

Livrable de projet D

Conceptualisation

GNG 1503

Équipe Delta

Ghantous Alessio

Halimi Anwar

Kitio Tegnyem Leonel.

Mohamed Houssein, Mahad

Mohamed Omar

Boisvert Simon

Théberge Valérie

9 février 2025

Résumé

Ce livrable présente l'ensemble des concepts préliminaires élaborés par l'équipe Delta pour résoudre le problème posé dans la phase d'idéation du projet de conception. Ces concepts sont basés sur l'étalonnage et la liste de critères de conception établie. L'équipe Delta a travaillé d'arrache-pied pour développer ces idées initiales, en tenant compte des différents aspects et exigences du projet.

Table de matière:

Résumé	2
Table de matière:	3
1. Introduction:.....	5
2. Limite de concept:.....	5
3. Concept suggérer pour chaque sous-système:	6
3.1. Sous-système 1 (Capteur) :	6
3.1.1. Anwar :.....	6
3.1.2. Omar :.....	6
3.1.3. Simon :.....	6
3.1.4. Mahad :.....	6
3.1.5. Alessio :	7
3.1.6. Leonel :	7
3.1.7. Valérie :.....	8
3.2. Sous-système 2 (Logiciel) :	8
3.2.1. Anwar :	8
3.2.2. Omar :	8
3.2.3. Simon :	8
3.2.4. Mahad :.....	9
3.2.5. Alessio :.....	9
3.2.6. Leonel :.....	9
3.2.7. Valérie :	10
3.3. Sous-système 3 (Affichage) :	10
3.3.1. Anwar :	10
3.3.2. Omar :	10
3.3.3. Simon :	11
3.3.4. Mahad :.....	11
3.3.5. Alessio :.....	11

3.3.6. Leonel :	11
3.3.7. Valérie :	11
4. Esquisses:	12
4.1. Sous-système 1 (Capteur) :	12
4.1.1. Puce RFID :	12
4.1.2. Laser :	12
4.2. Sous-système 2 (Logiciel) :	13
4.3. Sous-système 3 (Affichage) :	14
5. Matrice décisionnelle :	16
6. Avantage et inconvénient du Système de Détection par Émetteur Laser :	16
6.1 Avantages	16
6.3 Inconvénients	17
7. Référence:	18
8. Trello :	18

1. Introduction:

Dans ce livrable, le but est de trouver un ensemble de concepts préliminaires des trois sous-systèmes qui composeront le système souhaité que nous envisageons de concevoir pour le client en se basant sur l'étalonnage technique (livrable C). Ces concepts seront analysés et évalués pour déterminer les avantages et les inconvénients de chaque concept par rapport à nos critères de sélection prédéfinis. Après cette évaluation, nous convergerons vers une solution unique qui répondra au mieux à nos objectifs et exigences techniques.

2. Limite de concept:

On a définis nos sous-systèmes comme étant le ou les capteurs, le côté logiciel et l'affichage du résultats. Il est nécessaire que le sous-système de soit minimalement précis avec les données recueillis (+/- 1 sec maximum), en plus de donner des données fiables, donc gardant un minimum d'exactitude. Reste plus que ces mêmes composantes soient en dessous de 100\$ (idéalement moins pour laisser assez de lousse pour d'autre dépense comme le matériel, etc.) et on a les critères d'un sous-système interchangeable au niveau préliminaire.

Pour en décrire le sous système capteur, il est primordial que cela est assez automatique de façon qu'il y aura le moins d'incertitudes possibles, c'est-à-dire, lorsque les voitures franchissent la ligne en même temps il ne devrait avoir aucun problème d'enregistrement. De plus, ceci devrait être capable de connecte au logiciel Arduino Uno afin que les résultats de la course puissent être traité par le logiciel et affiché clairement avec notre système de choix.

Le sous-système de logiciel doit pouvoir interagir avec le Arduino Uno (ce fut une suggestion des plus convergeant au sein des membres de l'équipe). De ce fait, il devra logiquement pouvoir accéder à composante auxiliaire de celle-ci (tel que le langage C+). Le langage du coding importe peu tant qu'il accomplit la fonction qui lui est assigné, communiquant les bonnes données à la bonne place au sous-système suivant (l'affichage).

Le sous-système d'affichage devrait utiliser les résultats recueillis via l'Arduino Uno pour ensuite convertir cela en format propre ou nous pouvons clairement voir toutes les infos importantes de la course, que sa soit les temps des voitures, leur parcours et progrès dans la course et même le nombre de tours effectués. L'idée générale serait alors d'utiliser une sorte d'écran pour démontrer les détails mentionnés en forme de tableaux, graphiques ou classements avec une connexion fiable à l'Arduino et des mises à jours rapides pour assurer une efficacité et qualité haute.

3. Concept suggérer pour chaque sous-système:

3.1. Sous-système 1 (Capteur) :

3.1.1. Anwar :

Système de Chronométrage Laser à Détection Multi-Angulaire: Le système repose sur l'utilisation de **quatre lasers ou plus**, placés stratégiquement autour de la **ligne d'arrivée** et dans différents angles afin de couvrir **toute la zone adjacente**. Un **laser principal** est positionné directement sur la ligne d'arrivée pour détecter le franchissement, tandis que les autres lasers sont placés sous des **angles étudiés** afin d'analyser précisément la **position et l'orientation des véhicules** au moment du passage.

3.1.2. Omar :

Système de surveillance caméra : Une ligne de 2 mètres constitué d'un ruban papier d'une couleur assez distincte de l'arrière-plan pour l'effet contrastuel pour la caméra. Tout changement d'une section du ruban de papier (d'une largeur minimale) déclenche la prise d'une photo qui serait fourni à but d'entrée pour le programme. Chaque voiture aura un ruban de papier d'une couleur différente attaché à l'avant de chaque'un des véhicules à but d'identification. La photo déclenche un algorithme du programme à plusieurs condition (else if et for) automatisant le processus d'affichage du temps de tours, accompagné de l'identité du participant accomplissant le temps.

3.1.3. Simon :

Système de laser de distance : En placent des lasers de capteur de distance a la ligne de départ/arriver. Ils vont être capable de déterminer la distance entre le point initiale du laser et la voiture qui coupe le laser va donner un message a la partie logicielle du produit. Alors avec ces distances que les laser captent, on peut assigner des intervalles sur la largeur de la piste a la ligne d'arriver pour chaque différente voiture. Comme posséder une voiture dans l'intervalle de 0 à 50 cm, puis 50 à 100 cm, 100 à 150 cm et puis 150 à 200 cm. En plus, ce capteur serait utilisé avec une boîte d'Arduino pour être capable d'utilise un logiciel que je connais déjà comment utiliser.

3.1.4. Mahad :

Sous-système au Module RFID et capteurs infrarouge ou lumière: nous utilisons des modules RFID pour identifier chaque voiture et des capteurs infrarouges ou de lumière pour détecter leur passage à la ligne de départ/arrivée. Les lecteurs RFID sont placés de manière stratégique sur la piste pour détecter les tags RFID fixés sur chaque voiture. Lorsque les voitures passent devant les lecteurs RFID, le microprocesseur Arduino reçoit un signal indiquant quelle voiture a franchi la ligne. Simultanément, les capteurs infrarouges ou de lumière aident à confirmer le passage des voitures en enregistrant un changement dans les niveaux de lumière lorsque les voitures passent. Cela garantit que chaque tour est correctement comptabilisé et que le temps de passage est enregistré avec précision.

3.1.5. Alessio :

Sous-système de Laser Diode + Camera : Les lasers diodes au ligne d'arrivée seront capable de capter et enregistrer les voitures lorsqu'ils franchissent la ligne. Ils seront orientés de façon que chaque voiture pourra être détecter sans se faire bloquer. Le camera serve comme assistant pour aider a identifier lesquels voitures son lesquels et pour avoir l'option de "replay" afin d'analyser et vérifier les résultats de la course.

3.1.6. Leonel :

Système 4EML à 4 émetteurs-récepteurs laser

Le système 4EML est un système de détection basé sur des faisceaux laser (émetteur et récepteur laser Arduino), un dispositif installé sur la toiture des véhicules et une interface de traitement des données avec Arduino.

Le système repose sur trois sous-systèmes principaux :

- **Un sous-système de détection de présence (des véhicules,)** composé de deux portiques parallèles, support des **quatre émetteurs et récepteurs laser, placés** de part et d'autre de la piste.

En effet, a chaque passage d'un véhicule sur la ligne d'arriver, un faisceau laser spécifique a ce véhicule sera couper, et l'Arduino interprètera cette information comme une présence.

- **Un sous-système de marquage de présence**, il s'agit d'un objet transparent montée sur la toiture de chaque voiture, et ayant une partie précise opaque (non transparente), unique pour chaque véhicule, qui servira de signature, car cette partie sera à hauteur de l'un des faisceaux laser et pourra couper un unique faisceau laser, nous permettant ainsi de marquer sa présence
- **Détection** : Lorsqu'un véhicule traverse la ligne d'arriver, il coupe un faisceau laser spécifique, permettant d'identifier le véhicule et noter son passage.
- **Sous-système de calibrage :**

Pour calibrer les positions des émetteurs-récepteurs laser et des marqueurs de présence, voici une proposition :

1. **Alignement des portiques** : Utilise un niveau laser pour aligner les portiques parallèlement à la piste.
2. **Réglage des hauteurs** : Ajuste la hauteur des émetteurs-récepteurs pour qu'ils correspondent aux marqueurs de présence sur les véhicules.
3. **Test de détection** : Fais passer un véhicule test équipé du marqueur de présence pour vérifier que chaque faisceau est correctement coupé et détecté.

3.1.7. Valérie :

Système de puces RFID: Les puces RFID permettent d'identifier chaque voiture de façon unique avec des ondes radio. Chaque véhicule aura une puce RFID unique qui sera détectée par des capteurs (antennes par exemple) placés au début et à la fin de la course. Un microcontrôleur Arduino peut se connecter au lecteur RFID pour traiter les données en temps réel au besoin. Ce sous-système peut fonctionner de manière autonome et peut facilement être intégré à divers types de capteurs pour assurer un chronométrage optimal et une flexibilité dans la gestion des données.

3.2. Sous-système 2 (Logiciel) :

3.2.1. Anwar :

Mon concept pour le sous-système logiciel repose sur l'utilisation d'un Arduino Uno pour traiter les signaux binaires envoyés par les lasers de détection. L'application Arduino IDE sera utilisée pour programmer en langage C, permettant d'exploiter un tableau dynamique afin de stocker et organiser les données recueillies. Chaque voiture sera représentée sous forme de matrice, avec une première ligne indiquant le nombre de tours effectués et une seconde ligne enregistrant les temps de passage. Une file d'attente structurera ces données pour un affichage et un traitement efficace. La gestion du chronométrage se fera à l'aide de la bibliothèque `#include <timer.h>`, permettant de mesurer précisément le temps de course et de déterminer les écarts entre les participants.

3.2.2. Omar :

Mon sous-système logiciel est constitué de quelques simples fonctions "custom", permettant l'analyse et l'interprétation de données visuelles à partir de l'élément déclencheur (détection d'un objet de couleur vive traversant le ruban de papier), avec une ligne d'impression de code printf (avec identité et données du temps de tour variable) indiquant un message enthousiaste du participant avec son temps et nombre de tours. Le tout est possible grâce à un système Arduino Uno de base.

3.2.3. Simon :

Mon concept pour le sous-système de logiciel serait un système qu'on utilise en laboratoire pour la partie logiciel quand on utilise une boîte Arduino. L'application de Arduino Ide serait bien utile pour utiliser du code C que la majorité de nous connaît déjà comment utiliser ce langage de programmation. Dans ce code, on peut placer un tableau d'array puis, le placer dans une file qui agit comme un tableau pour afficher les données que les capteurs envoient. Aussi, pour avoir une

horloge dans ce code on peut utiliser la fonction `#include <timer.h>` et puis placer le bon code pour commencer arrêter et compter le nombre de tour que les voitures on faites.

3.2.4. Mahad :

Sous-système logiciel Arduino Uno (codage): Le cœur du système est le microprocesseur Arduino Uno, qui orchestre toutes les opérations en exécutant le code qui traite les données des capteurs et des lecteurs RFID. Le code commence par initialiser les différents composants, tels que le module RFID, les capteurs infrarouges et l'afficheur LCD. Il surveille ensuite en continu les signaux provenant des capteurs et des lecteurs RFID. Lorsque l'Arduino détecte qu'une voiture a franchi la ligne de départ/arrivée, il enregistre l'heure de passage et met à jour le nombre de tours effectués par cette voiture. Le code gère également le traitement des données, telles que le calcul des temps au tour et la détermination du gagnant. En outre, le logiciel inclut une fonctionnalité de réinitialisation pour préparer le système à une nouvelle course après la fin de chaque compétition.

3.2.5. Alessio :

Un concept logiciel possible serait surtout d'utiliser l'Arduino Uno afin que nous puissions enregistrer et chronométrer les voitures. Chaque fois qu'une voiture coupe le faisceau laser, la photocellule correspondante enregistre une interruption. Arduino Uno calcule le temps écoulé depuis le départ ou le dernier tour, selon le mode choisi (course complète ou tours individuels).

- Modes de fonctionnement :
 - Mode Course : Enregistre le temps total depuis le départ pour chaque voiture qui franchit la ligne.
 - Mode Tours : Enregistre le temps pour chaque tour individuel pour chaque voiture.

Pour la programmation il y aura plusieurs parties clés:

- Initialisation : Définir les pins des entrées (pour les photocellules) et des sorties (pour l'affichage et les alertes sonores).
- Boucle principale : Surveiller les interruptions des faisceaux lasers et calculer les temps en fonction de l'heure actuelle moins l'heure du dernier passage.

3.2.6. Leonel :

Étant donné que nos capteurs émetteurs-récepteurs laser sont compatibles avec Arduino, nous allons utiliser une **carte Arduino Uno** pour gérer la détection des véhicules, l'enregistrement des données et l'affichage des résultats. Voici comment le système fonctionnera :

1. **Programmation de l'Arduino :**

- a. À chaque réception d'un signal **LOW** (interruption du faisceau laser) par un ou plusieurs récepteurs, l'Arduino identifiera le véhicule concerné.
- b. Pour chaque tour, le système **incrémentera la position** du véhicule détecté.
- c. Le temps de passage à la ligne d'arrivée sera **mesuré et enregistré**.

2. **Affichage des informations :**

- a. Les données collectées (position, temps de passage, etc.) seront **affichées sur le moniteur série** de l'Arduino.
- b. Ces informations pourront ensuite être **exportées vers un fichier Excel** pour une analyse plus approfondie.

3.2.7. Valérie :

Système avec Arduino Uno: Ce concept est capable de traiter (et afficher) les données par lui-même. Le Arduino Uno calcule les temps de passage des voitures et les enregistre. Les résultats sont immédiatement affichés sur un écran quelconque. Lorsqu'un capteur détecte qu'une voiture passe, l'Arduino enregistre l'heure du passage avec `timeStart = millis()` et à la fin de la course, lorsque la voiture arrive à la fin de course, le capteur détecte à nouveau l'événement, et l'Arduino enregistre le temps de ce deuxième passage avec `timeEnd = millis()`. L'Arduino calcule le temps écoulé entre les deux passages en soustrayant `timeStart` de `timeEnd`.

3.3. Sous-système 3 (Affichage) :

3.3.1. Anwar :

Pour garantir une visibilité optimale, nous utiliserons un **écran LCD 20x4** ou un **affichage OLED**, connectés à l'Arduino, permettant d'afficher simultanément plusieurs informations essentielles telles que le nombre de tours, les temps de passage et le classement en direct. Le programme Arduino mettra à jour l'affichage après chaque passage de voiture sur la ligne d'arrivée, en intégrant les données recueillies par les lasers. Une fois la course terminée, un résumé s'affichera avec le **temps total de chaque voiture** ainsi que le **nom du gagnant**.

3.3.2. Omar :

L'affichage peut-être quelque chose d'aussi simple qu'une interface de base, sur un écran d'ordi connecté par USB au Arduino, puis effectuant la fonction miroir par détection d'écran sur le même réseau. Cet écran peut lui-même être un autre ordi ou bien une télévision selon les besoins du client.

3.3.3. Simon :

D'ailleurs comme méthode d'affichage je pensais à faire une page web qui utiliserait un fil USB connecter au Arduino Uno pour avoir une interface graphique avec processing et affichage de temps réels. On peut faire ceci avec le code d'Arduino ou installer un processeur sur l'ordinateur utiliser pour utiliser un code un peu plus différent sans être trop compliquer et détailler pour mélanger le client.

3.3.4. Mahad :

Sous-système d'affichage des résultats: L'affichage des résultats est essentiel pour informer les participants et les spectateurs des performances des voitures en temps réel. Nous utilisons un afficheur LCD 16x2 connecté à l'Arduino pour montrer des informations telles que le nombre de tours effectués par chaque voiture, les temps au tour et, finalement, le gagnant de la course. Le code Arduino met à jour l'afficheur LCD à chaque tour, fournissant des données précises et instantanées sur la progression de la course. Lorsque la course est terminée, l'afficheur LCD montre un résumé des résultats, y compris le temps total de chaque voiture et la voiture gagnante. Cette configuration simple et efficace permet à tous les participants de suivre facilement le déroulement de la course et de connaître immédiatement les résultats.

3.3.5. Alessio :

Pour afficher efficacement les résultats d'une course en utilisant on pourra utiliser un écran LCD ou OLED. Un écran LCD ou OLED connecté à l'Arduino peut afficher les informations de base telles que le numéro du véhicule, le temps de tour, le temps total et le classement. Chaque nouvelle ligne peut afficher les informations d'une voiture différente, et l'écran pourrait défiler si le nombre de véhicules est supérieur à la capacité d'affichage. Pour une visibilité accrue lors d'événements en extérieur ou à grande échelle, un tableau de bord composé de grands afficheurs à segments LED peut être utilisé. Chaque afficheur pourrait montrer le temps d'un véhicule particulier ou le meilleur temps du tour actuel. En intégrant une connectivité filaire à l'Arduino, les données pourraient être envoyées en temps réel à un ordinateur ou une tablette où une application dédiée affiche les résultats de manière plus détaillée. L'interface pourrait inclure un tableau des temps de tour pour chaque véhicule, des graphiques de progressions des véhicules et un résumé des meilleurs temps par véhicule.

3.3.6. Leonel :

Sous-systèmes d'affichage des résultats simple en utilisant la synchronisation entre le traitement d'information par l'Arduino et l'affichage des données via une feuille Excel, que nous concevons pour que cela ressemble à un logiciel correct. Connection via HDMI d'un écran à l'ordinateur central.

3.3.7. Valérie :

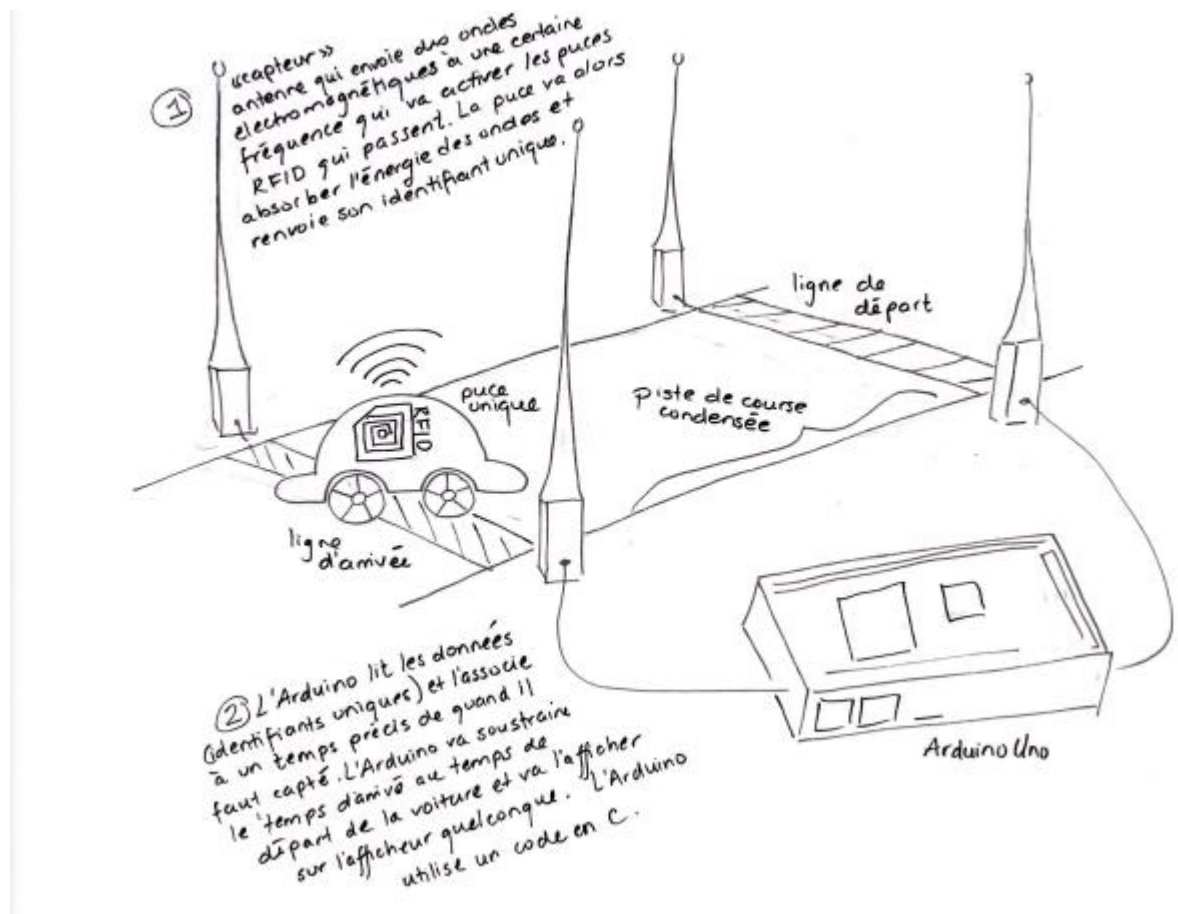
Écran LED: Un écran LED matriciel de 64 LEDs ou plus (connecté à un Arduino) permet d'afficher en temps réel (au besoin) les résultats de la course. Ce système utilise une base de

données statique (un tableau prérempli) pour afficher les temps de passage: À chaque passage, l'Arduino peut afficher les temps sur l'écran LED en utilisant une bibliothèque comme LedControl par exemple. Il est aussi possible d'ajouter un mode replay, où les spectateurs et juges peuvent revoir l'évolution de la course, avec les données passées et un graphique qui décrit la course.

4. Esquisses:

4.1. Sous-système 1 (Capteur) :

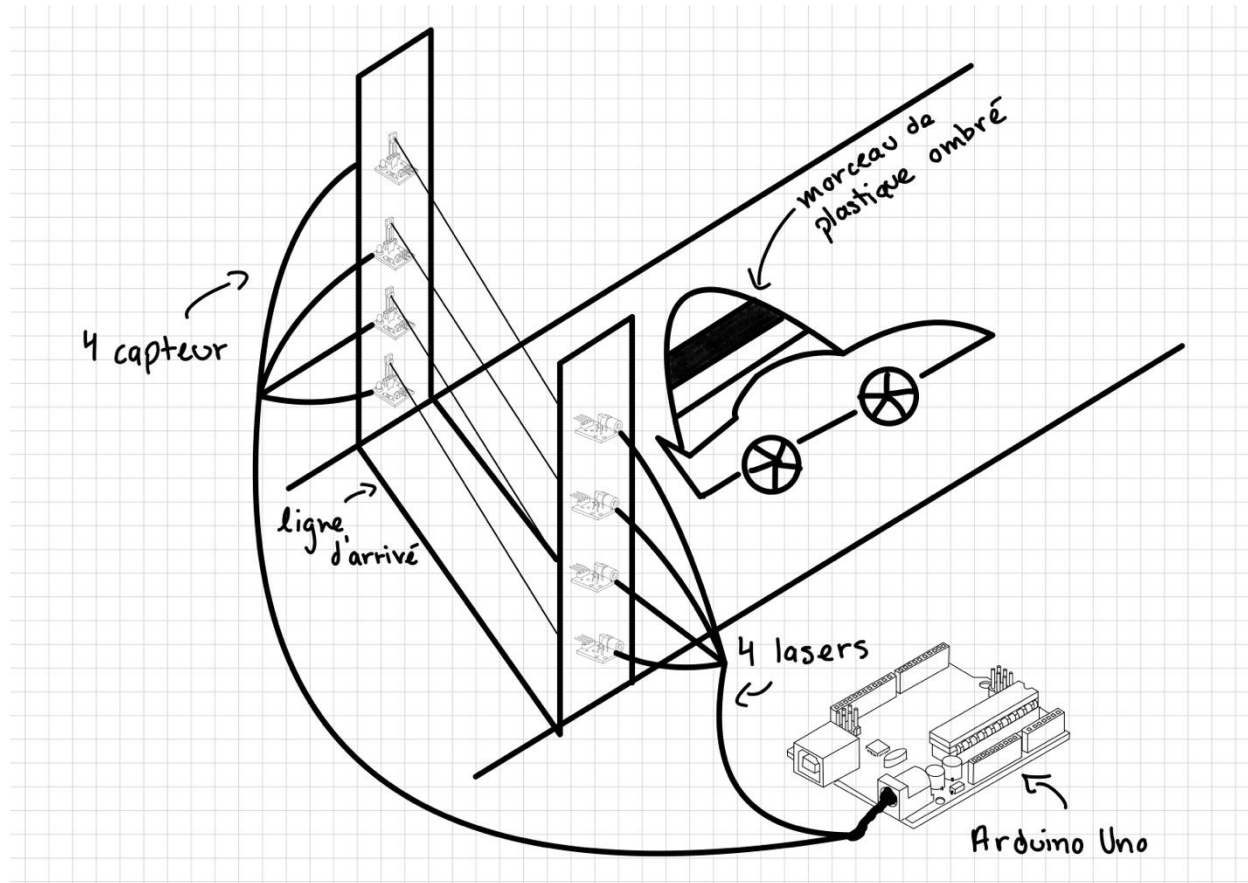
4.1.1. Puce RFID :



4.1.2. Laser :

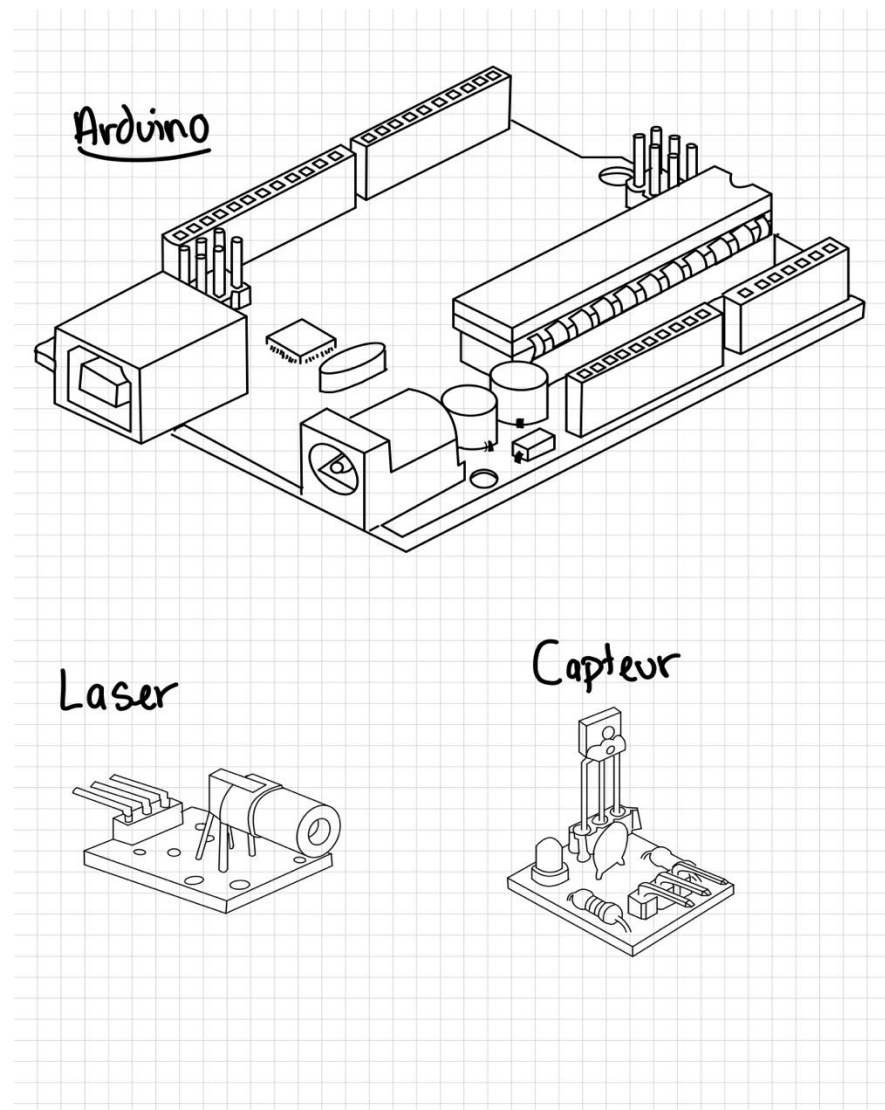
Pour ce concept il y a un laser et un capteur pour chaque voiture. Puis, la voiture a un morceau de plastique dessus le haut de la voiture qui est divisé en 4 parties ou que 1 est noircis et les trois autres sont claires. Chaque voiture a une section différente noircie pour que les lasers ne passent pas au travers et puis quand le capteur ne reçoit pas le laser, il va envoyer un message au Arduino

pour lui dire que tel voiture a traversé la ligne d'arrivée. Aussi, chaque laser est associé avec une voiture, donc, quand le laser est bloqué il envoie le signal au Arduino.



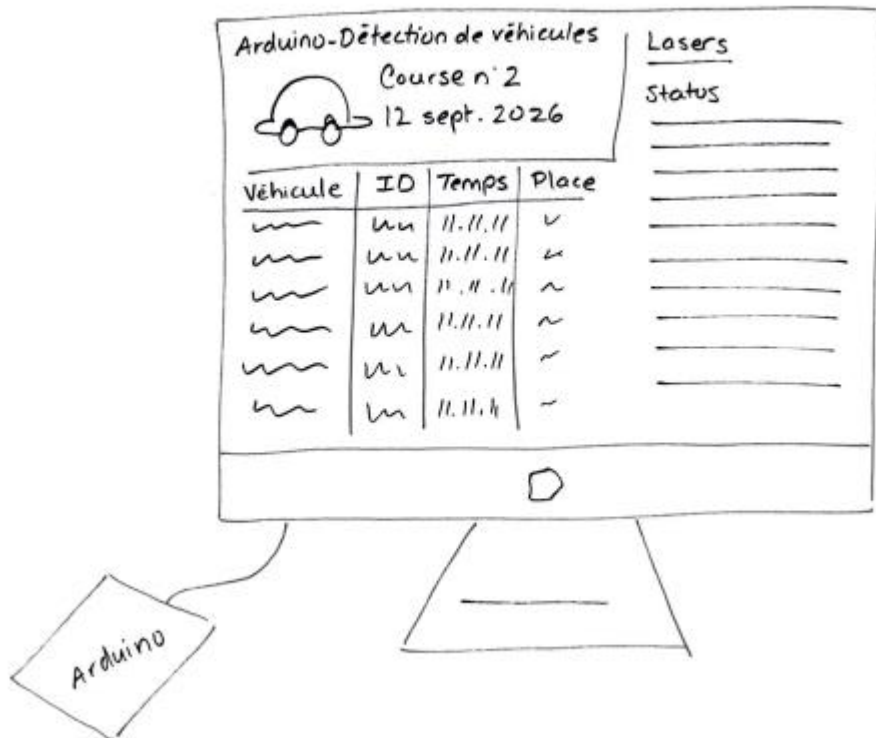
4.2. Sous-système 2 (Logiciel) :

L'Arduino Uno va être utilisé pour la partie codage et logiciel de notre produit. Il va servir à placer une horloge dans le code pour avoir des temps sur les tours que les voitures font et les associer aux bonnes voitures. En plus, dans ce code on peut placer un tableau de type array et puis le placer dans un document pour faire l'affichage des résultats de la course. Sur temps réel et garder les temps en mémoire pour revoir les résultats à la fin de la course.



4.3. Sous-système 3 (Affichage) :

Site web :



5. Matrice décisionnelle :

		Capteur						Logiciel						Affichage								
	Importance (Poids)	A	O	L	S	V	M	A	A	O	L	S	V	M	A	A	C	L	S	V	M	A
		n	m	e	i	a	a	n	n	m	e	i	a	a	n	n	e	a	i	a	a	e
Budget	5	1	3	4	4	5	4	4	2	5	4	5	4	4	4	2	5	4	5	4	4	3
Simplicité	5	1	5	5	5	5	3	3	3	3	4	4	4	5	3	3	4	4	5	5	4	3
Portabilité	4	2	5	4	5	4	4	4	4	5	4	5	4	4	5	4	3	4	4	4	4	3
Connectivité	4	5	5	3	4	3	4	4	3	5	4	5	4	4	5	3	3	4	3	4	4	5
Fiabilité	4	5	4	4	4	4	4	4	4	5	3	4	3	4	4	4	4	4	3	4	4	5
Total		58	96	89	81	94	83	83	70	104	84	104	83	91	97	85	88	90	93	88	82	

6. Avantage et inconvénient du Système de Détection par Émetteur Laser :

6.1 Avantages

- (i) **Précision élevée**
 - a. Le laser offre une détection instantanée et précise du passage des véhicules.
 - b. La séparation verticale des faisceaux permet d'identifier chaque voiture individuellement.
- (ii) **Simplicité et coût abordable**
 - a. Utiliser des capteurs laser avec Arduino est plus économique que des systèmes de caméras à haute vitesse ou des capteurs RFID avancés.
 - b. Le système repose sur des composants standards facilement accessibles.

(iii) Fiabilité et rapidité

- a. Le laser fonctionne en temps réel et ne nécessite pas de traitement d'image complexe.
- b. Pas d'interférence avec d'autres fréquences radio ou systèmes GPS.

(iv) Adaptabilité et évolutivité

- a. Il est possible d'ajouter plus de faisceaux pour détecter davantage de véhicules.
- b. Peut être adapté pour différents types de courses et pour plusieurs compétiteurs supplémentaires

(v) Facilité d'intégration avec un logiciel de gestion des résultats`

- a. Les données sont directement exploitables par Arduino et peuvent être affichées sur Excel.

6.3 Inconvénients

i. Sensibilité aux conditions environnementales

- a. La pluie, le brouillard ou la poussière peuvent perturber les faisceaux laser.
- b. Un mauvais alignement des capteurs peut fausser la détection.

ii. Complexité d'installation et de calibration

- a. Le positionnement précis des portiques et des faisceaux doit être bien calibré.
- b. Le marquage des véhicules doit être parfaitement aligné avec les lasers, et épouser la forme aérodynamique des toitures des véhicules

iii. Limitations en cas de forte vitesse

- a. Si un véhicule roule à très grande vitesse, il pourrait y avoir un risque de floue dans la détection (négligeable si bien calibré).

iv. Risque de perturbation entre plusieurs faisceaux

- a. Si plusieurs véhicules passent en même temps, les multiples réfractions des rayons laser émisnnnnnn pourrait poser un problème, et ne pas rejoindre le bon récepteur laser, et causer ainsi un mal fonctionnement.

v. Nécessité d'un marquage spécial sur les véhicules

- a. Chaque voiture doit être équipée du même type de toiture, et avoir la même hauteur, ce qui demande une installation supplémentaire.

vi. Encombrement, o a ce jour, pour détecter 4 véhicules, il faut 4 récepteurs, quand sera t'il quand il faudra 18 compétiteurs, prendrions-nous 18 capteurs et émetteur laser?

Conclusion :

Dans cette étape du projet de conception, Nous avons d'abord rassemblé divers concepts créatifs pour les trois sous-systèmes du système générale en se basant sur l'étalonnage technique, en prenant en compte les aspects techniques que ces concepts pourraient présenter lors des essais. Ensuite, après avoir analysé tous les concepts préliminaires proposés par chaque membre de l'équipe, nous nous sommes convergées vers la solution qui correspond le mieux avec nos critères ciblés dans le livrable précédent.

7. Référence:

[1]

“Free CAD Designs, Files & 3D Models | The GrabCAD Community Library,” *Grabcad.com*, 2024. <https://grabcad.com/library/arduino-uno-rev3-microcontroller-board-1> (accessed Feb. 08, 2025).

[2]

“Amazon.fr,” *Amazon.fr*, 2015. <https://www.amazon.fr/Sgerste-capteur-Non-modulator-%C3%A9metteur-r%C3%A9cepteur/dp/B07JFTFLWM> (accessed Feb. 08, 2025).

[3]

DHM, “KY-008 Module capteur laser Arduino,” *Compass DHM projects*, 2025. <https://www.dhm-online.com/fr/modules-arduino/2059-ky-008-module-capteur-laser-arduino.html> (accessed Feb. 08, 2025).

8. Trello :

