

# Rapport de Projet

---

## ALFOX

Projet réalisé par

Verseillie Yoann

Torreilles Adrien

Castanet Guilhem

De Saint Denis Gaëtan



---

# Sommaire

Introduction	3
Présentation de l'entreprise	4
Localisation géographique	4
Historique	5
Présentation du projet	6
Introduction au projet	6
Logiciel de gestion de projet : TimePerformance	8
Logiciel de Développement : Arduino	9
Versionner le code	9
Logiciel de versionnage de code	10
Schéma de fonctionnement d'un logiciel de versionnage de code	11
Logiciel choisi	12
Séparation du Projet :	14
Diagramme de classe Alfox 2	17
partie technique	18

---

# Introduction

Le projet qui nous a été proposé est un projet demandé par la société Alcis qui souhaite intégrer à ses véhicules un système de récupération de données et de localisation. Nous avons commencé en février. Notre groupe est composé de quatre personnes, VERSEILLIE Yoann, TORREILLES Adrien, DE SAINT DENIS Gaëtan, CASTANET Guilhem.

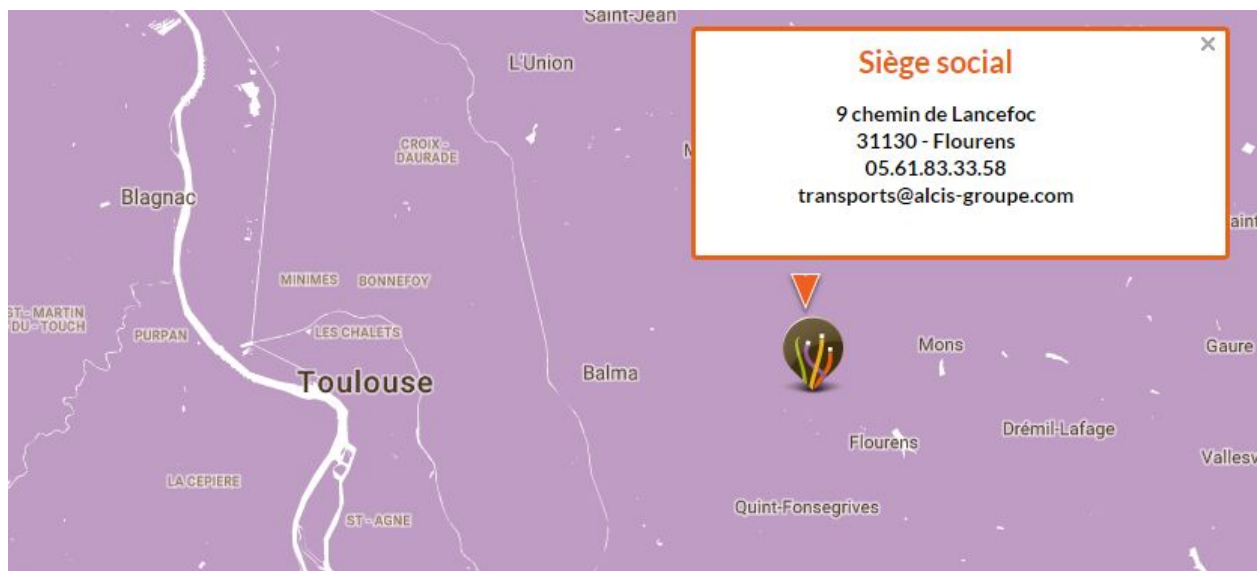
---

# Présentation de l'entreprise

La société qui nous à commandé le projet est ALCIS, une entreprise spécialiste du transport de personnes. ALCIS Groupe met à la disposition des particuliers et des professionnels, en groupe ou en individuel, sa flotte de plus de 450 véhicules. Depuis peu Alcis loue des véhicules à des chauffeurs VTC.

## Localisation géographique

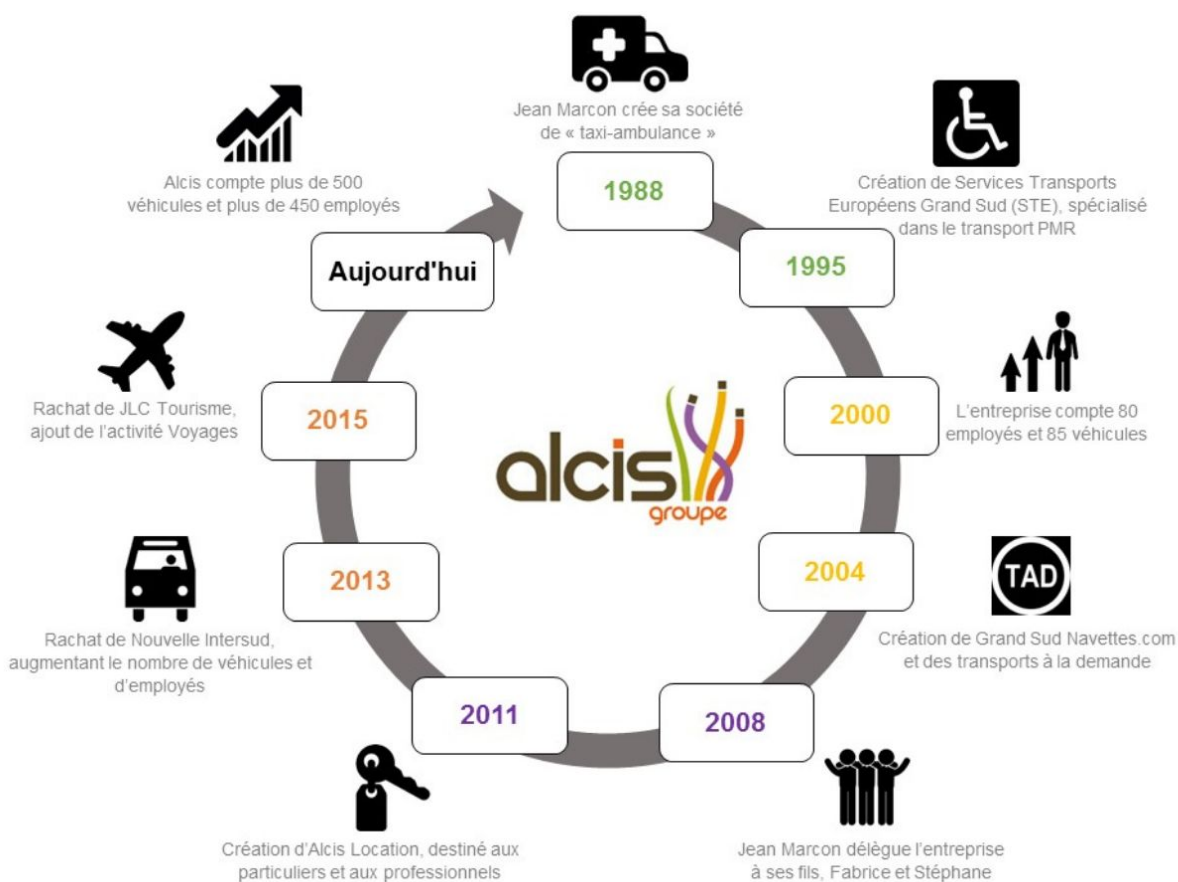
Alcis-groupe est basé sur toulouse.



## Historique

Alcis groupe a été créé par Jean Marcon en 1988. Cette société était spécialisée dans le transport de personnes handicapées. C'est en 2000 que la société Alcis devient une société de transport à la demande avec les premiers TAD.

Le schéma ci-dessous présente l'entreprise au fil des années :



---

# Présentation du projet

## Introduction au projet

ALCIS est une société de transport de personnes qui a récemment étendu son domaine d'activité à la location de voiture pour les chauffeurs VTC. Suite à des incidents lors des locations (Non respect des contrats, dépassement de la limite de kilométrage, impayé, vol de voiture, etc), ALCIS a décidé de chercher une solution pour résoudre ces problèmes. Leur souhait était de pouvoir équiper chaque véhicule d'un objet connecté qui permettait de récupérer les données voulues ainsi que de localiser la voiture si besoin est.

Alcis souhaite avoir un moyen d'obtenir l'état et la localisation des véhicules loués afin de répondre à certains problèmes :

- Le vol du véhicule → Une localisation précise afin de retrouver le véhicule.
- Respect du contrat → Une localisation approximative via SigFox afin de vérifier que le véhicule ne sorte pas de la zone contractuelle.
- L'oubli de maintenance (Vidange, niveaux d'huile, plaquettes de freins...) → Une récupération de l'état du véhicule en temps réel.

Contraintes	Solutions
Le client veut avoir un accès simple et pratique aux données	Un site web sur lequel les données des différents véhicules sont disponibles. Le site web est lié à une base de données et ces dernières sont récupérées grâce à un module Sigfox qui sert de communication entre le boîtier placé dans la voiture louée et les serveur ALCIS.
Le client veut deux types de localisations différentes, une approximative afin de s'assurer que le chauffeur ne sort pas de la zone (Ce qui permet aussi de ne pas donner au chauffeur le sentiment d'être espionné) et une précise afin de retrouver le véhicule en cas de problème.	La localisation approximative est récupérée directement par Sigfox puis envoyé au client tandis qu'un GPS permet d'obtenir la localisation précise quand nécessaire.
Le client veut un suivi en temps réel des consommations et données par véhicules.	Les données du véhicule sont récupérées via la prise diagnostic de ce dernier et sont transmises du module OBDII au boîtier en Bluetooth.

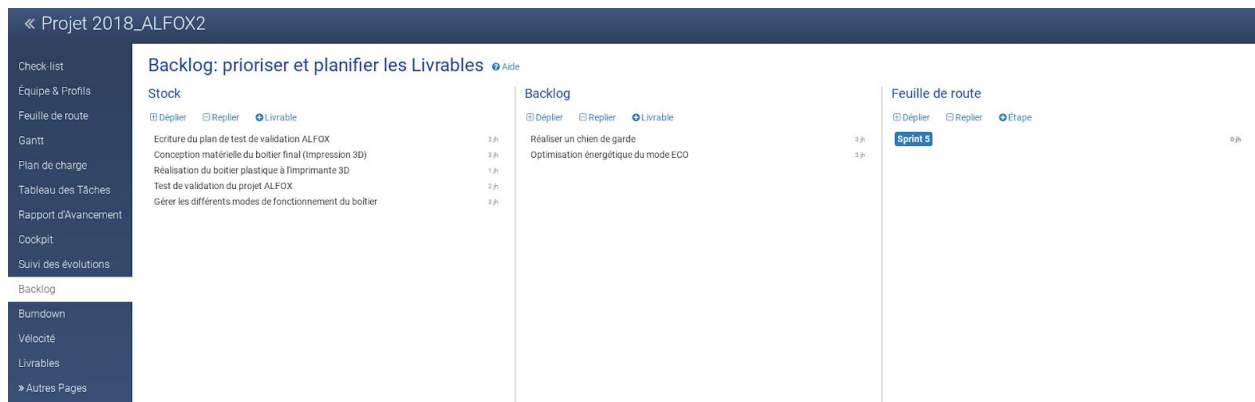




## Logiciel de gestion de projet : TimePerformance

TimePerformance est un logiciel qui permet une gestion de projet automatisée, une visibilité facilitée et un gain de temps conséquent.

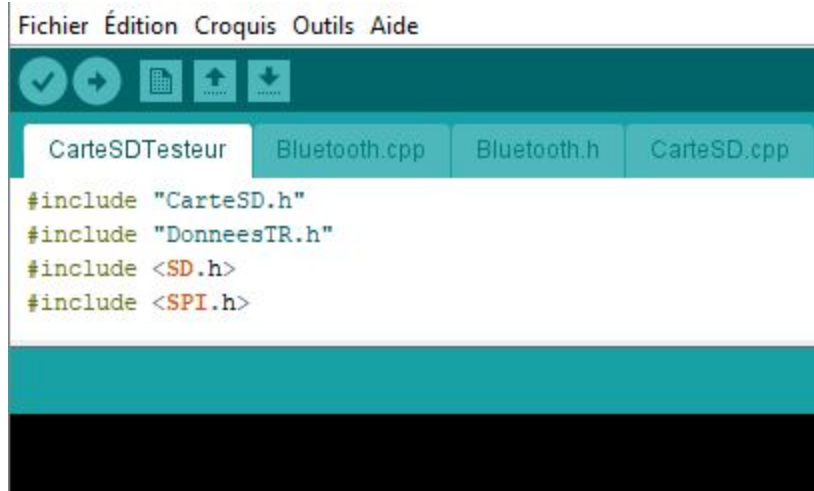
Basé sur la méthode SCRUM, il permet de rassembler de nombreux outils dans un seul logiciel, ainsi il remplace les post-it par un tableau des tâches et nous génère automatiquement un Burndown chart, qui représente l'avancée du projet.



---

## Logiciel de Développement : Arduino

Voici l'environnement de développement d'arduino, c'est le logiciel que nous utilisons pour développer notre code pour la carte arduino, nous ne pouvons pas utiliser n'importe quel logiciel car arduino as des compilateur spéciaux en fonction des processus des carte utilisée



---

## Versionner le code

En raison des nombreuses tâches à effectuer, notre projet à été divisé en deux groupes de trois et quatres personnes. Pour gérer le code nous avons dû trouver une solution pour travailler ensemble. On a donc opté pour un logiciel de version de code.

### Logiciel de versionnage de code

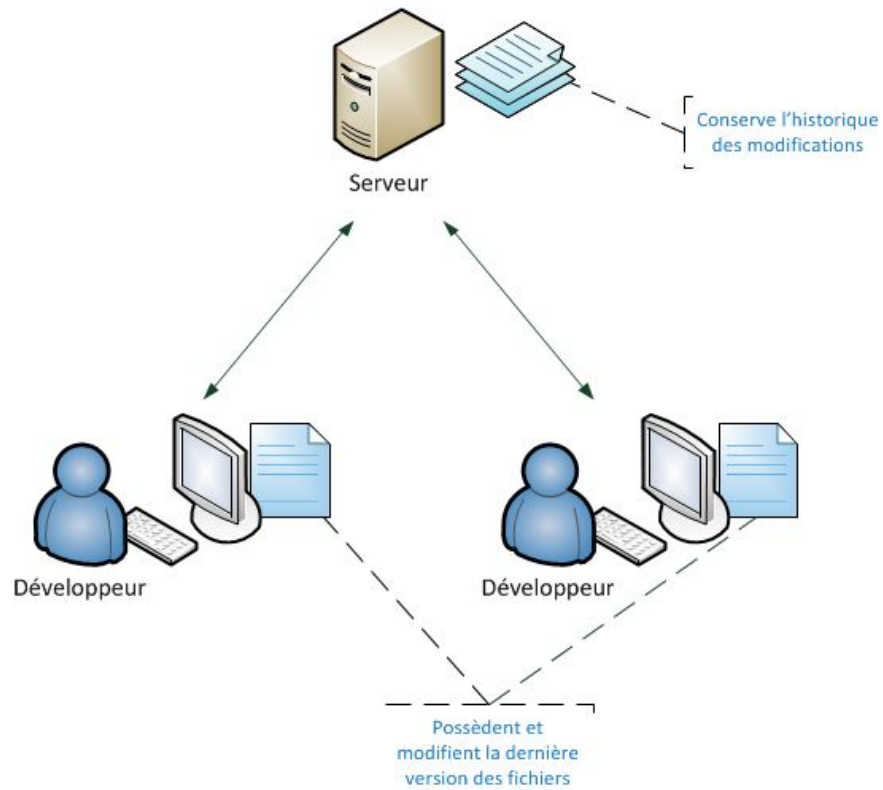
Les logiciels de gestion de versions sont utilisés principalement par les développeurs. En effet, ils sont quasi exclusivement utilisés pour gérer des codes sources, car ils sont capables de suivre l'évolution d'un fichier texte ligne de code par ligne de code. Ces logiciels sont fortement conseillés pour gérer un projet informatique. ils retiennent qui a effectué chaque modification de chaque fichier et pourquoi. Ils sont par conséquent capables de dire qui a écrit chaque ligne de code de chaque fichier et dans quel but :

- si deux personnes travaillent simultanément sur un même fichier, ils sont capables d'assembler (de fusionner) leurs modifications et d'éviter que le travail d'une de ces personnes ne soit écrasé.

Ces logiciels ont donc par conséquent deux utilités principales :

- suivre l'évolution d'un code source, pour retenir les modifications effectuées sur chaque fichier et être ainsi capable de revenir en arrière en cas de problème ;
- travailler à plusieurs, sans risquer de se marcher sur les pieds. Si deux personnes modifient un même fichier en même temps, leurs modifications doivent pouvoir être fusionnées sans perte d'information.

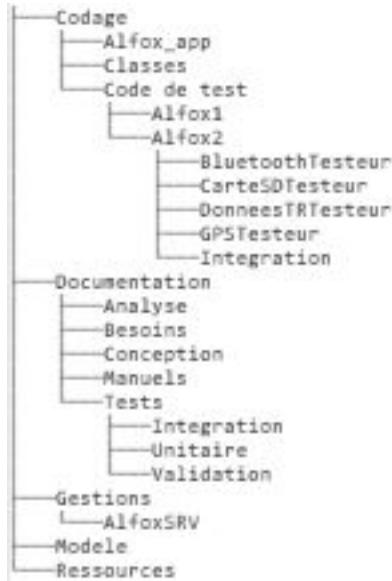
## Schéma de fonctionnement d'un logiciel de versionnage de code



---

## Logiciel choisi

Nom	Description
Git	Très puissant et récent, il a été créé par Linus Torvalds, qui est entre autres l'homme à l'origine de Linux. Il se distingue par sa rapidité et sa gestion des branches qui permettent de développer en parallèle de nouvelles fonctionnalités.
Bazaar	Un autre outil, complet et récent, comme Mercurial. Il est sponsorisé par Canonical, l'entreprise qui édite Ubuntu. Il se focalise sur la facilité d'utilisation et la flexibilité.
Mercurial	Plus récent, il est complet et puissant. Il est apparu quelques jours après le début du développement de Git et est d'ailleurs comparable à ce dernier sur bien des aspects.
SVN	Probablement l'outil le plus utilisé à l'heure actuelle. Il est assez simple d'utilisation, bien qu'il nécessite comme tous les outils du même type un certain temps d'adaptation. Il a l'avantage d'être bien intégré à Windows avec le programme Tortoise SVN, là où beaucoup d'autres logiciels s'utilisent surtout en ligne de commande dans la console.
CVS	C'est un des plus anciens logiciels de gestion de versions. Bien qu'il fonctionne et soit encore utilisé pour certains projets, il est préférable d'utiliser SVN (souvent présenté comme son successeur) qui corrige un certain nombre de ses défauts, comme son incapacité à suivre les fichiers renommés par exemple.



L'architecture de ce GIT nous permet de se retrouver facilement sur des ordinateurs qui ne sont pas les nôtres et à la fois nous permet de nous aider à savoir les différentes tâches sans passer par le time performance.

Le GIT nous permet d'avoir toutes les classes aux mêmes endroits ce qui facilite les tests d'intégration et nous a permis de faire une intégration continue. De plus sur le git il y a toujours la dernière version de chaque classe. Le GIT peut aussi servir de sauvegarde en cas de perte de donnée.

---

## Séparation du Projet :

Le projet a été séparé en deux parties : ALFOX 1 qui s'occupe de la communication SIGFOX ainsi que la gestion de la base de donnée et du site WEB, notre groupe ALFOX 2 s'occupe de la récupération des donnée depuis la voiture ainsi que de la localisation GPS et du stockage sur la carte SD ainsi que la gestion de l'espace de stockage.

travail alfox 1

travail alfox 2

---

## partie présentation globale/compréhension projet

Pour ce projet le cahier des charge nous demandait un moyen basse consommation et peu coûteux pour l'envoi des données de la voiture. Dans cette optique nous nous sommes tournés vers les meilleures technologies actuelles sur le marché. Nous avons d'abord pensé à la communication GSM que nous utilisons tous les jours à travers nos téléphone portable, puis nous nous sommes tournés vers des technologies qui étaient plus économiques, c'est pourquoi nous nous sommes penchés vers SIGFOX et LORA qui sont deux concurrents qui fournissent des services similaires.

Moyen de communication	SIGFOX	LORA	GSM
Transmission bidirectionnelle	Non(enfin très limitée)	Oui	Oui
Distance de transmission	15Km	15 km(rural) 3 km(urbain)	de 10 m à 35 km
Précision de la localisation	5km	500m	10m
Nombre de messages par jours	140 message montant 4 message descendant	24	150
Débit	10 kbps	100 bps	12 kbps
Bande de fréquence	868MHz	868MHz	downlink 925-960MHz uplink 880-915MHz
Consommation	50 $\mu$ W	50 $\mu$ W	820 mW
Coût	1€ par mois	2€ par mois	3€ par mois

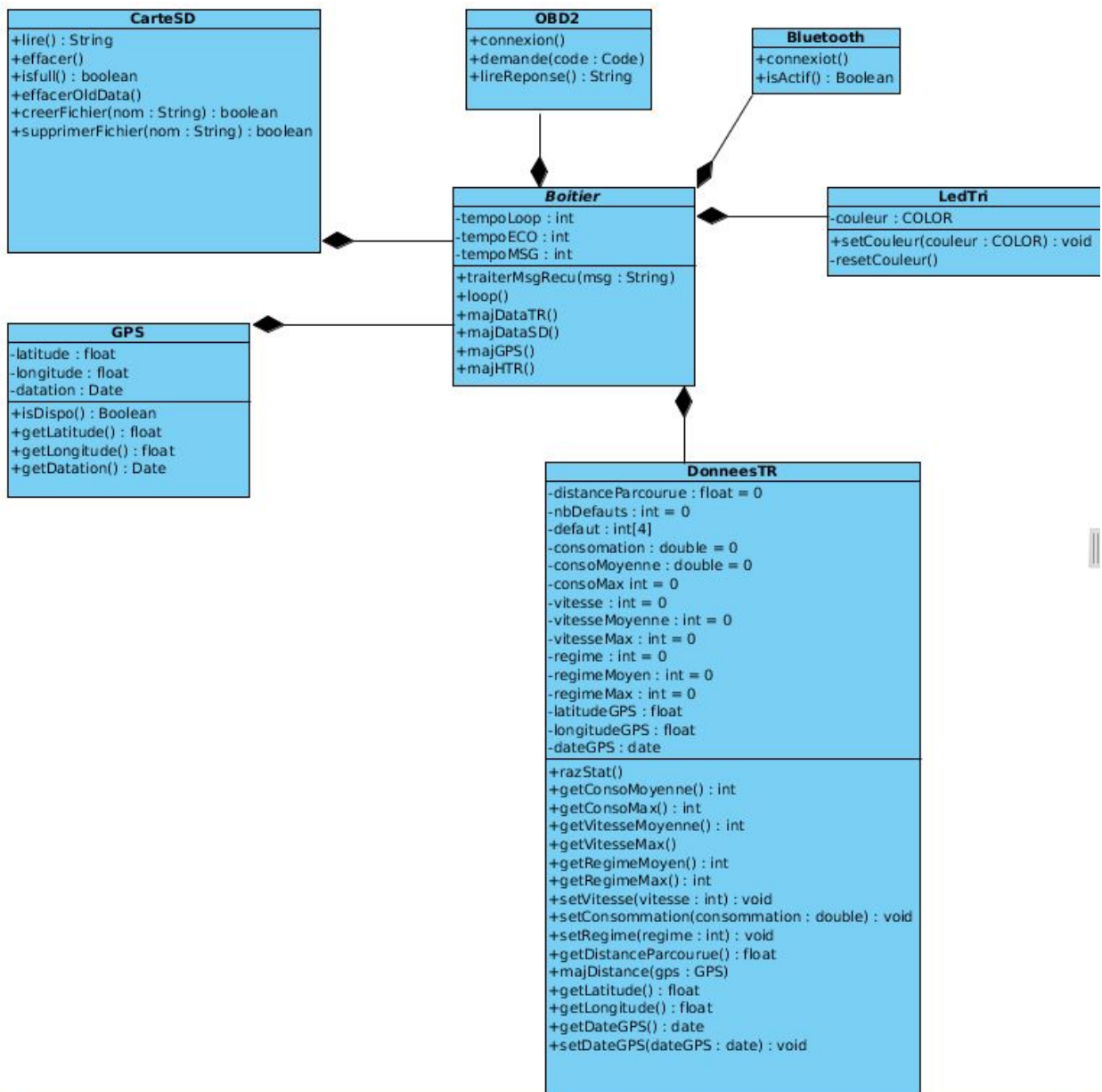
Pour ce projet nous avons choisi un module SIGFOX pour la communication entre le boîtier et alcis car l'abonnement est moins cher



---

que le GSM ou LORA, le module a une faible consommation équivalent à LORA mais avec une distance de transmission bien meilleure en milieu urbain. De plus Sigfox utilise une modulation UNB (ultra narrow band) ce qui signifie que la bande de fréquence pour l'envoi d'un message est comprise entre 868.034Mhz et 868.226MHz

## Diagramme de classe Alfox 2



---

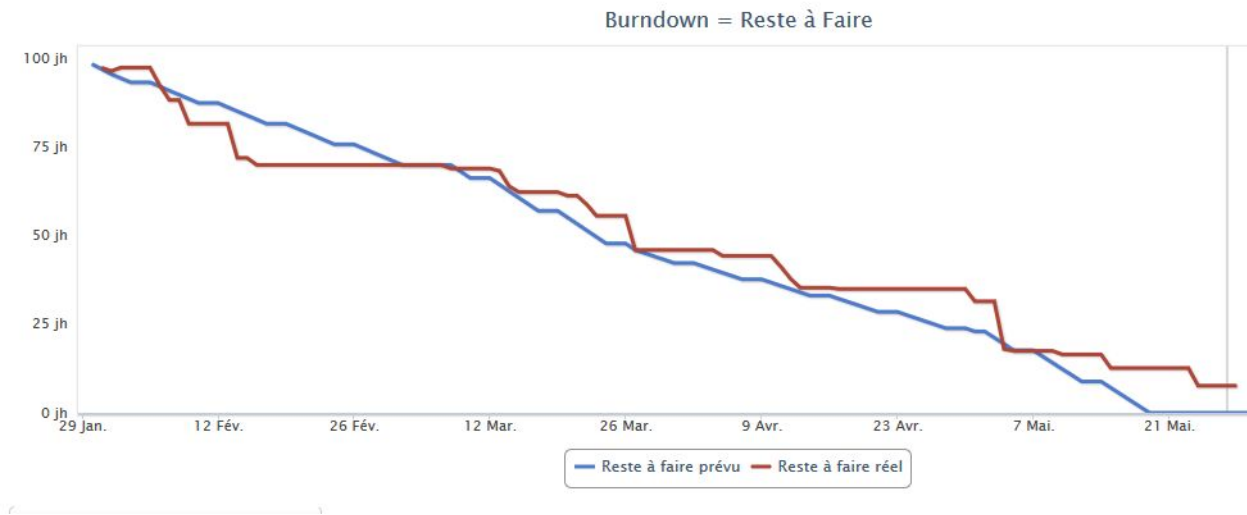
## partie technique

Pour ce projet nous avons besoin d'une carte de développement. nous avons le choix entre une carte Raspberry et une carte Arduino. La carte Raspberry ayant un O.S., la carte consomme nettement plus. Nous nous sommes donc tournés vers le choix le plus économe en énergie c'est à dire une carte arduino. Pour ce projet, vu que nous utilisons plusieurs classes dont certaines ne peuvent pas être arrêtées en cours, il nous a fallu faire un ordonnanceur. Il nous permettait de changer de méthode et d'attendre l'interruption pour récupérer les données recherchées. La carte nécessitait donc une certain RAM pour pouvoir gérer l'ordonnanceur.

type de carte	MKRFOX1200	UNO	MEGA
tension de sortie des pins I/O	3.3V	5V	5V
nombre de port série	1	1 (mais il est possible d'avoir des port série logiciel)	8
ram	32KB	2KB	8KB
mémoire morte	256KB	1KB	4KB
vitesse de l'horloge	48 MHz	16MHz	16MHz
prix	35€( mais le module SIGFOX est intégré)	20€	35€

Vu que nous utilisons un module SIGFOX la carte MKRFOX 1200 était très intéressante. De plus c'est une carte qui a une RAM suffisante pour l'ordonnanceur mais pour le projet nous savons que nous avons besoin d'un port série pour le GPS et d'un pour le module bluetooth il nous a donc fallu trouver un moyen de simuler les port série pour la carte MKRFOX 1200.

## Burndown Tout le projet



Nous avons séparé notre projet en "Sprints" dont la durée moyenne est de 3 semaines :

- Sprint 1 : Analyse du projet et du matériel nécessaire, réduction des risques puis début de la programmation.
- Sprint 2 : Avancée des programmes et analyse des données nécessaires. Un léger retard est visible durant le milieu du Sprint car nous avons mal séparés les tâches, ce qui résulte en une chute drastique du travail restant en fin de Sprint.
- Sprint 3 : Un changement de matériel a provoqué un retard. En effet, suite à un changement de carte (passage d'une Arduino Uno à une MKRFOX1200), nous avons dû mettre à jours tous nos programmes précédemment rédigés (Retard qui a été rattrapé depuis).
- Sprint 4 : Intégration des classes

