et peuvent apporter le support nécessaire.

## 10.3 Guides d’installation, de configuration et fichier à lire

Le développement des fonctionnalités futures et la maintenance sont effectués par des membres du club étudiant. Les documents concernant l’architecture de l’application sont donc nécessaires pour aider à comprendre l’implémentation effectuée. Cependant, aucun guide d’installation n’est fourni puisque l’application consiste en une application pour iOS standard.

La configuration du logiciel est effectuée dans un fichier XML à la demande du club. En effet, les membres changent souvent et le format XML est connu de plusieurs et facile à lire. Cette méthode doit donc être utilisée.

# Annexes

## A Attributs des caractéristiques

Les attributs des caractéristiques sont des critères pour aider à l’analyse de la rentabilité, des risques et de la priorité de chaque caractéristique du logiciel. Chaque attribut présente différents niveaux à considérer selon la description proposée.

### État

|  |  |
| --- | --- |
| Proposé | La caractéristique est proposée, mais n’a pas encore été approuvée par les parties prenantes. |
| Approuvé | La caractéristique est approuvée par les parties prenantes. |
| Incorporé | La caractéristique est incluse dans le produit. |

### Bénéfice

|  |  |
| --- | --- |
| Faible | La caractéristique apporte peu de valeur ajoutée au produit et n’est pas nécessaire à son bon fonctionnement. |
| Moyen | La caractéristique apporte une valeur ajoutée additionnelle au produit, mais n’est pas critique à son bon fonctionnement. |
| Élevé | La caractéristique apporte une valeur ajoutée importante au produit et est essentielle à son bon fonctionnement ou à la réalisation de ses tâches. |

### Effort

|  |  |
| --- | --- |
| Faible | La réalisation de la caractéristique nécessite un effort de moins de 20 heures-personnes. |
| Moyen | La réalisation de la caractéristique nécessite un effort entre 20 et 40 heures-personnes. |
| Élevé | La réalisation de la caractéristique nécessite un effort de plus de 40 heures-personnes. |

### Risque

|  |  |
| --- | --- |
| Faible | La technologie utilisée et la méthode d’implémentation sont connues et bien maîtrisées. |
| Moyen | La technologie utilisée est récente ou la méthode d’implémentation nécessite une attention particulière. |
| Élevé | La technologie utilisée est nouvelle et peu éprouvée ou la méthode d’implémentation est complexe et demande une analyse plus complète. |

### Stabilité

|  |  |
| --- | --- |
| Faible | Les exigences concernant la caractéristique ont de fortes chances de changer ou le bon fonctionnement de la caractéristique n’a pas d’impact sur le fonctionnement général du système. |
| Moyen | Les exigences concernant la caractéristique sont susceptibles de changer ou le bon fonctionnement de la caractéristique a un impact sur le fonctionnement général du système sans toutefois compromettre son exécution. |
| Élevé | Les exigences concernant la caractéristique ont peu de chance de changer et le bon fonctionnement de la caractéristique a un impact critique sur le fonctionnement général du système et peut compromettre son exécution. |

### Priorité

|  |  |
| --- | --- |
| Utile | La caractéristique apporte des fonctionnalités accessoires au système. Son inclusion dans le produit a peu d’impact sur la satisfaction du client et sur l’utilisation du système. |
| Important | La caractéristique apporte des fonctionnalités supplémentaires au système. Son inclusion dans le produit peut influencer la satisfaction du client, mais son absence n’empêche pas l’utilisation du système. |
| Critique | La caractéristique est primordiale au fonctionnement du système. Il est nécessaire de l’inclure en priorité dans le produit pour assurer la totale satisfaction du client et son absence pourrait empêcher l’utilisation du système. |