



ÉCOLE CENTRALE CASABLANCA  
PROJET LEARNING BY DOING  
RAPPORT PLBD 4

---

SMART RAMP

---



*Equipe projet:*

AFFOVEHOUNDE Panel  
AL AYACHI Chaima  
BADOLO Christian Thomas  
BORO Adama

*Encadrant:*  
*BELHBOUB Anouar*

## Table des matières

<b>1 Remerciements</b>	<b>3</b>
<b>2 Résumé</b>	<b>4</b>
<b>3 Abstract</b>	<b>5</b>
<b>4 Introduction</b>	<b>6</b>
<b>5 Développement du projet</b>	<b>7</b>
5.1 Plannification . . . . .	7
5.2 Contexte et problématique . . . . .	9
5.3 Etat de l'art . . . . .	11
5.3.1 Solutions existantes . . . . .	11
5.3.2 Solution choisie . . . . .	15
5.4 Mise en place de la solution . . . . .	16
5.4.1 Application web . . . . .	16
5.4.2 Prototype physique . . . . .	18
5.5 Réalisation et tests . . . . .	21
5.5.1 Prototpe physique . . . . .	21
5.5.2 Application web . . . . .	25
5.6 Résultats obtenus, commentaires et analyses . . . . .	27
<b>6 Conclusion générale</b>	<b>29</b>
<b>7 Références</b>	<b>30</b>
<b>8 Annexes</b>	<b>31</b>
8.1 Codes de la simulation tinkercad . . . . .	31
8.2 Codes pour le prototype . . . . .	34

## Table des figures

1	Charte de projet . . . . .	7
2	Diagramme de Gantt . . . . .	8
3	Diagramme de Gantt . . . . .	8
4	Importance de la ville de Casablanca dans l'économie Marocaine . . . . .	9
5	Population de Casablanca par rapport aux grandes villes africaines . . . . .	10
6	Feux tricolores intelligents . . . . .	11
7	Système de tarification de la congestion . . . . .	12
8	Application Waze . . . . .	12
9	Parking intelligent . . . . .	13
10	Interconnectivité des véhicules autonomes . . . . .	13
11	Compteur de rampe . . . . .	14
12	Avantages et inconvénients des solutions existantes . . . . .	15
13	Simulation sur Tinkercad . . . . .	21
14	Feu inactif . . . . .	21
15	Feu actif . . . . .	22
16	Détection des véhicules . . . . .	22
17	Fonctionnement des feux . . . . .	23
18	Branchements avec l'ESP32, les capteurs et les feux . . . . .	23
19	Support de la route . . . . .	24
20	Installation de la route . . . . .	24
21	Code pour l'interface . . . . .	25
22	Code pour les couleurs . . . . .	26
23	Page d'inscription . . . . .	27
24	Page de connexion . . . . .	27
25	Interface . . . . .	28
26	Code simulation tinkercad 1 . . . . .	31
27	Code simulation tinkercad 2 . . . . .	32
28	Code simulation tinkercad 3 . . . . .	32
29	Code simulation tinkercad 4 . . . . .	33
30	Code simulation tinkercad 5 . . . . .	33
31	Code prototype 1 . . . . .	34
32	Code prototype 2 . . . . .	35
33	Code prototype 3 . . . . .	35
34	Code prototype 4 . . . . .	36
35	Code prototype 5 . . . . .	37

## 1 Remerciements

Nos remerciements vont en premier lieu à l'endroit de l'**Ecole Centrale Casablanca**, pour nous avoir offert l'opportunité d'entreprendre le projet LEARNING BY DOING qui nous a permis d'appliquer nos différents acquis théoriques et expérimentaux, d'acquérir par nous-même de nouvelles connaissances tout en développant notre aptitude à travailler en équipe. De plus, nous remercions l'école pour le choix du thème de l'année «Smart City et développement durable en Afrique» qui nous a fait réfléchir sur un sujet important pour l'avenir de l'Afrique.

Nous tenons également à exprimer notre profonde gratitude à notre tuteur, Monsieur **Anouar Belhboub** et notre coach d'ADPL Madame **Malak El Oudghiri** pour leur disponibilité, leurs précieux conseils et leur soutien indéfectible tout au long de notre projet.

Nous tenons à remercier chaleureusement nos enseignantst et nos différents encadrants dans ce projet, à savoir :

- Monsieur **Khalid DAHI** pour la bonne organisation du déroulement du projet ;
- L'équipe du FABLAB qui n'a pas hésité à nous aider dans la fabrication de notre prototype, en particulier, Monsieur **Abdelaziz DAGHOURI** et Madame **Maha ANNOUKOUBI** ;
- Tout le corps professoral de l'Ecole Centrale Casablanca, pour les savoirs dispensés et les connaissances à nous transmises.

Nous tenons enfin à remercier particulièrement notre collègue **Saad ADLANI** qui a commencé cette aventure avec nous mais qui n'a pas pu rester jusqu'à son terme.

## 2 Résumé

Le Projet Learning By Doing (PLBD) de cette année a pour thème "Smart city et développement durable en Afrique". Ayant à cœur d'impacter le développement en Afrique et compte tenu de la place importante qu'il y tient, notre projet est axé autour du transport et de la mobilité intelligente. Nous nous intéressons plus particulièrement à la congestion routière, en se fixant pour objectif de réduire les embouteillages sur les grandes axes de Casablanca. Pour cela notre projet vise à réguler le flux de véhicules qui entrent sur ces axes par ce qu'on appelle des "rampes". D'où le nom de notre projet "Smart Ramp". Nous avons dans un premier temps travaillé sur le prototype physique que nous avons repensé et amélioré tout au long des dernières semaines. Nous devions détecter les véhicules afin de déterminer le flux sur l'autoroute et ainsi y laisser l'accès ou non aux véhicules sur la rampe à l'aide du feu qui se trouve à la fin de celle ci. En fonction du flux, nous pouvons également indiquer à l'entrée de la rampe, si celle-ci est ouverte ou fermée.

Dans un second temps, nous avons développé une application web afin d'indiquer aux usagers qui souhaitent entrer sur une autoroute les rampes libres d'accès.

Dans l'ensemble, notre projet réduit le nombre de ralentissements des conducteurs pour le trafic fusionné sur l'autoroute. Ceci réduit les perturbations du trafic dues aux ralentissements et aux collisions ce qui aide à maintenir des vitesses moyennes sur les autoroutes plus élevées et, par conséquent, contribue à réduire les temps de trajet.

### 3 Abstract

The theme of this year's Learning By Doing Project (PLBD) is "Smart city and sustainable development in Africa". As a part of our commitment to impacting Africa's development, our project focuses on transport and smart mobility. We are particularly interested in the problem of traffic congestion. Therefore, We've set ourselves the goal of reducing traffic jams on Casablanca's main roads. To achieve this, our project aims to regulate the flow of vehicles entering on these roads by means of socalled "ramps". Hence the name of our project "Smart Ramp". Initially, we worked on the physical prototype, which we redesigned over the last few weeks. We had to detect vehicles in order to determine the flow of traffic on the freeway, and whether or not to allow vehicles on the ramp using the traffic light at the end of the ramp. Depending on the flow, we can also indicate at the ramp entrance whether the ramp is open or closed.

Secondly, we developed a web application in order to indicate to users wishing to enter a freeway which ramps are open.

Overall, our project reduces how often and how much drivers slow down for the merging traffic resulting in fewer disruptions to freeway traffic, including slowdowns and collisions. This helps keep average freeway speeds higher, helping reduce travel times.

## 4 Introduction

Dans le but de développer les compétences de ses élèves ingénieurs de première année et de les aider à mettre en pratique leurs acquis, l'Ecole Centrale Casablanca a mis en place le « Project Learning By Doing » qui s'inscrit dans sa vision d'inventer le monde de demain en formant des ingénieurs responsables et innovants. Le thème choisi pour cette année étant « Smart City Et Développement durable en Afrique», notre groupe de PLBD, « Les TranStarks », a décidé de travailler sur le sous-thème de « Transport et mobilité intelligente » et s'est donné comme tâche de résoudre l'un des problèmes réels liés au transport à Casablanca : la congestion routière. Le secteur du transport représentait 6,5% du PIB du Maroc en 2021. Par ailleurs, la congestion routière coûte 0.6 Litres de plus en carburant par 100 kilomètres parcourus. Cela montre combien il serait bénéfique pour le développement de fluidifier le trafic routier. Nous verrons dans la suite de ce rapport de façon plus détaillée l'impact du transport sur le développement.

Nous tacherons de résumer l'ensemble des étapes par lesquels nous sommes passés pour la réalisation du projet. Nous commencerons d'abord par la gestion du projet par notre équipe ainsi que la planification du projet tout au long de l'année, ensuite nous expliquerons le contexte du projet et la problématique choisie et présenterons également notre solution quant à la problématique de la fluidification du transport routier. Nous parlerons des différents tests réalisés et présenterons les résultats de notre travail. Enfin, ce rapport sera clôturé par un bilan des différents points abordés, les difficultés rencontrées et les enseignements tirés.

## 5 Développement du projet

### 5.1 Plannification

Au début de notre projet LBD, nous avons commencé par des recherches bibliographiques afin d'appréhender la sous-thématique de transport et mobilité intelligente en la confrontant aux réalités du continent Africain. Nos recherches nous ont permis non seulement de comprendre l'importance du secteur de transport dans le développement, mais également d'avoir une idée des problèmes que rencontre ce secteur, particulièrement en Afrique. Nous nous sommes alors lancés dans ce projet mais en commençant d'abord par la planification des étapes de celui-ci.

Dans le cadre de cette planification, nous avons commencé par la réalisation de la charte de notre projet résumant les résultats de notre recherche et présentant ce que nous comptons faire à travers notre projet.

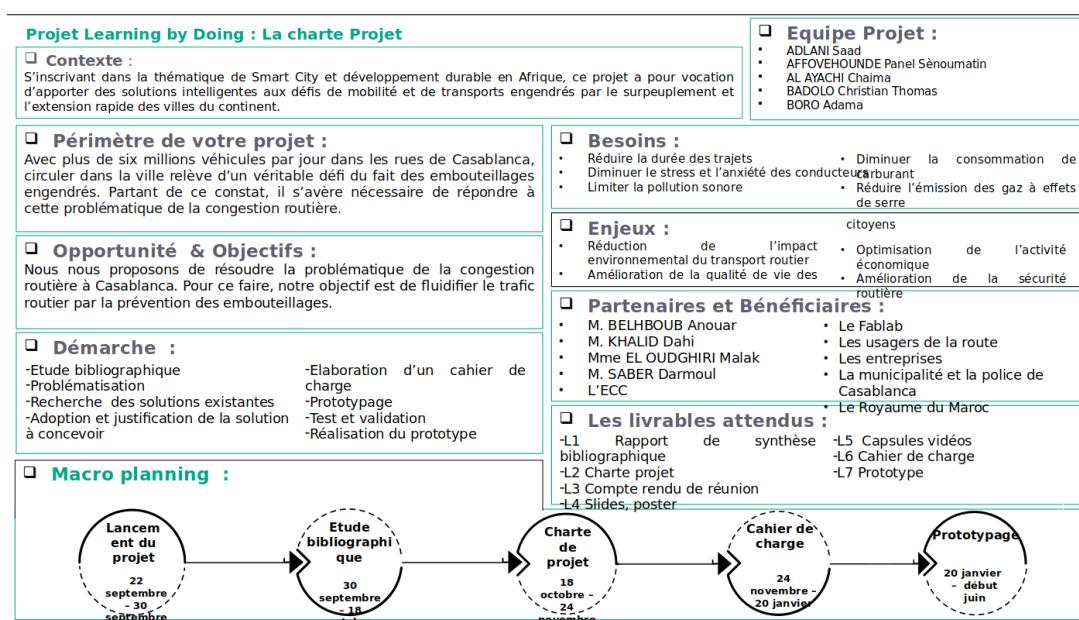


FIGURE 1 – Charte de projet

Pour mener à bien un projet, une bonne plannification est très importante. Les grandes phases par lesquelles est passé notre projet sont : l'étude bibliographique, l'élaboration de la charte de projet, l'élaboration du cahier des charges, et le prototypage. La partie prototypage est subdivisée elle-même en des sous parties : l'élaboration du prototype, les tests et la présentation du prototype lors de la foire qui aura lieu cette année le lundi 26 juin.

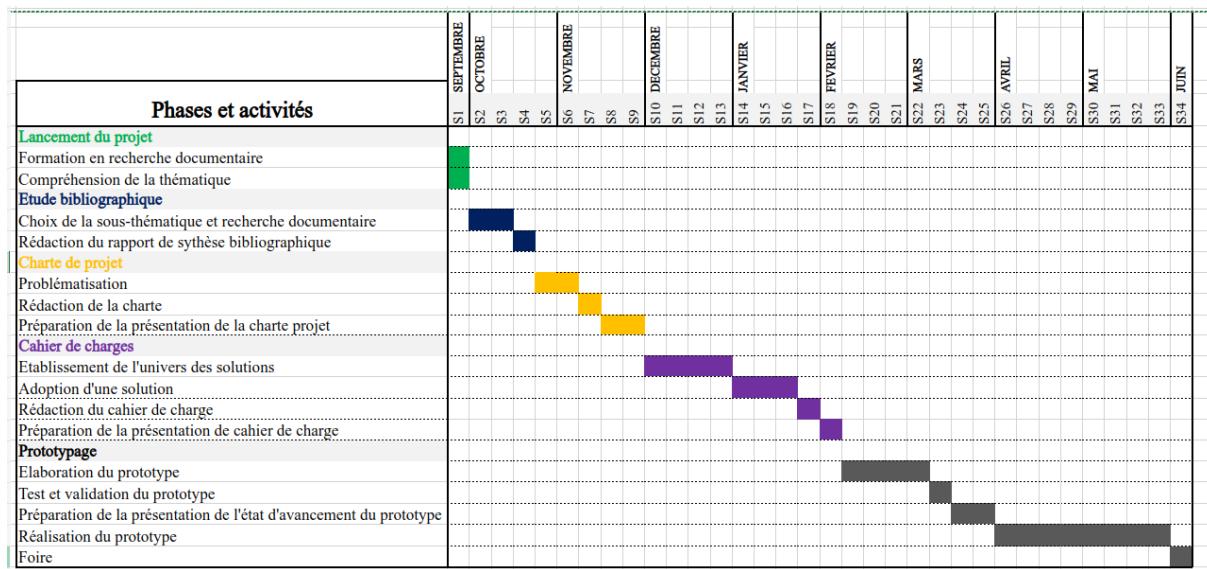


FIGURE 2 – Diagramme de Gantt

Enfin, dans l'objectif d'être efficaces, nous avons départagé les responsabilités entre les membres de l'équipe :

Membres	Rôles
BORO Adama	Chef de projet
AFFOVEHOUNDE Panel	Responsable de la planification
AL AYACHI Chaima	Responsable de la recherche
BADOLO Christian	Responsable de la transcription
ADLANI Saad	Responsable de la communication

FIGURE 3 – Diagramme de Gantt

Il faut cependant notifier que notre collègue ADLANI Saad n'a pas mené ce projet à son terme avec nous pour des raisons personnelles. L'équipe s'est bien adaptée à cette situation afin qu'elle n'impacte pas son succès. Les précieux conseils et l'accompagnement de notre encadrant Monsieur BELHBOUB Anouar et de notre coach d'ADPL Mme EL OUDGHIRI Malak ont été également d'une grande importance pour mener à bien ce projet.

## 5.2 Contexte et problématique

Les pays africains font face à une urbanisation annuelle très rapide et incontrôlée estimée à plus de 3,5% par ONU-habitat. A cela s'ajoute une forte croissance de la population dont la majeure partie est concentrée dans les villes. On estime à 58% le taux d'augmentation de la population africaine entre 2000 et 2017 contre 19% pour le reste du monde[1]. Cela entraîne une extension des villes et le rallongement des distances de déplacement des populations citadines. Les villes du continent se retrouvent ainsi confrontées à de nombreux problèmes de transport et de mobilité. Par ailleurs, ces problèmes entraînent des répercussions sur de nombreux secteurs d'activité économique. En effet, les difficultés de déplacement de la population active empiètent sur la productivité économique des pays et la qualité de vie des citoyens. Les secteurs de l'énergie et de l'environnement ne sont pas épargnés car une grande partie de la consommation énergétique et de la pollution de l'air est due au transport. On note une augmentation du taux d'émission de dioxyde de carbone (CO<sub>2</sub>) dans les embouteillages qui contribue à faire du Maroc le responsable de 2% des émissions mondiales de CO<sub>2</sub> en 2019[2] et une consommation de 3,5 fois de plus en carburant que lors d'un transport fluide. Le poids du transport est aussi considérable dans l'économie des états africains. En 2020, il représentait 15% des recettes fiscales de l'Etat Marocain et 6% du PIB [3] et il crée un rallongement de la durée des trajets équivalent à 0,1% du PIB [4].

Les images suivantes montrent le poids économique de la ville de Casablanca au Maroc et sa position en terme de population parmi les plus grandes villes d'Afrique.

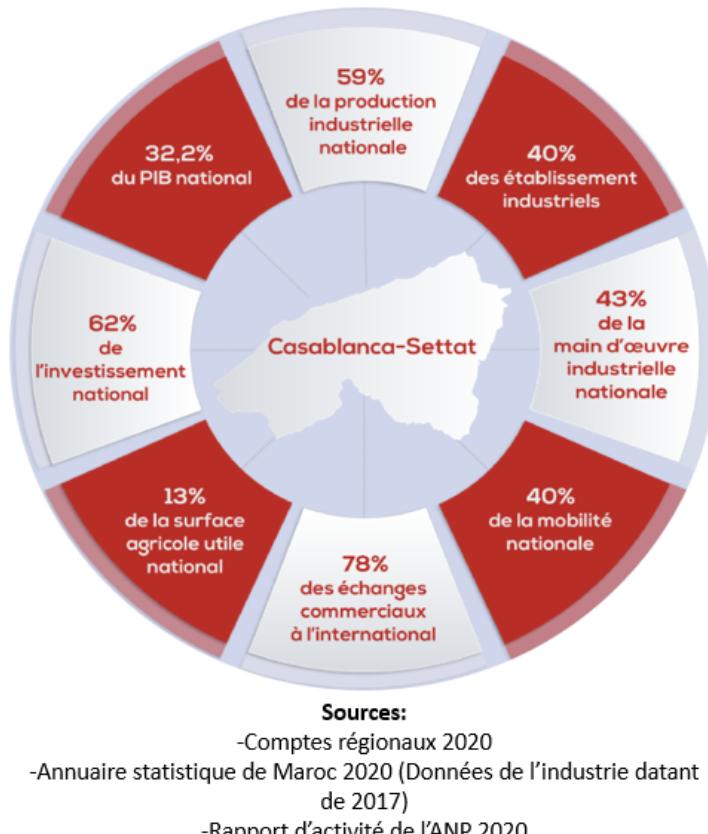


FIGURE 4 – Importance de la ville de Casablanca dans l'économie Marocaine

Rang	Ville	Population (habitants)
1	Le Caire	24 439 785
2	Lagos	22 829 561
3	Gauteng (Johannesburg – Pretoria)	14 717 040
...		
18	Casablanca	4 270 750

FIGURE 5 – Population de Casablanca par rapport aux grandes villes africaines

Il se pose alors la question suivante : Comment fluidifier le trafic routier dans la ville de Casablanca ? Dans le cadre de notre projet nous nous sommes concentré sur les entrées des autoroutes qui sont de grands points d'embouteillage dans les grandes villes.

## 5.3 Etat de l'art

### 5.3.1 Solutions existantes

Avant de nous lancer dans l'élaboration de notre solution, nous avons procédé à une analyse des solutions qui existaient déjà. Plusieurs solutions sont utilisées dans le but de répondre à la problématique de la congestion routière. Nous en avons recensé quelques unes ici :

#### Feux tricolores intelligents

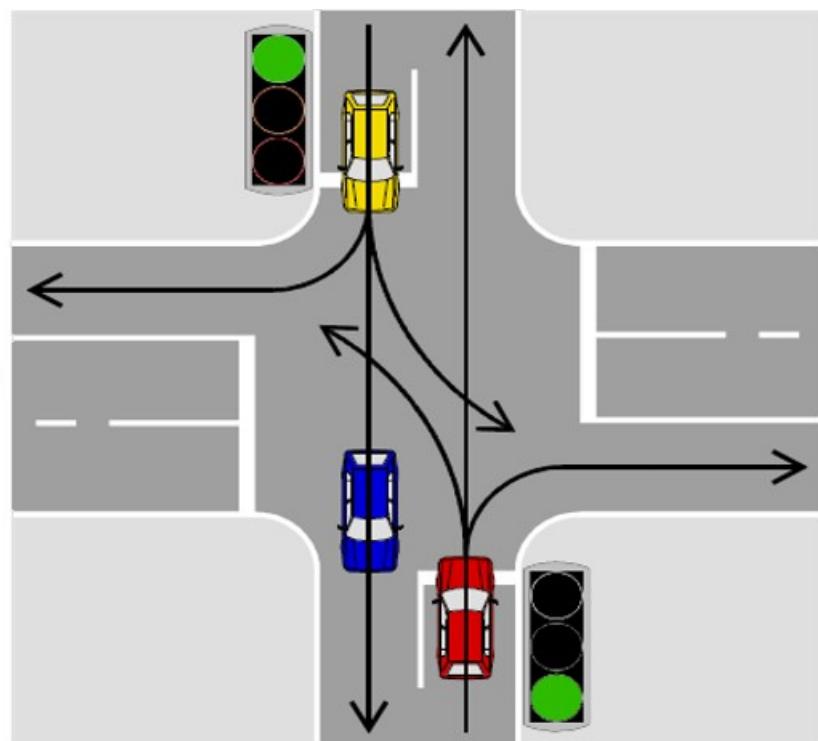


FIGURE 6 – Feux tricolores intelligents

Les feux tricolores intelligents sont une évolution des feux tricolores traditionnels. Ils utilisent des technologies avancées pour améliorer l'efficacité de la circulation routière. En outre, ils permettent de réguler la circulation en utilisant des plans de feux adaptés en fonction des besoins de la circulation en temps réel. Ces feux intelligents fonctionnent en utilisant des capteurs pour détecter les véhicules et les piétons, ainsi que des algorithmes pour analyser les données de circulation en temps réel. Ils peuvent alors ajuster les phases de feux en fonction de la densité de la circulation, de l'heure de la journée et de nombreux autres facteurs pour minimiser les retards et les embouteillages. Les feux tricolores intelligents ont aussi la capacité de communiquer avec d'autres infrastructures de transport comme les panneaux de signalisation et les systèmes de transport en commun. Cela permet notamment de mieux coordonner la circulation routière. Ils peuvent également utiliser des informations sur les véhicules autonomes pour assurer leur respect des règles de circulation et améliorer la sécurité routière.

#### Tarification de la congestion



FIGURE 7 – Système de tarification de la congestion

La tarification de la congestion est la politique consistant à facturer aux conducteurs des frais pour l'utilisation de certaines routes où le risque d'encombrement est élevé. Cette pratique dissuade ainsi de nombreux conducteurs d'emprunter certaines voies [5]. Le paiement se fait de manière automatique. Les tarifs sont déduits de la vitesse des voitures sur l'autoroute que l'on obtient à l'aide de la technologie de perception électronique(capteurs) et varient généralement selon l'heure de la journée. Par ailleurs, ces tarifs peuvent être fixés à l'avance ou définis de façon dynamique, c'est-à-dire qu'ils peuvent augmenter ou diminuer d'une minute à l'autre pour garantir que les voies sont pleinement utilisées sans interruption du flux de trafic.[6]

### Application de suivi du trafic routier : Waze

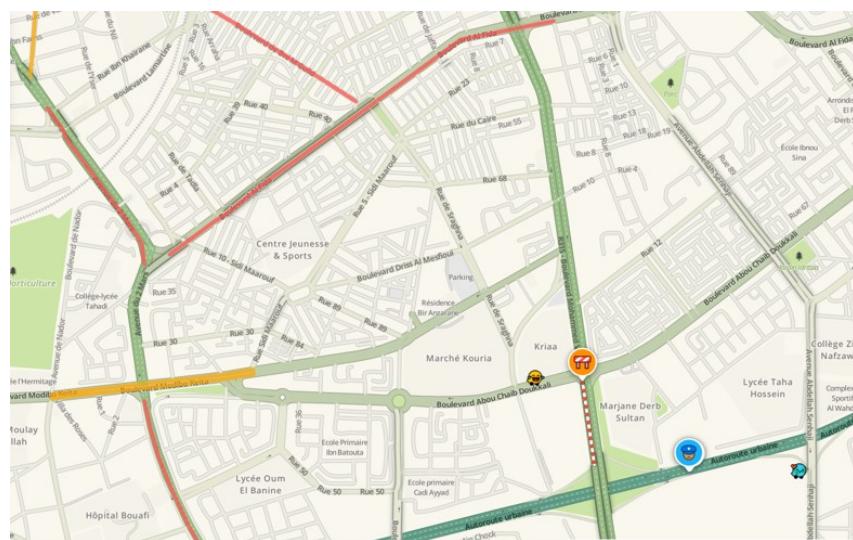


FIGURE 8 – Application Waze

Les applications de suivi du trafic routier ont pour objectifs la fluidification et la surveillance en temps réel du trafic ainsi que la détection et la signalisation des incidents. Waze est une application mobile de trafic et de navigation communautaire née d'un chercheur israélien en 2008 et rachetée par Google en 2013. Le fonctionnement de Waze est basé sur un système de réseau communautaire. Les conducteurs sont connectés les uns

aux autres et contribuent ensemble à améliorer l'expérience de conduite de chacun. Les utilisateurs indiquent les accidents, les embouteillages, les radars de polices, les routes barrées (...) sur Waze. L'application combine ces données et compare les itinéraires possibles afin de cartographier l'évolution du trafic en temps réel. [7]

### Parkings intelligents

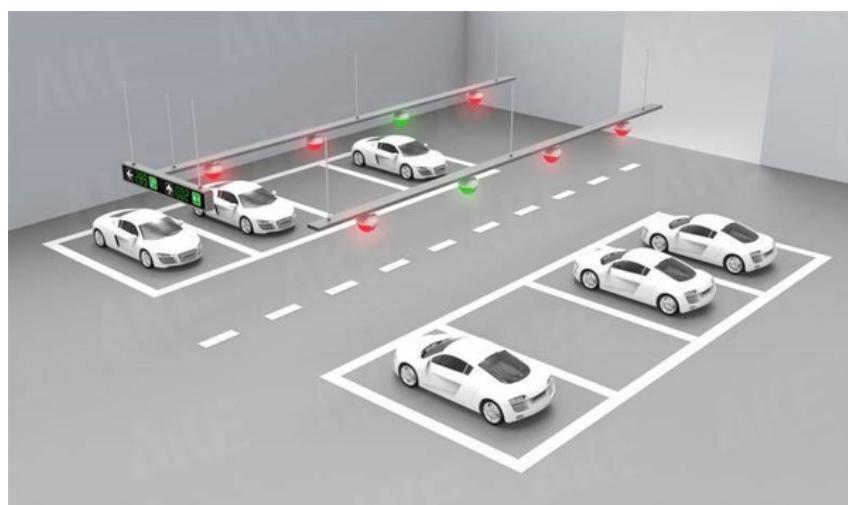


FIGURE 9 – Parking intelligent

Le smart parking est une solution de stationnement qui utilise des technologies avancées pour améliorer l'expérience de stationnement. Ces technologies comprennent des capteurs installés sur les places de stationnement ainsi qu'aux entrées et sorties des parkings. Ils collectent des données sur la disponibilité des places. Ces données sont ensuite transmises à une application mobile pour les conducteurs qui peuvent alors consulter les options de stationnement disponibles, les tarifs et les emplacements des places de parking à proximité. Cela permet aux conducteurs de faire un choix de stationnement facile et rapide. [8]

### Voitures autonomes

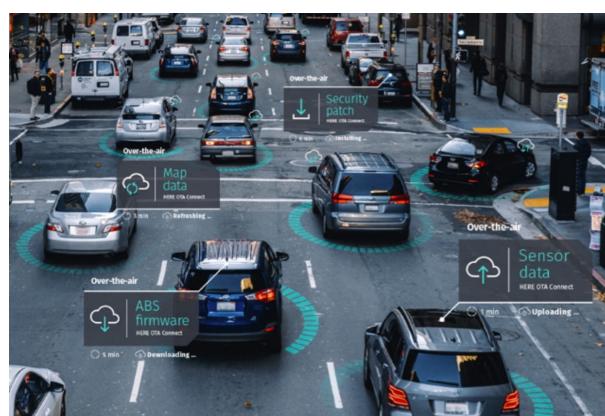


FIGURE 10 – Interconnectivité des véhicules autonomes

Le principe de fonctionnement des voitures autonomes est basé sur la combinaison de

technologies telles que la vision par ordinateur, les capteurs, la navigation GPS, la cartographie et l'apprentissage automatique. Ainsi, ces voitures sont capables de percevoir et comprendre leur environnement, planifier leur parcours et prendre des décisions de conduite [9]. La cartographie de l'environnement se fait grâce aux capteurs tels que les lidars, les radars et les caméras. Les données collectées par les capteurs sont utilisées pour créer une vision 3D afin de planifier des parcours sûrs et efficaces. Les véhicules autonomes peuvent également communiquer avec d'autres véhicules et infrastructures de transport pour partager des informations sur la circulation et améliorer la coordination de la circulation.

### Compteur de rampe

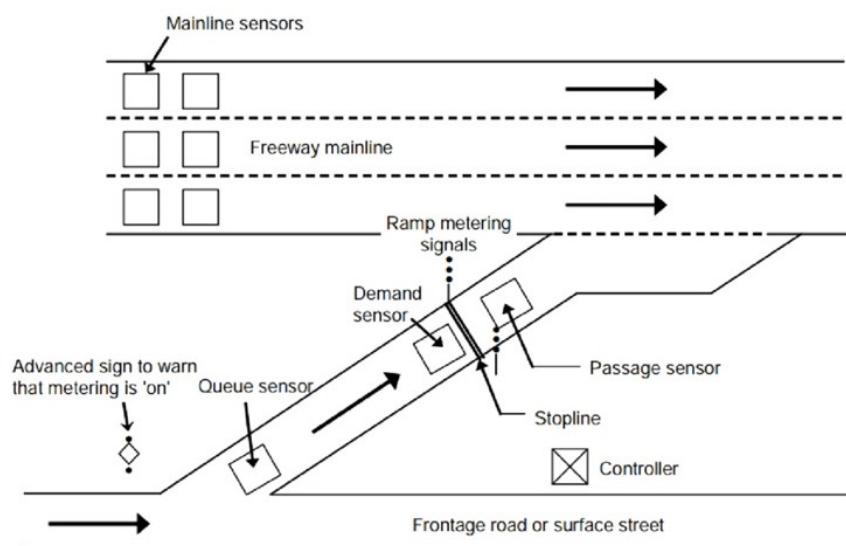


FIGURE 11 – Compteur de rampe

Les compteurs de rampe sont des feux de circulation installés sur les bretelles d'autoroute pour réguler l'entrée des véhicules sur les autoroutes. Ils réduisent la congestion en gérant la quantité de véhicules entrant et en évitant la formation de pelotons qui rendent difficile la fusion sur l'autoroute. Les véhicules qui viennent d'une artère adjacente forment une file d'attente derrière la ligne d'arrêt avant d'être autorisés à entrer sur la voie principale, généralement à un rythme qui est fonction du volume et de la vitesse du trafic sur la voie principale à ce moment-là. [10]

Les avantages et inconvénients des différentes solutions sont regroupées dans le tableau suivant :

Solutions	Avantages	Inconvénients
Feux tricolores intelligents	-Connectivité -Adaptation à la circulation piétonnière	-Non prise en charge des incidents -Coût d'installation et de maintenance élevée
Tarification de la congestion	-Moins de retards pour les véhicules prioritaires -Utilisation accrue du covoiturage	-Impact sur les personnes à faible revenu -Création de nouvelles voies congestionnées
Application de suivi du trafic routier : Waze	-Gratuité et simplicité d'utilisation -Pas nécessairement besoin d'internet	-Problème de véracité des données fournies -Distraction des utilisateurs
Parkings intelligents	-Réduction de la densité de la circulation -Réduction des émissions de CO2 -Amélioration de l'expérience client	Elaboration et maintenance coûteuse
Voitures autonomes	-Optimisation des déplacements -Respect scrupuleux des règles de conduite	-Risque de piratage informatique -Non prise en charge de certaines circonstances naturelles
Compteur de rampe	-Réduction des risques d'accroches et d'accidents entre véhicules -Faible coût de réalisation	-Débordement sur les routes adjacentes -Non respect du feu de la bretelle

FIGURE 12 – Avantages et inconvénients des solutions existantes

### 5.3.2 Solution choisie

La plupart des solutions existantes n'ont qu'un impact occasionnel et assez limité sur la congestion routière. Notamment, en dehors des cas extrêmes où l'un des axes routiers est à faible circulation pendant que l'autre est à circulation denses, les feux intelligents sont réduits à des feux classiques. En outre, d'autres se montrent assez contraignantes pour les usagers à l'image de la tarification de la congestion qui en réalité ne fait que déplacer le problème vers d'autres routes et tend à privilégier les plus nantis des automobilistes. Quant à la voiture autonome, son impact reste très limité sur la régulation du trafic dans une ville telle que Casablanca. En effet, son potentiel ne peut s'exprimer réellement que s'il peut interagir avec les autres voitures qui doivent donc être aussi autonomes alors que la quasi-totalité des véhicules sont non-autonomes dans la capitale économique. Enfin, la mise en œuvre de certaines de ces solutions s'avère assez difficile. C'est l'exemple des parkings qui en plus d'être situés à des endroits stratégiques pour être efficaces se montrent très couteux en raison de tout le dispositif requis pour leur fonctionnement intelligent.

Ainsi, dans le souci de proposer une solution efficace, peu coûteuse, peu contraignante et simple de réalisation, nous avons opté pour le compteur de rampe comme solution pour la suite de nos travaux.

## 5.4 Mise en place de la solution

### 5.4.1 Application web

Dans le but de mener à bien notre projet, nous avons eu à recourir à plusieurs méthodes et technologies numériques, qu'on peut principalement regrouper en des langages de programmation, des logiciels pour le codage ainsi que des logiciels pour assurer et faciliter le fonctionnement des cartes utilisés dans ce projet. Ainsi nous allons commencer par nous intéresser aux langages de programmations utilisés :

#### Python



Le langage C est un langage de programmation très populaire et largement utilisé. Il est souvent considéré comme un langage de bas niveau car il offre un contrôle précis sur le matériel et la mémoire. Le langage C a été développé à l'origine dans les années 1970 pour écrire le système d'exploitation UNIX, ce qui en fait un langage très performant et efficace pour les applications système et embarquées. Avec un ESP32, qui est un microcontrôleur puissant et polyvalent, le langage C peut être utilisé pour développer une large gamme de projets. Nous choisissons de travailler avec le langage C car il permet de contrôler et de communiquer avec les périphériques matériels tels que les capteurs, l'écran LCD, les feux et les interfaces de communication (comme le Wi-Fi et le Bluetooth) de l'ESP32.

#### HTML



Le langage HTML (HyperText Markup Language) est le langage informatique descriptif conçu pour représenter les pages web. Il permet, comme son nom l'indique, d'écrire de l'hypertexte et aussi de pouvoir introduire des fichiers multimédias dans un contenu. Après

les nombreuses modifications qu'il a subi depuis sa création en 1990, nous sommes arrivés à la dernière version d'HTML, HTML5 qui est une version plus aboutie d'HTML. Le langage HTML est le plus connu pour tout ce qui touche à la représentation de site web. Il reste exclusif à cette utilisation puisqu'il a été créé dans ce but-là, c'est d'ailleurs le langage le plus utilisé au monde pour la représentation de site web.

Mondialement reconnu pour la création et la représentation de pages web, et avec des tutoriels disponibles en ligne, ce langage va nous permettre créer et représenter le contenu de l'application web et sa structure.

## CSS



Le langage CSS ("Cascading Style Sheets") est un langage informatique permettant de mettre en forme des fichiers et pages web (HTML ou XML). Depuis son apparition en 1990, ce langage s'est imposé comme une référence mondiale lorsqu'il s'agit de mise en forme de pages web. Pouvant être utilisé avec le langage HTML, la combinaison de ces deux langages offre aux programmeurs plusieurs possibilités ainsi qu'une facilité d'utilisation importante. Ce qui fait de ce langage un incontournable pour la mise en forme de pages web.

Il se présente alors comme indispensable pour la partie numérique de notre solution portant sur l'application web pour sa popularité dans la mise en forme de pages web et le gain de temps qu'il nous procure.

## Javascript



JavaScript est un langage de programmation populaire utilisé principalement pour développer des applications web interactives. Il permet d'ajouter des fonctionnalités dynamiques et d'interagir avec les éléments d'une page web. JavaScript peut être exécuté côté client, directement dans le navigateur, ou côté serveur avec des plates-formes telles que Node.js. Il offre des fonctionnalités telles que la manipulation du DOM, la gestion des événements, l'interaction avec les API, la création de requêtes AJAX et la manipulation des données JSON. JavaScript est un langage interprété et orienté objet, avec une syntaxe similaire à d'autres langages de programmation comme Java ou C++.

Ce langage va nous permettre de rendre l'application web interactive, dynamique et de faciliter la communication avec le serveur sur l'ESP 32.

## Tinkercad



Tinkercad est une plateforme en ligne populaire pour la simulation et la modélisation 3D, principalement utilisée dans le domaine de l'électronique et de la robotique. Elle permet aux utilisateurs de créer des circuits virtuels en utilisant une interface conviviale, en ajoutant des composants électroniques tels que des résistances, des LED, des capteurs, des microcontrôleurs, etc. Les simulations réalisées dans Tinkercad permettent de tester et de valider le fonctionnement des circuits avant de les implémenter dans des projets réels. Grâce à sa facilité d'utilisation et à sa bibliothèque riche en composants, Tinkercad est un outil précieux pour les amateurs, les étudiants et les professionnels, offrant la possibilité de créer, d'apprendre et de partager des modèles et des simulations électroniques.

Nous allons l'utiliser pour effectuer la simulation de notre branchement.

### 5.4.2 Prototype physique

Pour notre prototype physique mettant en exergue notre solution, nous avons utilisé plusieurs éléments :

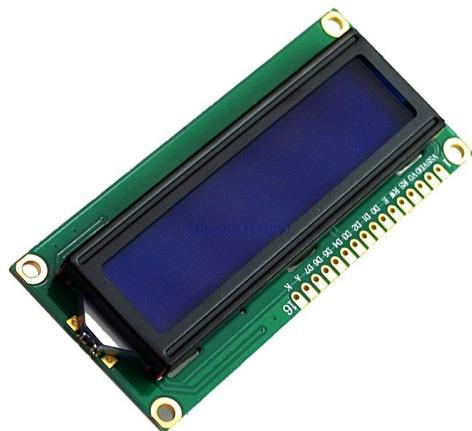
#### Capteurs de proximité NPN



Les capteurs de proximité NPN (Near-Proximity Non-Polarized) sont des dispositifs largement utilisés dans de nombreuses industries pour détecter la présence ou l'absence

d'objets à proximité sans nécessiter un contact physique direct. Ils sont faciles à installer, offrent une sortie compatible avec les systèmes logiques, et sont résistants aux conditions environnementales difficiles. Leur fonctionnement sans contact, leur intégration aisée et leur fiabilité en font des choix populaires pour des applications où la détection précise de la proximité est nécessaire. Nous les utiliserons pour détecter les véhicules sur notre route.

### Ecran LCD



Les écrans LCD offrent une interface simple pour afficher des messages, des données ou des informations en temps réel. Ils sont faciles à utiliser et peuvent être programmés pour afficher différentes combinaisons de caractères et de symboles, offrant ainsi une flexibilité. Nous en utiliserons un ici pour indiquer au passagers l'état de la rampe à laquelle ils comptent accéder.

### Carte ESP32



Les cartes ESP32 sont des cartes de développement basées sur le microcontrôleur ESP32, offrant une puissance de traitement élevée, une connectivité Wi-Fi et Bluetooth, une grande capacité d'E/S et une flexibilité de programmation. Elles sont largement utilisées dans les projets électroniques et l'IoT, offrant des possibilités de développement avancées.

grâce à leur puissance, leur connectivité et leur large gamme de fonctionnalités intégrées. Il nous servira à commander les capteurs et actualiser en temps réel les informations de l'application web.

### Feux bicolores



Il s'agit de feux que nous avons conçu au FABLAB et qui nous serviront à contrôler l'entrée des véhicules sur la route principale.

## 5.5 Réalisation et tests

### 5.5.1 Prototypage physique

#### Simulation

Pour la réalisation de notre prototype, nous avons d'abord commencé par effectuer une simulation des branchements :

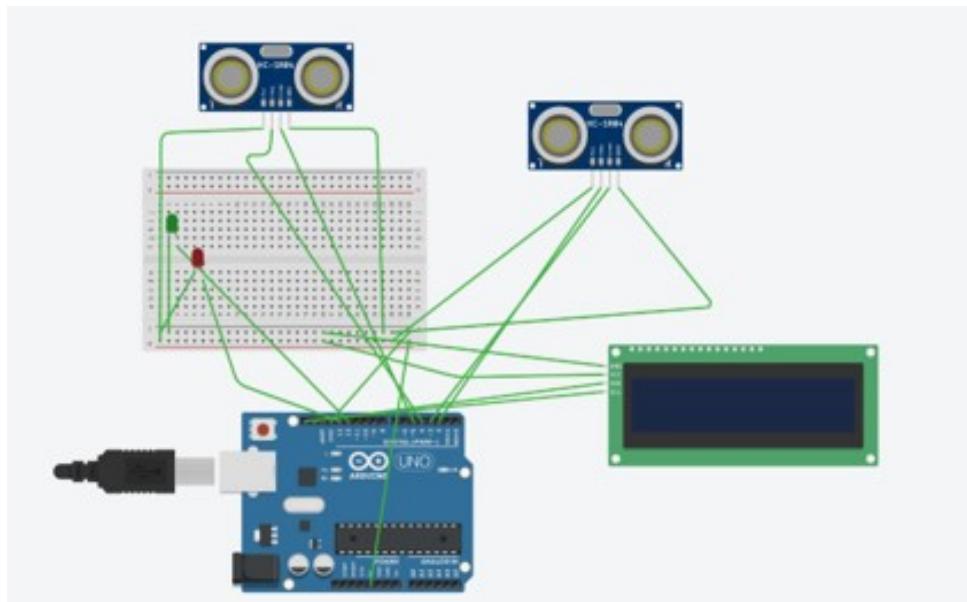


FIGURE 13 – Simulation sur Tinkercad

Pour cette simulation a effectué avec des capteurs ultrasons du fait de l'indisponibilité des capteurs NPN sur Tinkercard.

Les images qui vont suivre montrent les différentes simulations que nous avons effectué avec Tinkercad

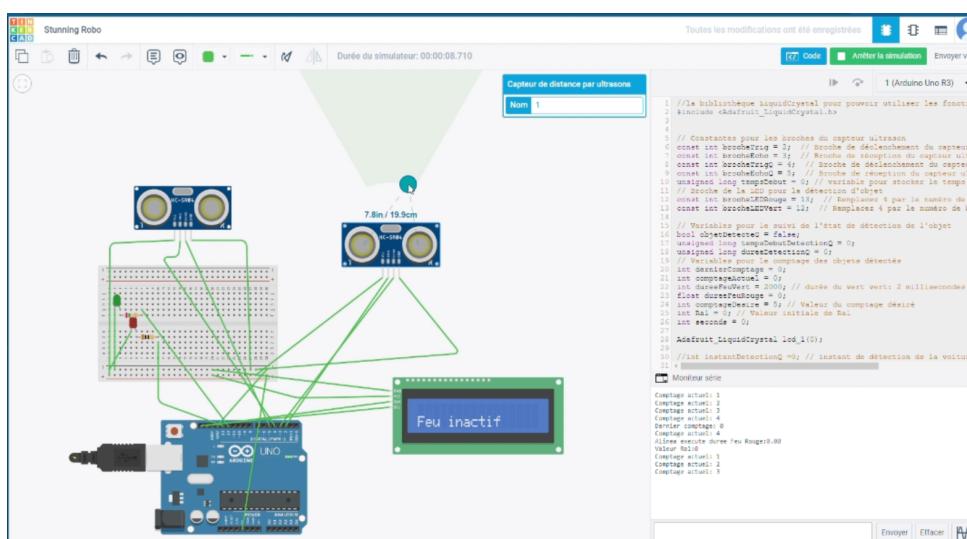


FIGURE 14 – Feu inactif

Dans cette configuration, le nombre de véhicules détectés sur la route principale implique que les feux de la rampe ne sont pas nécessaires.

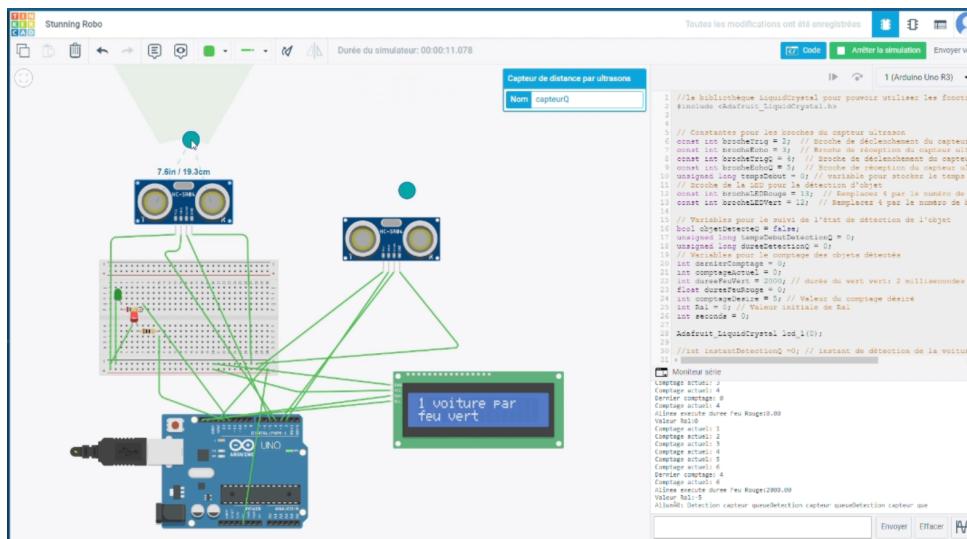


FIGURE 15 – Feu actif

Dans le cas présent, le nombre de véhicules détectés est supérieur à celui qu'on a mis par défaut donc le feu s'active.

## Branchements

Après avoir fait les différentes simulations, nous sommes passés aux branchements. Nous avons effectué différents branchements pour vérifier le fonctionnement de nos composantes.

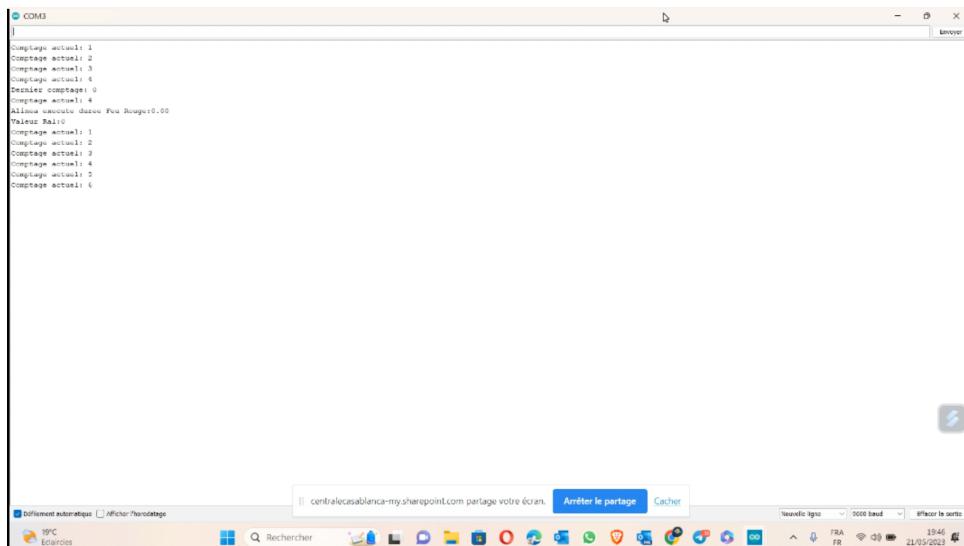


FIGURE 16 – Détection des véhicules

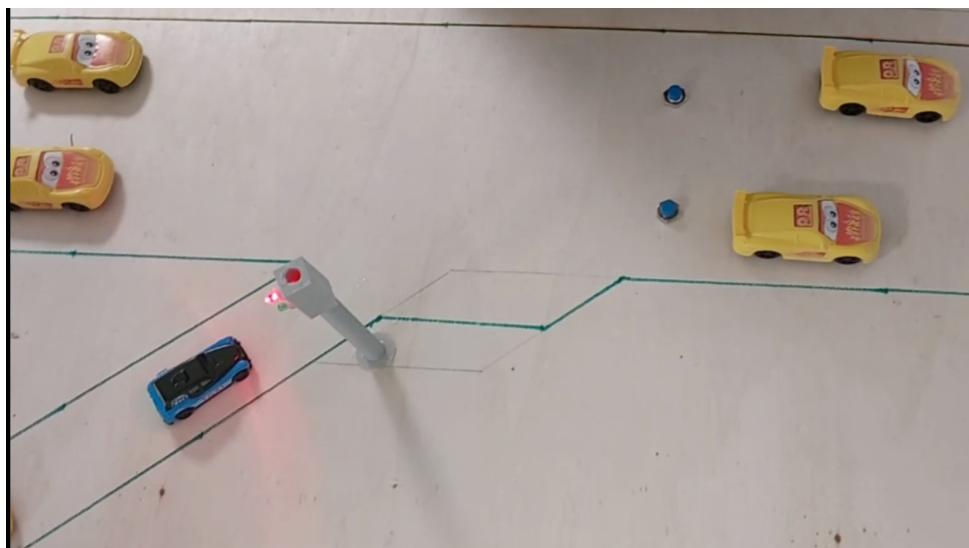


FIGURE 17 – Fonctionnement des feux

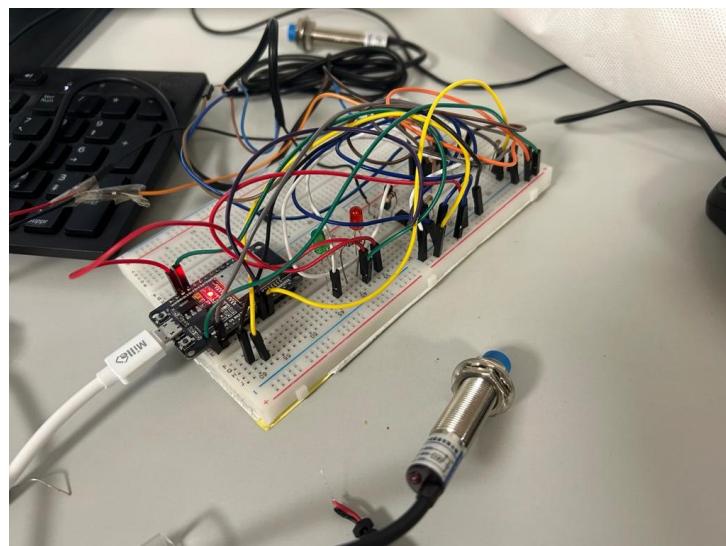


FIGURE 18 – Branchements avec l'ESP32, les capteurs et les feux

Après avoir vérifié que nos feux et nos capteurs fonctionnaient bien, nous avons installé la route et les alentours mais également le support du prototype qui contiendra le circuit.



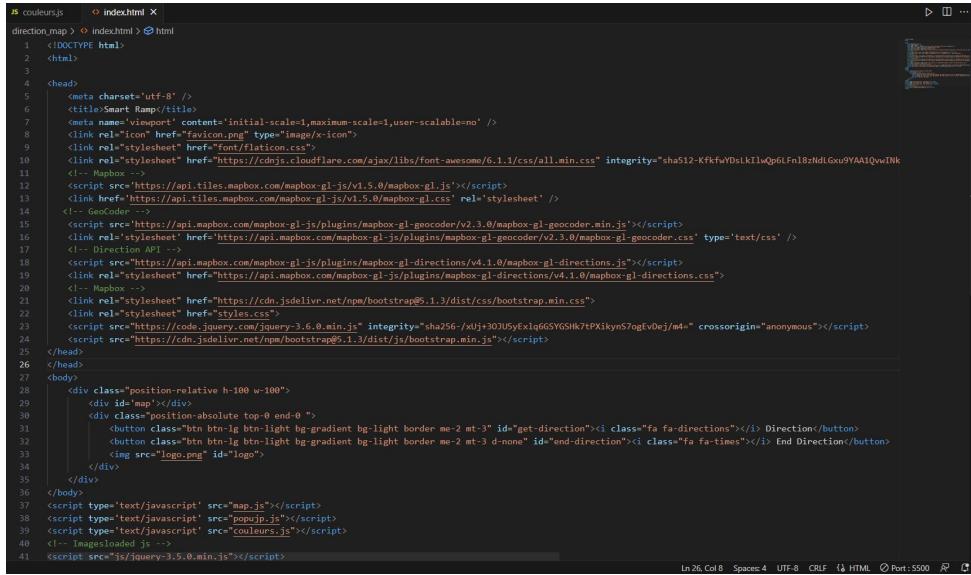
FIGURE 19 – Support de la route



FIGURE 20 – Installation de la route

### 5.5.2 Application web

Pour notre application web, nous avons utilisé plusieurs codes des langages de HTML, CSS, C et Javascript. Voici quelques uns de ces codes.



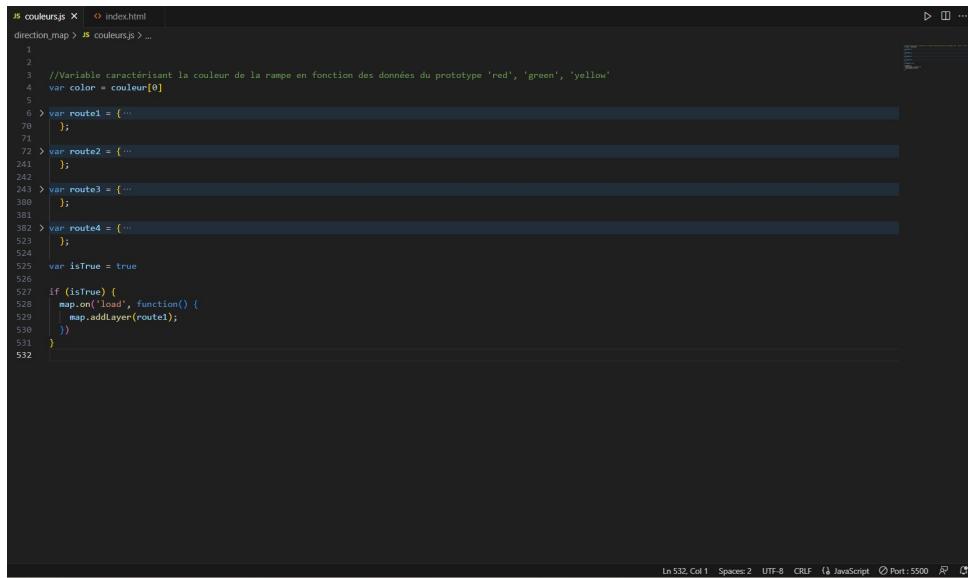
```

4 couleurs.js  0 index.html x
direction_map > 0 index.html > 0 index.html
1   <!DOCTYPE html>
2   <html>
3   </html>
4   <head>
5     <meta charset="utf-8" />
6     <title>Smart Ramp</title>
7     <meta name="viewport" content="initial-scale=1,maximum-scale=1,user-scalable=no" />
8     <link rel="icon" href="favicon.png" type="image/x-icon">
9     <link rel="stylesheet" href="font/fontawesome.css">
10    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css" integrity="sha512-XKfwYDskIluQp6lFl8zIdLGu9YAA1QwInK
11    <!-- Mapbox -->
12    <script src="https://api.tiles.mapbox.com/mapbox-gl-js/v1.5.0/mapbox-gl.js"></script>
13    <link href="https://api.tiles.mapbox.com/mapbox-gl-js/v1.5.0/mapbox-gl.css" rel="stylesheet" />
14    <!-- Bootstrap -->
15    <script src="https://api.mapbox.com/mapbox-gl-js/plugins/mapbox-gl-geocoder/v2.3.0/mapbox-gl-geocoder.min.js"></script>
16    <link rel="stylesheet" href="https://api.mapbox.com/mapbox-gl-js/plugins/mapbox-gl-geocoder/v2.3.0/mapbox-gl-geocoder.css" type="text/css" />
17    <!-- Direction API -->
18    <script src="https://api.mapbox.com/mapbox-gl-js/plugins/mapbox-gl-directions/v4.1.0/mapbox-gl-directions.js"></script>
19    <link rel="stylesheet" href="https://api.mapbox.com/mapbox-gl-js/plugins/mapbox-gl-directions/v4.1.0/mapbox-gl-directions.css">
20    <!-- Mapbox -->
21    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css">
22    <link rel="stylesheet" href="styles.css">
23    <script src="https://code.jquery.com/jquery-3.6.0.min.js" integrity="sha256-/xJj+3OIJ+yXiXqEl9iL/5fC4OI+5V8zGCJYV0J/n" crossorigin="anonymous"></script>
24    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js"></script>
25    <!-- Imagedloaded JS -->
26    <script src="jquery-3.5.0.min.js"></script>
27  </head>
28  <body>
29    <div class="position-relative h-100 w-100">
30      <div id="map"></div>
31      <div class="position-absolute top-0 end-0">
32        <button class="btn btn-lg btn-light bg-gradient bg-light border me-2 mt-3" id="get-direction"><i class="fa fa-directions"></i> Direction</button>
33        <button class="btn btn-lg btn-light bg-gradient bg-light border me-2 mt-3 d-none" id="end-direction"><i class="fa fa-times"></i> End Direction</button>
34        
35      </div>
36    </div>
37  </body>
38  <script type="text/javascript" src="map.js"></script>
39  <script type="text/javascript" src="popup.js"></script>
40  <script type="text/javascript" src="couleurs.js"></script>
41  <!-- Imagedloaded JS -->
42  <script src="jquery-3.5.0.min.js"></script>

```

FIGURE 21 – Code pour l'interface

Ce code nous permet d'afficher dans l'interface, la carte d'une zone. Nous l'avons obtenu avec l'api mapbox.



```

1
2
3 //Variable caractérisant la couleur de la rampe en fonction des données du prototype 'red', 'green', 'yellow'
4 var color = couleur[0]
5
6 > var route1 = {...
79
71
72 > var route2 = {...
241
242
243 > var route3 = {...
388
389
390 > var route4 = {...
523
524
525
526
527 var.isTrue = true
528
529 if (istru...
530   map.on('load', function() {
531     map.addlayer(route1);
532   })
533 })
534

```

Ln 532, Col 1   Spaces: 2   UTF-8   CRLF   JavaScript   Port: 5500  

FIGURE 22 – Code pour les couleurs

Ce code nous permet de mettre des couleurs sur les rampes en fonction du nombre de véhicules qui l'empruntent. Nous l'avons rédigé nous-même.

## 5.6 Résultats obtenus, commentaires et analyses

Pour ce qui est de l'application web, les images suivantes montrent le rendus obtenu.

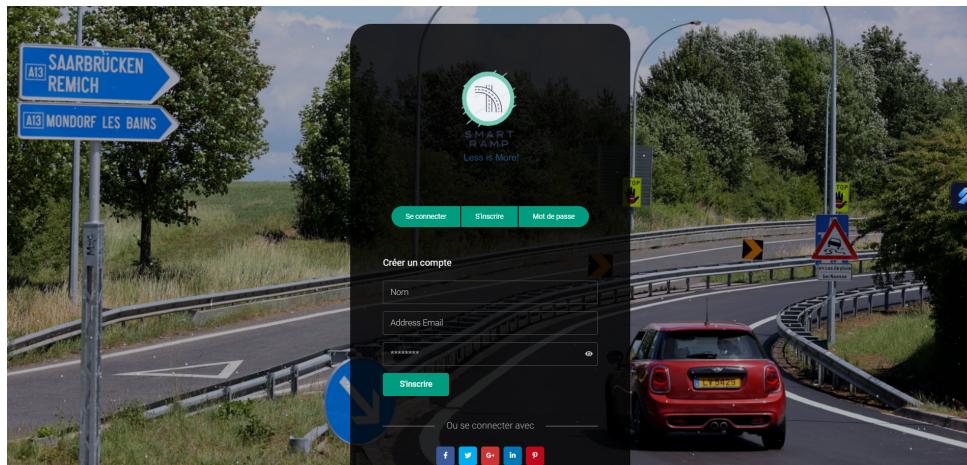


FIGURE 23 – Page d'inscription

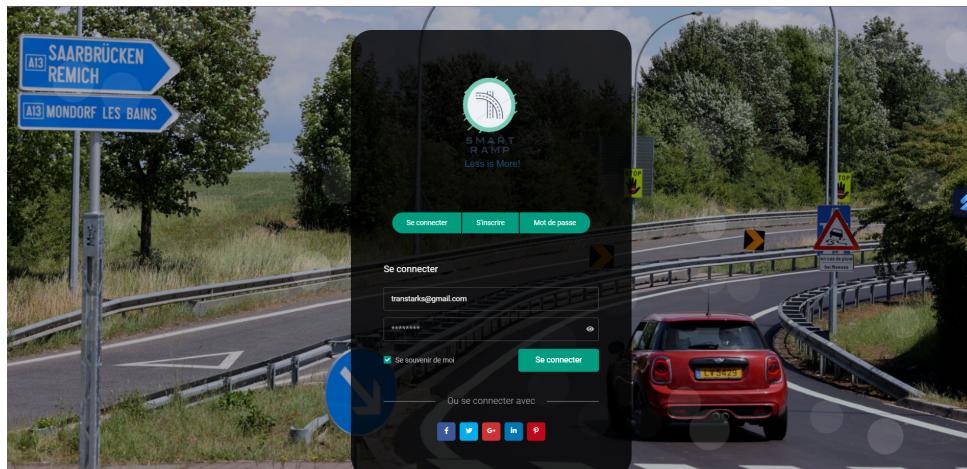


FIGURE 24 – Page de connexion

En ce qui concerne le prototype physique, lors de nos essais, nous avons réussi à déterminer le flux de véhicule sur la route principale. Ensuite en fonction de ce flux, nous activons le feu et nous la faisons alterner de façon à faire passer un véhicule à la fois. Pour ce cas de figure , nous affichons également sur l'écran LCD le message suivant «Un véhicule par feu vert». De plus, en fonction du flux, nous pouvons détecter s'il y a un trop grand nombre de véhicule sur la route principale. Ajouté à celà la détection de la fin de la rampe, nous pouvons déterminer si la rampe est pleine et la notifier sur l'écran LCD( Embouteillage/bretelle fermée). Toutes ces informations sont stockées dans la base de données et sont utilisées par l'application web pour indiquer aux utilisateurs les rampes les plus propices d'accès. La mise en place de notre solution viendra pallier aux défauts des autres solutions tels que la fiabilité des données(comparativement à waze), la prise en charge des incidents (Feux tricolores simples) le coût de l'élaboration .

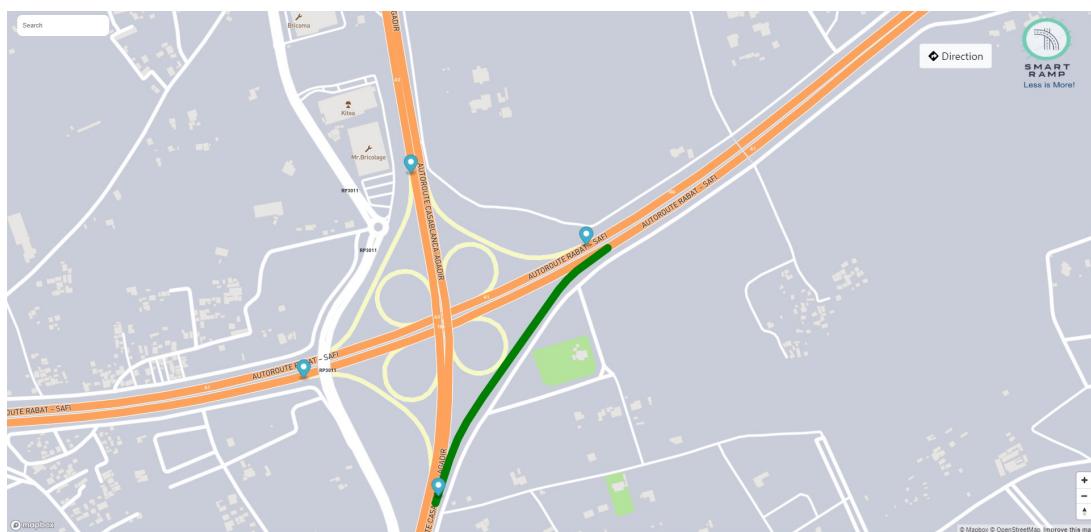


FIGURE 25 – Interface

## 6 Conclusion générale

Depuis quelques années, l'école confie aux élèves de première année un projet ayant une thématique bien définie et ils ont pour charge de développer ce projet par eux-même tout au long de l'année. Le Project Learning By Doing est l'occasion pour les futurs ingénieurs centraliens de travailler en équipe sur un sujet de grande importance et de développer leurs capacités.

Le thème de cette année était "Smart city et développement durable en Afrique" et notre équipe s'est penché sur la problématique de la congestion routière qui a un grand impact sur nos sociétés. Pour répondre à cette problématique, nous proposons une solution qui permettra de réduire la congestion sur les grands axes. Le "Smartramp" utilise les nouvelles technologies et une application web afin d'atteindre cet objectif. Cette solution s'avère efficace et est facile à implémenter. Cependant, il faudra travailler à son adoption par la population et à l'amélioration des algorithmes. Dans le futur on pourra y faire des ajouts comme la suggestion de voies alternatives sur le site et la synchronisation de plusieurs rampes.

Au cours de ce projet, nous avons fait face à plusieurs difficultés comme le départ de notre camarade qui nous a obligé à repenser la distribution des rôles et à avoir des tâches supplémentaires.

## 7 Références

### Références

- [1] Dominique Tabutin and Bruno Schoumaker. La démographie de l'afrique subsaharienne au xxie siècle. bilan des changements de 2000 à 2020, perspectives et défis d'ici 2050. *Population*, 75 :169–295, 2020.
- [2] World Bank. Co2 emissions per capita. <https://data.worldbank.org/indicator/EN.ATM.CO2E.PC?locations=MA>, 2020. Dernière visualisation le 17 juin 2023.
- [3] Médias24. Comment le secteur du transport et de la logistique est impacté par la crise du covid - médias24. <https://medias24.com/2020/10/13/>, 2020. Dernière visualisation le 17 juin 2023.
- [4] United Nations, Economic Commission for Latin America and the Caribbean, Alberto Bull. *Traffic congestion : the problem and how to deal with it*. CEPAL, 2003.
- [5] <https://www.intechopen.com/chapters/53623>. Dernière visualisation le 21 juin 2023.
- [6] <https://ops.fhwa.dot.gov/congestionpricing/>. Dernière visualisation le 21 juin 2023.
- [7] <https://support.google.com/waze/community?hl=fr>. Dernière visualisation le 21 juin 2023.
- [8] <https://www.myflexipark.be/fr/smart-parking>. Dernière visualisation le 21 juin 2023.
- [9] <https://www.aldautomotive.ma/qui-sommes-nous-/news/articleid/4500/voiture-autonome-fonctionnement-et-avantages>. Dernière visualisation le 21 juin 2023.
- [10] <https://wsdot.wa.gov/travel/operations-services/ramp-meters>. Dernière visualisation le 21 juin 2023.

## 8 Annexes

### 8.1 Codes de la simulation tinkercad

```

1 //la bibliothèque LiquidCrystal pour pouvoir utiliser les fonctionnalités de l'écran LCD
2 #include <Adafruit_LiquidCrystal.h>
3
4
5 // Constantes pour les broches du capteur ultrason
6 const int brocheTrig = 2; // Broche de déclenchement du capteur ultrason
7 const int brocheEcho = 3; // Broche de réception du capteur ultrason
8 const int brocheTrigQ = 4; // Broche de déclenchement du capteur ultrason
9 const int brocheEchoQ = 5; // Broche de réception du capteur ultrason
10 unsigned long tempsDebut = 0; // variable pour stocker le temps de début
11 // Broche de la LED pour la détection d'objet
12 const int brocheLEDRouge = 13; // Remplacez 4 par le numéro de broche réel que vous utilisez
13 const int brocheLEDVert = 12; // Remplacez 4 par le numéro de broche réel que vous utilisez
14
15 // Variables pour le suivi de l'état de détection de l'objet
16 bool objetDetecteQ = false;
17 unsigned long tempsDebutDetectionQ = 0;
18 unsigned long dureeDetectionQ = 0;
19 // Variables pour le comptage des objets détectés
20 int dernierComptage = 0;
21 int comptageActuel = 0;
22 int dureeFeuVert = 2000; // durée du vert vert: 2 millisecondes
23 float dureeFeuRouge = 0;
24 int comptageDesire = 5; // Valeur du comptage désiré
25 int Ral = 0; // Valeur initiale de Ral
26 int seconds = 0;
27
28 Adafruit_LiquidCrystal lcd_1(0);
29
30 //int instantDetectionQ =0; // instant de détection de la voiture par le capteur de queue
31 //int dureeDetectionQ =0; // durée de détection de la voiture par le capteur de queue
32 bool estRalCalcule = false; // Variable pour contrôler le calcul de Ral
33 bool etatLed = false; // état du feu bicolore (allumé/éteint)
34
35
36 // Variables pour le suivi de l'état de la détection d'objet
37 bool objetDetecte = false;
38 //bool objetDetecteQ = false;
39
40
41 // Variables pour le suivi du temps
42 unsigned long dernierTempsiteur;
43 unsigned long tempsActuel;
44 unsigned long previousTime = 0;
45 unsigned long interval = 2000; // Intervalle de deux secondes
46
47 // Fonction ALINEA
48 int ALINEA() {
49     Serial.println("Alinea execute ");
50 }
51 Moniteur série

```

FIGURE 26 – Code simulation tinkercad 1

```

47 // Fonction ALINEA
48 int ALINEA() {
49     Serial.print("ALinea execute ");
50     int difference = comptageActuel - dernierComptage;
51     if (comptageActuel >= comptageDesire && !estRalCalcule) {
52         Ral = difference;
53         estRalCalcule = true; // Marque que Ral a été calculé
54     }
55     if (estRalCalcule) {
56         Ral += inc( 7 * (dernierComptage - comptageDesire));
57         dureeFeuRouge = (10000- (Ral*dureeFeuVert))/abs(Ral);
58     }
59     if (dureeFeuRouge<=1000){
60         dureeFeuRouge =0;
61         // éteindre les LED
62         digitalWrite(brocheLEDRouge, LOW);
63         digitalWrite(brocheLEDVert, LOW);
64     }
65     if ( estRalCalcule && dureeFeuRouge ==0){
66         estRalCalcule= false;
67         Ral = 0;
68     }
69 }
70 return dureeFeuRouge;
71 }
72 //Vérifie si la fil d'attente est saturée
73 /*void QueueOverride(){
74     if (distanceQ <= 20 && !objetDetecteQ ){
75         instantDetectionQ =int(millis());
76         objetDetecteQ = true;
77         Serial.print("duree feux rouge Queue nulle:");
78         Serial.println(dureeFeuRouge);
79     }
80     else{
81         if (objetDetecteQ && !distanceQ <= 20){
82             dureeDetectionQ = int(millis()- instantDetectionQ);
83             objetDetecteQ = false;
84         }
85         if (dureeDetectionQ )>= dureeFeuRouge && dureeFeuRouge !=0){
86             dureeFeuRouge = (dureeFeuRouge * 2)/3;
87             Serial.print("duree feux rouge Queue:");
88             Serial.println(dureeFeuRouge);
89         }
90     }
91 }*/
92 //Allumage et extinction de la led
93 void ledOnOff () {
94     if (!etatLed && tempsActuel - tempsDebut >= dureeFeuVert && dureeFeuRouge > 0) {
95         // si la LED est éteinte et le temps d'extinction est écoulé
96         digitalWrite(brocheLEDRouge, HIGH); // allumer la LED rouge
97         digitalWrite(brocheLEDVert, LOW); // éteindre la LED verte
98         etatled = true; // mettre à jour l'état du feu bicolore
99         Serial.print("Allumé: ");
100        tempsDebut = tempsActuel; // stocker le temps de début
101    }
102    else {
103        if (etatLed && tempsActuel - tempsDebut >= dureeFeuRouge && dureeFeuVert > 0) {
104            // si la LED est allumée et le temps d'allumage est écoulé
105            digitalWrite(brocheLEDRouge, LOW); // éteindre la LED
106            digitalWrite(brocheLEDVert, HIGH); // allumer la LED
107            Serial.print("Eteint: ");
108            etatled = false; // mettre à jour l'état de la LED
109            tempsDebut = tempsActuel; // stocker le temps de début
110        }
111    }
112 }
113 void setup() {
114     // Configure les broches du capteur ultrason
115     pinMode(brocheTrig, OUTPUT);
116     pinMode(brocheEcho, INPUT);
117     pinMode(brocheTrigQ, OUTPUT);
118     pinMode(brocheEchoQ, INPUT);
119     // Configure la broche des LED
120     pinMode(brocheLEDRouge, OUTPUT);
121     pinMode(brocheLEDVert, OUTPUT);
122     // message initial du LCD
123     // Initialise le temporisateur
124     dernierTemps = millis();
125     // Initialise la communication série pour afficher les résultats
126     Serial.begin(9600);
127 }
128 */
129 }
130 }
131 */
132 void loop() {
133     // Déclenche une impulsion sur la broche de déclenchement du capteur ultrason
134     digitalWrite(brocheTrig, LOW);
135     delayMicroseconds(2);
136     digitalWrite(brocheTrig, HIGH);
137     delayMicroseconds(10);
138     digitalWrite(brocheTrig, LOW);
139     //gestion de la duree du feu vert
140     //dureeFeuRouge = 1000;
141 }

```

FIGURE 27 – Code simulation tinkercad 2

```

54 //Allumage et extinction de la led
55 void ledOnOff () {
56     if (!etatLed && tempsActuel - tempsDebut >= dureeFeuVert && dureeFeuRouge > 0) {
57         // si la LED est éteinte et le temps d'extinction est écoulé
58         digitalWrite(brocheLEDRouge, HIGH); // allumer la LED rouge
59         digitalWrite(brocheLEDVert, LOW); // éteindre la LED verte
60         etatled = true; // mettre à jour l'état du feu bicolore
61         Serial.print("Allumé: ");
62         tempsDebut = tempsActuel; // stocker le temps de début
63     }
64    else {
65        if (etatLed && tempsActuel - tempsDebut >= dureeFeuRouge && dureeFeuVert > 0) {
66            // si la LED est allumée et le temps d'allumage est écoulé
67            digitalWrite(brocheLEDRouge, LOW); // éteindre la LED
68            digitalWrite(brocheLEDVert, HIGH); // allumer la LED
69            Serial.print("Eteint: ");
70            etatled = false; // mettre à jour l'état de la LED
71            tempsDebut = tempsActuel; // stocker le temps de début
72        }
73    }
74 }
75 void setup() {
76     // Configure les broches du capteur ultrason
77     pinMode(brocheTrig, OUTPUT);
78     pinMode(brocheEcho, INPUT);
79     pinMode(brocheTrigQ, OUTPUT);
80     pinMode(brocheEchoQ, INPUT);
81     // Configure la broche des LED
82     pinMode(brocheLEDRouge, OUTPUT);
83     pinMode(brocheLEDVert, OUTPUT);
84     // message initial du LCD
85     // Initialise le temporisateur
86     dernierTemps = millis();
87     // Initialise la communication série pour afficher les résultats
88     Serial.begin(9600);
89 }
90 */
91 void loop() {
92     // Déclenche une impulsion sur la broche de déclenchement du capteur ultrason
93     digitalWrite(brocheTrig, LOW);
94     delayMicroseconds(2);
95     digitalWrite(brocheTrig, HIGH);
96     delayMicroseconds(10);
97     digitalWrite(brocheTrig, LOW);
98     //gestion de la duree du feu vert
99     //dureeFeuRouge = 1000;
100 }
101 */
102 void Moniteur serie()
103 {
104     Serial.println("Moniteur série");
105 }

```

FIGURE 28 – Code simulation tinkercad 3

```

130  digitalWrite(brocheFeuRouge, LOW);
139  //gestion de la duree du feu vert
140  if(dureeFeuRouge == 0){
141    dureeFeuVert = 0;
142    Ral =0; // on réinitialise Ral à 0.
143    estRalCalculé = false;
144  }
145  else{
146    dureeFeuVert = 2000;
147  }
148
149  // Mesure la durée de l'impulsion sur la broche de réception du capteur ultrason
150  unsigned long duree = pulseIn(brocheEcho, HIGH);
151  // Vérifie si un objet est détecté à la queue
152  if (detecterObjetQ()) {
153    Serial.print("Detection capteur queue");
154    // Si l'objet vient d'être détecté, enregistre le temps de début de détection
155    if (!objetDetecteQ) {
156      objetDetecteQ = true;
157      tempsDebutDetectionQ = millis();
158    }
159  }
160  else {
161    if (!detecterObjetQ()){
163      // Si l'objet n'est pas détecté, réinitialise les variables
164      if (objetDetecteQ) {
165        objetDetecteQ = false;
166        dureeDetectionQ = millis() - tempsDebutDetectionQ;
167
168        // Affiche la durée de détection en millisecondes
169        Serial.print("Duree de detection : ");
170        Serial.print(dureeDetectionQ);
171        Serial.println(" millisecondes");
173      }
174    }
175  }
176  if (dureeFeuRouge < 1000 ){
177    unsigned long currentTime = millis(); // Obtenir le temps actuel
178    if (currentTime - previousTime >= interval){
179      previousTime = currentTime; // Mettre à jour le temps précédent
180
181      lcd_1.begin(16, 2);
182      lcd_1.setCursor(0, 1); // Positionne le curseur à la colonne 0, ligne 1
183      lcd_1.print("Feu inactif");
184    }
185  }
186  if (dureeFeuRouge > 0){

```

FIGURE 29 – Code simulation tinkercad 4

```

223  Serial.print("Dernier comptage: ");
224  Serial.println(dernierComptage);
225  serial.print("Comptage actuel: ");
226  serial.println(comptageActuel);
227  dureeFeuRouge = ALINEA(); // Appelle la fonction ALINEA pour mettre à jour Ral
228  dernierComptage = comptageActuel;
229  /digitalWrite(brocheLEDRougeVert, LOW);
230  serial.print("duree Feu Rouge:");
231  serial.println(dureeFeuRouge);
232  serial.print("Valeur Ral:");
233  serial.println(Ral);
234
235
236  //digitalWrite(brocheLEDRougeVert, HIGH);
237  //digitalWrite(brocheLEDRouge, HIGH);
238
239  //delay(dureeFeuVert);
240  dernierComptage = comptageActuel;
241  comptageActuel = 0;
242
243  // Met à jour le temporisateur
244  dernierTemporisateur = tempActuel;
245  }
246  if (dureeDetectionQ >= dureeFeuRouge && dureeFeuRouge != 0){
247    serial.print("Duree feu rouge non ajustee : ");
248    serial.print(dureeFeuRouge);
249    dureeFeuRouge = dureeFeuRouge*2/3;
250    dureeDetectionQ = 0;
251    serial.print("Duree feu rouge ajustee : ");
252    serial.print(dureeFeuRouge);
253  }
254  }
255  // Fonction pour détecter la présence d'un objet
256  bool detecterObjetQ() {
257    // Envoie une impulsion de déclenchement au capteur ultrason
258    digitalWrite(brocheTrigQ, LOW);
259    delayMicroseconds(2);
260    digitalWrite(brocheTrigQ, HIGH);
261    delayMicroseconds(10);
262    digitalWrite(brocheTrigQ, LOW);
263
264    // Mesure la durée de l'impulsion sur la broche de réception
265    unsigned long dureeQ = pulseIn(brocheEchoQ, HIGH);
266    float distanceQ = dureeQ * 0.0343;
267
268    // Vérifie si un objet est détecté en fonction de la durée mesurée
269    return (distanceQ < 200);
270  }

```

FIGURE 30 – Code simulation tinkercad 5

## 8.2 Codes pour le prototype

```
C:\> Photo rapport > C:\rapport.c
1  #if defined(ESP32)
2  #include <WiFi.h>
3  #include <FirebaseESP32.h>
4  #elif defined(ESP32D6)
5  #include <ESP8266WiFi.h>
6  #include <FirebaseESP8266.h>
7  #endif
8  //la bibliothèque LiquidCrystal pour pouvoir utiliser les fonctionnalités de l'écran LCD
9  #include <Adafruit_LiquidCrystal.h>
10 // start of settings for LCD1602 with I2C
11 #include <LiquidCrystal_I2C.h>
12 // Set the LCD address to 0x27 for a 16 chars and 2 line display
13 LiquidCrystal_I2C lcd(0x27, 16, 2);
14
15 //Provide the token generation process info.
16 #include <addons/tokenhelper.h>
17
18 //Provide the RTDB payload printing info and other helper functions.
19 #include <addons/RTDBHelper.h>
20
21 /* 1. Define the WiFi credentials */
22 #define WIFI_SSID "TranStarks"
23 #define WIFI_PASSWORD "0123456789"
24
25 //For the following credentials, see examples/Authentications/SignInAsUser/EmailPassword/EmailPassword.ino
26
27 /* 2. Define the API Key */
28 #define API_KEY "atK2PtSnpkwEcqQIX4a0ZtQkogPjvGExc9PsYg"
29
30 /* 3. Define the RTDB URL */
31 #define DATABASE_URL "https://rmpesmart-default.firebaseio.com/ //<databaseName>.firebaseio.com or <databaseName>.<region>.firebasedatabase.app
32
33
34 //Define Firebase Data object
35 FirebaseDatabase fbdo;
36
37 FirebaseAuth auth;
38 FirebaseConfig config;
39
40
41 String main="";
42
43 //Int a, b, x, y;
44 // Constantes pour les broches du capteur ultrason
45
46 const int brocheIndSensor = 19; // Broche du capteur de la voie 1
47 const int brocheIndSensor_bis = 17; // Broche du capteur de la voie 2
48 const int brocheIndSensorQ = 18; // Broche de réception du capteur ultrason
```

FIGURE 31 – Code prototype 1

```
C:\> Photos rapport > C rapport.c > ...
51 // Broche de la LED pour la détection d'objet
52 const int brocheLEDOrange = 13;
53 const int brocheLEDVert = 12;
54 const int brocheLEDQueue = 14;
55
56
57 // Variables pour le suivi de l'état de détection de l'objet
58 bool objetDetecteQ = false;
59 unsigned long tempsDebutDetectionQ = 0; // instant à partir duquel le véhicule a été détecté par le capteur de la queue
60 unsigned long dureeDetectionQ = 0; // mesure la durée de détection d'un véhicule par le capteur de queue
61 unsigned long tempsDebutDetection = 0; // instant à partir duquel le véhicule a été détecté par le capteur de la voie principale
62 unsigned long tempsInDetection = 0; // instant à partir duquel le véhicule a été détecté par le capteur de la voie principale
63
64 unsigned long tempsDebutDetection_bis = 0; // instant à partir duquel le véhicule a été détecté par le capteur de la voie principale
65 unsigned long tempsInDetection_bis = 0; // instant à partir duquel le véhicule a été détecté par le capteur de la voie principale
66 unsigned long dureeDetection_bis = 0; // mesure la durée de détection d'un véhicule par le capteur de la voie principale
67 unsigned long dureeDetection_bis = 0; // mesure la durée de détection d'un véhicule par le capteur de la voie principale
68
69 // Variables pour le comptage des objets détectés
70 int Flux_precedent = 0;
71 int Flux_moyen = 0;
72 int comptageActuel_voie1 = 0;
73 int comptageActuel_voie2 = 0;
74 int dureeFeuVert = 2000; // durée du vert vert: 2 millisecondes
75 int debut_execution = 0;
76 int fin_execution = 0;
77 float dureeFeuRouge = 0; // Valeur du comptage désiré
78 int Ral = 0; // Valeur initiale de Ral
79 int seconds = 0;
80
81 Adafruit_LiquidCrystal lcd_1(0);
82
83 //int instantDetectionQ=0; // instant de détection de la voiture par le capteur de queue
84 //int dureeDetectionQ=0; // durée de détection de la voiture par le capteur de queue
85 bool estRalCalcule = false; // Variable pour contrôler le calcul de Ral
86 bool etatled = false; // état du feu bicolore (allumé/éteint)
87
88 bool embouteillage = false; // Variable indiquant s'il y a embouteillage ou pas
89
90 // Variables pour le suivi de l'état de la détection d'objet
91 bool objetDetecte = false;
92 bool objetDetecte_bis = false;
93
94 //bool objetDetecteQ = false;
95 // Variables pour le suivi du temps
96 unsigned long dernierTempsActuel;
97 unsigned long previousTime = 0;
98 unsigned long interval = 2000; // Intervalle de deux secondes
```

FIGURE 32 – Code prototype 2

```
C:\> Photos rapport > C rapport.c > ...
98 unsigned long interval = 2000; // Intervalle de deux secondes
99
100 // Fonction ALINEA
101 int ALINEA() {
102     Serial.print("Alinea execute ");
103     int difference = Flux_moyen - Flux_precedent;
104     if (Flux_moyen >= comptageDesire && !estRalCalcule) {
105         Ral = difference;
106         estRalCalcule = true; // Marque que Ral a été calculé
107     }
108     if (estRalCalcule) {
109         Ral += int(7 * (flux_precedent - comptageDesire));
110         dureeFeuRouge = (10000 - (Ral*dureeFeuVert))/abs(Ral);
111     }
112     if (dureeFeuRouge <= 1000){
113         dureeFeuRouge = 0;
114         // éteindre les LED
115         digitalWrite(brocheLEDOrange, LOW);
116         digitalWrite(brocheLEDVert, LOW);
117         digitalWrite(brocheLEDQueue, LOW);
118
119     }
120     if (estRalCalcule && dureeFeuRouge == 0){
121         estRalCalcule = false;
122         Ral = 0;
123     }
124     return dureeFeuRouge;
125 }
126
127 //Allumage et extinction de la led
128 void ledOnOff () {
129     if (!etatled && tempsActuel - tempsDebut >= dureeFeuVert && dureeFeuRouge > 0) {
130         // si la LED est éteinte et le temps d'extinction est écoulé
131         digitalWrite(brocheLEDOrange, HIGH); // allumer la LED rouge
132         digitalWrite(brocheLEDVert, LOW); // éteindre la LED verte
133         digitalWrite(brocheLEDQueue, HIGH);
134         etatled = true; // mettre à jour l'état du feux bicolore
135         Serial.print("Allumé: ");
136         tempsDebut = tempsActuel; // stocker le temps de début
137     }
138     else {
139         if (etatled && tempsActuel - tempsDebut >= dureeFeuRouge && dureeFeuVert > 0) {
140             // si la LED est allumée et le temps d'allumage est écoulé
141             digitalWrite(brocheLEDOrange, LOW); // éteindre la LED
142             digitalWrite(brocheLEDVert, HIGH); // allumer la LED
143             digitalWrite(brocheLEDQueue, HIGH);
144             Serial.print("Eteint: ");
145             etatled = false; // mettre à jour l'état de la LED
146         }
147     }
148 }
```

FIGURE 33 – Code prototype 3

```
C:\> Photos rapport > C:\rapport.c > ...
146     etatLed = false; // mettre à jour l'état de la LED
147     tempsDebut = tempsActuel; // stocker le temps de début
148 }
149 }
150 }
151
152
153
154 void setup()
155 {
156     debut_execution = millis();
157
158
159     lcd.begin();
160     lcd.backlight();
161     // Configure les broches du capteur ultrason
162     // Configure la broche des LED
163     pinMode(brocheLEDRouge, OUTPUT);
164     pinMode(brocheLEDVert, OUTPUT);
165     pinMode(brocheLedQueue, OUTPUT);
166     // message initial du LCD
167
168     // Initialise le temporisateur
169     dernierTemps = millis();
170     // Initialise la communication série pour afficher les résultats
171     //Serial.begin(9600);
172     Serial.begin(115200);
173     delay(2000);
174     WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
175     Serial.print("Connecting to Wi-Fi");
176     while (WiFi.status() != WL_CONNECTED)
177     {
178         Serial.print(".");
179         delay(300);
180     }
181     Serial.println();
182     Serial.print("Connected with IP: ");
183     Serial.println(WiFi.localIP());
184     Serial.println();
185
186     Serial.printf("Firebase Client v%s\\n\\n", FIREBASE_CLIENT_VERSION);
187
188     //Assign the api key (required)
189     config.api_key = API_KEY;
190
191     config.database_url = DATABASE_URL;
192
193
```

FIGURE 34 – Code prototype 4

```
C:\> Photos rapport > C rapport.c > ...
193
194
195 //Please make sure the device free Heap is not lower than 80 k for ESP32 and 10 k for ESP8266,
196 //otherwise the SSL connection will fail.
197 ///////////////////////////////////////////////////////////////////
198
199
200 Firebase.begin(DATABASE_URL, API_KEY);
201
202 //Comment or pass false value when WiFi reconnection will control by your code or third party library
203 // Firebase.reconnectWiFi(true);
204
205 Firebase.setDoubleDigits(5);
206
207 }
208
209 void loop()
210 {
211
212 //gestion de la duree du feu vert
213 if (dureeFeuRouge == 0){
214     dureeFeuVert = 0;
215     Ral =0; // on réinitialise Ral à 0.
216     estRalCalcule = false;
217 }
218 else{
219     dureeFeuVert = 2000;
220 }
221
222 // Mesure la durée de l'impulsion sur la broche de réception du capteur ultrason
223 // Vérifie si un objet est détecté à la queue
224 if (digitalRead(brocheIndSensorQ) == LOW) {Serial.print("Detection capteur queue");
225 // Si l'objet vient d'être détecté, enregistre le temps de début de détection
226 if (!objetDetecteQ) {
227     objetDetecteQ = true;
228     tempsDebutDetectionQ = millis();
229 }
230 else {
231     if (digitalRead(brocheIndSensorQ) == HIGH){
232         // Si l'objet n'est pas détecté, réinitialise les variables
233         if (objetDetecteQ) {
234             objetDetecteQ = false;
235 }
```

FIGURE 35 – Code prototype 5