

Projet ROBOCUP

Parcours Innovation

GAZUI Laetitia - DECULTOT Martin - MOCAER Hugo - LUC Valentin - MONCLUS Louis

TABLE DES MATIERES :

I/- PARTENARIAT ET ORGANISATION	2
A. Partenariat	2
1. Elektrons Libres	2
2. Cohabit	2
B. Qui sommes-nous et quel est notre projet ?	3
1. Membres	3
2. Projet Robocup SSL.....	3
→ Essentiel des règles et fonctionnement :	4
3. Organisation Interne	5
→ Répartition des tâches :	5
→ Notre planning :	5
→ Outils utilisés :	6
C. Nos Inspirations	7
1. NAMEC (Nouvelle-Aquitaine Mécatronique Club)	7
2. Tigers Manheim.....	7
II/- LE PROJET	8
A. Mécanique.....	8
1. Etude du robot v1 : NAMEC	8
→ À la recherche des composants du robot :	9
→ Fonctionnement du Robot :	10
→ Conception des pièces et usinage :	12
→ Moulage résine (Polyuréthane PU) :	13
2. Amélioration du robot en v2, inspirée des NAMEC et des Tigers.....	15
→ Déplacement des cartes électroniques moteurs :	15
→ Modification du carénage :	16
→ Modifications et tests du dribbleur :	Erreur ! Signet non défini.
→ Modifications du kicker :	16
→ Modifications des roues :	17
B. Electronique.....	18
C. Informatique Embarquée	18
D. Software.....	19
1. Ce qu'on a fait	19
2. Ce qu'on a prévu de faire	20
3. Comment on le fait.....	21
III/- CONCLUSION	24
IV/- ANNEXES.....	25

I/- Partenariat et Organisation

A. Partenariat

1. Elektrons Libres



Elektrons Libres est une association, loi 1901, qui réunit des jeunes, parents et formateurs. Créée en septembre 2019 à Pau, elle a pour vocation de faciliter l'accès aux sciences à tous les jeunes de l'agglomération paloise, de favoriser leur mobilité à l'international, de renforcer leur identité européenne, tout en leur permettant de participer à des concours et en les soutenant dans leur orientation professionnelle.

Les jeunes à l'initiative de la création d'Elektrons Libres se sont rencontrés dans le cadre de projets auxquels ils ont participé au sein du lycée Saint-Cricq : concours de robotique, projet d'astronomie, séminaires scientifiques en Allemagne et en Écosse, etc.

En créant cette association, ces lycéens souhaitent partager leurs expériences et intégrer des jeunes extérieurs à leur établissement afin que ces opportunités soient offertes au plus grand nombre.

C'est Elektrons Libres qui nous a proposé de participer à la Robocup au sein de l'association. Ainsi nous avons accès à leur réseau d'information et leur partenaire, Cohabit.

2. Cohabit



Cohabit est une association regroupant le fablab et le technoshop de l'université de Bordeaux. Ainsi Cohabit regroupe un espace collaboratif de fabrication numérique au travers du fablab et un technoshop pour le développement de projets technologiques innovants.

L'association propose des ateliers à destination des jeunes, mais également des professionnels ou particuliers sur des concepts technologiques.

Au travers de cette association, nous avons accès au **club innovation** de l'université de Bordeaux. Cette entité est à l'origine de rassemblement Elektrons libres/Cohabit et de la formation de l'équipe. C'est également le club innovation qui nous permet d'assister à des cours d'informatique embarquée et de software en rapport avec la Robocup SSL ainsi qu'à des conférences en rapport avec les nouvelles technologies.

B. Qui sommes-nous et quel est notre projet ?

1. Membres

Nous sommes 10 élèves de l'école d'ingénieur ESME Sudria Bordeaux, dont 5 en parcours innovation :

- En parcours innovation :



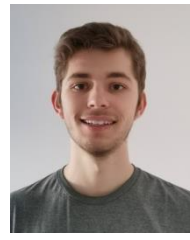
GAZUI Laëtitia
Leader



DECULTOT Martin
Mécanique



MOCAER Hugo
Mécanique



LUC Valentin
Software

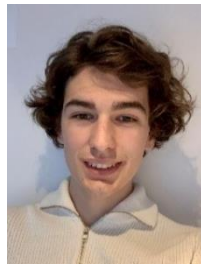


MONCLUS Louis
Software

- Hors parcours innovation :



DARNAUD Louis
Mécanique



DE-FARIA Tenessy
Info Embarquée



CHUKRY Clément
Info Embarquée



SUREAU-SAMIT
Arthur
Info Embarquée



COUTHURES
Aurélien
Software

2. Projet Robocup SSL

La **Robocup** est la plus grande compétition de Robotique et d'intelligence artificielle au monde. Elle rassemble chaque année plus de 3500 participants et autant de robots venants de 45 pays différents. De même, des milliers de chercheurs venus du monde entier viennent assister à des conférences dans le but de partager connaissances, savoir-faire, idées et technologies.

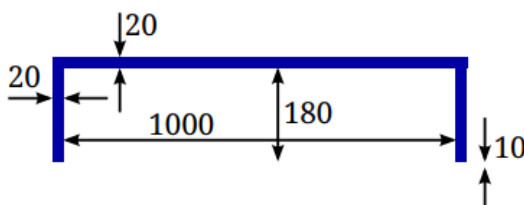
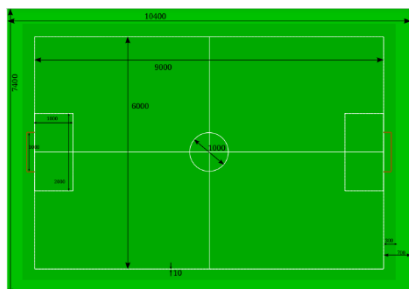
C'est une initiative scientifique internationale dont le but est de faire progresser l'état de l'art des robots intelligents. Lors de sa création en 1997, la mission initiale était d'aligner une équipe de robots capable de gagner contre les champions de la Coupe du monde de football humain d'ici 2050.

La Robocup contient de nombreuses compétitions différentes réparties par ligues : Soccer, Rescue, @Home, Junior. Et dans chaque ligue, il existe plusieurs compétitions.

Nous avons choisi de participer à la compétition SSL (small size league) de la ligue Soccer, compétition se concentrant sur le problème de coopération et de contrôle intelligent de plusieurs robots dans un environnement dynamique avec un système de contrôle centralisé : l'ordinateur contrôle et envoie continuellement des ordres aux robots.

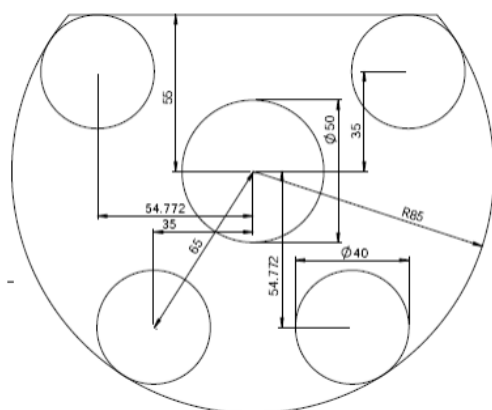
→ Essentiel des règles et fonctionnement :

La SSL est un **match de football** se déroulant entre deux équipes ayant chacune six robots. Chaque robot doit être conforme aux dimensions spécifiées dans les règles F180 : le robot doit pouvoir entrer dans un cercle de 18cm de diamètre et ne doit pas dépasser 15cm de hauteur. Les robots jouent au football avec une balle de golf orange sur un terrain recouvert de tapis vert de 9 m de long sur 6 m de large (voir image 1 ci-dessous).

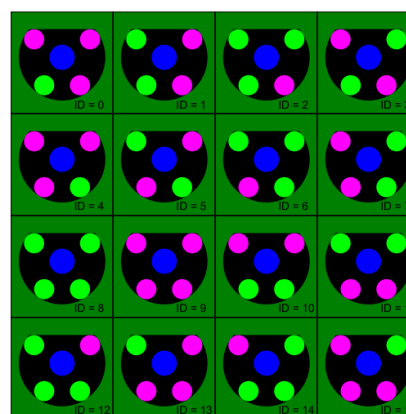


Tous les objets sur le terrain sont suivis par un système de vision standardisé appelé SSL Vision, un projet open source maintenu par la communauté de la ligue. Ce dernier traite les données fournies par quatre caméras attachées à une barre de caméra située à 4m au-dessus de la surface de jeu. Ces caméras envoient les images à un ordinateur coordinateur central qui traite les images et renvoie aux deux équipes la position, vitesse et orientation de chaque robot sur le terrain à chaque instant t (environ toutes les quelques millisecondes en simulation).

Pour avoir l'orientation du robot, l'ordinateur analyse les images de chaque robot grâce à un identifiant couleur composé de 5 cercles : un central représentant la couleur de l'équipe (bleue ou jaune), les quatre autres utilisés pour reconnaître chaque robot dans l'équipe (voir ci-dessous).



Placement conforme des identifiants couleur sur un robot



Signification des identifiants couleur pour l'ordinateur central

Les ordinateurs de chaque équipe sont utilisés pour le traitement nécessaire à la coordination et au contrôle des robots. Les communications sont sans fil entre l'ordinateur de chaque équipe et les robots et utilisent des émetteurs / récepteurs radio commerciaux dédiés.

En ce qui concerne les fonctionnements de certaines parties précises du robot (comment tirer dans la balle ? Comment le robot se déplace-t-il ? Comment attrape-t-il la balle pour dribbler ?) sont expliqués dans le paragraphe II/A. "Mécanique" en page 10.

3. Organisation Interne

→ Répartition des tâches :

Au vu de la densité de travail et de la taille du projet, nous sommes répartis sur 4 pôles : Mécanique, Electronique, Informatique embarquée et Software.

- Au pôle **Mécanique**, les étudiants étudient la modélisation 3D puis réalisent la fabrication du robot. Cela passe par toutes les étapes : utilisation du logiciel de modélisation (Inventor et Solidworks), élaboration de tableurs de budget répertoriant tous les composants à fabriquer, acheter ou à faire usiner, et l'impression des pièces 3D et de découpe laser au fablab.
- Au pôle **Electronique**, les étudiants apprennent à comprendre et modifier des schémas électroniques. Nous n'avons aucun étudiant travaillant vraiment là-dessus pour le moment, laissant ce pôle à deux membres expérimentés de l'association Elektrons Libres.
- Au pôle **Informatique embarquée**, les étudiants programment la partie embarquée du robot, comme par exemple les moteurs sur la carte électronique Maple Mini.
- Au pôle **Software**, les étudiants programment "l'intelligence stratégique" des robots. Sans faire de l'IA, ils travaillent sur des parcours d'arbres pour la prise de décision d'action de chaque robot dans le but d'une stratégie collective.

→ Notre planning :

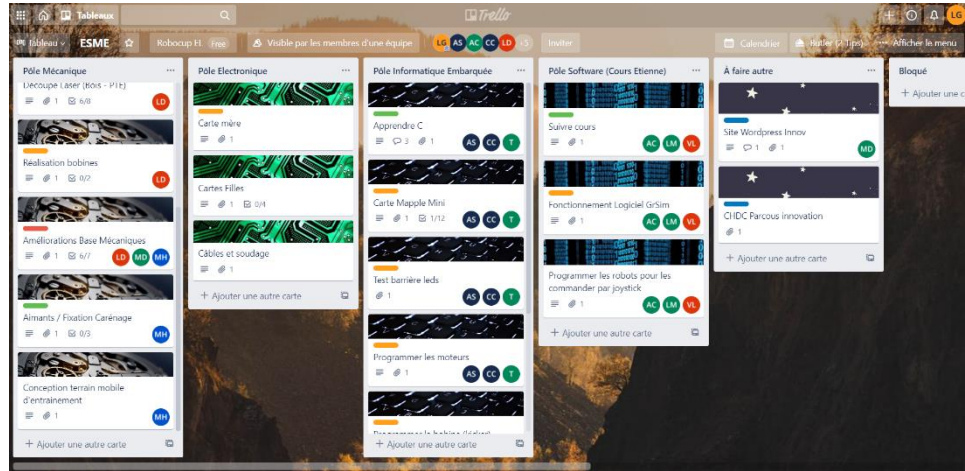
- Les **mardis soir** de 18h à 20h, les élèves s'occupant de la partie **Informatique embarquée** suivent les cours de Grégoire Passsault, professeur à l'Enseirb Matméca, à l'IUT et à l'université de Bordeaux et également membre de l'équipe Rhoban (quadruple championne du monde en ligue Humanoid Soccer de la Robocup).
- Les **mercredis après-midi** de 14h à 18h nous avons le parcours innovation où nous avançons le projet.
- Les **jeudis après-midi**, lorsque la situation le permet, nous venons au fablab de l'ESME continuer le projet.
- Les **jeudis soir**, nous avons des réunions hebdomadaires de l'équipe qui intègre aussi des étudiants Cohabit ou Elektrons libres, totalisant une trentaine de participants.
Le même soir après la réunion, le pôle **Mécanique** tient une réunion pour échanger sur l'avancée du travail et se répartir les tâches de la semaine.
De même, de son côté le pôle **Software** suit les cours d'Etienne Schmitz, étudiant en dernière année à l'Enseirb (filière robotique), membre de l'équipe NAMEC et qui enseigne la SSL au sein du club innovation de l'IUT de Bordeaux .

En plus, nous travaillons chacun de notre côté pour effectuer diverses tâches à réaliser dans la semaine, comme par exemple le moulage en résine des dribbleurs que l'on ne peut faire à l'école.

→ Outils utilisés :

- **Trello** : un outil de gestion de projets en ligne partagé entre les collaborateurs.

Cela permet de nous organiser entre les différents pôles et de suivre notre avancée générale et respectant des dates clés.



- **Teams** : Plateforme Collaborative de télétravail utilisée par l'école.

- **Discord** : Discord est un logiciel d'échanges très pratique par ses sous-groupes de discussions textuelles ou vocales permettant de parler d'un seul sujet par conversation sans créer un groupe pour chaque sujet. Initialement créé pour les gamers et pouvant accueillir de très nombreux membres dans un même groupe, il est utilisé de façon plus universelle dorénavant.



C'est la plateforme d'échanges de toute notre l'équipe, utilisée également pour suivre les cours proposés par le club innovation.

- **Google Drive** : Stockage et partage de fichiers.

On l'utilise pour partager tous nos fichiers administratifs, nos tableurs de commandes et tout fichier relevant de la mécanique.



- **GitHub** : un service web d'hébergement et de gestion de développement de logiciels, utilisé couramment pour l'hébergement de codes informatiques. Nous partageons nos codes informatiques en *Software* par ce service web.



- **GitLab** : GitLab est un logiciel libre basé sur git proposant les fonctionnalités d'un wiki, un système de suivi des bugs, l'intégration continue et la livraison continue. C'est le moyen de partage utilisé par l'équipe NAMEC.



- **OnShape** : Logiciel de conception assistée par ordinateur en ligne partagé entre collaborateurs.



- **Inventor** : Logiciel de modélisation 3D distribution Autodesk disponible librement pour les étudiants.



- **GrSim** : Logiciel fourni par la Robocup et remanié par les NAMEC pour simuler informatiquement le mouvement des robots lors d'un match de foot.

C. Nos Inspirations

La Robocup repose avant tout sur le partage et la collaboration dans le but de faire avancer la technologie. Alors, participer à la compétition impose à tous les participants de publier leurs recherches en open source une fois le championnat du monde passé.

Ainsi, cette année nous nous inspirons principalement de 2 équipes : les *NAMEC* (Nouvelle Aquitaine Mécatronique Club) (Français) et les *Tigers Mannheim* (Allemands)

1. NAMEC (Nouvelle-Aquitaine Mécatronique Club)



L'équipe *NAMEC* a été lancée en 2017 avec le soutien de l'équipe *Rhoban* (quadruple championne du monde en Humanoïd size de la Robocup) et d'enseignants chercheurs du LaBRI (Laboratoire Bordelais de Recherche en Informatique). L'objectif de ce projet régional est d'allier pédagogie innovante et collaboration entre enseignants chercheurs autour de la robotique.

L'équipe a participé à la Robocup au cours des deux dernières années, atteignant les quarts de finale l'année passée.

Nous sommes en étroite relation avec leur équipe, notamment avec Etienne Schmitz, leader de l'équipe entière, Grégoire Passault nous donnant des cours d'informatique embarquée et Patrick Félix, à l'origine de la création de notre nouvelle équipe. Les *NAMEC* étant la seule équipe française en SSL, il leur tenait à cœur de soutenir une nouvelle équipe.

Ainsi, l'équipe *NAMEC* a accepté de nous prêter un de leurs robots utilisé lors de la Robocup 2019 et nous a permis d'accéder à leurs fichiers mécaniques, électroniques et informatiques afin de nous lancer dans la compétition.

2. Tigers Mannheim



Les *Tigers Mannheim* est une équipe de Robocup SSL allemande fondée en 2011 qui s'est classée en haut du classement lors des dernières Robocup.



Après avoir étudié le robot *NAMEC*, nous nous inspirons également des *Tigers* pour ses composants placés par "modules" plus pratiques en cas de casse et ses recherches théoriques poussées, écrites et transmises aux autres équipes.

Ainsi, les *Tigers* nous ont beaucoup inspiré lors de la conception mécanique de notre propre robot.

II/- Le Projet

A. Mécanique

1. Etude du robot v1 : NAMEC

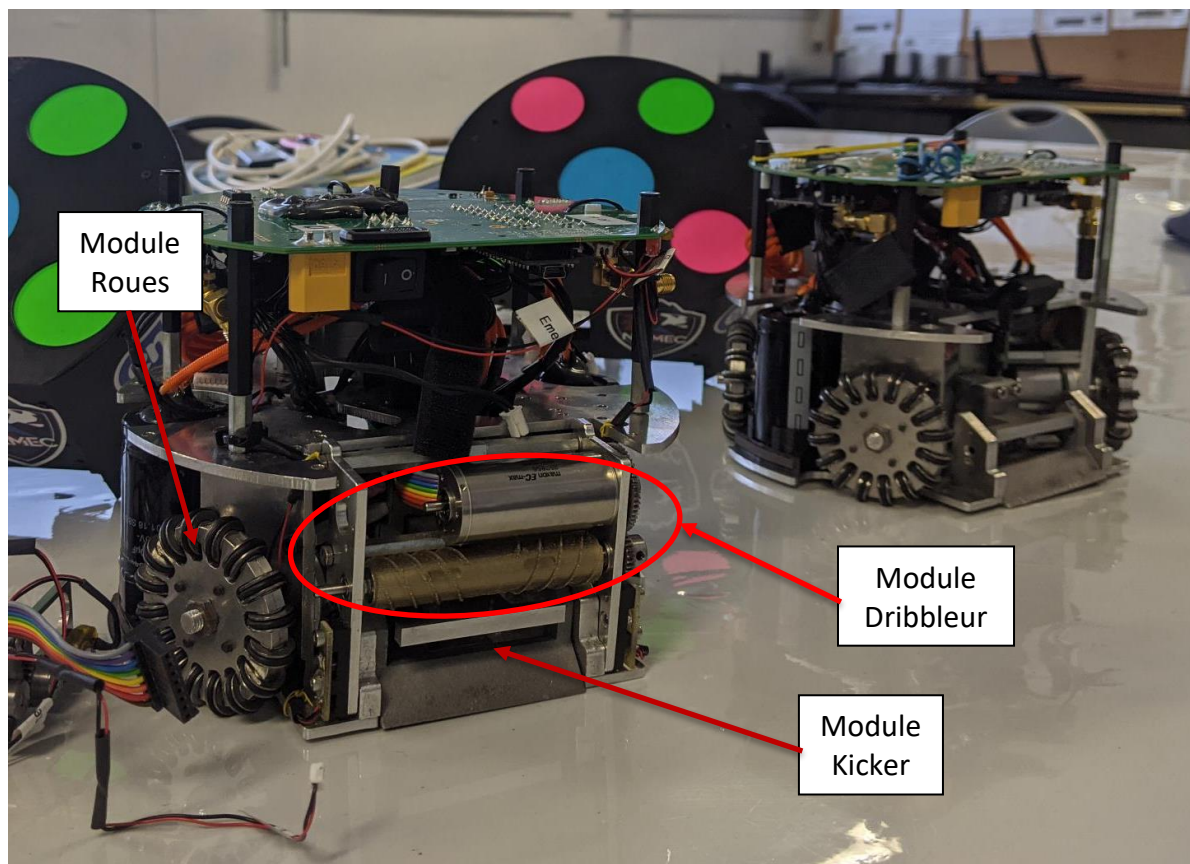
Lorsque l'on s'est lancé dans le projet, l'équipe NAMEC de Bordeaux nous a prêté un de leurs robots ainsi que l'accès à leur logiciel de modélisation 3D : OnShape.

Ainsi, on a pu commencer à l'étudier et réfléchir aux éventuelles améliorations en fonction des problèmes existants dont nous ont fait part l'équipe.

Tout d'abord, nous avons effectué un travail de sous-traitance : les NAMEC n'ayant pas de fichier ou liste répertoriant toutes les pièces utilisées sur leur robot, nous avons dû tout reprendre à zéro en analysant chaque composant présent sur la modélisation 3D. Ce fut un travail long et fastidieux car beaucoup de composants relevant de la visserie manquaient à l'appel et de nombreuses autres pièces étaient très spécifiques et étaient très difficiles à trouver dans le commerce.

Néanmoins, ce travail nous permit d'analyser chaque petite partie du robot et de comprendre son fonctionnement général.

Voici le robot NAMEC :



→ À la recherche des composants du robot :

Comme énoncé précédemment, tout d'abord, nous avons effectué un travail de sous-traitance : nous avons dû décortiquer le robot prêté par l'équipe NAMEC et lister la moindre pièce utilisée. Le but est de connaître toutes les pièces et les provenances s'il est nécessaire d'en recommander en cas de casse notamment.

Il s'agit d'un travail lourd et fastidieux réalisé, dans notre cas, à postériori. Cela nous permet néanmoins de comprendre l'importance de l'organisation au sein d'une équipe.

Ainsi on créa 3 tableaux regroupant respectivement les composants mécaniques à usiner (impression 3D, découpe laser, découpe CNC, moulage résine et impression sur lit de poudre), les composants électroniques (moteurs, condensateurs, batterie...) et les composants relevant de la visserie (vis, roulements, barres métalliques...) à commander sur internet.

Voici par exemple un extrait de la liste visserie :

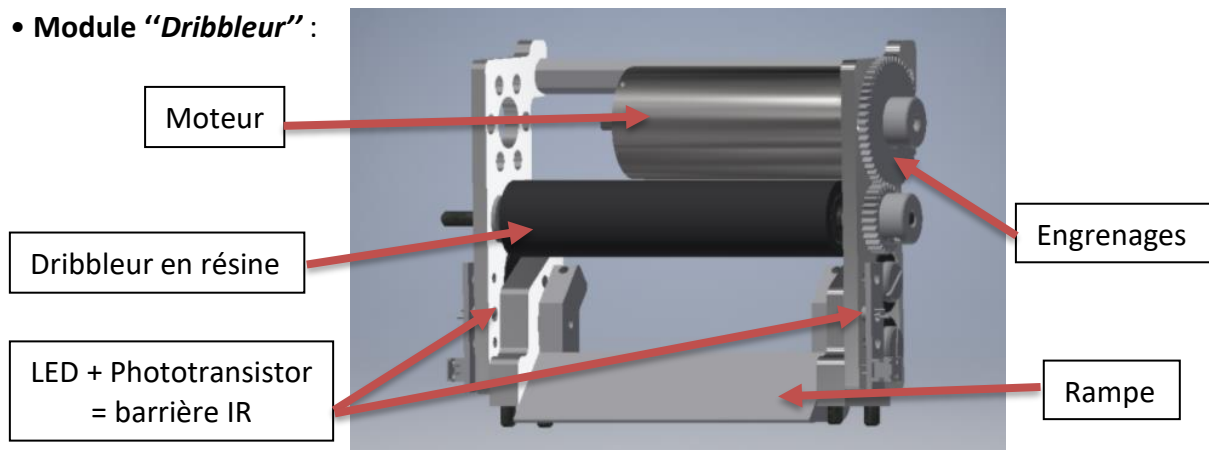
Liste Composants Visserie											
Index	Nom / Référence	Quantité	Description	Type	Commentaires	Quantité dans l'article	Nombre d'article à acheter	Prix Unitaire (€)	Prix Total (€)	Liens achat	Images
11	m3_5mm spacer	4	Entretroises M3 5mm - soutenant la carte elec mère	Visserie		5	1	12,2	12,2	https://www.amazon.com/vorgel-Aluminium-Standoff-Fastener-Quadropier/dp/B01M5G4H5Q	
11bis	Vis-spacer M3_10+6mm	4	Vis-entretroises M3 10+6mm - fixant les entretroises soutenant la carte elec mère et ayantant le rôle d'entretoise avec le rasot.	Visserie		100	1	17	17	https://www.amazon.com/ZXHAO-Female-Thread-Standoff-Pillar/dp/B07LSRZY4/ref=pp_srch_tw?pf_rd_p=1147d8cfe1&pf_rd_r=W3+10mm+space&tw=ss&qid=1603387726&s=s-8-14	
11tris	vis spacer	8	Vis M3*6mm (nylon) - fixant les entretroises soutenant la carte elec mère par le bas *ref : CSHSM-MT3B-M3-6	Visserie		1	10	0,68	6,8	https://fr.misumi-europe.com/vona2/detail/221000561365/?HissuCode=CS+SM-ST3B-M3-M3&PNSearch=CS+SM-ST3B-M3-M3&WSearch=CS+SM-ST3B-M3-M3&searchFlowresults2type	
12	A5 1474	4	Ecrus M6, diam 6mm (acier) - fixation roues *ref : LBNR6	Visserie		1	5	0,34	1,7	https://fr.misumi-europe.com/vona2/detail/11030020540/?HissuCode=LBNR6&PNSearch=LBNR6&WSearch=LBNR6&searchFlowresults2type	
13	A5 1420	29	Vis M3*8mm (acier inoxydable) - fixations roues *ref : S3-B-P	Visserie		1	30	0,25	7,5	https://fr.misumi-europe.com/vona2/detail/221005614842/?CategorySpec=00000230837%3a%3amq00000001977837%090000230780%3a%3amq00000001977700%090000230944%3a%3abq0000000230778%3a%3aa9	
14	oring_od13	64	O'Ring Rubber OD13*ID9.0 - anneaux flexibles sur roulements à bille sur roues	Visserie		20	4	5,49	21,96	https://www.amazon.com/2DPex-ID9.0-CS3.0mm-Butylnitrile-Rubbe/dp/B07QDB1K3?rmi=1	
15	SPWF3	128	Rondelles Plates M3 (référence misumi : SPWF3) - fixation autour roulements à billes sur roues *ref : SPWF3	Visserie		1	130	0,15	19,5	https://fr.misumi-europe.com/vona2/detail/110300252910/?HissuCode=SPWF3	
16	SZV3-14	64	Roulement à bille ID3*OD14 (référence misumi : SZV3-14)	Visserie		1	65	7,46	484,9	https://uk.misumi-europe.com/vona2/detail/110302025220/?HissuCode=SZV3-14	
17	3_B	68	Goupille cylindrique ISO 2338 M6*3mm - axes roulements à bille	Visserie		200	1	7,47	7,47	https://www.wegartdealer.com/ArticleDetails.aspx?ANVR=4564-293	
20-21	collet_screw / collet_nut	4	Vis + entretoise permettant de fixer la roue à l'axe moteur	Visserie	Option d'achat: Prop Shaft Ø 6mm	5	1	6,26	6,26	https://fr.aliexpress.com/item/32704164501.html	
22	LBMFN3-32_r	18	Entretoise M3 32mm (référence misumi : LBMFN3-32) - fixation cartes elec moteur	Visserie		1	20	3,36	67,2	https://fr.misumi-europe.com/vona2/detail/110300367370/?HissuCode=LBMFN3-32&PNSearch=LBMFN3-32&WSearch=LBMFN3-32&searchFlowresults2type	
22bis	Vis M3*8mm	36+4+4	Vis M3*8mm - fixation entretroises n°22, n°36 et n°47 *ref : CSH-ST-M3-8	Visserie		1	50	0,16	8	https://fr.misumi-europe.com/vona2/detail/221000551288/?HissuCode=CS+ST-M3-8&PNSearch=CS+ST-M3-8&WSearch=CS+ST-M3-8&searchFlowresults2type	
27-28	washer_BA0414	4+2	Rondelles OD14*ID4 (référence misumi : BA0414) - permet la rotation du moteur sans abimer *ref : BA0414	Visserie		1	6	5,74	34,44	https://uk.misumi-europe.com/vona2/detail/110300119770/?HissuCode=B0414	
26	m3_10mm spacer	4	Entretoise M3*15mm (petit soutien carte)	Visserie		4	5	0,55	2,7	https://fr.misumi-europe.com/vona2/detail/221004967004/?HissuCode=M3+10mm+spacer	

On a identifié les fournisseurs par couleur afin de faciliter les commandes et mis des descriptions et des images pour que chacun sache de quoi on parle lorsque l'on mentionne un composant et où il se situe.

Ce travail nous a permis de réaliser qu'en phase de création, cette tâche doit être intégrée pleinement au projet afin de faciliter la maintenance et la compréhension. Cela sera également pratique pour les élèves qui voudront intégrer le projet : la compréhension du fonctionnement du robot en sera accélérée sans avoir à solliciter les "anciens".

→ Fonctionnement du Robot :

• Module “Dribbleur” :



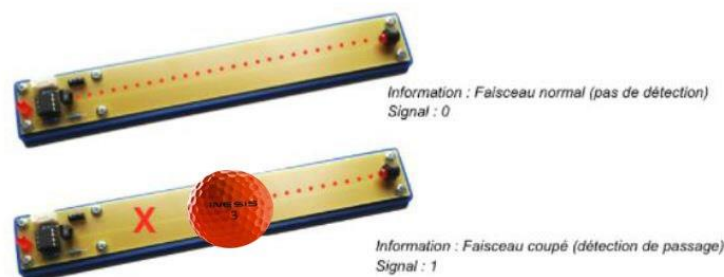
Le module est composé d'un moteur qui transmet une rotation par l'intermédiaire d'engrenages à une “gomme” fabriquée grâce à méthode de moulage de résine polyuréthane (PU).

Cet assemblage permet au robot de “dribbler”, c'est-à-dire garder la balle pendant qu'il se déplace. La résine est la solution utilisée par la plupart des équipes SSL car elle absorbe le choc de la réception de la balle tout en ayant la dureté nécessaire pour conserver, “attraper” continuellement la balle. En effet, la balle est “attrapée” par le filetage présent sur la surface du dribbleur. Ce dernier doit être assez solide pour ne pas se rompre au premier choc.

En dessous du dribbleur, on peut apercevoir une rampe contre laquelle la balle reste en contact. Cette rampe n'est pas fixe et peut permettre au robot de faire un lob quand elle est tapée par derrière (nous verrons cela dans le module “kicker”).

De même, une led + phototransistor infrarouge sont présents de part et d'autre des armatures du module pour détecter la présence de la balle :

→ Fonctionnement de la barrière Infrarouge (LED + phototransistor) :

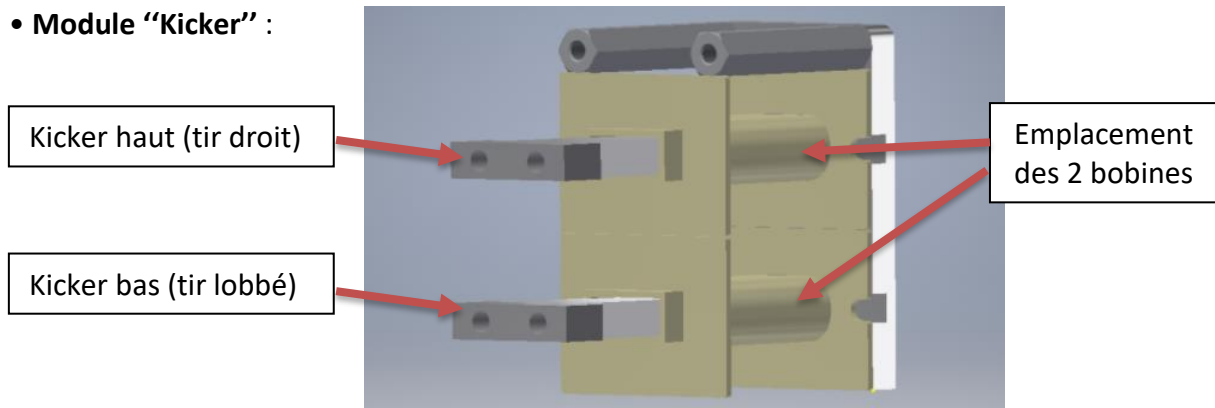


La barrière IR utilise un principe booléen : un émetteur émet un rayon infrarouge d'un côté et de l'autre, un récepteur reçoit (ou pas) ce rayon.

- Si le récepteur reçoit le rayon, *False* est renvoyé : la balle n'est pas entre les capteurs
- Si le récepteur ne reçoit pas le rayon, *True* est renvoyé : la balle est entre les capteurs.

Ce système permet de vérifier la présence de la balle devant le kicker pour ensuite pouvoir tirer, faire une passe ou encore un lob...

• **Module “Kicker” :**

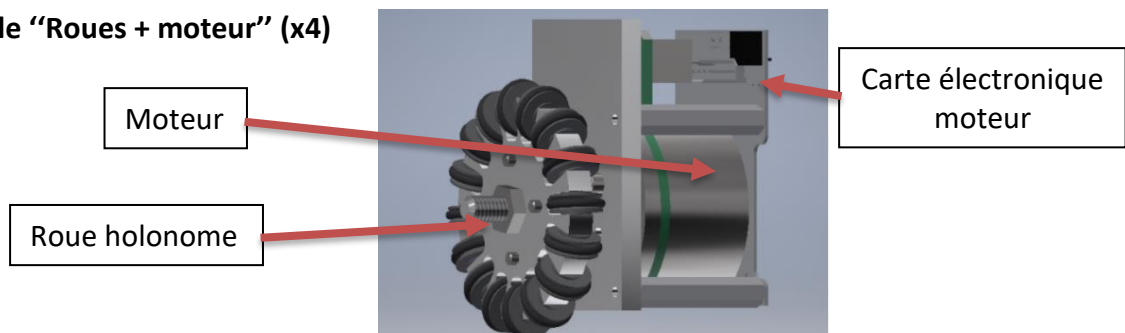


Le module est composé de 2 bobines enroulées autour des tubes qui vont propulser les 2 kickers (haut et bas) vers l’avant à très grande vitesse grâce à une pièce d’acier au centre qui va être projetée vers l’avant par la force électromagnétique créée par les bobines. Les bobines étant alimentées par les 2 condensateurs latéraux sur le robot.

Une fois la balle frappée, le système est ramené à sa position initiale grâce à 2 élastiques situés à l’arrière de chaque kicker.

Cet assemblage permet au robot de frapper la balle : soit par un tir normal ou une passe (kicker du haut), soit en lobbant grâce au kicker du bas qui va taper dans la rampe (voir image dans le module “dribbler”) qui va à son tour soulever la balle en lob.

• **Module “Roues + moteur” (x4)**



Le module est composé d’une roue holonome (roue omnidirectionnelle) reliée à un moteur.

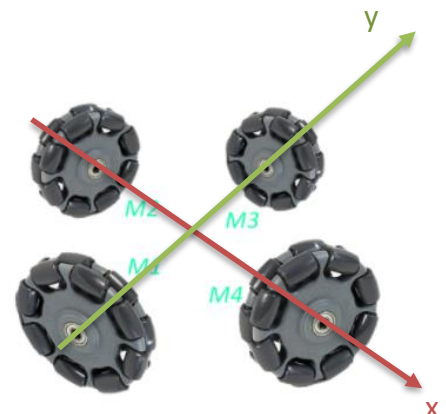
Cet assemblage est présent 4 fois pour les 4 roues du robot. Les modules étant indépendants, cela permet d’avoir un mouvement omnidirectionnel du robot ainsi qu’une plus grande flexibilité de mouvement.

→ **Principe des Roues Holonomes :**

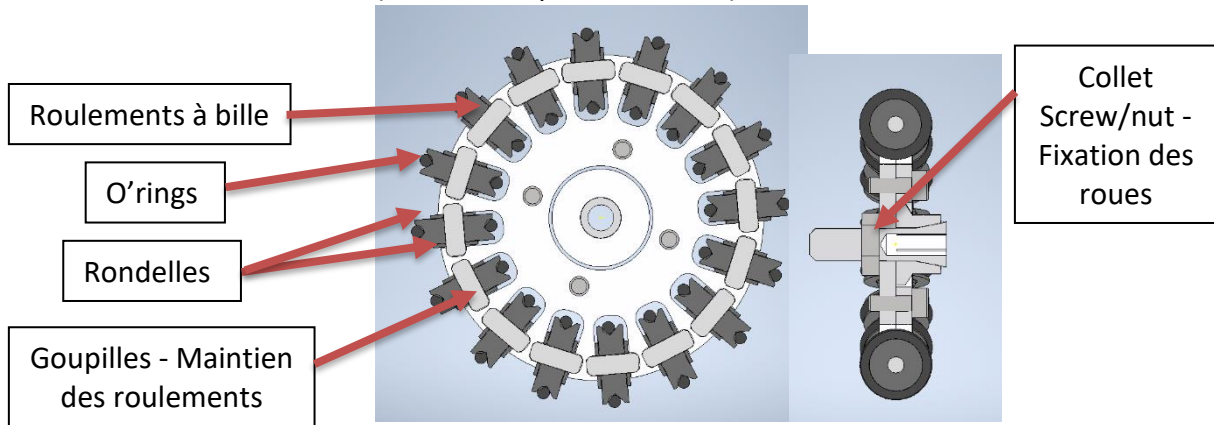
Pour avancer selon l’axe x, on met en marche M1 et M3 qui vont tracter le robot alors que M2 et M4 restent à l’arrêt. Les roulements présents perpendiculairement à la surface des roues permettent un glissement sans frottement.

Pour tourner sur lui-même, toutes les roues seront actionnées.

Pour aller dans la direction formant un angle de 45° avec x et y, toutes les roues seront en mouvement également.



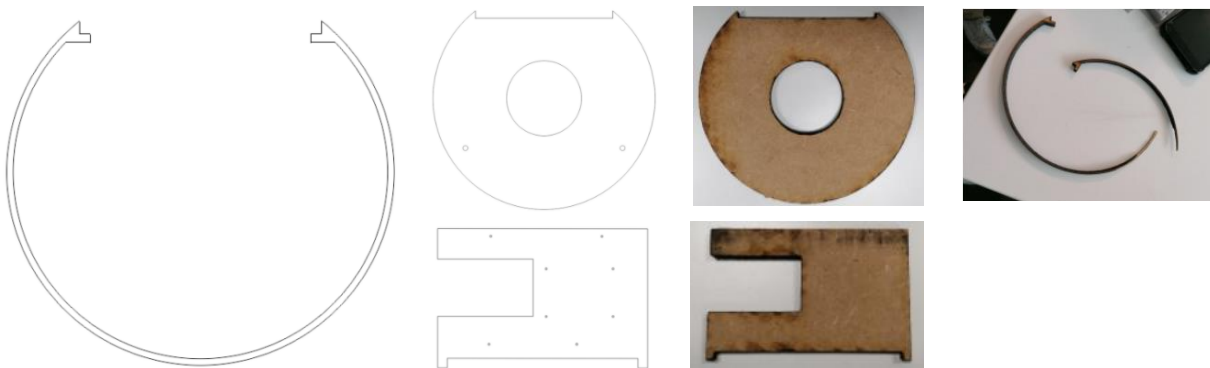
→ **Constitution des Roues :** (vues en coupe face/latéral)



→ **Conception des pièces et usinage :**

Une fois toutes les pièces répertoriées, il ne nous reste plus qu'à les usiner !

• **Découpe Laser (Bois et PEHD) :**



Les premières pièces effectuées (en bois) furent celles servant au maintien du carénage et servant de "toit" au robot.

Tout d'abord, on a dessiné ces pièces sur le logiciel Corel Draw, un logiciel de dessin pour découpe laser. Il a ensuite été possible de les découper sur des plaques de 9mm d'épaisseur au laser CO2 grâce à Jérémie Degorce-Dumas, élève en 2ème année à l'ESME et maîtrisant déjà la machine.

Nous avons eu un problème sur une pièce qui sert à maintenir la forme arrondie du carénage. C'est une pièce très peu solide de base, avec 3mm de largeur sur la surface supérieure du cerceau. Or, pour découper à travers la plaque en bois de 6 mm d'épaisseur, le laser doit passer 10 fois sur la pièce. Après la découpe, elle a été fragilisée alors qu'elle était déjà peu solide et s'est donc rapidement cassée.

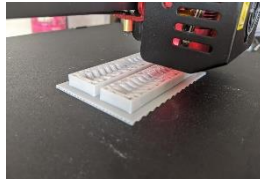
Nous avons donc augmenté la longueur de la surface supérieure pour avoir un cerceau plus solide, capable de supporter le carénage.

En ce qui concerne la découpe du PEHD (PolyEthylène Haute Densité) servant pour le carénage noir, nous avons dessiné la pièce et lancé la commande de la plaque. Dès réception, nous commencerons à usiner.

• Impression 3D PLA :

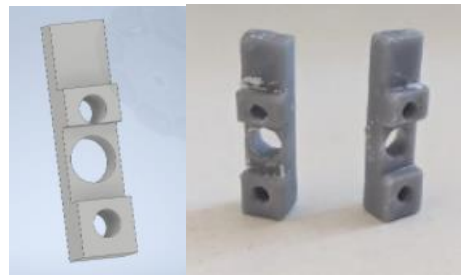
Les premières pièces que nous avons fabriquées furent celles en PLA :

- Les moules du dribbleur pour ensuite pouvoir couler la résine.
- La pièce "support élastique" qui tient les élastiques permettant le retour du module kicker à sa position d'origine.
- Les supports maintenant les condensateurs qu'on a modifiés pour enlever le système d'attache du carénage ne fonctionnant pas.



• Impression 3D Résine :

À partir du robot NAMEC, nous avons prélevé les mesures d'une toute petite pièce servant à maintenir la LED et le phototransistor. Nous l'avons recréée puis modifiée sous *Inventor* pour enfin la faire imprimer par Impression 3D de résine.



• Découpe CNC :

La découpe CNC en aluminium est sous-traitée par le *Technoshop de Cohabit*, nous leur avons envoyé des pièces à usiner et nous attendons de les recevoir prochainement.

→ Moulage résine (Polyuréthane PU) :

Le moulage en résine va permettre de créer les dribbleurs. Le processus expérimental doit être rigoureusement suivi pour éviter toute déformation du moule ou erreur.



• **Matériel** : 1 moule (male/femelle), 1 axe dribbleur, 6 boulons, du démoulant PU, papier ponce, perceuse, résine, récipient, agitateur en métal, 3 seringues, gants, lunettes, masque.

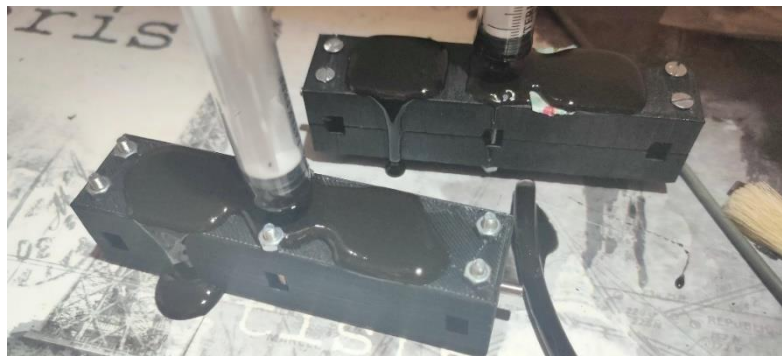
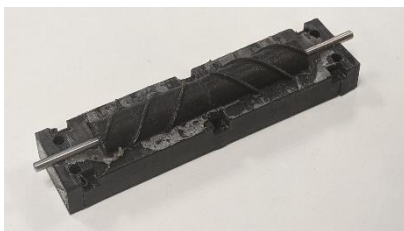
• **Préparation** :

- Ponçage des surfaces de contact entre les 2 moules pour éviter une fuite de produit.
- Perçage si les trous ont été bouchés au cours de l'impression 3D (manque de précision)
- Application d'une à deux couches de démoulant PU et le laisser sécher pour créer une surface qui n'accrochera pas la matière. (faire attention, c'est toxique)
- Vissage des moules l'un contre l'autre sans oublier de positionner l'axe du dribbleur auparavant. (faire attention à mettre les moules dans le bon sens)

• **Résine** :

- La résine est composée de 2 parties : Part A et Part B. Pour le premier test, on met des quantités égales des deux parties. On modifiera ces quantités plus tard en fonction des résultats.
- Utiliser une seringue pour prélever chacune des proportions de Part A/B pour les mettre dans un récipient puis agiter avec une barre métallique le mélange.
- A l'aide de la 3ème seringue, prélever le mélange obtenu et l'injecter dans le moule jusqu'à débordement par les événements. (Attention à ne pas mettre de bulles)

Images du
moulage/démoulage



Echantillon test effectué

• **Améliorations** à faire pour une meilleure efficacité :

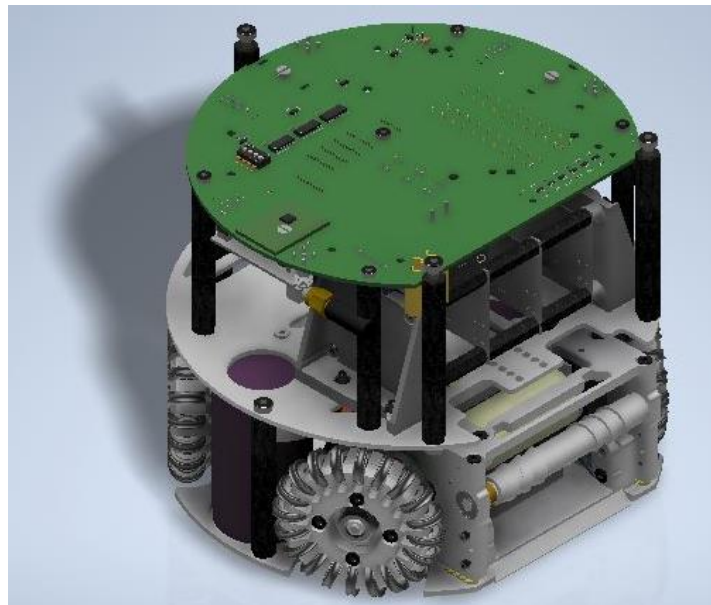
- Rajouter un détrompeur sur les moules pour ne pas se tromper de côté
- Rajouter un emplacement pour la tête de vis
- Modifier la taille des trous avant l'impression 3D
- Ponçage de l'axe

De plus, des tests du robot NAMEC ont débuté à Elektrons Libres grâce à un étudiant effectuant un stage dans l'association ce mois de janvier.

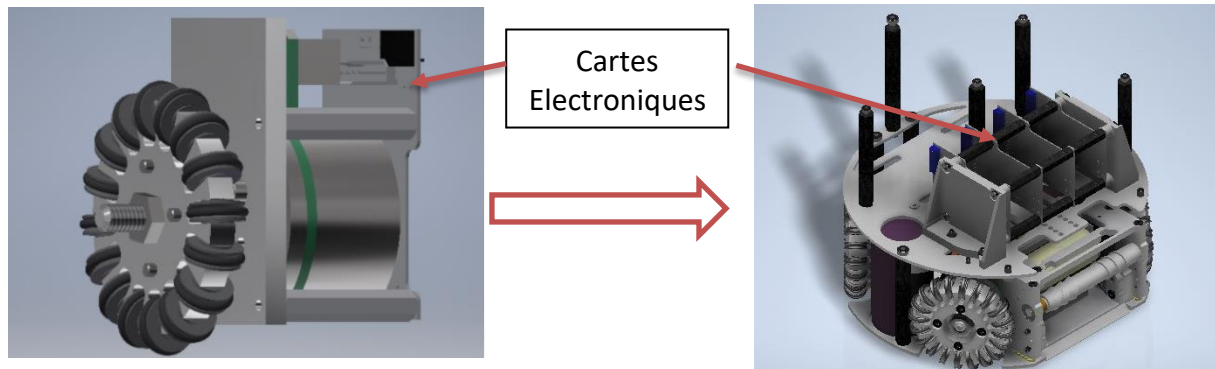
2. Amélioration du robot en v2, inspirée des NAMEC et des Tigers

Le robot que l'on a étudié jusqu'à présent a déjà été en partie testé par l'équipe NAMEC lors de la compétition à Sydney en 2019. Ainsi, ils ont pu nous faire parvenir certains problèmes qu'ils ont rencontrés, ce qui nous a permis d'essayer de chercher des solutions mécaniques pour pallier cela.

Voici la version d'amélioration finale :



→ Déplacement des cartes électroniques moteurs (problème d'interférence électromagnétique, meilleur accès et en vue d'améliorations futures) :



Le premier changement est la localisation des cartes des moteurs qui se trouvaient dans la version NAMEC derrière chaque moteur dans la partie basse en les regroupant toutes dans la partie haute du robot.

D'une part cela nous donne un meilleur accès aux cartes : en soulevant la partie avant du carénage nouvellement conçu pour s'ouvrir, on peut flasher les cartes rapidement et avec facilité en cours de compétition.

De même, cela diminue les interférences électromagnétiques, problème soulevé par les NAMEC qu'ils avaient résolu de manière informatique.

Enfin, pour pouvoir flasher encore plus vite les cartes moteur, on a l'idée de les regrouper rassembler 2 à 2, ce qui nous permettra de ne pas avoir à réfléchir de nouveau à l'emplacement des cartes.

→ **Modification du carénage (problème de maintien du carénage et pour faciliter l'accès aux cartes électroniques) :**

Suite au déplacement des cartes électroniques, on devait modifier le carénage pour avoir accès facilement à ces cartes. De même, le problème soulevé par les NAMEC comme quoi leur système de maintien du carénage réalisé par 3 aimants ne marchait pas, cela nous permet de changer cette solution en rajoutant des attaches sous forme de clips bloquants, se débloquent en appuyant dessus (type poignée télescopique de valise)

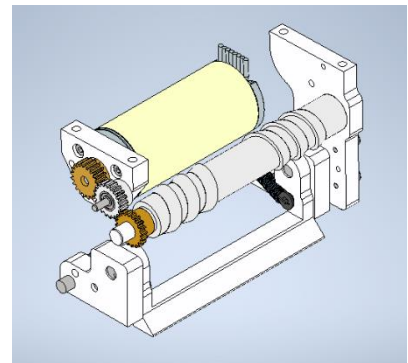


De même, en s'inspirant des Tigers, on a eu l'idée de tester l'impression d'un carénage imprimé en 3D PTGE accompagné d'un ruban autocollant qui a un double rôle : le premier qui relève de l'esthétique et de la publicité pour les sponsors, le second plus mécanique, aidant à la redistribution des forces entre les couches de l'impression 3D. Cette solution est à tester pour approbation.

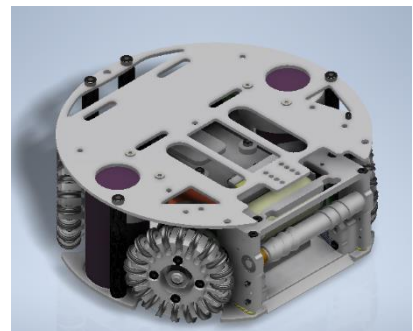
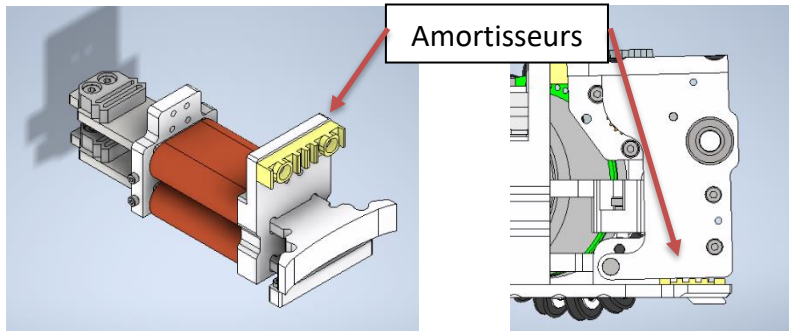
→ **Modifications et tests du dribbleur :**

Le dribbleur des NAMEC est récupéré de celui des Tigers. De même, nous l'améliorons cette année en reprenant la dernière version des Tigers.

Nous avons néanmoins commencé à effectuer des tests d'efficacité en faisant varier le pas du filetage, la présence d'une butée ou pas et le nombre de spires.



→ **Modifications du kicker :**



On souhaite modifier le kicker en utilisant l'expérience des Tigers. Le nouveau est maintenant un peu incurvé, permettant de mieux accompagner le tir. Les bobines sont aussi plus plates et stables.

De même, la rampe de lob est plus petite et plus proche du sol, réduisant le choc lors de l'arrivée de la balle sur la butée et améliorant donc sa réception. En outre, étant plus petite et moins lourde, la puissance du kicker bas est décuplée pour le lob, ce qui le rend plus efficace et donne aux robots un avantage conséquent face aux robots adverses.

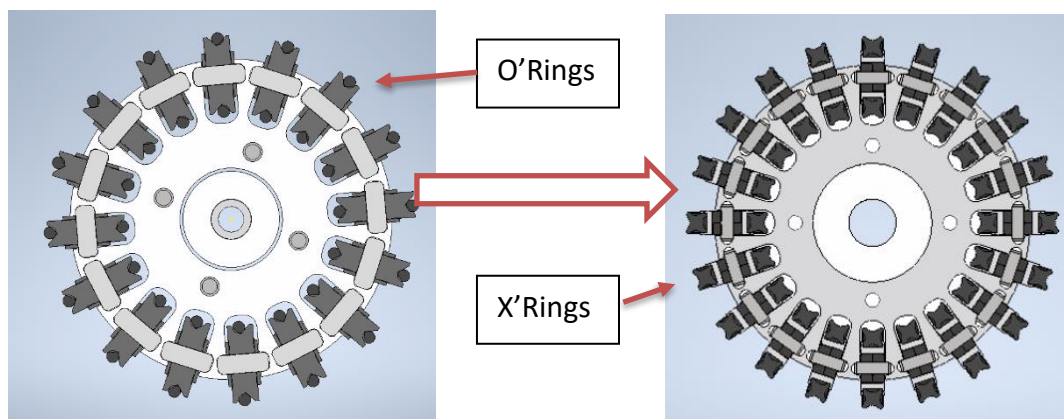
Le réarmement de la rampe est également plus efficace de par la gravité et un système de ressort. Il y a également un rajout de maintiens anti-vibrations au contact de la plaque du bas et au module dribbler.

Ainsi, le nouveau module kicker vise à réduire les vibrations par le biais d'amortisseurs de manière à empêcher un contact choc entre 2 parties de métal en fin de course. Il est également moins encombrant et plus simple d'assemblage car il y a moins de pièces et des bobines moins encombrantes.

Toutes ces modifications sont néanmoins à tester pour prouver leur efficacité.

Enfin, on souhaite remplacer les élastiques ramenant le module kicker à son état initial par un ressort. En effet, un élastique se détend rapidement et nous perdons en précision de puissance. Ainsi, avec un ressort, nous jouerons sur la modification de la raideur pour mettre plus ou moins de puissance dans nos tirs. On pense que cette solution mécanique est plus fiable qu'un résultat informatique qui s'élève finalement approximatif.

→ Modifications des roues :



Nous inspirant des études de l'équipe des Tigers, nous avons décidé de doter les robots de roues 20 X'rings, au lieu des 16 O'rings. La différence vient que les O'rings sont des cercles à bout rond, alors que les X'rings sont des cercles à bouts en croix (X). Cela augmente le nombre de contacts possibles de manière à réduire les vibrations que subit le robot dûes à son déplacement et à améliorer son adhésion au sol.

Ces modifications sont une nouvelle fois à tester.

Toutes nos livraisons sont attendues au début du second semestre. Nous pourrons ainsi assembler un robot et faire les tests prévus.

B. Electronique

Cette année, l'équipe NAMEC nous prête son travail effectué sur les cartes électroniques et le microcontrôleur du robot. Ainsi, notre équipe a directement commandé ces cartes à l'entreprise LeCati qui a conçu les cartes des NAMEC.

Néanmoins, cela n'empêche pas certains membres du partenariat de travailler sur les schémas électroniques, cherchant à comprendre comment cela fonctionne et surtout pour effectuer les modèles mathématiques du robot.

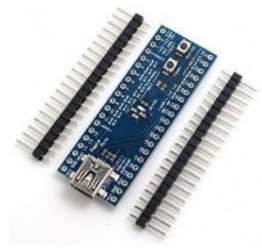
C. Informatique Embarquée

Trois élèves hors parcours innovation se sont positionnés sur l'informatique embarquée. Comme le travail en mécanique a été un peu stoppé (en attente de réception composants), certains élèves les ont rejoints.

Les cours d'informatique embarquée n'ayant débuté que le mardi 15 décembre, à ce jour seulement 2 cours ont été donnés. L'équipe n'a pas pu encore beaucoup avancer dans ce domaine.

Nous utilisons le microcontrôleur Maple Mini qui est l'une des premières cartes de microcontrôleur de la série des ARM Cortex-M3. Sa bibliothèque n'est plus renouvelée ni mise à jour, mais de nombreuses entreprises et particuliers (notamment en Chine) l'utilisent encore pour sa polyvalence, sa petite taille et sa puissance d'STM32.

Ainsi, NAMEC fait de même en maintenant à jour les bibliothèques de la carte.



Nous utilisons l'IDE VisualStudioCode, avec l'extension platformio, un environnement de développement qui peut se substituer à l'IDE Arduino traditionnel. Puis grâce à notre professeur d'informatique embarquée, on peut utiliser mbed sur platformio ce qui nous permet de flasher et compiler nos codes sans difficulté.

De plus, nous codons en C.

Voici un exemple de code faisant clignoter la LED présente sur la carte Maple Mini et affichant le message "Bonjour la Team ESME !" dans le moniteur série.

Pour afficher le texte dans le moniteur série, on utilise Putty, un émulateur de terminal.

```
main.cpp • dfu_maple.py test.txt
src > main.cpp > ...
1  #include <USBSerial.h>
2
3
4  DigitalOut led(PB_1);           // PB_1 = pin 33
5  DigitalOut usbDisconnect(PB_9); // PB_9 = USB
6
7  int main()
8  {
9      USBSerial usbSerial;
10     usbDisconnect = 0;
11
12     while (true) {
13         usbSerial.printf("Bonjour la Team ESME !\r\n");
14         led = 0;
15         ThisThread::sleep_for(1000);
16         led = 1;
17         ThisThread::sleep_for(1000);
18     }
19 }
20
```

D. Software

1. Ce qu'on a fait

Pendant de nombreuses semaines, la partie Software a consisté en la prise en main du logiciel de simulation grSim, permettant de simuler une partie de football SSL.

Au début, l'objectif était seulement de déplacer le robot, c'est-à-dire le faire bouger selon une direction ou le déplacer à un point précis sur le terrain, ce qui était compliqué !

Dès le début, nous avons été confrontés à un problème : il existait 2 repères sur le terrain :

- Le repère t correspondant au repère du terrain, il est fixe à chaque instant et l'origine est le centre du terrain.
- Le repère r correspondant au repère du robot, celui-ci n'est pas fixe à chaque instant et bouge en fonction de l'orientation du robot (que l'on notera θ). Il est constitué d'un axe tangent et d'un axe normal.

Ainsi, pour exprimer une direction dans le repère r , il sera nécessaire d'utiliser une matrice de rotation.

Notons les éléments suivants :

- \vec{v}_t : Coordonnées du vecteur vitesse exprimé dans le repère t .
- \vec{v}_r : Coordonnées du vecteur vitesse exprimé dans le repère r .
- $R_{r/t}$: Matrice de rotation dans le repère r au repère t .

$$\vec{v}_r = R_{r/t} * \vec{v}_t = R_{-\theta} * \vec{v}_t$$

Alors :

$$R_{-\theta} = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}$$

Par la suite, nous avons mis en place une succession d'actions coordonnées : plusieurs robots effectuant des actions en simultanée.

De plus, nous avons appris à déplacer le robot selon les coordonnées d'un autre robot ou de la balle.

Puis plus tard, nous avons appris à le faire tirer dans la balle.

En parallèle, on a développé des formations qui nous servirons (ou pas) lors d'un match.

Et enfin, dernièrement, nous avons appris à faire bouger un robot selon un tableau de points. En déclarant un tableau de coordonnées (avec une position en x et en y), le robot parcourt ainsi les points successifs.

Vous pouvez visionner cette vidéo résumant tout cela : <https://youtu.be/NGoJvpRjY5M>

2. Ce qu'on a prévu de faire

Depuis quelque temps, nous nous consacrons au **pathfinding** ou « recherche de chemin » en français pour l'évitement d'obstacle. Car, pour l'instant le robot se déplace d'un point de départ à un point d'arrivée, en ligne droite sans faire attention aux autres robots autour de lui. Le but, est ainsi de créer un algorithme qui élabore la trajectoire du robot en prenant en compte la position des autres robots et de la balle en temps réel.

Pour ce faire, le terrain sera vu en forme de grille (ou tableau) composé de colonnes et de lignes. Puis, nous allons rajouter dans cette grille les obstacles, le point d'arrivée et le point de départ.

Enfin, l'algorithme va se charger de trouver un chemin sans passer par les obstacles. Il en existe aujourd'hui déjà beaucoup d'existants (A*, RRT Connect...) mais il est nécessaire pour nous de les comprendre.

Pour la suite, nous avons prévu plusieurs choses :

- Continuer et approfondir l'algorithme d'évitement d'obstacle.
- Mettre en place un robot "gardien", c'est-à-dire un robot qui s'occupera de défendre les buts. On a notamment comme idée d'orienter la position du gardien en fonction de la position et de l'orientation de l'attaquant.
- Envisager la possibilité d'un tir en courbe, mais un problème va sûrement se poser au niveau de la mécanique du robot. En effet, pour l'instant le kicker du robot ne peut pas pivoter et peut seulement tirer droit. On travaillera avec les étudiants du pôle mécanique pour mettre en place cette option mais se sera sûrement un projet à plus long terme.
- A moyen terme, le développement d'une stratégie, avec la mise en place de passes entre les robots et l'autonomie complète des robots afin qu'il puisse jouer un vrai match de foot par eux-même.
- Mise en place d'un filtre afin de filtrer les données des robots. En effet, lors de la RoboCup, il y a une barre de caméras centrales permettant aux robots de recevoir les données des différents robots (position, orientation, vitesse...). Et les données transmises ont des parasites qu'il faut filtrer.

→ Comment fonctionne un filtre ? Pourquoi en utiliser un ?

Pour faire simple, les données de vision réelle ou de grSim (position des robots) peuvent avoir du bruit. Il faut donc prévoir un moyen de recevoir des données cohérentes.

Exemple :

S'il y a une balle qui est capté en $x = -3, y = -1$ et que la vision capte une balle en $x = 2, y = 2$. Imaginez l'accélération que va avoir la balle.

Voilà pourquoi on a besoin de filtre.

Dans notre cas, nous récupérons en entrée les valeurs (à savoir la position, la vitesse, et l'orientation des robots entres autres) et modifions/supprimons ces dernières en fonction du filtre puis nous retournons ses valeurs.

On utilisera principalement deux filtres :

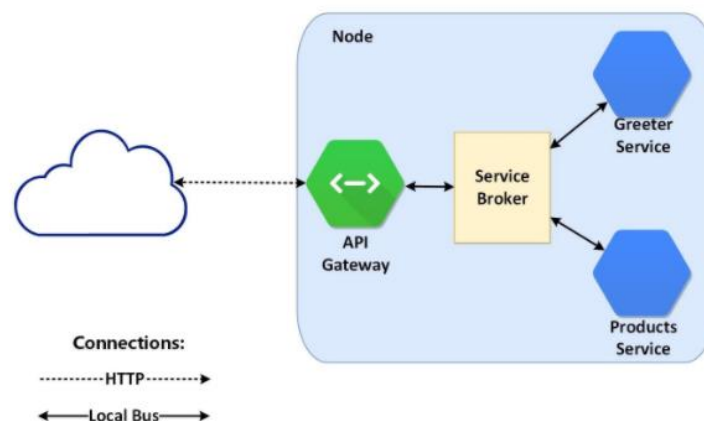
- **Le Filtre de seuillage de confiance** : Si le taux de confiance d'une donnée de la vision est inférieur à une valeur minimale, alors on supprime cette donnée.
- **Le Masque** : Masque certaines données comme une moitié de terrain. Cela sera utile lors des entraînements les jours de compétition. Chaque équipe se voit remettre un seul côté de terrain. Un filtre est alors nécessaire pour filtrer la partie adverse et éviter que nos robots empiètent la partie adverse.

3. Comment on le fait

L'architecture est la forme générale d'un logiciel, elle doit être adaptée à l'utilisation du logiciel ainsi qu'à la méthode de gestion de celui-ci. Certaines architectures sont conçues pour l'efficacité en essayant de minimiser les calculs pour gagner en vitesse tandis que d'autres sont conçues pour leur modularité ou encore leur simplicité de programmation.

Nous utilisons **NodeJS** (une plateforme logiciel de Javascript), et en particulier le framework Moleculer. En réalité, nous utilisons le langage Typescript qui est, pour faire simple, un sur-ensemble (syntaxique), de Javascript.

Moleculer :



En quelques mots, **Moleculer** est un Framework (ensemble d'outils et de composants logiciels à la base d'un logiciel ou d'une application) open source de micro-services pour Node.js.

Un service est un simple module JavaScript contenant une partie d'une application complexe. Il est isolé et autonome, ce qui signifie que même s'il se met hors ligne ou tombe en panne, les autres services ne seront pas affectés.

Un nœud est un processus simple du système d'exploitation fonctionnant sur un réseau local ou externe. Une seule instance d'un nœud peut héberger un ou plusieurs services.

Service Broker est le cœur de Moleculer. Il est responsable de la gestion et de la communication entre les services (locaux et distants). Chaque nœud doit disposer d'une instance de Service Broker.

Une passerelle API est un élément (de programmation) qui orchestre et coordonne le traitement des différentes requêtes dans une architecture de micro-services.

Cette passerelle est un service faisant tourner un serveur (HTTP...) Elle traite les demandes entrantes, puis renvoie les réponses appropriées.

→ Pourquoi avoir choisi le Framework Moleculer (et Typescript) pour développer les robots ? Quels sont ses avantages et ses inconvénients par rapport à d'autres langages tel que le C++ ?

Ce qu'il faut savoir c'est que, à la création de la compétition SSL, les enseignants qui créèrent les premiers robots participaient déjà à une autre ligue de la Robocup. Et dans cette ligue, ils utilisaient le C++. C'est donc pour cette raison que, historiquement, le C++ a été choisi.

Avant la RoboCup de Sydney, en 2019, l'équipe NAMEC est partie du code publié en open source par le comité de la RoboCup mais a vite été confrontée à un problème.

En effet, l'utilisation du C++ est complexe et il est facile de faire des erreurs, ce qui peut prendre beaucoup de temps pour programmer. C'est ainsi qu'un programmeur de l'équipe NaMec a décidé de commencer à utiliser le Framework NodeJS puis cette idée a été reprise par Etienne, notre leader, qui a finalement décidé de nous l'enseigner pour la Robocup.

Toutefois, le C++ a un avantage indéniable par rapport à Node.js : il est extrêmement performant. Mais, en réalité, de nos jours, avec nos ordinateurs, un programme performant ne changera rien le jour du match (ou du moins très peu). Et, au final, il faut plutôt avoir un choix ratio temps développement/performance dans ce cas-là. Evidemment, Node.js est clairement vainqueur sur le C++ là-dessus.

L'approche micro-service est aussi intéressante. Elle permet de créer des nœuds de langage différent si on le souhaite. Donc, si on observe des soucis de performance avec Node.JS, il suffit de séparer le travail en plusieurs sous-nœuds ou tout simplement de prendre un langage plus performant comme le GO.

Avec cette approche on peut également se permettre de connecter plusieurs ordinateurs ensemble, pour qu'il se partage le travail (un ordinateur tourne network et bots-gateway et un autre game-data, msb, bots-control).

On est moins limité que l'approche monolithique du C++ (où le projet n'est traité qu'à partir d'un seul ordinateur).

Il y a **six nœuds** qui cohabitent entre eux, chacun étant spécialisé dans un domaine. Ces nœuds communiquent entre eux et avec l'extérieur via un protocole.

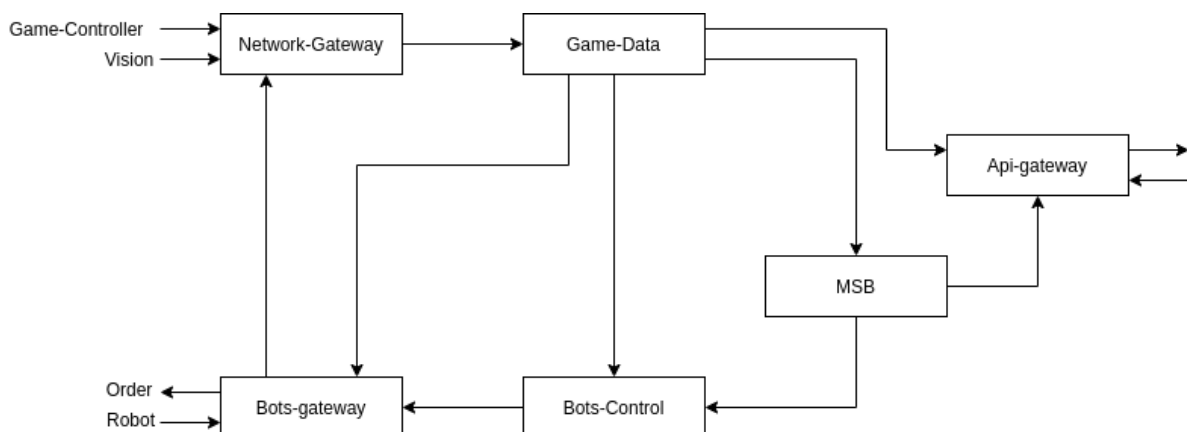


Diagramme représentant les différents nœuds du système ainsi que leurs communications.

Voici un récapitulatif des principaux nœuds et entrées sorties du système :

- **Le Game-Controller** est en quelque sorte l'arbitre du match, il indique les moments importants du match.
- **La vision** est la position des différents éléments sur le terrain, il est envoyé sous le format protobuf, spécialisé dans la communication entre les programmes.
- **Le nœud Network-Gateway** reçoit les données et les stocke.
- **Le nœud Game-Data** est un filtre permettant de filtrer les données brutes du Network-Gateway.
- **MSB** est un service permettant l'élaboration de stratégies pour le contrôle des robots.
- **L'API-Gateway** est une passerelle API.
- **Bots-control** permet de transformer les ordres du MSB pour chaque robot en instructions compréhensibles par les robots (vitesse angulaire, normale et tangente, frappe de kicker...).
Par exemple le service MoveTo en fait partie.
- **Bots-Gateway** envoie les instructions aux robots et reçoit leurs données pour les transmettre au Network-Gateway.

→ Pourquoi ce Framework plutôt qu'un autre ?

Il y en a d'autres sous Node.js mais celui-ci est plus complet et est très stable.

→ Pourquoi avoir choisi du parcours d'arbre (avec des conditions) plutôt que le l'IA par exemple ?

Historiquement, NAMEC utilise ça. L'approche d'arbre nous permet en revanche de savoir ce qu'on fait. L'IA étant plus "opaque", c'est-à-dire que l'on ne sait pas forcément ce que fait le robot contrairement à une approche d'arbre avec des conditions (if, else...) où l'on sait précisément pourquoi le robot fait cela.

Néanmoins, certains élèves de l'équipe en école d'ingénieur d'informatique souhaitent travailler sur du machine Learning.

III/- Conclusion

Ce projet nous a apporté beaucoup de compétences, à la fois techniques et humaines.

Techniquement, les élèves du pôle mécanique ont appris à modéliser en 3D sur Inventor, à dessiner des pièces sur Corel Draw, à utiliser les machines présentes au fablab (imprimantes 3D et découpeuse laser). Ils ont également appris à s'organiser et chercher des composants de manière efficace sur des sites professionnels (maxon motor, misumi,...).

De même, les élèves travaillant sur l'informatique ont appris le fonctionnement de l'architecture micro-service et ont compris ses avantages et ses inconvénients. Ils manipulent également les langages Javascript et Typescript qu'ils acquièrent au fur et à mesure.

Humainement, cette expérience nous apporte beaucoup. Que ce soit en travail d'équipe, organisation, respect de délais, mais également en menant à bien ce projet concret en ayant pour but final la participation à la Robocup à Bordeaux en juin !

→ Remerciements

Nous souhaitons remercier l'association Elektrons Libres, qui nous a permis de participer à ce projet.

Nous remercierons de même son partenaire, Cohabit ainsi que le club innovation nous permettant chaque semaine d'apprendre et innover autour de la Robocup.

Nos remerciements vont également à notre école, l'ESME Sudria Bordeaux, son directeur M.Crouzil ainsi que le responsable du FabLab M.Guillemard qui nous soutiennent dans ce projet complexe depuis le début.

Un grand merci à Patrick Félix et surtout à Etienne Schmitz, leader de l'équipe entière qui se démène chaque jour pour coordonner et satisfaire les besoins de chaque pôle, que ce soit en terme de commandes, dossiers administratifs ou problèmes informatiques et qui trouve également le temps de donner des cours de Software et de mener des conférences autour de la robotique !



IV/- Annexes

- Un lien Google Drive donnant accès aux fichiers suivant :
 - Résumé en français des règles de la SSL
 - Notice d'installation de grSim et d'un dual boot Ubuntu

https://drive.google.com/drive/folders/1sA1x16VK1wc-dBBk1QlpG-cocq1k_cEH?usp=sharing

- Sources :

<https://elektronlibres.fr/>

<https://www.iut.u-bordeaux.fr/cohabit/>

<https://ssl.robocup.org/>

<https://ssl.robocup.org/rules/>

<https://namec.fr/>

<https://www.tigers-mannheim.de/index.php?id=14>

- Quelques Images :

