

Fluidification du trafic via des feux tricolores dynamiques
PPE 1827
Valorisation Publication

Cahier des charges fonctionnel et technique

Green'n GO

Table des matières

I.	Objet	2
II.	Justification du système	3
1.	Contextualisation	3
2.	Justification du besoin initial	5
3.	État de l'art	6
III.	Valorisation du système	8
IV.	Spécifications techniques	13
1.	Modélisation des feux	13
2.	Choix des composants électroniques	15
3.	Dimensionnement des panneaux photovoltaïques	17
4.	Modèles mathématiques	20
5.	Modélisation graphique	23
V.	Conception de l'application de démonstration	24
VI.	Rédaction du papier de recherche	25
VII.	Feuille de route	25
VIII.	Références bibliographiques	27

I. Objet

L'objectif de notre étude est de proposer une méthode de fluidification du trafic routier. Cette étude se focalisera sur une intersection régie par des feux de circulation en prenant en compte deux types de véhicules : ceux à caractère prioritaire ainsi que les véhicules lambda. Ce projet s'inscrit dans le cadre du PPE (projet pluridisciplinaire en équipe) d'ING4 de l'ECE Paris. Nous avons choisi de valoriser ce projet via une démarche de recherche aboutissant éventuellement sur une publication scientifique.

Dans un premier temps, nous développerons un algorithme de fluidification du trafic répondant à l'objectif énoncé ci-dessus. Dans un second temps, nous réaliserons une maquette implémentant ledit algorithme afin de démontrer son applicabilité.

Ce projet s'inscrira dans le mouvement de l'IoT afin de construire la ville intelligente de demain, l'objectif final étant de participer à une conférence scientifique grade A/B sur le thème de l'IoT et de la smart-city telle que I2C.

II. Justification du système

1. Contextualisation

À l'heure où les villes deviennent des mégapoles à cause de la croissance démographique et de l'exode rural massif au niveau mondial, le trafic routier urbain est en explosion. D'autant plus que l'augmentation du niveau de vie dans les pays émergents, tels que l'Inde ou la Chine engendrant l'acquisition exponentielle de véhicules par ces populations, impose que la cohabitation entre les automobilistes, les piétons, les cyclistes et les nouveaux types de transports émergents (trottinettes électriques, gyroroue) soit repensée. Tous ces phénomènes sociaux et technologiques créent de nouveaux défis dans la vie urbaine d'aujourd'hui et de demain. Dans cette problématique transversale se cache une source d'inquiétude d'autant plus importante qu'elle peut être vicieuse.

En effet, chaque année, le trafic congestionné pèse sur le quotidien de tous. Source de pollution, on observe une augmentation de 13% des émissions de CO₂ à Paris entre 2013 et 2030 dues en grande partie aux embouteillages. Mais les bouchons pèsent aussi sur le portefeuille : les embouteillages sont à l'origine d'une hausse de 50% des coûts liés aux embouteillages pour les automobilistes du monde entier amenant la facture annuelle à 4123€ par an pour un Parisien lambda d'ici 2030 (contre 2883 aujourd'hui). Ces coûts sont liés à l'augmentation de consommation de carburant par les automobilistes qui gaspillent leur plein dans les bouchons. Mais d'autres coûts dans des sphères plus hautes que le simple contribuable sont également générés. Dans l'optique de la taxe carbone européenne, ce surplus d'émissions de CO₂ représente un coût certain pour l'État français : on l'évalue à 22 milliards d'ici 2030 pour la France.

Au-delà de l'aspect économique et environnemental, l'aspect social n'est pas négligeable. Les bouchons augmentent l'irritabilité des utilisateurs par les retards qu'ils causent, et diminue la sécurité sur la route. Les embouteillages sont source de dangers pour les véhicules légers (motos, cyclomoteurs, scooters) et piétons amenant le nombre d'accidents corporels en juillet 2018 à 1507, selon la DREIA. Ainsi, les points énoncés ci-dessus montrent que le trafic urbain devient un obstacle à l'élaboration d'un monde compatible avec le développement durable.

Dans l'optique de la ville intelligente de demain, notre projet s'inscrit dans un courant privilégiant la smart-city et la technologie au service de l'Homme et de l'environnement. Ainsi, nous comptons nous reposer sur l'IoT et nos connaissances pour offrir une démonstration de ce que pourra être la solution aux embouteillages de demain dans un système intégrable, efficace et éco-responsable répondant aux besoins de la ville du futur.

Pour conforter notre perception du besoin dans la Smart City de feux tricolores dynamiques, nous avons réalisé un benchmark de ce qui existait comme feux tricolores intelligents.

Aujourd’hui, en termes de feux tricolores intelligents, nous avons peu de modèles disponibles. semblables à ce que nous proposons.

Ainsi, nous trouvons principalement des feux tricolores à alimentation solaire. Cependant, l’énergie solaire ne permet pas de les alimenter complètement. Elle permet simplement de réduire la consommation en électricité en rechargeant au fur et à mesure la batterie dont ils dépendent.

IBM a déposé un brevet basé sur la collecte d’informations sur le trafic grâce à des caméras. IBM rentre ensuite dans un algorithme employant l’intelligence artificielle pour réguler l’évolution des deux tricolores réduisant ainsi le temps d’attente aux feux. Le brevet définit sa solution comme suit : “The present invention relates generally to the field of traffic control, and more particularly to controlling a traffic signal through cognitive computing that incorporates real time data at an intersection.” (« L’invention en question est en lien avec la gestion du trafic routier en général, et plus particulièrement avec le contrôle des feux tricolores grâce à l’information cognitive qui intègre le traitement des données d’une intersection en temps réel. »).

Un autre dispositif permet de contrôler le comportement du feu tricolore en fonction du comportement des conducteurs : ce sont des feux avec un radar de vitesse intégré. Si un automobiliste dépasse la vitesse instaurée (prenons 50 km/h), le feu passe au rouge pour forcer le conducteur à ralentir. Dans le cas contraire, le feu reste vert. Cela permet de faire ralentir les automobilistes à l’approche de carrefours compliqués ou aux alentours de zones sensibles (écoles, hôpitaux, etc.).

Ensuite, nous disposons d’outils de régulation de trafic utilisant l’IoT. On trouve ainsi une multitude de caméras qui communiquent via des canaux Ethernet, en Bluetooth ou en WiFi (donc particulièrement sensibles aux interférences) avec des feux. Ces caméras peuvent être de simples caméras vidéos qui envoient leurs informations à un algorithme et qui selon le résultat, envoie les instructions de régulation au feu. Cependant, elles présentent de nombreux inconvénients : elles sont sensibles au contre-jour (on ne peut donc les installer partout), détection difficile la nuit et les intempéries troubulent leur acuité ou visibilité. La plupart de ces installations sont onéreuses (pouvant monter jusqu’à 36 000€).

2. Justification du besoin initial

Comme précisé plus haut, les embouteillages sont une épée de Damoclès pour le développement durable de nos futures smart-cities. Ainsi, l'enjeu de la gestion des embouteillages est clair mais les moyens pour le faire le sont beaucoup moins. En effet, le périmètre du sujet n'étant pas exhaustif et aucun client ne pouvant nous exprimer clairement et simplement son besoin, il nous a fallu définir le périmètre de notre projet nous-même. De même, nous devrons transposer la problématique sous forme de besoins concrets qui puissent être adressés simplement dans le cadre d'un projet PPE qui se veut tout de même ambitieux.

Partant d'un constat clair, nous avons remarqué que la majeure partie du trafic se congestionne au niveau des intersections, notamment celles régies par des feux de circulation. Il semble, en effet, que les automobilistes perdent leur bon sens et s'aventurent aveuglément sur le croisement quitte à rester bloqués, tant que le feu est vert.

Notre état de l'art, démarrant du rapport de stage de Jérémy PETIT, stagiaire au sein du laboratoire de recherche de l'ECE en 2017, nous a permis de comprendre l'enjeu global que joueront les feux de circulations dans les intersections. Étant stratégiquement implantés puisque directement présents sur les points névralgiques des embouteillages, ils agissent en tant qu'agents de régulation puisque capables de stopper ou autoriser le flux de circulation.

Notre solution se porte donc sur cet objet qu'est le feu de signalisation. Notre but sera de le rendre intelligent, en le connectant au réseau grâce au potentiel de l'IoT et lui conférant la capacité, via son passage du rouge au vert, de réguler les flux circulatoires afin de dégager les intersections.

3. État de l'art

Afin d'amorcer notre projet, nous avons mené des recherches concernant des études scientifiques liées à notre sujet. À l'issue de ce processus, nous avons sélectionné dix textes qui nous semblaient les plus pertinents à analyser. Ces derniers sont listés ci-dessous. Une analyse de chacun d'eux est disponible dans notre rapport sur l'état de l'art.

- [1] Petit J., Zitouni R., & Georges Laurent., Réseaux IoT pour le contrôle des feux de signalisation dans les villes intelligentes (2017), IEEE International Smart Cities Conference, ISC2 2018., 2018.
- [2] Tubaishat M., Shang Y., & Shi H., Adaptive traffic light control with wireless sensor networks, Consumer Communications and Networking Conference, CCNC 2007. 4th IEEE (pp. 187-191), 2007.
- [3] Yousef K. M., Al-Karaki M. N., & Shatnawi A. M., Intelligent traffic light flow control system using wireless sensors networks. J. Inf. Sci. Eng., 26(3), 753-768, 2010.
- [4] J. Cunha, C. Cardeira and R. Melício, "Traffic Lights Control Prototype Using Wireless Technologies", Renewable Energy and Power Quality Journal, pp. 1031-1036, 2016.
- [5] M. Bagheri, M. Siekkinen and J. Nurminen, "Cloud-Based Pedestrian Road-Safety with Situation-Adaptive Energy-Efficient Communication", IEEE Intelligent Transportation Systems Magazine, vol. 8, no. 3, pp. 45-62, 2016.
- [6] K. Dresner and P. Stone, "A reservation-based multiagent system for intersection control", IFAC Proceedings Volumes, vol. 37, no. 8, pp. 136-141, 2004.
- [7] Sundar, R., Hebbar, S. and Golla, V. (2015). Implementing Intelligent Traffic Control System for Congestion Control, Ambulance Clearance, and Stolen Vehicle Detection. IEEE Sensors Journal, 15(2), pp. 1109-1113.
- [8] P. Bellavista , F. Caselli, A. Corradi and L. Foschini, "Cooperative Vehicular Traffic Monitoring in Realistic Low Penetration Scenarios: The COLOMBO Experience", Sensors, vol. 18, no. 822, 2018.
- [9] M. Collotta, G. Pau, G. Scatà and T. Campisi, "A dynamic traffic light management system based on wireless sensor networks for the reduction of the red-light running phenomenon", Transport and Telecommunication, vol. 15, no. 1, pp. 1-11, 2014.
- [10] H. Li, M. Tsukada, F. Nashashibi and M. Parent, "Multivehicle Cooperative Local Mapping: A Methodology Based on Occupancy Grid Map Merging", IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 5, pp. 2089-2100, 2014.

Après avoir listé les points forts et les points faibles de chacun de ces textes, nous avons pu explorer les éventuelles pistes d'amélioration que nous pourrions apporter à un tel système de feux de circulation connectés, à partir du tableau récapitulatif ci-dessous.

	Acquisition de données en temps réel	V-to-X	Priorité aux véhicules d'urgence	Prise en compte du gabarit des véhicules	Scalabilité du modèle	Modèle applicable sur voies multiples	Efficacité énergétique	Sécurité des Protocoles
[1]	✓	✓	✓					
[2]	✓			✓		✓	✓	
[3]	✓	✓		✓	✓	✓	✓	
[4]	✓					✓	✓	✓
[5]	✓	✓				✓	✓	
[6]	✓	✓				✓		
[7]	✓	✓	✓					✓
[8]	✓	✓						
[9]	✓			✓		✓	✓	✓
[10]	✓	✓				✓		
Green'n GO	✓	✓	✓	✓			✓	✓

À mesure de notre avancement dans le projet, nous avons été amenés à consulter un certains nombre d'autres publications scientifiques afin d'aller plus loin dans nos spécifications techniques.

III. Valorisation du système

Ainsi, le feu intelligent représente le cœur de notre projet. Nous nous reposerons sur une approche de la circulation par des théories de file d'attente. En effet, notre état de l'art se base sur plusieurs recherches montrant l'efficacité de cette considération dans l'approche théorique du sujet.

Notre rendu final prendra en compte que ce PPE n'est pas un projet de recherche à part entière malgré sa nature ambitieuse et complexe. Ainsi, notre rendu se présentera sous la forme d'un rapport écrit et d'un prototype visant à être présenté lors d'une conférence.

Nous avons ici une double mission. La première est celle de démontrer que notre modèle théorique fonctionne ainsi que notre algorithme testé et répondra de manière concrète à la problématique. Il sera implémenté dans une application au design travaillé dotée d'une interface simple d'utilisation et accessible au plus grand nombre afin de rendre un format présentable à n'importe quel public. Les prototypes (mock-ups) de cette application seront donnés par la suite.

Dans un second temps, nous rendrons le projet palpable, via une maquette physique. Celle-ci se compose d'une intersection à laquelle se trouveront plusieurs feux rouges, alimentés par une batterie visant à autonomiser le circuit en cas de panne ou de sous-exposition des panneaux photovoltaïques qui serviront à alimenter cette batterie. Ces panneaux seront dimensionnés selon plusieurs critères : la localisation choisie (Paris), les différents types de rayonnement tout au long de l'année ainsi que la dimension de la batterie qui alimente le système.

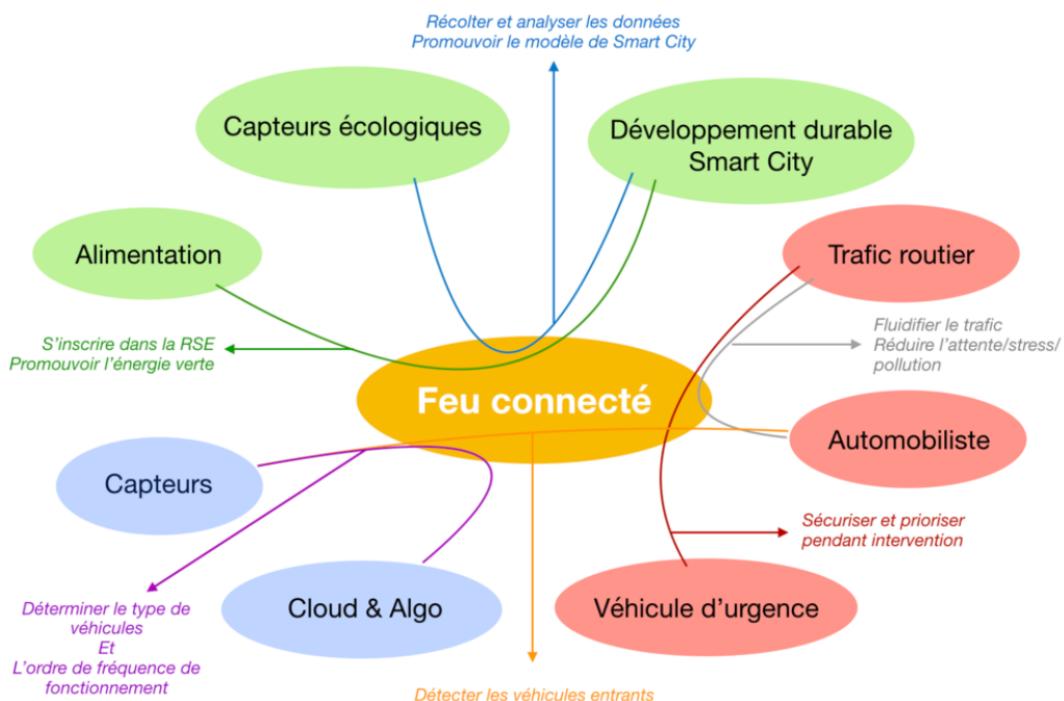


Diagramme APTE de notre projet

Afin d'effectuer une analyse fonctionnelle de notre projet, nous avons réalisé un diagramme pieuvre à l'aide de la méthode APTE comme présenté ci-dessus. Pour ce faire, nous avons établi les différents éléments extérieurs reliés aux feux connectés par des fonctions de service.

Dans un premier temps nous avons déterminé le trafic routier, l'automobiliste et le véhicule d'urgence comme entités à partir desquelles nous avons pu dégager deux fonctions principales. La première étant de fluidifier le trafic, réduire l'attente, le stress et la pollution, elle relie le trafic routier et l'automobiliste. La deuxième concerne les véhicules d'urgence et le trafic routier, elle permet de sécuriser et de prioriser les véhicules d'urgence pendant leurs interventions.

Les capteurs vont nous permettre de détecter les automobilistes, ces données seront envoyées à l'algorithme et au cloud afin de déterminer le type de véhicule entrant et par la suite la fréquence de fonctionnement du feu tricolore.

Dans le cadre du développement durable et des Smart Cities, nous pensons que l'alimentation de nos feux doit s'inscrire dans la RSE et promouvoir une énergie verte. Pour ce faire nous pouvons également opter pour des capteurs écologiques qui nous permettront de récolter et d'analyser les données de l'intersection. Le schéma fonctionnel ci-dessus présente un cas d'usage simple au niveau d'un feu de circulation. La voiture s'avance vers le feu. Elle est détectée par des capteurs sans fil positionnés au niveau de ce dernier. Les informations de la voiture sont ensuite transmises, par le biais de Remotes Zolertia, vers le système central sur lequel tourne un script python ainsi qu'un algorithme de traitement des données. Ces données sont ensuite envoyées au Cloud pour stockage.

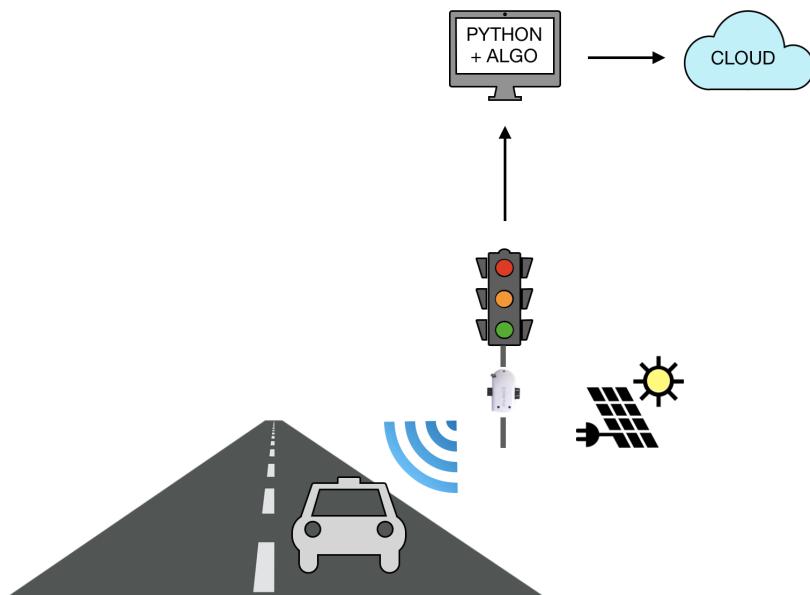


Illustration globale du fonctionnement de notre système à l'échelle d'un feu

Ce prototype vise à montrer la faisabilité et la viabilité de la solution que nous proposons. C'est pourquoi elle se décompose en 2 temps : premièrement une maquette de petite nature qui montrera rapidement que notre algorithme est capable de discerner différents types de véhicules. Il s'agira notamment de distinguer les véhicules d'urgence et les véhicules lambda via des capteurs sensibles qui donneront l'ordre de passage au feu vert dans un cas prédéfini (approche d'un véhicule d'urgence par exemple).

Dans cette maquette, deux véhicules miniatures seront présents à une intersection : l'un sera une voiture lambda et l'autre un véhicule d'urgence. Ces usagers se trouvent sur une intersection régie par des feux alimentés via un panneau photovoltaïque hors réseaux. Bardés de capteurs au choix (détecteur RFID, capteur de présence, température, pollution), ils seront reliés via un cloud à une unité centrale décideuse qui, à la réception des données, donnera l'ordre de passage au vert pour le véhicule d'urgence.

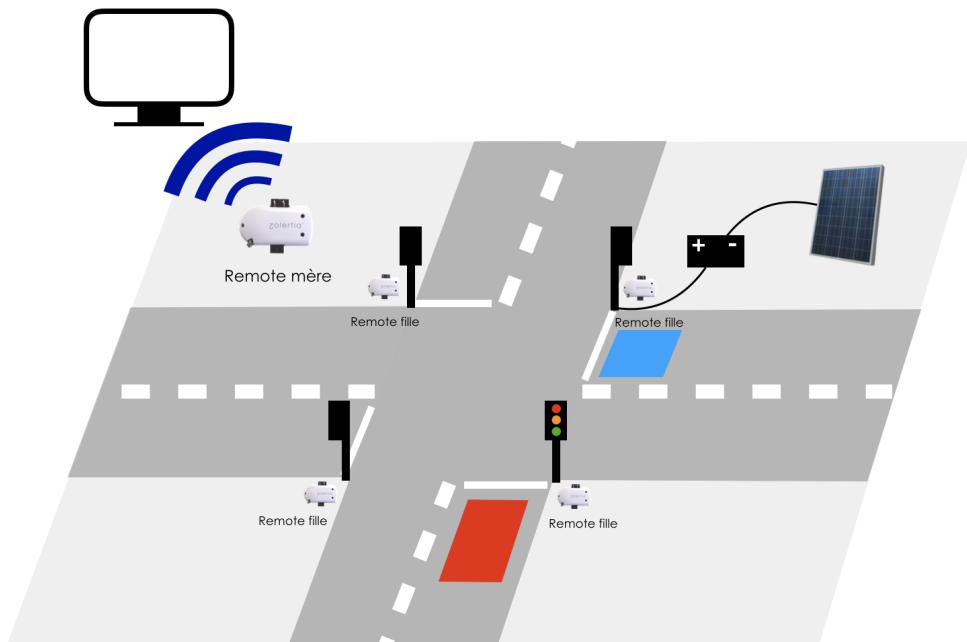


Illustration globale du fonctionnement de notre système à l'échelle du carrefour

Le schéma de fonctionnement de notre système peut être résumé en un diagramme SADT, qui représente les entrée et sorties du système. Servant à avoir une idée globale du système vu de l'extérieur, celui peut se représenter par le schéma qui suit :

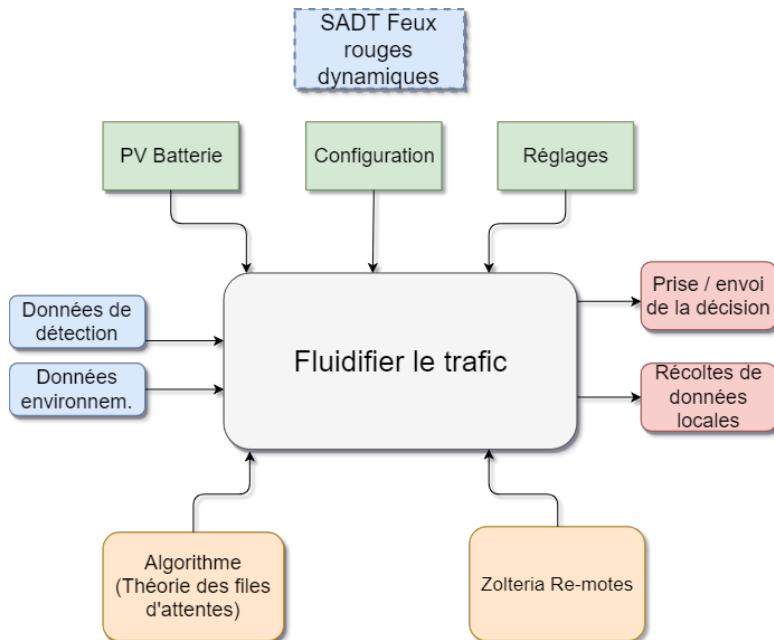


Diagramme SADT de notre projet

On voit que les données des capteurs sont envoyées au système par les Re-Motes, celui-ci se charge de traiter ces données via l'algorithme afin de décider de l'ordre qu'il enverra en sortie aux feux (passage au rouge/vert ou ne rien faire).

Des données environnementales sont aussi récoltées et envoyées pour stockage dans un cloud, dans lesquelles elles pourront servir de matière première au Big Data.

L'alimentation en énergie se fera par des panneaux photovoltaïques reliés à des batteries permettant l'autonomie et le pilotage du système pour une hypothétique implémentation en smart grid. Cette alimentation indépendante nous permet de simplifier l'implantation de la maquette, ainsi que de rendre le système plus sécurisé en cas de panne d'approvisionnement de courant.

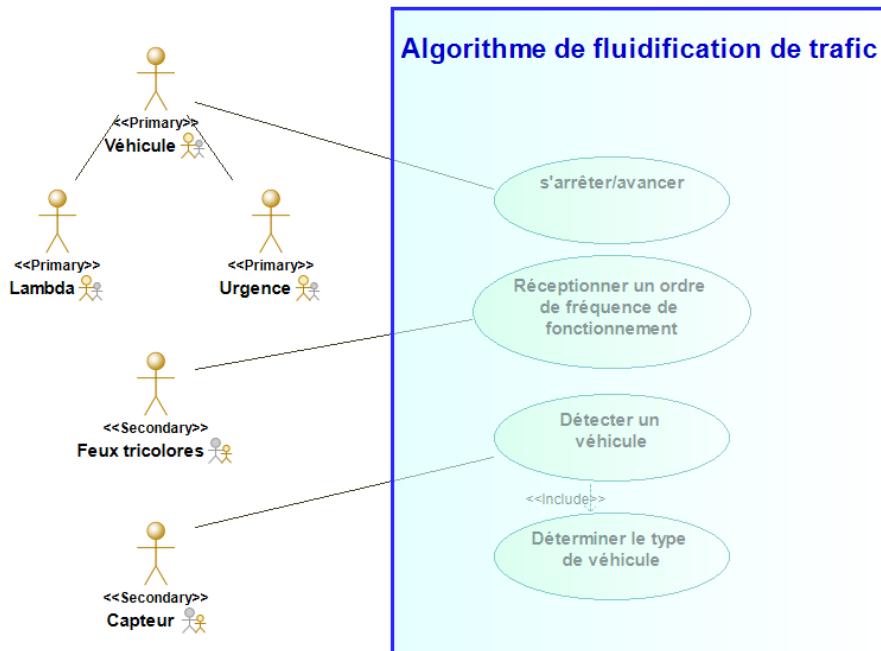


Diagramme de définition du besoin

Afin de conceptualiser le système, nous avons établi un modèle UML dans lequel a été défini l'algorithme de fluidification de trafic comme système. Nous distinguons trois acteurs différents : le feu tricolore, le capteur et le véhicule. Les acteurs interagissant avec le système, nous avons décrit les principales actions de chacun de ces acteurs.

Dans un premier temps, le capteur détecte le véhicule entrant et envoie ces données au remote mère qui récupère l'ensemble des données de l'intersection afin de les envoyer au système central. Celui-ci va par la suite traiter les données réceptionnées à l'aide d'une base de données afin de déterminer le type du véhicule entrant et plus exactement son possible caractère urgentiste. À partir des informations obtenues des remotes, l'algorithme détermine et envoie en réponse la meilleure fréquence de fonctionnement aux feux tricolores. Les feux mettent donc en application cette réponse, permettant par la suite au véhicule d'avancer ou non. Dans le cas d'un véhicule d'urgence, le feu passe au vert afin de favoriser son passage.

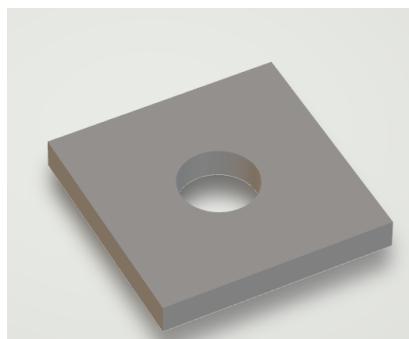
IV. Spécifications techniques

1. Modélisation des feux

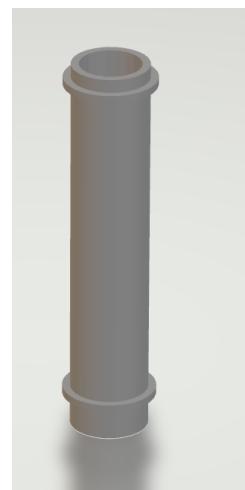
Ayant pour objectif cette année d'effectuer une maquette de démonstration assez grande, d'une hauteur approximative de 30 cm, nous avons choisi d'imprimer nos feux en 3D. Ainsi nous sommes particulièrement libres pour ce qui est du choix de la taille de chaque pièce et nous pouvons dimensionner l'ensemble en fonction de nos besoins en pièces électroniques (câblage, les, ...).

En vue de faciliter encore plus le montage, nous avons choisi de diviser nos feux en quatre pièces indépendantes (les unités sont en cm) :

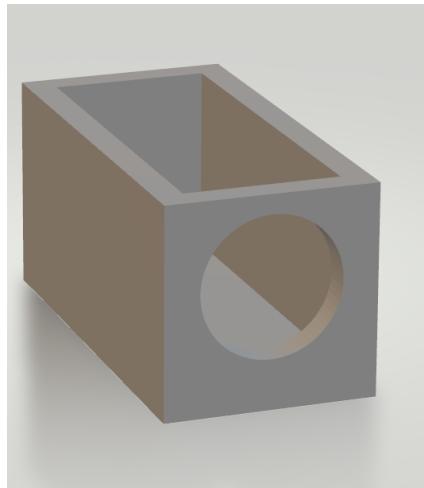
1. Le socle (1/7.5/7.5) constituera la base de notre feu. En son centre se situe une perforation pour pouvoir y emboîter le tube du feu rouge.
2. Le tube quant à lui fait une hauteur de 12.5 cm pour un diamètre de 3 cm. Il est muni d'anneaux pour améliorer la solidité de la structure une fois montée et collée.
3. Le boîtier, élément qui abritera les led (3.75/7.5/7.5), lui aussi comporte une perforation pour encastre le tube du feu.
4. Pour finir le capot du boîtier, muni de trois perforations, une pour chaque led (0.25/7.5/3.75). Cette pièce a été ajoutée pour pouvoir, avec facilité, intégrer les led à la structure.



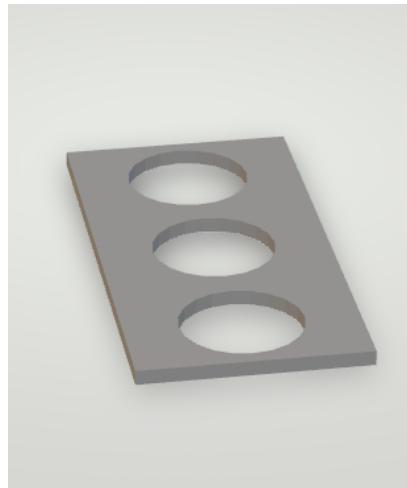
1. *Socle*



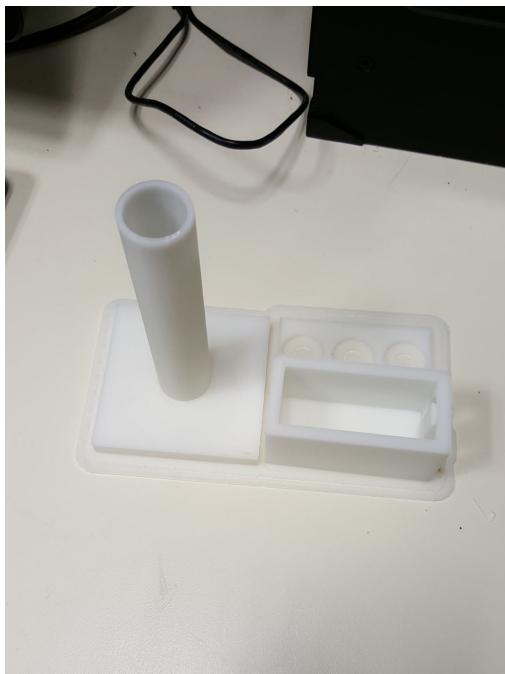
2. *Tube*



3. Boîtier



4. Capot du boîtier



Feu après impression



Feu assemblé

2. Choix des composants électroniques

A. Choix du CPU

Il est maintenant important de définir les critères qui nous permettront de choisir notre CPU.

Notre maquette nous impose d'avoir une unité de traitement qui nous permet de recevoir des entrées sorties, analogique et numérique afin de détecter l'arrivée de véhicules ainsi que le critère urgent ou non des véhicules en question et le gabarit de ceux-ci.

De plus, notre décision d'alimenter le système entièrement via des énergies vertes et de leur offrir une autonomie conséquente nous oblige à opter pour des technologies de faible consommation.

Ainsi nos choix se sont dirigés vers 2 technologies. La première étant les Arduino, simple d'utilisation et open source, celle-ci a attiré notre attention très vite.

La seconde technologie est moins accessible au public, les Re-Mote de Zolertia sont destinées au prototypage IoT d'application industrielle.

Cependant, notre CPU doit pouvoir transmettre des informations via des protocoles sécurisés et réagir en temps réel.

Nous avons choisi les protocoles IEEE 802.15.4 et 6LowPan car largement utilisés et sécurisés. et un OS léger, basé sur Linux nous permet de gérer le temps réel, contrairement à l'Arduino qui appelle le langage C, qui bien que rapide n'est pas optimal pour notre application.

Nous récapitulons les critères retenus dans le tableau qui suit :

	Temps réel	Basse consommation	Protocole sécurisé	Transceiver inclus
Re-Mote	X	X	X	X
Arduino	~	~	X	N/A

B. Choix des capteurs

Notre projet se situant dans l'IoT, il nécessite une écoute permanente de notre système, et une interaction avec le monde extérieur via des capteurs adaptés. Ainsi, notre système vise à détecter l'arrivée de véhicules et à en connaître le nombre.

Au delà de ça, notre optique d'intégration dans le développement durable, nous comptons utiliser les feux rouges comme des acteurs de la récolte de données environnementales.

Capteurs de température, de pression, d'humidité et de pollution de l'air.

Notre démonstration intègre donc un capteur de température qui vise à démontrer la faisabilité du concept dont nous présentons les données techniques :



Interface: compatible Grove
 Alimentation: 2,6 à 3,4 Vcc
 Consommation: 1,5 mA
 Température de service: 40 à 85°C
 Température de l'objet: 40 à 115°C
 Résolution de mesure: 0,02°C
 Précision: ±1°C
 Dimensions: 20 x 40 x 9,6 mm
 Poids: 8 g
 Protocole: I2C
 Non compatible avec Tinker Kit
 Référence Fabricant: 101020077

Nous avons choisi un capteur qui consomme très peu, et consiste en une simple thermistance Grove pour une connectique facile. Le protocole I2C compatible avec nos Zolertia rend simple la communication et la précision bien que faible et répond à nos attentes au niveau d'une démonstration.

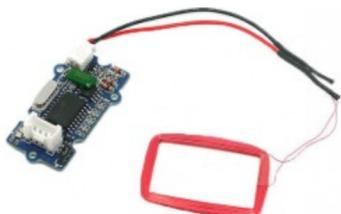
Un capteur de pression vient détecter la présence du véhicule et donner le nombre de voitures dans la file d'attente au feu.



Alimentation: 4,3 à 5,3 Vcc
 Consommation maxi: 500 µA
 Impédance de sortie: 1 k
 Distance de détection: 12,7 mm maxi
 Epaisseur maxi du matériau: 12 mm
 Température de service: -40°C à +85°C
 Dimensions: 38 x 32 x 8 mm.
 Module prêt à l'emploi.

Les mêmes critères ont été retenus : consommation, connectique, protocole de données et taille.

Les véhicules d'urgence viennent ajouter une composante supplémentaire, il est nécessaire de pouvoir les reconnaître et les différencier du trafic usuel. Ainsi nous avons opté pour un capteur RFID passif, qui détectera l'arrivée d'un véhicule d'urgence équipé d'un récepteur.



Alimentation: 5 vcc
 Sorties:
 - Uart: TTL, 9600 bauds, 8 bits de données et 1 bit de stop
 - Wiegand: sortie 26 bits (24 de données et 2 de vérification)
 Fréquence: 125 kHz
 Portée de détection: 7 cm maxi
 Dimensions:
 - carte: 40 x 20 x 15 mm
 - capteur: 43 x 33 x 3 mm

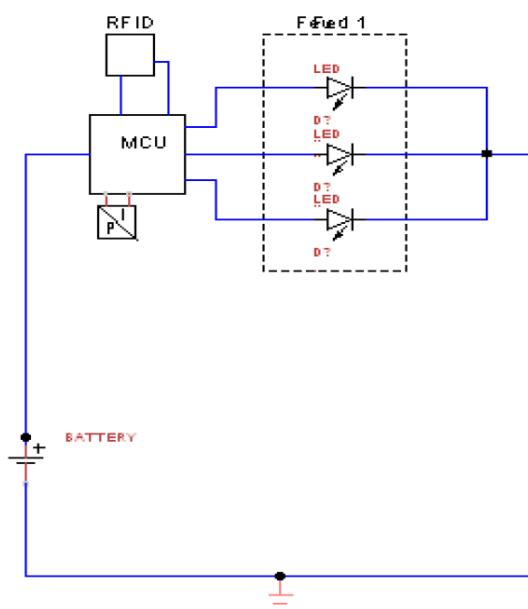
La simulation du carrefour nécessite des lumières, nous avons pour cela choisi de simples led CMS (RGB) pour leur polyvalence et leur basse consommation.



3. Dimensionnement des panneaux photovoltaïques

L'alimentation du système se fait donc via des panneaux photovoltaïques reliés à des batteries d'une capacité dépendante de plusieurs facteurs notamment l'autonomie requise et la consommation du système.

Il convient de définir les caractéristiques de consommation de notre système et donc de dresser un schéma électrique simple :



Après avoir clarifié le système électrique, il convient de dresser la consommation du système. Les éléments sont résumés dans le tableau qui suit :

Composant	Voltage	Amperage	Puissance	H de fonctionnement	consommation Wh	Consommation Ah
Re-Mote (CPU)	3,6	0,02	0,072	24	1,728	0,48
Re-Mote (TRANSCIEVER)	3,6	0,04	0,144	24	3,456	0,96
Led CSM	3,2	0,02	0,064	24	1,536	0,48
Capteur RFID Actif	3,3	0,026	0,0858	1	0,0858	0,026
Capteur RFID Inactif	3,3	Ignoré	Ignoré			
Capteur pression	3,3	Ignoré	Ignoré			
Capteur Temp	3,3	Ignoré	Ignoré			
Conso Système Wh/j		6,8058		amperage du système A		0,106
Conso Système Ah/j		1,946		Consommation Ah		2,544
Irradiance paris decembre	W/m ² /j	1360		Jours d'autonomie		1
Heure équivalente 1000 W/m ²		1,36		Profondeur de décharge max :		0,7
Puissance Crete du module	Wc	5,004265		Coefficient T°		0,7
Pertes temp et low irradiance		0,09				
pertes optiques		0,03		Capacité batterie retenue Ah		5,19
other		0,1		Voltage retenu V		12
Produit des efficacité		0,79443				
Puiss crete nette nécessaire		6,032991				

IC	Operating Range	Current Consumption	Notes
MSP430f2617	1.8V to 3.6V	0.1µA 0.5µA 0.5mA < 10mA	OFF Mode Standby Mode Active Mode @ 1MHz Active Mode @ 16MHz
CC2420	2.1V to 3.6V	<1µA 20µA 426µA 18.8mA 17.4mA	OFF Mode Power Down IDLE Mode RX Mode TX Mode @ 0dBm
ADXL345	1.8V to 3.6V	0.1µA 40µA to 145µA	Standby Active Mode
M25P16	2.7V to 3.6V	1µA 4mA to 15mA	Deep Power Down Active Mode
TMP102	1.4V to 3.6V	1µA 15µA	Shutdown Mode Active Mode

Table 1 — Approximate Current Consumption of Z1 circuits.

Nous détaillons ici les consommations de la carte Re-Mote dont les caractéristiques se trouvent dans la Datasheet de la Re-Mote. Le MPU MSP430f2617 consomme entre 1 mA et 10mA. Par souci de simplification, nous prendrons la valeur de 10mA car nous considérons que la carte sera utilisée au maximum de sa fréquence de calcul. Nous prendrons donc 20 mA de consommation pour les 2 Re-Motes de notre maquette.

Du côté du transceiver, on se doit de détailler le calcul : le transceiver CC2420 est en écoute constante soit 18mA minimum. Nous provisionnons 20 mA/transceiver, soit 40 mA pour les deux.

On ignore la consommation des capteurs de pression RFID passif ainsi que le capteur de température car elles sont négligeables par rapport aux autres.

D'après notre schéma électrique, si la Re-Mote se comporte comme un circuit en dérivation, nos capteurs et led nécessitent au minimum 3.3 V. Cependant, ne connaissant pas le comportement du circuit interne à la Re-Mote, nous avons opté pour 12V dans le cas où le système réagirait comme un circuit en série.

Le système est fait pour tourner 24h/24h pour respecter la contrainte temps réel que nous nous sommes imposée.

Après avoir déterminé les composants ainsi que leur consommation journalière, nous pouvons dimensionner notre système. Cependant, ayant une visée d'application réelle, il est important de ne pas négliger les pertes dues aux différents composants.

Nos chiffres proviennent du travail d'un enseignant de l'ECE, M. Jean José Wanegue [Source : Cours de Jean-José Wanegue Energie photovoltaïque, D'après Anne Labouret et Michel Villoz ©].

Nous avons choisi 2 jours d'autonomie afin de garder des proportions raisonnables. Nous avons enfin choisi d'autoriser la descente jusqu'à 30% de batterie (au lieu de 50% en général) pour des questions de proportions encore une fois.

Notre système est évidemment sur-dimensionné pour parer à toute éventualité.

Cela nous mène à une batterie de 12V pour 7Ah environ.

Pour les panneaux photovoltaïques, il faudra que ceux-ci fournissent une tension de 12V pour la charge de la batterie et que l'ensemble du module photovoltaïque fournisse 6 Wc.

Nous avons opté pour un panneau de 10 Wc. En effet, des raisons de budget (achat par pack) nous ont poussé à choisir un panneau de 10 Wc d'une taille de 30 * 60 cm soit 0,18 m².

4. Modèles mathématiques

On s'appuiera sur des modèles mathématiques probabilistes afin de développer notre algorithme de modélisation du trafic, en particulier la théorie des files d'attente.

Tout d'abord, nous allons introduire quelques notions au sujet de la configuration du carrefour considéré. Une illustration de ce dernier est présentée dans la figure suivante. Il s'agit d'un carrefour à quatre voies, dotées chacune d'un unique feu de circulation.

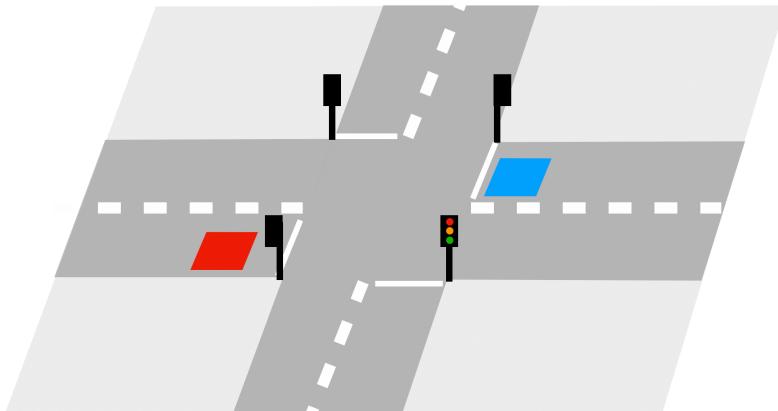
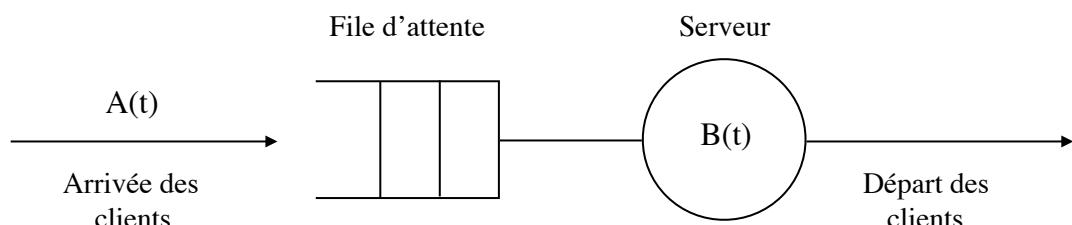


Illustration 2D du carrefour étudié

Avant d'étudier le fonctionnement de celui-ci, nous allons introduire quelques notions de base relatives à la théorie des files d'attente. Il est possible de modéliser simplement une file d'attente comme suit :



Modélisation d'une file d'attente

On considère une file d'attente par le modèle suivant : les clients arrivent au guichet selon un processus exponentiel de Poisson, que l'on explicitera ultérieurement. Ici, on assimile les clients aux voitures engagées sur la voie considérée et le guichet au feu de circulation présent sur cette voie. La file d'attente est finie, elle est limitée par la longueur de la voie. On considère qu'une voie débute après que l'on ait quitté le carrefour. On considère que le véhicule a été traité une fois qu'il s'est engagé sur une autre voie, c'est-à-dire une fois qu'il a quitté le carrefour et non seulement une fois qu'il a dépassé le feu. On part du principe que les clients ne peuvent pas renoncer, c'est-à-dire quitter la file avant d'être servis et ne peuvent pas non plus changer de file une fois engagés. Afin de simplifier notre analyse, nous ne prenons en compte ni le dépassement des véhicules au feu, ni les piétons. Cela implique que les véhicules dans chaque voie doivent traverser le carrefour en mode FIFO (« First in First out ») ou en d'autres termes : « Premier arrivé, premier servi ». On ne prend pas non plus en considération les évènements ayant lieu au niveau des voies antérieure et postérieure à notre voie d'étude. On considère aussi que lors du traitement d'un véhicule au feu, la file d'attente ne s'est pas agrandie.

Le temps de service des véhicules, c'est-à-dire le temps que met un véhicule de la file pour être traité, est également décrit par une loi de Poisson. Les temps de service sont des variables exponentielles identiques pour chaque véhicule et indépendantes du processus des arrivées. On note λ le paramètre des arrivées et σ le paramètre de la loi des temps de service.

On effectue une simulation à événements discrets. On considère que le temps de service ne dépend pas du type de client et que les voitures mettent toutes le même temps à être traitées par le serveur. Le temps de service ne dépend pas non plus du moment de la journée pendant lequel circulent les voitures. Le temps de service ne dépend pas des temps d'arrivée ni de la longueur de la file. Le service se fait à l'unité, c'est-à-dire que les véhicules sont traités un par un. Ce serveur est unique.

Dans cette première approche, nous ne prenons pas en compte dans notre étude le caractère prioritaire des véhicules d'urgence. Nous nous intéresserons à cette problématique ultérieurement lors de la phase d'établissement de notre algorithme.

Afin de modéliser notre file d'attente, nous avons recours à un processus de Poisson qui régit les arrivées et les départs de nos clients. Un processus de Poisson est un processus stochastique de Markov, selon lequel la distribution conditionnelle des états futurs, étant donnés les états passés et présents, ne dépend que de l'état présent.

Étude d'une file d'attente M/M/1

Soit $\{N_t, t \geq 0\}$ un processus de Poisson d'intensité λ . On note N_t le nombre de clients présents dans la file (et au guichet) à l'instant t . On suppose que N_0 est une variable aléatoire indépendante du processus des arrivées et des temps de service.

D'après la propriété d'absence de mémoire de la loi exponentielle, la durée avant l'arrivée du prochain client est une variable exponentielle de paramètre λ et le temps de service restant du client au feu est une variable exponentielle de paramètre σ . Soit $\lambda < \sigma$.

La loi qui définit le nombre de voitures dans la file à l'instant $t + s$, $s \geq 0$ sachant $N_t = i$ définie par est équivalente à la loi en $t = 0$ sachant $N_0 = i$.

On en déduit que le nombre moyen N de clients dans la file (longueur de la file) est

$$N = \frac{1}{2} + \frac{\lambda}{\sigma - \lambda}$$

et le temps moyen T qu'un client passe dans la file est

$$T = \frac{1}{2\sigma} + \frac{1}{\sigma - \lambda}$$

On ne souhaite pas agir directement sur le temps de service des véhicules au feu. On veut plutôt pouvoir diminuer la longueur de la file d'attente afin de diminuer le temps d'attente. Pour ce faire, on veut pouvoir agir sur le paramètre σ . En particulier, $N(\sigma)$ est une fonction inverse du paramètre σ . Ainsi en augmentant σ , $N(\sigma)$ diminue.

En réarrangeant les formules précédentes, on obtient

$$\sigma = \lambda \frac{2N + 1}{2N - 1}$$

On obtient une valeur du temps moyen T qu'un véhicule passe dans la file en fonction du nombre moyen N de véhicules

$$T = \frac{2N - 1}{\lambda} \left(\frac{1}{2N + 1} + \frac{1}{\lambda} \right)$$

On s'intéresse aux différents scénarios entre les véhicules engagés au niveau du carrefour. On identifie notamment des situations sans conséquences sur la gêne utilisateur et à l'inverse, des situations conflictuelles de croisement entre les véhicules. On illustre notre intersection en 2D dans la figure suivante :

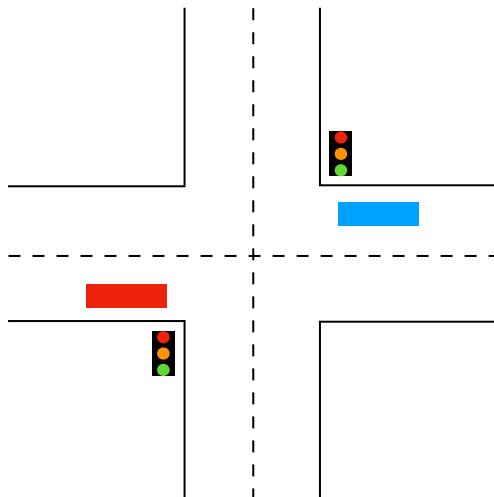
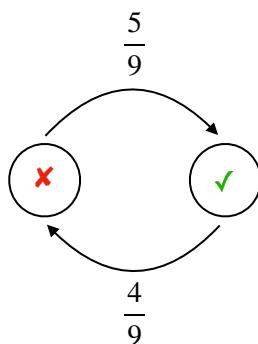


Illustration 2D du carrefour étudié

Deux véhicules sont présents sur le schéma, sur des voies qui se font face. Chacun d'eux est en attente à son feu respectif. On imagine que leurs feux passent simultanément au vert et par conséquent les véhicules s'engagent en même temps sur le carrefour. Dans le cas où les voitures tournent à leur gauche ou à leur droite, ou continuent tout droit, alors elles ne se croisent pas et il n'y a pas conflit. De même, si l'une des voitures choisit de continuer sa route tout droit et que l'autre décide de prendre la voie à sa droite, alors il n'y a pas de croisement. En revanche, pour tous les autres cas de figure, une situation de conflit a lieu. On les recense tous ces cas de figure dans le tableau suivant :

Décision du véhicule	Droite	Gauche	Gauche
Droite	✓	✗	✓
Gauche	✗	✓	✗
Tout Droit	✓	✗	✓

On peut représenter par une chaîne de Markov la probabilité de passer d'un état de conflit à un état de non conflit :



Chaîne de Markov de notre modèle de conflit

5. Modélisation graphique

La modélisation graphique de l'intersection se fera au moyen de l'outil Matlab. On choisit d'utiliser ce langage de programmation qui est particulièrement adapté à la réalisation d'interfaces graphiques de manière simple.

V. Conception de l'application de démonstration

Description de la navigation dans et entre les pages de l'application.

1^{ère} page : Nous y présentons le thème de l'application. Pour cela, nous avons modélisé, de façon élémentaire, un feu de signalisation. Nous avons ensuite introduit l'identité de notre démonstration et concept grâce à la bannière de présentation comportant le nom du concept.

Interactions :

Lorsque l'utilisateur clique sur « Lancer la démo », la voiture démarre, émet un petit nuage et roule vers la droite pour faire la transition vers la deuxième page. Cette transition prendra la forme d'un fondu entre le noir de la première page et le blanc de la deuxième page.

2^{ème} page : Nous avons réduit au maximum le questionnaire à remplir pour les conditions de modélisation de la situation d'embouteillage. Pourquoi cela? Limiter le nombre de clics et d'engagement forcé de l'utilisateur : un utilisateur qui doit remplir trop de champs de renseignements quitte 2 fois plus rapidement une application.

Interactions :

- L'utilisateur peut soit taper directement le nombre de véhicules dans le champ soit utiliser les flèches
- Une fois les champs remplis, l'utilisateur clique sur le bouton « Envoyer »
- S'affiche alors la voiture lors du chargement des données dans l'algorithme et le temps de compilation de ce dernier. Les trois petits nuages émis par la voiture s'affichent progressivement et recommence l'affichage progressif par cycle.
- L'utilisateur peut à tout moment revenir en arrière ou progresser grâce aux petites voitures en bas à gauche et à droite de l'écran.

3^{ème} page : Cette page permet la visualisation du résultat de compilation de l'algorithme. Ce schéma modélise simplement la situation voulue par l'utilisateur (nombres de véhicules remplis) et la régulation des feux tricolores en fonction de cette dernière. Les quatre points de couleur symbolisent les feux (la couleur variera en fonction de l'évolution des feux pour fluidifier le trafic).

Interactions :

- Le bouton en haut à droite permet de recommencer la simulation autant de fois que l'utilisateur le souhaite (remise à zéro de la situation puis évolution).
- Le bouton en bas à droite permet de revenir à l'écran d'accueil (1^{ère} page).

Cette simulation sera projetée en temps réel sur la maquette réelle grâce à un projecteur numérique.

Nous avons donc pensé cette application pour limiter le nombre de clics pour obtenir le résultat afin de maximiser le temps d'interaction. Nous avons choisi un design simple au vu du peu de temps de développement que nous disposerons. Ainsi, les efforts seront concentrés sur l'intégration de l'algorithme à l'application.

Enfin, l'application sera développée de sorte que ce soit une Progressive Web App pour plusieurs raisons :

- Limiter son espace de stockage sur le support numérique ;
- Améliorer le temps de chargement et l'expérience utilisateur ;
- Présenter une plus grande réutilisabilité.

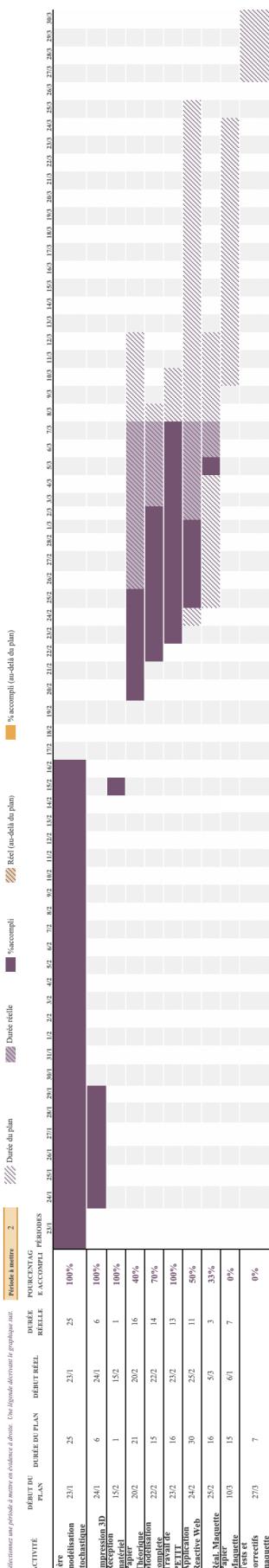
VI. Rédaction du papier de recherche

Afin de rédiger notre papier de recherche, nous utiliserons l'outil LaTeX.

VII. Feuille de route

Nous avons décidé de mettre notre feuille de route sous forme d'un planning Gantt et d'une matrice répartissant les responsabilités parmi les membres composant l'équipe. Celui-ci est disponible en page suivante.

Green'N GOPPE 1827



AR : Alexis Léa	Responsable	Auteur	Consulté	Informé	Terminé
AM : Ammaria	AR	AR/L/M	RZ	RZ/E	
AV : Alice	AR	AR/E	FABLAB	RZ	X
M : Michael	E	E/AM	RZ	RZ/AV	X
E : Eric	L	L/E	RZ	RZ/AV	
RZ : Rafaël Zitoon	E	E/AM/E	RZ	RZ/L/M	
matériel	E	E/FABLAB	RZ		
Réal. Maquette	AR	AR/L/M/E	RZ/E	RZ/AV	
Mat. papier	AV	AR/L/M	RZ/E	RZ	
mat. plastique	AM	AR/L/E	RZ/AV		
correctifs	E	AR/L/M	RZ/E	RZ	
Maquette	AV	AR/L/M	RZ/E	RZ	

VIII. Références bibliographiques

- [1] Couvier D., "Processus stochastiques", <http://www.prism.uvsq.fr/~mogue/cours-modelisation/CoursPS.pdf>
- [2] ENS, "Préparation à l'agrégation", 2008, https://www.ceremade.dauphine.fr/~rivoirar/File.pdf?fbclid=IwAR29hPQdCk5kMVGPtX3LB08ei4GjGKh1AvWEaCb_PvG6JRbqHic29kagSs
- [3] Université Laval, Christian Gagné, "Théorie des files d'attente", 2017, <http://vision.gel.ulaval.ca/~cgagne/enseignement/SETR/H2017/presentations/setr-sem12-filesattente.pdf>
- [4] Alexandre Blondin Massé, Laboratoire d'informatique formelle, Université du Québec, "Files d'attente", 2014, <http://www.lacim.uqam.ca/~blondin/files/8inf802/ete2014/files-attente.pdf>
- [5] Lignan A., "Zolertia RE-Mote platform", <https://github.com/Zolertia/Resources/wiki/RE-Mote>, 2016.
- [6] Zolertia, "Z1 Datasheet", http://zolertia.sourceforge.net/wiki/images/e/e8/Z1_RevC_Datasheet.pdf, 2010.
- [7] Caméras thermiques :
 "Caméra thermique intelligente ThermoCam2 V2X avec V2X | FLIR Systems", Flir.fr, 2018, <https://www.flir.fr/products/thermicam-v2x/>
 "TraficOne | FLIR Systems", Prod.flir.fr, 2018, <https://prod.flir.fr/products/trafione/?model=10-7070>
- [8] Feux tricolores solaires :
 C. Screen, "JEU DE 2 FEUX TRICOLORES DE CHANTIER TEMPO SOLAIRE - avec télécommande -", Ttsys.fr, 2018, <https://www.ttsys.fr/signalisation-des-chantiers/feux-tricolores-de-chantiers/jeu-de-2-feux-tricolores-de-chantier-tempo-solaire.htm>
- [9] Feux avec radar de vitesse :
 "Feu comportemental mobile - Virages.com", Virages.com, 2018, <https://www.virages.com/Signalisation-Lumineuse-de-Chantier/Feu-Comportemental-Mobile>

[10] Brevet d'IBM :

"US Patent # 9,965,951. Cognitive traffic signal control - Patents.com", Patents.com, 2018, <http://patents.com/us-9965951.html>

[11] Renseignements produits électroniques :

<https://www.conrad.fr/p/se037-1-pcs-1616256>

<https://www.gotronic.fr/art-capteur-de-temperature-grove-101020077-23843.html>

<https://www.gotronic.fr/art-lecteur-rfid-grove-125-khz-113020002-19038.html>

Sources pour les PWA :

[12] L. Chatillon, "Progressive Web Apps : tout savoir sur les applis mobiles nouvelle génération", Codeur Mag, 2019, <https://www.codeur.com/blog/progressive-web-apps/>

[13] "Progressive web apps : définition, avantages et faiblesses.", Blog pour le Développement d'Applications, web et jeux mobiles, 2019, <https://fr.yeeply.com/blog/progressive-web-apps/>

[14] "Progressive Web Apps | Web | Google Developers", Google Developers, 2019, <https://developers.google.com/web/progressive-web-apps/>