PROJET DE ROBOTIQUE

ROBOT SERVEUR

Membres du groupe : Héloïse LECOCQ
Thaïs LE GALLIC
Chef de projet : Louis PAUCHET

Enseignants : Frederic MARSEILLE & David ARROUCH

Année Scolaire : 2021 - 2022

INSA ROUEN NORMANDIE

INSA Rouen Normandie
Département Mécanique
685 Av. de l'Université
76800 Saint-Étienne-du-Rouvray

1 REMERCIEMENTS 3

2 INTRODUCTION 4

3 CHOIX DU ROBOT 5

3.1 ÉTAT DE L'ART 5

3.1.1 Le robot suiveur de ligne 5

3.1.2 Le robot suiveur de personnes (ou Robot Sherpa) 6

4 CAHIER DES CHARGES 7

4.1 EXPRESSION DU BESOIN 7

4.2 DECOMPOSITIONS EN FONCTIONS ELEMENTAIRES 8

4.3 LISTE DES FONCTIONNALITES DU ROBOTS 9

5 ORGANISATION 10

5.1 PLANNING DES OPERATIONS (DIAGRAMME DE GANTT) 10

5.2 ORGANISATION DES TACHES 11

6 CONCEPTION DU ROBOT 12

6.1 CONCEPTION MECANIQUE 12

6.1.1 Conception du châssis 13

6.1.2 Conception du monte-charge 14

6.1.3 Stabilité du robot 14

6.2 LISTE DU MATERIEL 16

6.3 CONCEPTION ELECTROTECHNIQUE 17

6.3.1 Objectif algorithmique du robot 17

6.3.2 Présentation des capteurs & détecteurs 17

6.3.3 Schéma du branchement 20

6.3.4 Gestion de l'alimentation 21

6.3.5 Programme Arduino 22

7 CONCLUSION 24

ANNEXE 1 ETUDE DE FAISABILITE DU ROBOT 29

ANNEXE 2 ETUDE DE STABILITE STATIQUE DU ROBOT 31

ANNEXE 3 ETUDE DE STABILITE DYNAMIQUE 32

ANNEXE 4 SUIVI DU PROJET 35

ANNEXE 5 PLANS CHASSIS 41

ANNEXE 6 SCHEMA ELECTRIQUE DU ROBOT 46

ANNEXE 7 CODE DU PROGRAMME FONCTIONNEL ET TESTE 47

ANNEXE 8 CODE THEORIQUE POUR LES TROIS CAPTEURS (NON TESTE) 50

ANNEXE 9 QUELQUES PHOTOS DE LA FABRICATION 54

1 REMERCIEMENTS

Tout d'abord nous tenons à remercier Mr Marseille pour sa bienveillance et ses précieux conseils dans le choix des capteurs mais également les idées de conception de notre robot.

Nous remercions également vivement Helene Rade et Pascal Williams, ainsi que l'équipe du CCEAS, pour l'investissement et le temps précieux que vous nous avez accordé pour répondre à nos questions, ouvrir la salle de projet et réaliser nos pièces sans oublier David Arrouch qui nous a aiguillé dans les possibilités d'impression 3D de nos pièces.

Le châssis de notre robot est conçu en aluminium et nous souhaitons donc remercier chaleureusement David Paresy pour ses conseils, son implication, sa réactivité et sa bienveillance, y compris pour nos demandes farfelues, ainsi que toute l'équipe du Centre Commun d'Usinage pour leur aide.

Enfin, nous tenons à remercier toutes celles et ceux qui nous ont aidés à tester le robot ou encore qui nous ont apporté un œil neuf essentiel dans l'élaboration du code Arduino du robot.

2 INTRODUCTION

Le projet robot est une matière proposée en MECA 3 d'une durée de 3h toutes les 2 semaines durant le semestre 6. Ainsi il nous est demandé de concevoir mécaniquement et algorithmiquement un robot. Ce projet nous demande une grande organisation au sein de notre groupe afin de lui donner vie. N'ayant que peu de compétence en langage Arduino ainsi qu'en conception mécanique, il s'agit bien d'un défi à relever en seulement 6 séances. Ce projet fait appel à notre imagination et notre curiosité. Nous avons décidé de concevoir un robot original : Un robot serveur.

Ce robot présente de nombreux défis techniques mais aussi algorithmique par sa présence de nombreux capteurs. Il s'agit d'un robot qui nous tient beaucoup à cœur par son aspect sympathique est ses dimensions impressionnantes. Ce robot est capable de suivre un individu muni d'un émetteur infrarouge et, par l'intermédiaire d'un monte-charge, apporter un objet.

Au travers de ce dossier, nous souhaitons vous faire partager comment nous avons mis en œuvre nos connaissances afin de réaliser ce robot. Dans un premier temps nous aborderons nos réflexions sur les fonctionnalités du robot serveur (avec un état de l'art et l'élaboration d'un cahier des charges), nous parlerons ensuite de la conception mécanique et l'utilisation de CATIA V5 pour nos différentes pièces puis de la conception algorithmique et l'étude des cartes Arduino.

Figure 1 : Illustration - Photo du robot assemblée

3 CHOIX DU ROBOT

3.1 État de l'art

Dans un premier temps, le choix du robot s'est porté essentiellement sur une recherche approfondie des différents robots existants dans le milieu de l'entreprise. Nous souhaitions en effet créer un robot utile. Un robot qui aide l'Homme dans son quotidien. Notre choix s'est donc orienté vers les robots qui interagissent avec l'utilisateur. Dans notre réflexion, nous avons tous les trois étés séduits par les robots à l'image de BB-8 ou R2-D2 de la saga Star Wars et nous nous sommes dit que ça pourrait être intéressant de concevoir un robot qui suivrait son « maître ». Enfin, s'il ne sait que suivre cela ne serait pas forcément intéressant nous avons donc réfléchi à concevoir un robot capable de présenter un objet sur un plateau.

Figure 2 : Illustration R2D2 (4)

Figure 3 : Illustration BB-8 (5)

Ainsi notre robot possède deux fonctions principales :

- Suivre son propriétaire
- Lui apporter un objet sur un plateau lorsque son propriétaire est devant lui

Notons que nous avons beaucoup aimé la notion de sphère dans le robot BB-8. Nous avons donc conçu notre robot avec deux demi-sphères.

Nous avons entrepris des recherches sur internet pour nous inspirer des robots suiveurs.

3.1.1 Le robot suiveur de ligne

Classique des projets robots menés à bien à l'INSA, le robot suiveur de ligne a de nombreuses applications industrielles notamment logistiques.

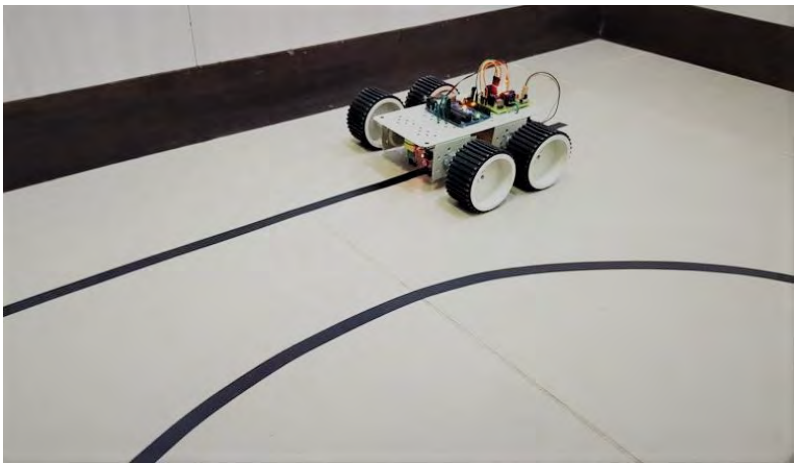


Figure 4 : Robot suiveur de ligne (1)

3.1.2 Le robot suiveur de personnes (ou Robot Sherpa)

Plusieurs sociétés ont mis au point des robots permettant d'assister les personnes devant faire de la manutention dans les entrepôts ou les facteurs lors de leurs tournées.

Figure 5 : Robot Sherpa

Ainsi la Société NORCAN a développé un robot capable de suivre les opérateurs dans les entrepôts en les affranchissant de transporter des charges lourdes ou de tirer un chariot ce qui cause des traumatismes. Ce robot, en mode « Follow-me » ne nécessite aucun équipement sur l'opérateur. (2)

La Poste française a depuis 2018, mis en place plusieurs robots suiveurs sur le territoire permettant aux facteur.ice.s de réaliser leurs tournées sans avoir à tirer leurs chariots. Ce robot identifie son « maître » grâce à une bague et dispose d'un ensemble de systèmes de sécurités lui permettant d'évoluer dans son environnement sans présenter de risque pour les populations. (3)

Figure 6 : Robot suiveur de La Poste (3)

4 CAHIER DES CHARGES

Le cahier des charges que nous avons proposé regroupe toutes les fonctionnalités que nous souhaitions intégrer pour permettre d'effectuer la mission d'un serveur de verre et de suiveur de personne.

4.1 Expression du besoin

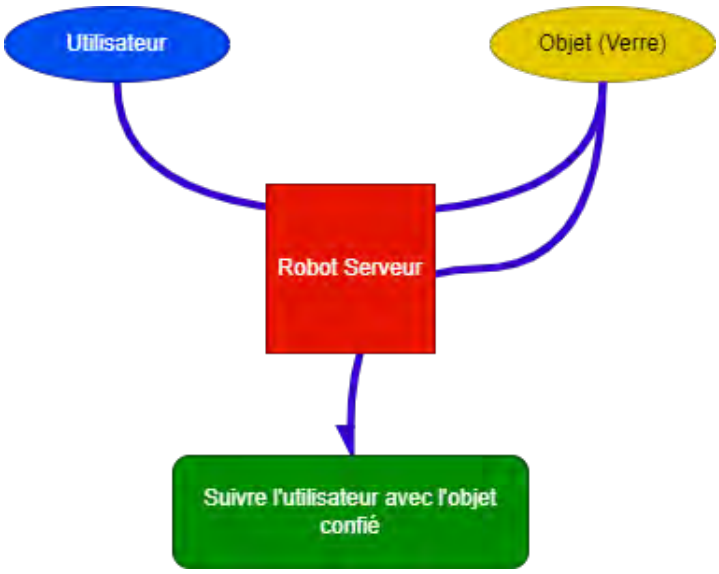


Figure 7 : Diagramme Bête à Cornes

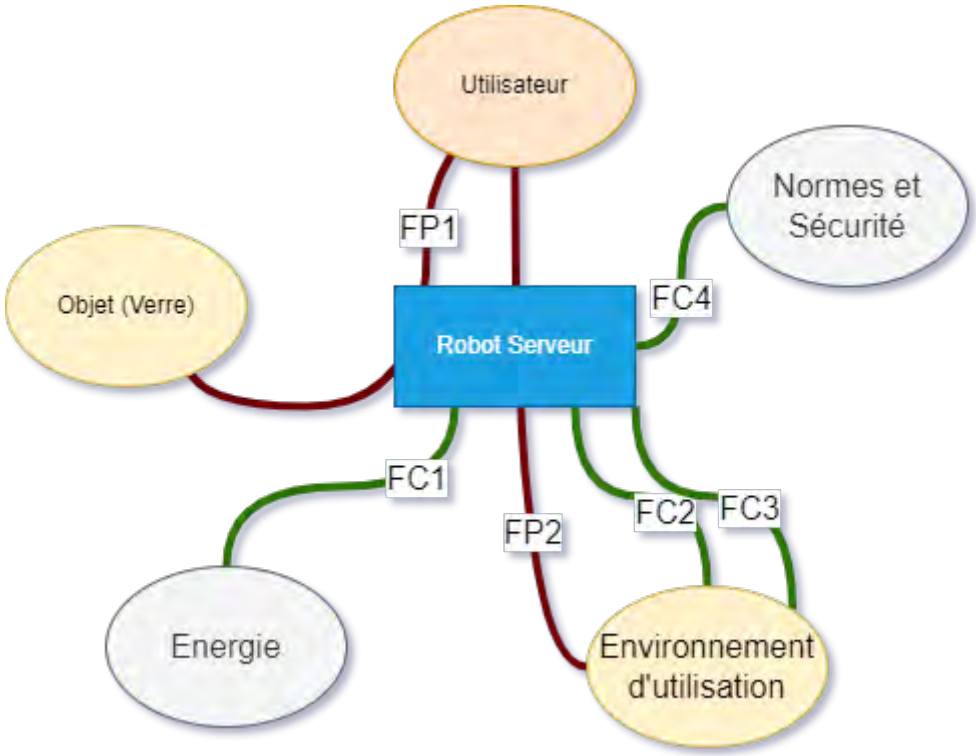
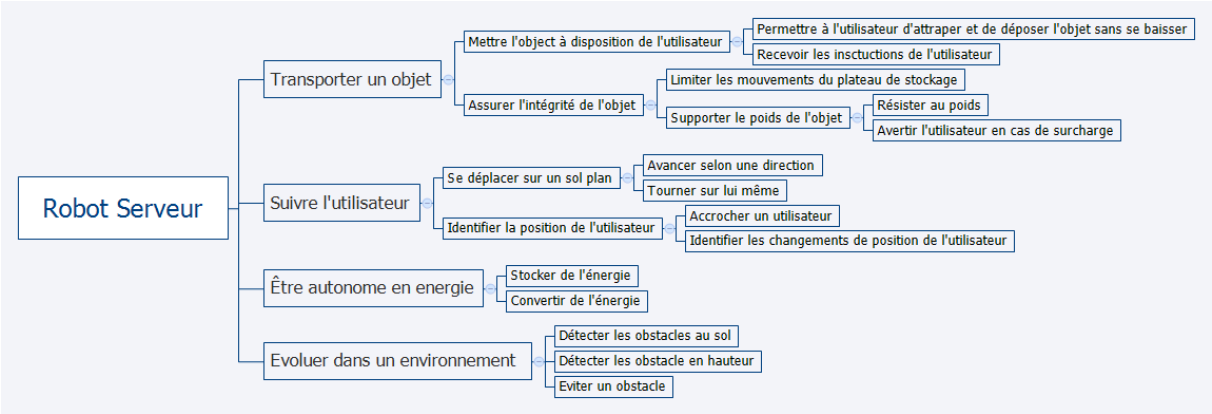


Figure 8 : Diagramme Pieuvre

FP1	**Transporter un objet confié par l'utilisateur et le servir**
FP2	Suivre l'utilisateur dans son environnement
FC1	Être autonome en énergie
FC2	Appréhender son environnement
FC3	Être discret au niveau sonore
FC4	Ne pas présenter de danger pour l'utilisateur, les autres personnes et l'environnement

4.2 Décompositions en fonctions élémentaires



4.3 Liste des fonctionnalités du robots

Les fonctions élémentaires présentés ci-dessus ont été traduite en fonctionnalités implémentés et présentés dans le tableau ci-dessous.

Tableau 1 : Liste des fonctionnalités implémentés

Référence	Description	Solution Technique Proposée
FE1	Mobilité	
FE1-1	Avancer dans une direction	Moteur, poids pour limiter le bascule-ment
FE1-1-1	Avancer à la vitesse d'une personne mar-chant (6,4 km/h)	Moteur
FE1-1-2	Rester immobile	Roulette de stabilité
FE1-2	Tourner sur lui-même	Pilotage des moteurs
FE1-3	Déplacer le plateau verticalement	Poulie, monte-charge, corde
FE2	Perception	
FE2-1	Captation des déplacements de l'utilisateur	Capteur IR
FE2-2	Détection d'obstacles	Capteur IR
FE2-5	Détection fin de course plateau	Détecteur à lame
FE2-6	Actionnement montée plateau	Gros bouton poussoir vert
FE3	Sécurité	
FE3-1-3	Arrêt obstacle	Capteur Sharp
FE3-2	Arrêt d'urgence manuel	Bouton On/Off
FE3-3	Frein de stationnement	Resistance moteur suffisante
FE3-4	Frein déplacement plateau	Resistance moteur suffisante
FE3-7	Maintien distance de sécurtié	Sharp détection d'obstacle
FE3	Resistance	
FE3-1	Support d'un objet de 300g	Plateau, Equerre de positionnement
FE4	Energie	
FE4-1	Stocker de l'énergie	Accu
FE4-1	Convertir et adapter l'énergie	Conditionneurs Moteur et Cartes
FE4-3	Protéger les cartes électroniques	Plastique de protection

5 ORGANISATION

5.1 Planning des opérations (diagramme de Gantt)

Nous avons établi plusieurs diagrammes de Gant et nous nous sommes adaptés aux aléas de la conception :

Figure 9 : Diagramme de Gantt prévisionnel

Figure 10 : Diagramme de Gantt réalisé

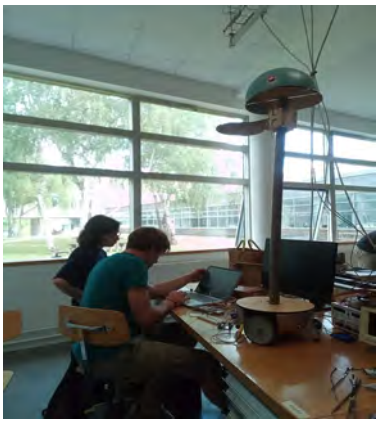


Figure 11 : Photo de l'équipe CAO à pied d'œuvre

Nous avons modifié de nombreuses fois notre diagramme de Gant. En effet, après de nombreuses heures sur nos pièces CATIA nous avons appris que le temps d'impression était beaucoup trop long. Il nous a fallu reconcevoir et repenser nos pièces en utilisant la découpeuse laser et la découpe jet d'eau.

Malgré les nombreux imprévus, nous avons finalement réussi à concevoir nos pièces dans les temps.

5.2 Organisation des tâches

Nous avons décidé de nous répartir les tâches afin d'être plus efficace. Ainsi, nous nous sommes répartis selon nos goûts et nos compétences.

Tableau 2 : Répartition des rôles de conception

	Conception CAO	**Conception CAO**	**Conception informatique**
Nom	Thaïs	Louis	Héloïse

La conception CAO comporte une grande partie d'élaboration des modèles sur CATIA mais également toute une partie « bricolage » avec notamment l'utilisation de la perceuse à colonne, poste à souder, pistolet à colle…

La conception Informatique comporte bien évidemment le codage du robot sous Arduino mais également le test des différents capteurs et leur compréhension à travers la lecture des documents techniques.

Tableau 3 : Répartition des pièces de CAO

Thaïs	**Louis**
Monte-Charge	Colonne du châssis
Plateau	Plaque du châssis
Yeux du robot	Maintien des moteurs
Roulette du monte-charge	Treuille
Maintien du cou du robot	Mise en place des roulettes
Équerres	Découpe sphères et position des roues
	Mise en place des capteurs et bouton

D'autre part Louis s'est assuré de la soudure des composants et du rangement des fils nous permettant de ne pas nous perdre dans les branchements.

Lorsque la conception mécanique a été terminé, Thaïs et Louis se sont concentrés sur le code Arduino qui prenait plus de temps. Héloïse a contribué également à l'assemblage du robot. Ainsi, le travail a été répartie de différentes façon au cours du semestre. Nous nous sommes aidés les uns avec les autres pour avancer plus rapidement. Cela nous a permis également de toucher un peu à tout et d'apprendre de nouvelle notion en informatique tout comme en mécanique.

6 CONCEPTION DU ROBOT

6.1 Conception mécanique

Notre robot est composé en 3 parties distinctes :

- Une demi-sphère en bas du robot composé du châssis et des roues ;
- Un tube central en PVC sur lequel coulisse un plateau ;
- Une demi-sphère en haut composant la tête et les yeux du robot ;

Dans un premier temps nous avons commencé par dessiner tout simplement sur un tableau blanc, une feuille de papier la forme générale de notre robot.

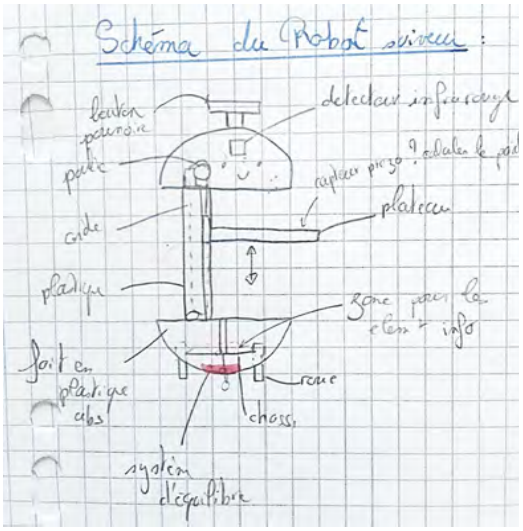


Figure 12 : Première esquisse du robot

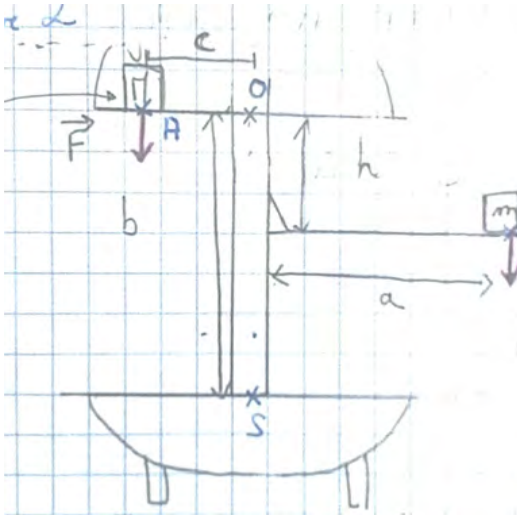


Figure 13 : Étude du positionnement du cou du robot

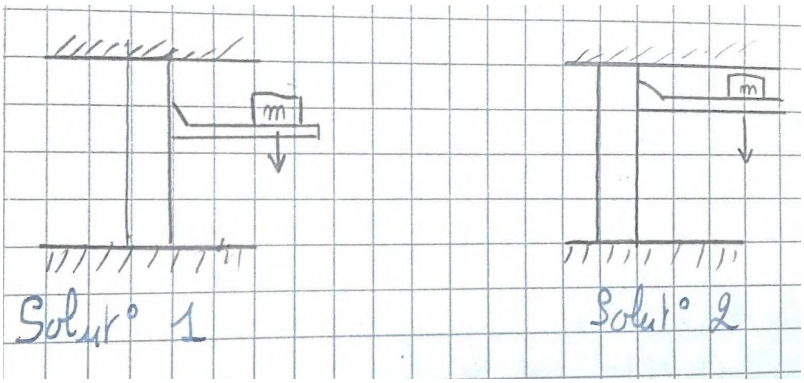


Figure 14 : Deux principales solutions de conception étudiés

Les deux solutions envisagées sur la Figure 14 diffèrent par le positionnement du tube par rapport au robot. Le premier cas présente un tube centré qui pourrait, par l'excentration du plateau, un déséquilibre qui serait compensé par l'excentration du tube présenté en solution n°2. L'étude statique présenté en ANNEXE 2 a montré que ce paramètre a relativement peu d'influence.

6.1.1 Conception du châssis

Pour concevoir le châssis, nous sommes partis de l'ensemble des composants à y installer pour finalement obtenir un châssis permettant de tout maintenir en place.

Avec la complexité de la demi-sphère comme support pour l'extérieur du bas du robot, nous sommes partis dans un premier temps sur un châssis imprimé en 3D. Malheureusement, il s'est avéré que le temps d'impression était bien trop important pour les capacités d'impressions de l'INSA et surtout au vu des problématiques de risque d'incendie lors d'impressions la nuit (1 jour et 10h estimé par le logiciel CURA sur une imprimante Ultimaker S5).

Un châssis en bois découpé au laser semblait un peu léger au vu de la charge que devra supporter le robot, ainsi nous nous sommes tournés vers le CCU et David Paressy a qui nous avons confié la fabrication des pièces que nous avions dessinées pour être réalisées en aluminium.

Le châssis est à présent composé d'une plaque de maintien dans laquelle sont fixés deux supports de moteurs bas un pour chaque roue, un support de moteur haut pour fixer le tambour du treuil et quatre colonnes permettant le maintien de la structure permettant qui supporte la colonne principale du robot.

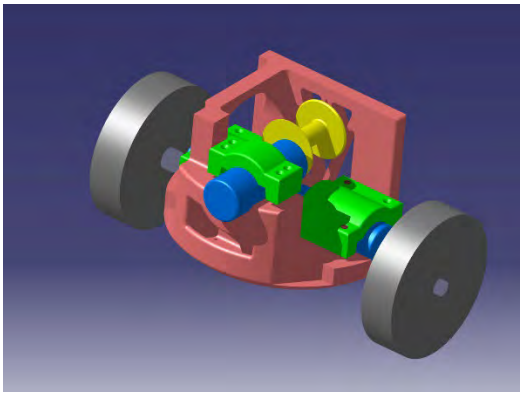


Figure 15 : Assemblage CATIA du châssis version 1

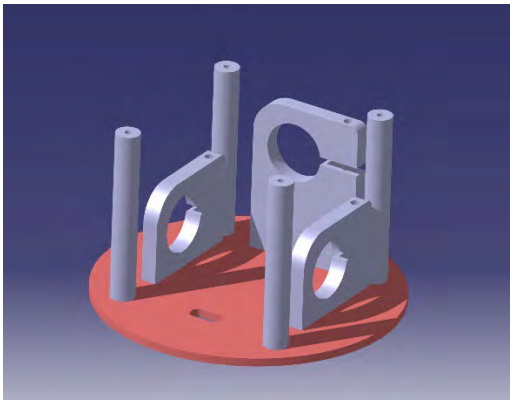


Figure 16 : Assemblage CATIA du châssis final

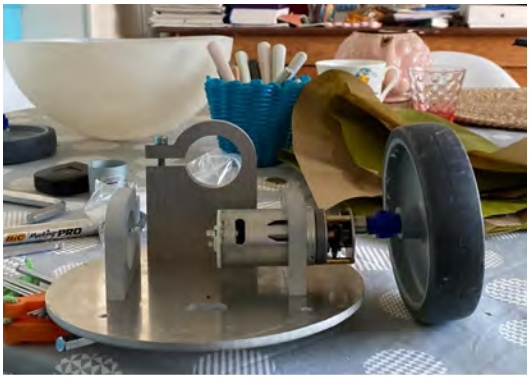


Figure 17 : Début du montage des éléments du châssis

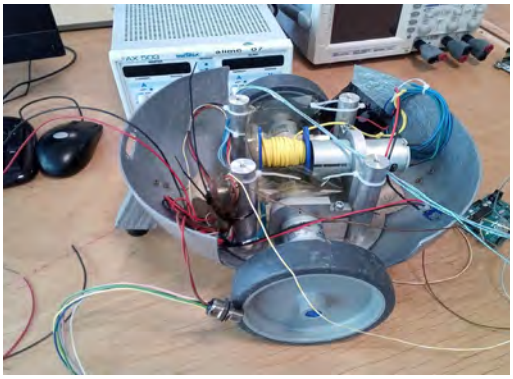


Figure 18 : Châssis assemblé avec tous les éléments

6.1.2 Conception du monte-charge

Nous avons choisi de faire un robot à taille humaine. Ainsi, le « cou » de notre robot fait 1mètre. Nous avons choisi un diamètre de 40mm afin de pouvoir imprimer notre monte-charge en 3D.

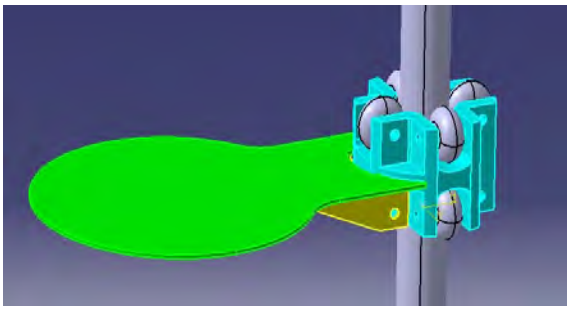


Figure 19 : Conception de la fonction service du robot

Figure 20 : Monte-charge imprimé et assemblé

Pour le monte-charge nous avons fait un système mécanique capable de soulever une masse au bout d'un plateau. Le monte-charge est composé de 4 entretoises reliant nos supports de roues. Afin d'éviter le basculement de notre système sous l'effet de la masse, nous avons mis en place un centrage court avec une hauteur du monte-charge de 80mm pour un diamètre intérieur de 64mm.

Figure 21 : CAO du monte-charge

Nous avons fait une étude de poids (ANNEXE 1) pour la charge appliquée au bord du plateau. Cela n'a finalement pas été nécessaire car le plateau est suffisamment solide et la masse sur ce dernier n'influence pas le poids du robot.

Le plateau est en médium d'épaisseur 3mm. Il est logé dans une fente prévue à cet effet dans le monte-charge. Pour assurer sa stabilité, il est supporté par deux équerres en médium également. Ces deux systèmes mécaniques permettent de mettre en position notre plateau et assurer sa stabilité

Enfin, notre monte-charge coulisse par l'intermédiaire de 8 roues de 40mm de diamètre. Elles sont placées au-dessus et en dessous des entretoises permettant d'assurer la stabilité du système.

6.1.3 Stabilité du robot

Nous sommes très tôt souciés des potentiels problèmes de stabilités engendré par le déplacement d'un robot de cette envergure et par conséquence du risque pour les usagers en cas de basculement de ce robot de plus 1m de haut et pesant plus de 10kg.

Nous avons donc d'abord réalisé l'analyse statique du problème (ANNEXE 2) qui a démontré que le robot était parfaitement stable à l'arrêt, y compris avec le verre en haut.

Nous avons également regardé ce qu'il se passait lorsque le robot freine ou accélère et nous sommes arrivés à la conclusion qu'en sous estimant largement la masse du robot, et en surestimant la hauteur du centre de gravité, nous avons toujours 10% de coefficient de sécurité avant le basculement (ANNEXE 3).

Cette deuxième étude révèle qu'il est indispensable de veiller à ne pas avoir une vitesse trop rapide, et si tel serait le cas, il est indispensable de programmer l'arrêt progressif des moteurs pour avoir la distance d'arrêt la plus longue possible.

6.2 Liste du matériel

Tableau 4 : Liste du matériel nécessaire

Origine	Description	Référence	Unité	P.U	Quantité	Prix Total
Stock CC EEAS	Roue grande		pcs	21.40 €	2	42.80 €
Stock CC EEAS	Roue folles		pcs	4.20 €	2	8.40 €
Stock CC EEAS	Moteur CC	RB350050-227	pcs	20.99 €	3	62.97 €
Stock CC EEAS	Carte Pilotage Moteur Propulsion	MC33926	pcs	85.00 €	1	85.00 €
Stock CC EEAS	Carte Pilotage Moteur Treuille	MC33926	pcs	85.00 €	1	85.00 €
Fabrication CC EEAS	Tambour Treuille (impression 3D)		pcs	10.00 €	1	10.00 €
F. MARSEILLE	Tube PVC		m	3.90 €	1	3.90 €
Louis	Ficelle		m	3.00 €	1	3.00 €
F. MARSEILLE	LED IR		pcs	0.82 €	5	4.10 €
Louis	Chapeau de maintient des émetteurs IR		pcs	4.00 €	1	4.00 €
F. MARSEILLE	Capteur infrarouge suivi		pcs	0.94 €	3	2.82 €
Stock CC EEAS	Capteur infrarouge détection obstacle		pcs	12.50 €	2	25.00 €
Commande CC EEAS	Poulie		pcs	8.00 €	1	8.00 €
Commande CC EEAS	Bouton poussoir rigolo énorme		pcs	2.50 €	1	2.50 €
Stock CC EEAS	Bouton On/Off		pcs	3.77 €	1	3.77 €
Fabrication F. MARSEILLE	Roulette chariot (impression 3D)		pcs	4.20 €	8	33.60 €
Fabrication F. MARSEILLE	Chariot monte charge (impression 3D)		pcs	40.00 €	1	40.00 €
Stock CC EEAS	Demi Sphère		pcs	9.40 €	2	18.80 €
Fabrication CC EEAS	Plateau		pcs	20.00 €	1	20.00 €
Stock CC EEAS	Fil électrique		m	0.25 €	10	2.50 €
Stock CC EEAS	Cable électrique multibrin souple		m	2.00 €	1.5	3.00 €
Stock CC EEAS	Boitier à piles		pcs	1.75 €	1	1.75 €
Stock CC EEAS	Piles (3.7V)		pcs	1.22 €	3	3.66 €
Stock CC EEAS	Carte Arduino Uno		pcs	21.00 €	1	21.00 €
F. MARSEILLE	Carte Arduino Nano		pcs	15.00 €	1	15.00 €
Stock CC EEAS	Contacteur de fin de courses		pcs	2.00 €	2	4.00 €
Stock CC EEAS	Résistance (valeur variables)		pcs	0.40 €	6	2.40 €
Divers	Fournitures diverse (visserie, colle, étain, peinture…)		forfait	30.00 €	1	30.00 €
Fabrication CCU	Chassis, Axe moteurs V2, Platine Capteur SHARP		forfait	300.00 €	1	300.00 €
Total					29	846.97 €

6.3 Conception électrotechnique

6.3.1 Objectif algorithmique du robot

Notre robot se déplace en suivant un émetteur infrarouge ; lorsqu'il arrive devant la personne qu'il doit servir, il s'arrête, puis si la personne souhaite boire un verre, elle appuie sur le bouton, le robot s'arrête alors fait monter le plateau avec le verre et attend que l'utilisateur rappuie sur le bouton pour redescendre et avancer uniquement lorsque le plateau est en bas.

6.3.2 Présentation des capteurs & détecteurs

Notre robot est constitué de deux types de capteurs, un récepteur infrarouge et un capteur d'obstacle ainsi qu'un détecteur.

6.3.2.1 Capteur Infrarouge

Ce capteur est composé d'une diode PIN et d'un préamplificateur qui sont assemblés sur une grille de connexion, l'époxy, le paquet contient un filtre IR.

Dans notre robot notre capteur infrarouge est alimenté par une tension de 5V (Vs sur la Figure 22) et en sortie celui-ci livre un signal composé de 0 lorsqu'il n'y a pas de rayonnement infrarouge et des salves de 1 et de 0 (les 1 sont symbolisés par 5V et les zéros sont par 0V) qui correspondent à la fréquence de salves de l'émetteur infrarouge.

MECHANICAL DATA

Pinning for TSOP382.., TSOP384..:

1 = OUT, 2 = GND, 3 = V_S

Figure 22 : Extrait de la documentation du capteur infrarouge

6.3.2.2 Capteur détection d'obstacle

Figure 23 : Capteur d'obstacle Sharp GP2Y0A21YK0F

Notre capteur de détection d'obstacle est un Sharp GP2Y0A21YK0F. Ce capteur envoie des ondes infrarouges, une partie de ces ondes sont réfléchies lorsqu'elles rencontrent un obstacle et elles sont réceptionnées par le récepteur infrarouge du SHARP. Selon la distance à laquelle l'obstacle est détecté, la tension en sortie du SHARP varie. La relation entre la distance et la tension de sortie du capteur est donnée par la courbe ci-dessous (Figure 25).

Dans notre robot le SHARP est alimenté par une tension de $5V$ (Figure 24) et en sortie celui-ci nous renvoie une tension comprise entre 0.3 et $5.3\ V$ (Figure 24).

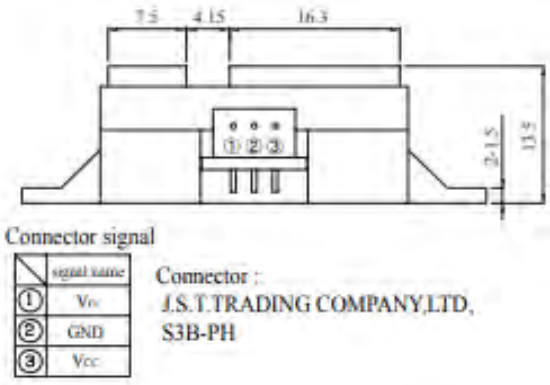


Figure 24 : Schéma du capteur SHARP GP2Y0A21YK0F

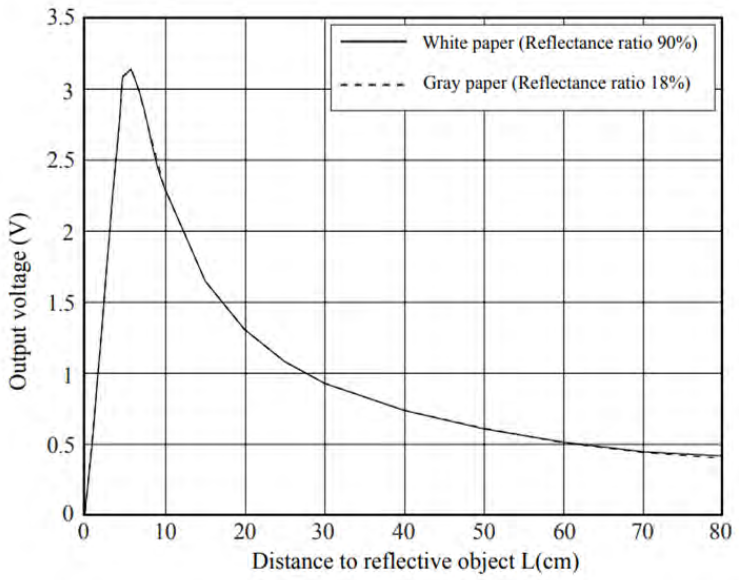


Figure 25 : Courbe caractéristique de la tension de sortie en fonction de la distance

6.3.2.3 Les détecteurs

A ces deux principaux capteurs, il faut rajouter les détecteurs, nous regrouperont ici le bouton et les fins de courses car ils fonctionnent sur le même principe.

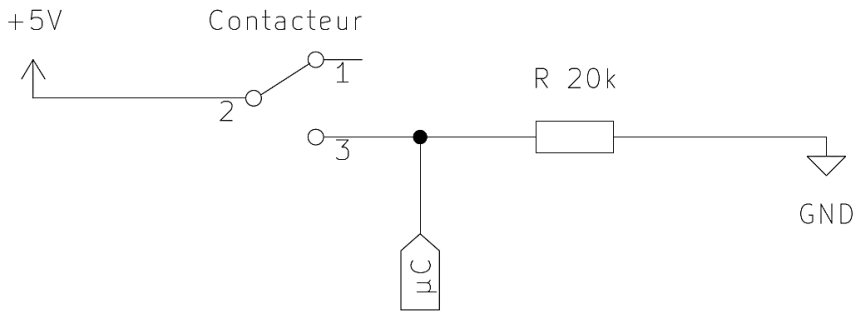


Figure 26 : Schéma câblage contacteur

L'objectif est de livre une tension nul sur la borne de la carte électronique sauf quand le bouton / détecteur est actionné. Pour être certain que la tension est bien $0V$, on ajoute une résistance dite de tirage à la masse qui permet de mettre la sortie au même potentiel que la masse lorsque le contact n'est pas appuyé (Figure 26). Le potentiel passera à $5V$ lorsque le contacteur sera actionné.

A noter que la seule différence entre les capteurs de fin de course et le bouton réside dans la présence de deux broches NC (Normally Closed) et NO (Normally Open) (en plus de la broche commune) sur le premier là où le second ne dispose que de deux broches, donc une commune et une NC.

6.3.3 Schéma du branchement

Pour nous permettre de nous y retrouver dans l'ensemble des composants du robot, nous avons réalisé un schéma complet de câblage du robot.

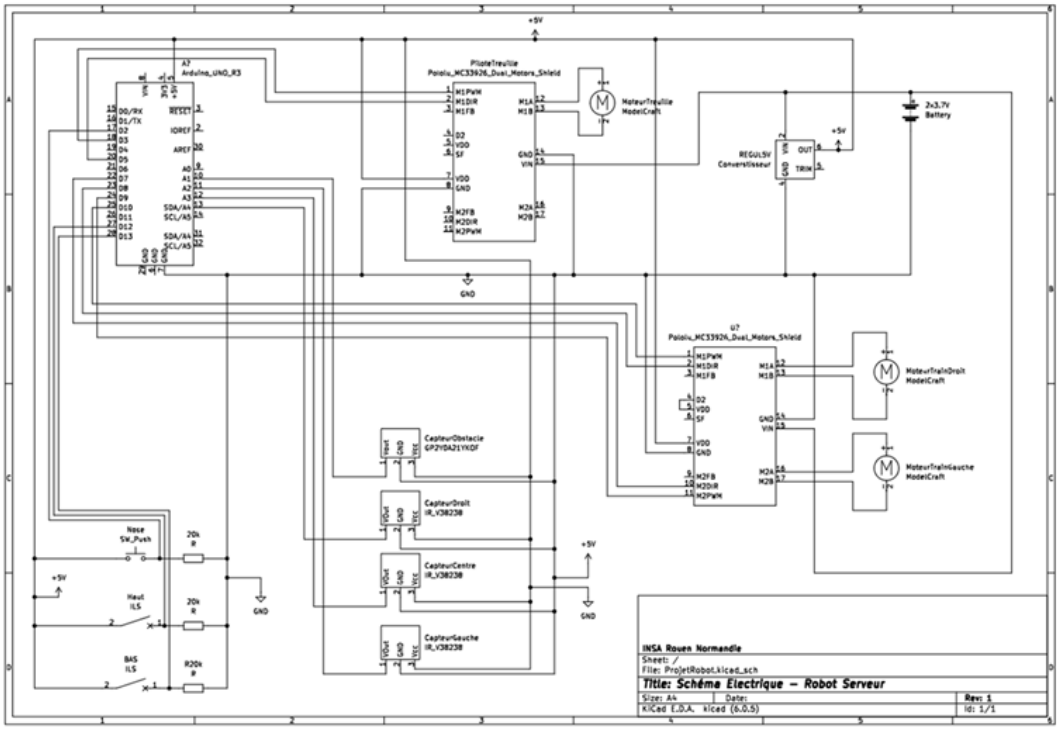


Figure 27 : Schéma électrique du robot disponible en ANNEXE 6

6.3.4 Gestion de l'alimentation

Pour alimenter les différents composants du robot, nous avons opté pour l'utilisation de trois accumulateurs de $3{,}7V$ montées en série pour avoir une alimentation principale de $11.1\ V$. Chaque accumulateur à une capacité de $3500mAh$ ce qui implique, du fait du montage en série des accumulateurs, une capacité totale de $3500mAh$.

Une capacité de $3{,}5Ah$ signifie que la batterie est capable de délivrer $3{,}5$ ampère pendant un heure avant d'être complétement déchargé. On peut donc utiliser cette information pour estimer la longévité de la batterie.

Nous reviendrons apprès sur l'alimentation des capteurs et cartes électroniques qui ne consomme pas beaucoup d'électricité comparé aux moteurs. Nous ne prendrons en compte que ces derniers pour estimer l'autonomie du robot.

La documentation fournie par le site internet Conrad (4), le courant maximum utilisé par le moteur est de $0.81A \approx 1A$. Nous allons considérer que deux moteurs fonctionnent en moyenne, ainsi la consommation des moteurs sera de $2A$.

Par une simple règle de trois, nous obtenons une durée de fonctionnement maximal de 105 minutes. Cette autonomie est sous-estimée à cause des hypothèses précédentes.

Revenons maintenant à l'alimentation des cartes en 5V, nous avons intégré un régulateur à notre robot qui prend une tension d'entrée de $11.1V$ et qui abaisse la tension au niveau souhaité réglable avec un potentiomètre trimmer intégré à la carte.

Pour finie, nous avons pris soin de protéger les différents composants du robot avec des feuilles plastiques pour éviter les faux contacts avec le châssis métallique.

6.3.5 Programme Arduino

6.3.5.1 Organigramme du programme

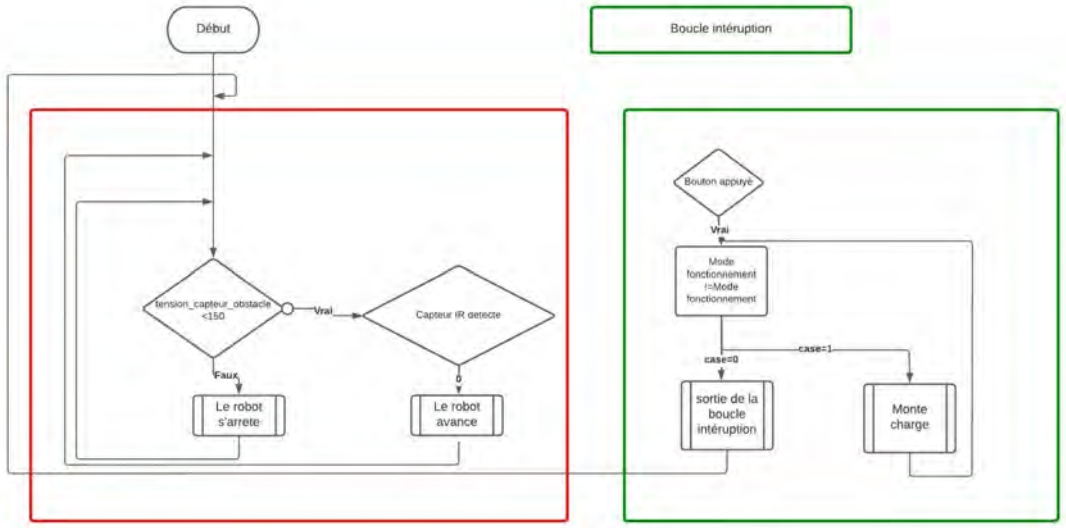


Figure 28 : Logigramme du programme

6.3.5.2 Présentation du programme

Notre programme est constitué de deux parties, la première partie est le mode avancer qui se charge de faire avancer le robot et la deuxième partie est le mode service qui se charge de la mobilité du plateau. Notre programme sort de la partie avancer à l'aide d'une boucle interruption.

1ère partie : Mode avancer

Avant de pouvoir avancer, notre robot vérifie s'il n'y a pas d'obstacle devant lui. Ce qui correspond dans le programme à la commande « tension_capteur_obstacle < 150 », à 150 cV est la tension que renvoie le capteur d'obstacle pour un obstacle à une distance environ de 20 cm.

Ensuite si notre robot ne détecte aucun obstacle, il vérifie que le capteur infrarouge reçoit un signal. Pour pouvoir détecter si le robot reçoit un signal infrarouge, le programme lit les trois dernières valeurs mesurées, et précédemment stockées dans un tableur, des capteurs IR. Si ces trois mesures sont différentes alors cela signifie que le robot reçoit le signal infrarouge. Si ces deux tests sont vérifiés le robot est autorisé à avancer.

2ème partie : Mode service

La deuxième partie est la partie service c'est-à-dire celle qui s'occupe de la mobilité du plateau.

Lorsque l'on appuie une fois sur le bouton, le plateau monte. Le système de remonté se termine lorsque le plateau touche le capteur fin de course du haut. Une fois en haut, le programme attend que l'utilisateur se serve et appuie une seconde fois sur le bouton. A ce moment, le plateau redescend jusqu'à toucher le capteur fin de course du bas.

Notons que nous utilisons une variable "ModeFonctionnement » qui pilote l'entrée et la sortie de la boucle interruption

6.3.5.3 Lecture des informations des capteurs infrarouges

Pour permettre une lecture efficace des informations issus des capteurs infrarouges, l'émission étant réalisée en slaves relativement courtes, nous utilisons une fonction permettant de stocker chaque lecture dans un tableau et de conserver les 100 dernières lectures des informations issus du capteur.

Pour prendre une décision, nous réalisons alors la somme de tous les éléments du tableau et pour confirmer une lecture sur le capteur, il faut que cette somme soit supérieure à un seuil fixé à 3 dans un premier temps.

Cela permet d'éviter les perturbations dues aux infrarouges émis ailleurs tout en étant certain de réceptionner le signal.

6.3.5.4 Gestion du monte-charge

L'algorithmique du monte-charge est relativement simple : on monte le plateau, on attend l'utilisateur pour le redescendre et repartir en mode suivi de personne. La difficulté réside dans le lancement de cette partie du programme, en effet l'utilisateur peut appuyer à n'importe quel moment.

Nous avons donc eu de nombreuses difficultés relatives à ce fonctionnement jusqu'à tomber sur de la documentation sur les interruptions. Le principe consiste à la suite d'un évènement sur un objet, de mettre en pause le programme principal pour exécuter une suite d'instruction immédiatement.

Dans cette suite d'instruction, nous avons utilisé une variable dont la valeur est inversée à chaque appui sur le bouton, puis d'appeler la fonction permettant de gérer le monte-charge uniquement lorsque cette variable est égale à 1.

6.3.5.5 Programme principale

Le programme principal, au départ complexe, a été simplifié pour faciliter la mise en place de quelque chose de fonctionnel pour le robot. Ainsi le *void loop()* se résume à deux structures « Switch / Case Of » qui d'abord vérifie qu'il n'y a pas d'obstacle. Si pas d'obstacle, on demande la direction de déplacement. Par manque de temps, nous n'avons pu intégrer que la fonctionnalité « avancer en ligne droite », non pas à cause de difficultés de programmation, mais à cause de problème mécanique dans la séparation des champs des 3 capteurs infrarouges que nous n'avons pas pu mener à son terme.

7 CONCLUSION

Ce projet nous a permis de découvrir la mécanique au sens littéral du terme. A la fois ingénieur, dans la conception des modèles, et technicien, dans la maîtrise des outils comme la perceuse à colonne ou le poste à souder, nous avons pu vêtir plusieurs casquettes au cours des 6 séances. Ce robot, nous y avons consacré tout notre temps et notre patience pour le voir grandir puis avancer au fil des mois. Aujourd'hui, il ne répond pas complètement à nos critères de bases. Par manque de temps, nous n'avons pas pu par exemple, lui mettre trois yeux. Malgré ça, nous somme tout de même fier d'avoir conçu notre petit robot (de 1 m de haut tout de même) en partant tout simplement d'une envie, d'une idée puis de deux coups de crayon Veleda sur un tableau blanc.

Nous avons aussi appris à travailler en groupe. Malgré quelques soucis de communication nous avons réussi à maintenir un lien et créer une bonne ambiance de travail entre nous. Nous avons essuyé les échecs de conception (notamment notre châssis qui avait été conçu en premier lieu en 3D) et nous avons su rebondir en nous réorganisant sans perdre la motivation de donner vie à notre robot serveur.

A travers ce projet, nous avons appris de nombreuses techniques de conceptions et de notions informatiques à travers les connaissances des uns et des autres. L'un des points négatifs du projet est notre ambition parfois trop grande. Les rêves pleins la tête, il n'est pas toujours évident de se mettre d'accord sur les fonctionnalités et les priorités du robot. Nous avons dû faire des concessions pour simplifier notre projet.

BIBLIOGRAPHIE

1. **GTadmin.** Réaliser un Robot suiveur de ligne avec une carte Arduino UNO. *gootrio.com.* [En ligne] 17 10 2021. [Citation : 18 02 2022.] https://gootrio.com/realiser-un-robot-suiveur-de-ligne-avec-une-carte-arduino-uno/.

2. **PEYRONNET - LE MARTIN, Nicole.** LE COACH SANTÉ DE VOS COLLABORATEURS, LE ROBOT SHERPA , VOUS SUIT PARTOUT ! [En ligne] [Citation : 18 02 2022.] https://ergonoma.com/2018/09/le-coach-sante-de-vos-collaborateurs-le-robot-sherpa-vous-suit-partout/.

3. **Fanchi, José.** ALF, ça bouge avec La Poste : Le robot-suiveur, une première à Ajaccio. *Corse Net Info.* [En ligne] 07 09 2018. [Citation : 18 02 2022.] https://www.corsenetinfos.corsica/ALF-ca-bouge-avec-La-Poste-Le-robot-suiveur-une-premiere-a-Ajaccio_a35222.html.

4. **Modelcraft RB350050-22723R Moteur à entraînement ultra puissant 12 V 50:1. *CONRAD.* [En ligne] [Citation : 30 05 2022.] https://www.conrad.fr/p/modelcraft-rb350050-22723r-moteur-a-entrainement-ultra-puissant-12-v-501-233132?gclid=Cj0KCQjw4uaUBhC8ARIsANUuDjWjYaOLKBQaZV6aVD6vBN6ScFyTMsF9p_cUslRU5ULSHLiQnLd4DLEaAi2-EALw_wcB&gclsrc=aw.ds&utm_campaign=shopping-feed&utm_content=.**

5. Star Wars - B7493 - R2-D2 Electronique. *Amazon.* [En ligne] [Citation : 31 05 2022.] https://www.amazon.fr/Star-Wars-B7493-R2-D2-Electronique/dp/B01L93GCTO.

6. BB-8. *Star Wars Wiki.* [En ligne] [Citation : 31 05 2022.] https://starwars.fandom.com/fr/wiki/BB-8.

TABLE DES ILLUSTRATIONS

Figure 1 : Illustration - Photo du robot assemblée 4
Figure 2 : Illustration R2D2 (4) 5
Figure 3 : Illustration BB-8 (5) 5
Figure 4 : Robot suiveur de ligne (1) 5
Figure 5 : Robot Sherpa 6
Figure 6 : Robot suiveur de La Poste (3) 6
Figure 7 : Diagramme Bête à Cornes 7
Figure 8 : Diagramme Pieuvre 7
Figure 9 : Diagramme de Gantt prévisionnel 10
Figure 10 : Diagramme de Gantt réalisé 10
Figure 11 : Photo de l'équipe CAO à pied d'œuvre 10
Figure 12 : Première esquisse du robot 12
Figure 13 : Étude du positionnement du cou du robot 12
Figure 14 : Deux principales solutions de conception étudiés 12
Figure 15 : Assemblage CATIA du châssis version 1 13
Figure 16 : Assemblage CATIA du châssis final 13
Figure 17 : Début du montage des éléments du châssis 13
Figure 18 : Châssis assemblé avec tous les éléments 13
Figure 19 : Conception de la fonction service du robot 14
Figure 20 : Monte-charge imprimé et assemblé 14
Figure 21 : CAO du monte-charge 14
Figure 22 : Extrait de la documentation du capteur infrarouge 17
Figure 23 : Capteur d'obstacle Sharp GP2Y0A21YK0F 18
Figure 24 : Schéma du capteur SHARP GP2Y0A21YK0F 18
Figure 25 : Courbe caractéristique de la tension de sortie en fonction de la distance 19
Figure 26 : Schéma câblage contacteur 19
Figure 27 : Schéma électrique du robot disponible en ANNEXE 6 20
Figure 28 : Logigramme du programme 22
Figure 29 : Visualisation de l'espace 54
Figure 30 : Marquage du châssis 54
Figure 31 : Peinture de la demi-sphère 54
Figure 32 : Assemblage et soudure des composants du châssis 54
Figure 33 : Montage du nez du robot 54
Figure 34 : Peinture de la tête 54
Figure 35 : Séance de travail 55
Figure 36 : Tournage des axes des roues en dernière minute 55
Figure 37 : Taraudage des trous pour vis pointaux 55

TABLE DES TABLEAUX

Tableau 1 : Liste des fonctionnalités implémentés **9**
Tableau 2 : Répartition des rôles de conception **11**
Tableau 3 : Répartition des pièces de CAO **11**
Tableau 4 : Liste du matériel nécessaire **16**

ANNEXES

Liste des annexes :

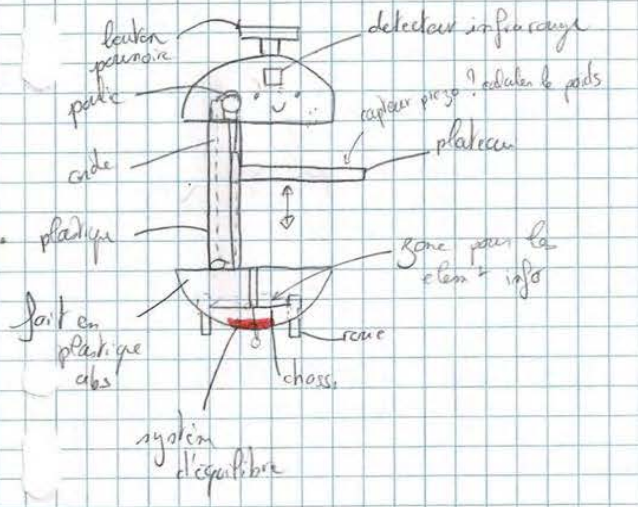
ANNEXE 1 ETUDE DE FAISABILITE DU ROBOT 29
ANNEXE 2 ETUDE DE STABILITE STATIQUE DU ROBOT 31
ANNEXE 3 ETUDE DE STABILITE DYNAMIQUE 32
ANNEXE 4 SUIVI DU PROJET 35
ANNEXE 5 PLANS CHASSIS 41
ANNEXE 6 SCHEMA ELECTRIQUE DU ROBOT 46
ANNEXE 7 CODE DU PROGRAMME FONCTIONNEL ET TESTE 47
ANNEXE 8 CODE THEORIQUE POUR LES TROIS CAPTEURS (NON TESTE) 50
ANNEXE 9 QUELQUES PHOTOS DE LA FABRICATION 54

ANNEXE 1 Etude de faisabilité du robot

Etude de Faisabilité du projet robot

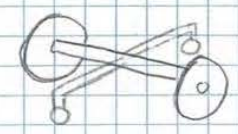
Robotbeauf

Schéma du Robot suivreur :



Système des roues

→ 2 roues motrices
→ 2 billes d'appuis



objectif
→ assurer l'équilibre de la boule
→ permettre au robot d'avancer

Note les roues sont sortie de la boule → bien peu les roues.

Les differents capteurs

→ detection obstacles
- objectif : detecter les obstacles et les contourner
- localisation : sur la surface de la demi sphère inferieur
- nombre : 5

technologie : IR ? US ? ou capteur, capteur IR

→ capteur detection de personne
objectif : suivre une lumière IR presente sur la personne
localisation : ds la tête du robot
nombre : 1 × 3
technologie : un recteur infrarouge

info transmise par un f.f à la sphère du bas (carte Arduino)

× × ×

tête milieu cherchera avoir la lumière au centre

→ mesure d'inclinaison ?

⇒

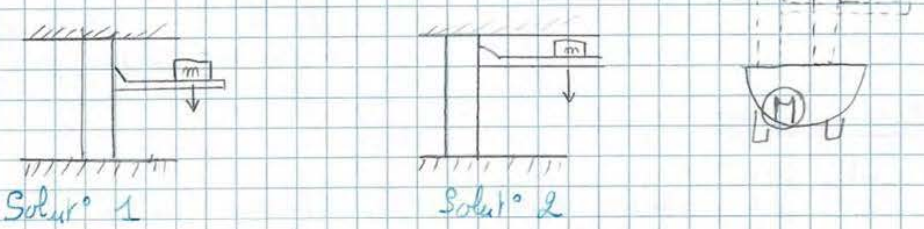
→ bouton poussoir
objectif : actionner la poulie et faire lever le plateau
localisation : en haut de la tête du robot
nb : 1
contrainte : accessible par utilisateur.
et éviter les risques de chute du robot

→ Interrupteur / Détecteur à lame
objectif : détecte lorsque le plateau est en haut.
et permet d'arrêter le moteur.
localisation : en haut du système de levage du plateau.
nb : 1

→ bouton on off

Etude mécanique du système.

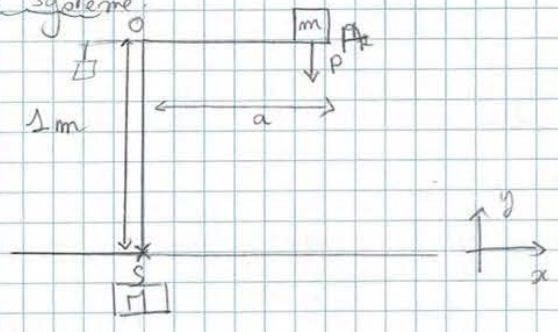
2 modèles sont étudiés, on va chercher à déterminer le ⊕ fixée et le ⊕ réalisable d'un pt de vue statique et mécanique.



Solut° 1 Solut° 2

Le problème : Où placer la masse M pour que le système soit équilibre statique ? Et quel est la valeur de M ?

2° système



$\vec{P_A} = m\vec{g}$

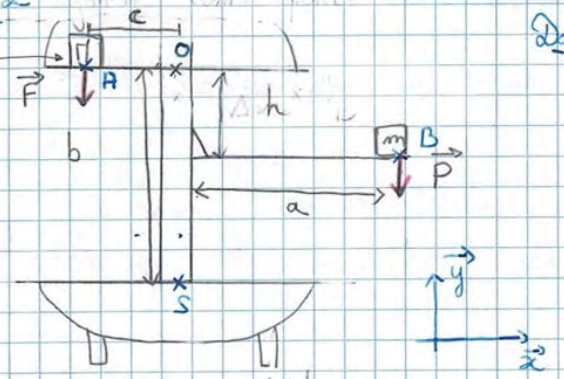
$M(A) = \vec{OA_1} \wedge \vec{P} = a\vec{x} \wedge m\vec{-g}$
$= -a\,mg\,\vec{z}$

$\vec{P_S} = M\vec{g}$

ANNEXE 2 Etude de stabilité statique du robot

Etude du système 2

Analogie → grue



Données

$\overrightarrow{OA} = c\,\vec{x}$

$\overrightarrow{OS} = -b\,\vec{y}$

$\overrightarrow{OB} = h\,\vec{y} + a\,\vec{x}$

Forces :

en B $\vec{P} = -mg\,\vec{y}$

en A $\vec{F} = -Mg\,\vec{y}$

1) Calcul du moment du poids m par rapport à O

$$\{\mathcal{T}_m\}_B = \begin{vmatrix} 0 & 0 \\ -mg & 0 \\ 0 & 0 \end{vmatrix}_{B,R}$$

Note $\overrightarrow{M}_B R = \overrightarrow{M}_A R + \overrightarrow{BA} \wedge R$

$\overrightarrow{M}_B = \overrightarrow{M}_A + \overrightarrow{BA} \wedge \vec{R}$ } Formule de Varignon

Donc $\overrightarrow{\mathcal{M}}_{O\,P_{oid}} = \overrightarrow{\mathcal{M}}_B P + \overrightarrow{OB} \wedge \vec{P}$

$$= 0 + \begin{pmatrix} a \\ h \\ 0 \end{pmatrix} \wedge \begin{pmatrix} 0 \\ -mg \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -amg \end{pmatrix}$$

Le moment du poids m sur axe O

car pas de rotation au point B de la masse m

2) On cherche la masse m du contre poid pour que le système soit en équilibre.

$\vec{F} = -Mg\,\vec{y}$

$\overrightarrow{\mathcal{M}}_O F = \overrightarrow{M}_A F + \overrightarrow{OA} \wedge \vec{F}$

$$= 0 + \begin{pmatrix} -c \\ 0 \\ 0 \end{pmatrix} \wedge \begin{pmatrix} 0 \\ -Mg \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ cMg \end{pmatrix}$$

Etude de l'équilibre

$\sum \mathcal{M}_O = 0$ Les moments sont au m point

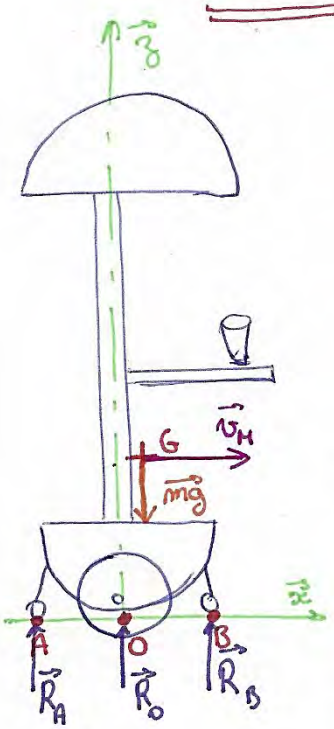
$\overrightarrow{\mathcal{M}}_O F + \overrightarrow{\mathcal{M}}_O P = 0$

$$\begin{cases} 0 \\ 0 \\ -amg + cMg = 0 \end{cases} \Leftrightarrow \boxed{M = \frac{amg}{cg} = \frac{am}{c}}$$

ANNEXE 3 Etude de stabilité dynamique

Stabilité dynamique

1/3



G centre de gravité :

A point de contact de la roulette avant

B point de contact de la roulette arrière

O point de contact du train propulsif.

$\vec{v}_M$ vitesse maximum du robot

$m\vec{g}$ poids du robot

$\vec{R}_A, \vec{R}_B, \vec{R}_O$ les résultantes des appuis sans frottements aux points A, B et O.

t_a temps d'accélération et/ou de décélération.

x_A distance d'arrêt.

→ translation en ligne droite selon $\vec{x}$

→ accélération et décélérations supposées constante

L'accélération en valeur absolue : $\gamma_x = \dfrac{v_M}{t_a}$

avec $t_a = \dfrac{2\,x_A}{v_M}$ donc $\boxed{\gamma_x = \dfrac{v_M^2}{2\,x_A}}$

→ On néglige le moment d'inertie du système, on considère toute la masse en G.

- $P = R_A + R_O + R_B$
- $F_{freinage} = F_{inertie}$

  $\parallel \qquad\qquad \parallel$

  $F_{freinage} = F_i$

→ Moment en B

$$\mathcal{M}_B = OB\,R_0 + AB\,R_A + z_G F_i = \left(\frac{AB}{2} - x_G\right) P$$

À la limite du basculement, $R_0 = 0$ et $R_A = 0$

donc la condition de basculement :

$$\boxed{z_G F_i < \left(\frac{AB}{2} - x_G\right) P}$$

$$F_i = \frac{v_H^2}{2 x_A} \cdot m \quad \text{et} \quad P = mg$$

donc la condition de basculement s'écrit

$$\boxed{\frac{z_G v_H^2}{2 x_A} < \frac{AB}{2} \cdot g}$$

On prend en ordre de grandeur :

$AB \simeq 0{,}3$ m

$z_G \simeq 0{,}2$ m

$v_H \simeq 0{,}3\ m.s^{-1}$

$x_A \simeq 0{,}05$ m

$m \simeq 10^{1}$ kg

$x_G \ll AB \Rightarrow x_G \simeq 0$ et $\frac{AB}{2} \simeq 0{,}15$ m

3/3

donc $\begin{cases} \dfrac{z_G v_M^2}{2x_M} = \dfrac{0{,}2 \times 0{,}3^2}{2 \times 0{,}05} = 0{,}2\ m^2.s^{-2} \\ \dfrac{AB}{2} g = 0{,}15 \times 10 \simeq 1{,}5\ m^2.s^{-2} \end{cases}$

$$\frac{z_G\ F_i}{\frac{AB}{2} \times P} = \frac{0{,}2}{1{,}5} = 0{,}13$$

$\Rightarrow$ coefficient de sécurité de 13 %

ANNEXE 4 Suivi du projet

COMPTE-RENDU DE SEANCE 11 mai 2022	
Objectifs de séance	- Récupération des pièces usinées du nouveau châssis - Récupération des éléments imprimé en 3D et assemblage monte-charge, électronique et châssis. - Premiers tests d'intégrations des composants au programme.
Bilan de la séance	- Châssis non disponible = impossible de l'assembler - Continuation des tests d'intégrations des composants et assemblages des composants électroniques.
Suivi des tâches	
Objectifs de la prochaine séance	- Montage du robot - Test du programme complet

COMPTE-RENDU DE SEANCE 27 avril 2022	
Objectifs de séance	- Récupération des pièces et assemblage du robot - Intégration des composants au robot - Pendant les vacances, finir de dessiner les yeux et le nez. - Test de l'algorithmique déjà réalisé et avance de l'algorithmique
Bilan de la séance	- Impossible d'imprimer le châssis (trop grande pièce) - Recherche d'une solution alternative et conception du nouveau châssis - Modification du monte-charge pour l'adapter à un tuyau de diamètre inférieur. - Continuation du dessin des yeux et du nez
Suivi des tâches	
Objectifs de la prochaine séance	- Récupération des pièces usinés du nouveau châssis - Récupération des éléments imprimé en 3D et assemblage monte-charge, électronique et châssis. - Premiers tests d'intégrations des composants au programme.

COMPTE-RENDU DE SEANCE 30 mars 2022	
Objectifs de séance	- Validation du châssis - Validation du monte-charge - Envoie des pièces à imprimer - Continuer le développement logiciel

Bilan de la séance	- Validation du châssis et monte-charge avec quelque indications de modifications
	  o Ajout emplacement pour écrou dans l'impression
	  o Ajout emplacement pour aiment pour capteur de fin de course
	- Début finalisation circuit alimentation de l'émetteur
	- Finalisation monte-charge

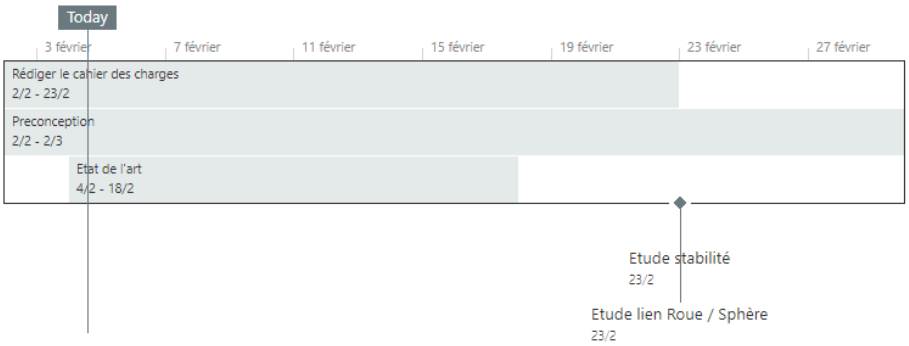
Suivi des tâches	
Objectifs de la prochaine séance	- Récupération des pièces et assemblage du robot - Intégration des composants au robot - Pendant les vacances, finir de dessiner les yeux et le nez. - Test de l'algorithmique déjà réalisé et avance de l'algorithmique

COMPTE-RENDU DE SEANCE 16 mars 2022	
Objectifs de séance	- Finalisation de la conception du châssis et du chariot de guidage - Validation de la solution de localisation de l'utilisateur (IR / US / Radio ?) - Création et test du pilotage des moteurs et orientation du robot. - Avance de la DAO du châssis à imprimer
Bilan de la séance	- Validation de la technologie de suivi : IR & test des émetteurs et récepteur IR - Avance du dessin du monte-charge et création adapteurs d'axe entre moteur et roue. - Analyse et Etude du fonctionnement de Shield moteur.
Suivi des tâches	
Objectifs de la prochaine séance	- Validation du châssis - Validation du monte-charge - Envoie des pièces à imprimer - Continuer le développement logiciel

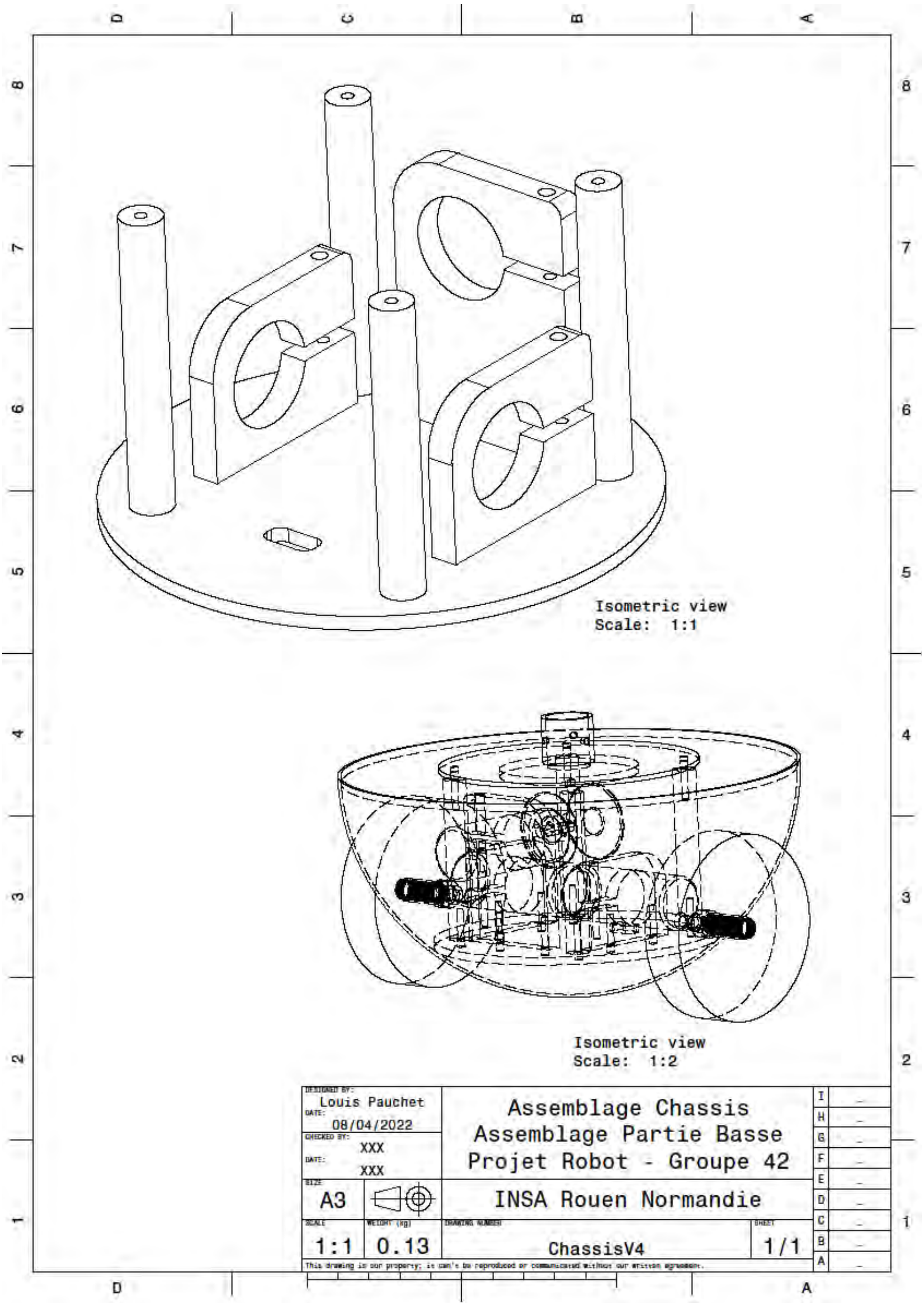
COMPTE-RENDU DE SEANCE 2 mars 2022	
Objectifs de séance	- Valider le cahier des charges - Réaliser le bon de commande - Début de la conception du châssis et du support de plateau. - Travail algorithmique préliminaire.
Bilan de la séance	- Choix des principaux composants réalisé - Préparation du bon de commande pour finalisation avant la fin de la semaine - Début de la conception du châssis
Suivi des tâches	
Objectifs de la prochaine séance	- Finalisation de la conception du châssis et du chariot de guidage - Validation de la solution de localisation de l'utilisateur (IR / US / Radio ?) - Création et test du pilotage des moteurs et orientation du robot.

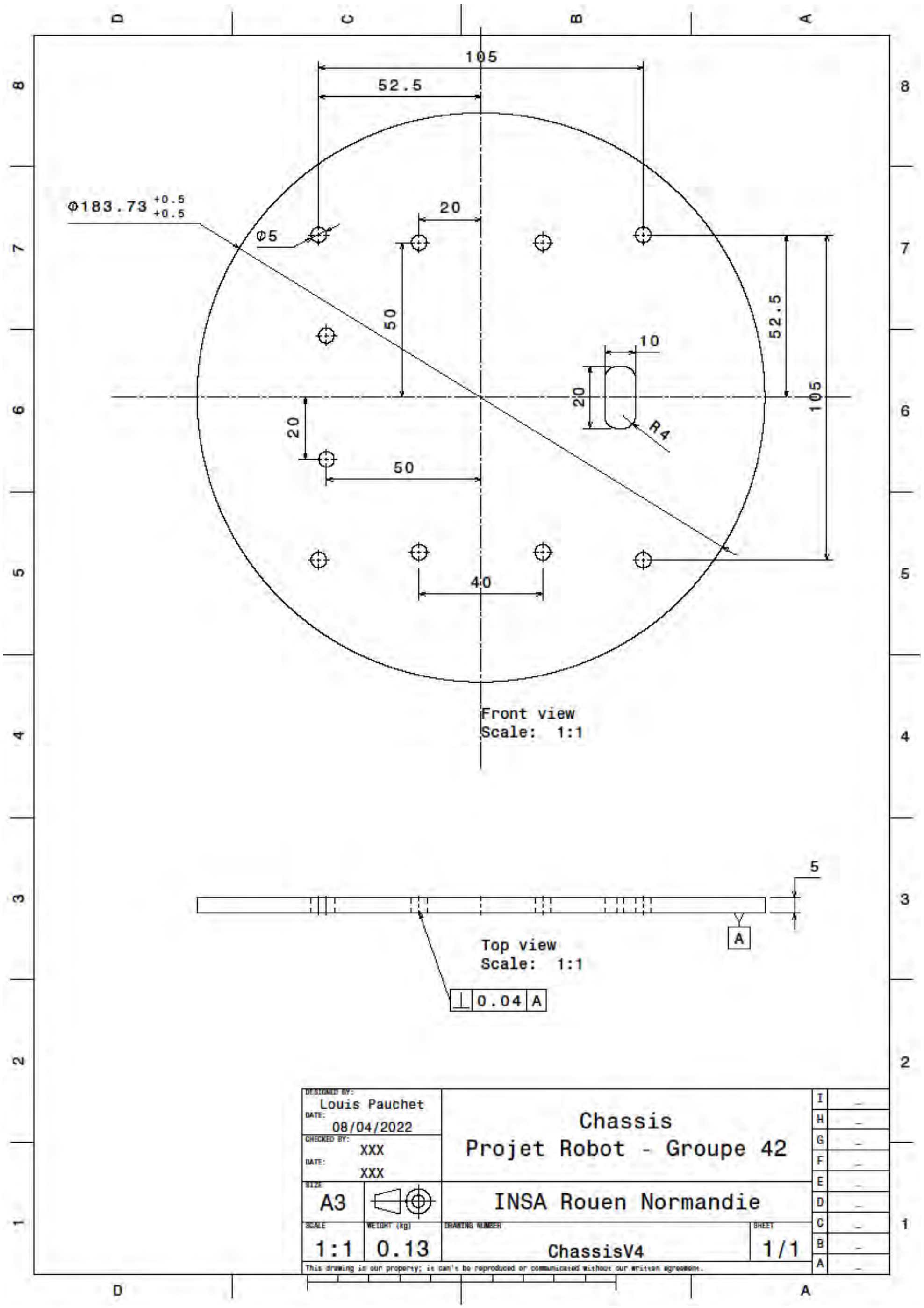
COMPTE-RENDU DE SEANCE

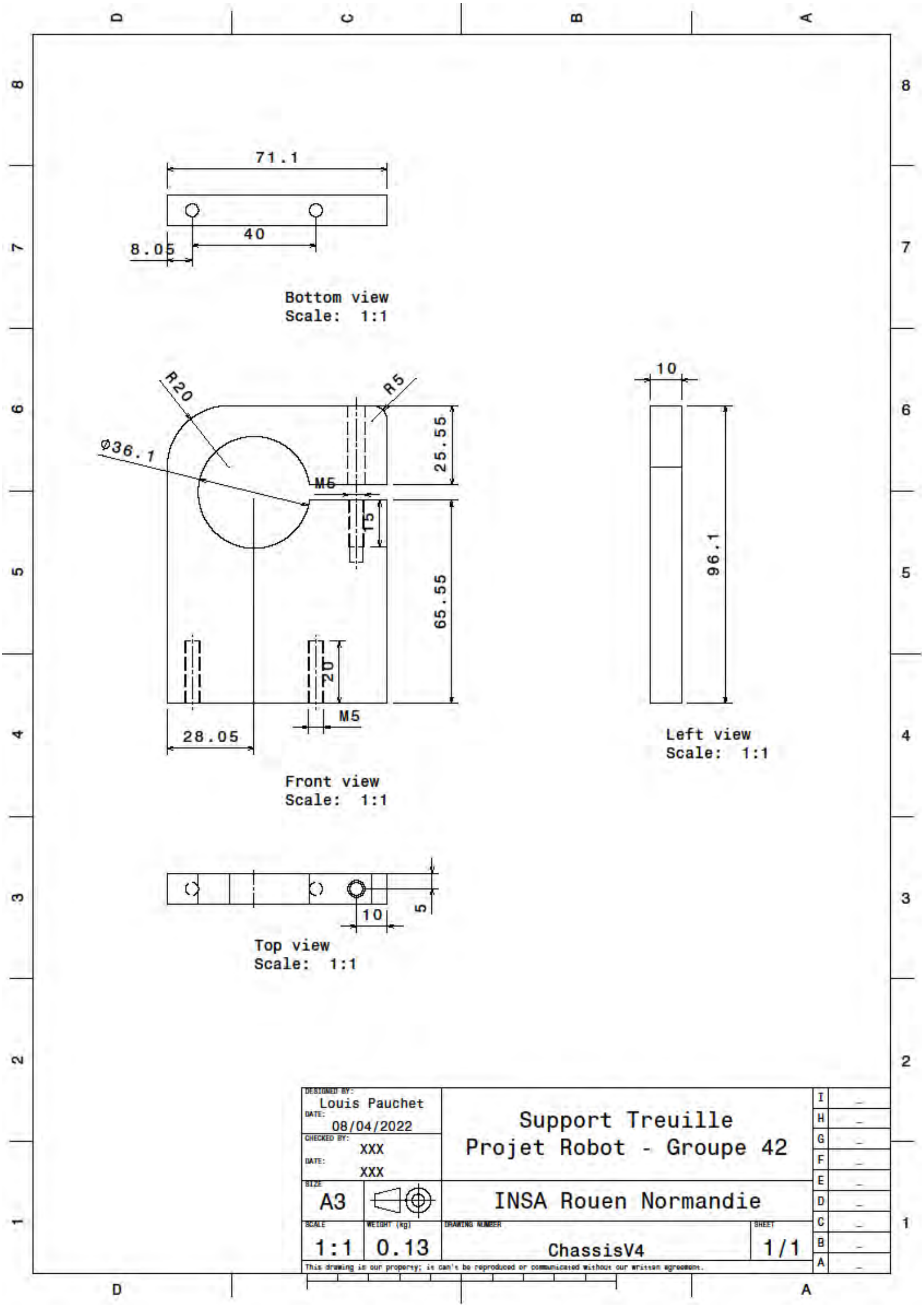
2 février 2022

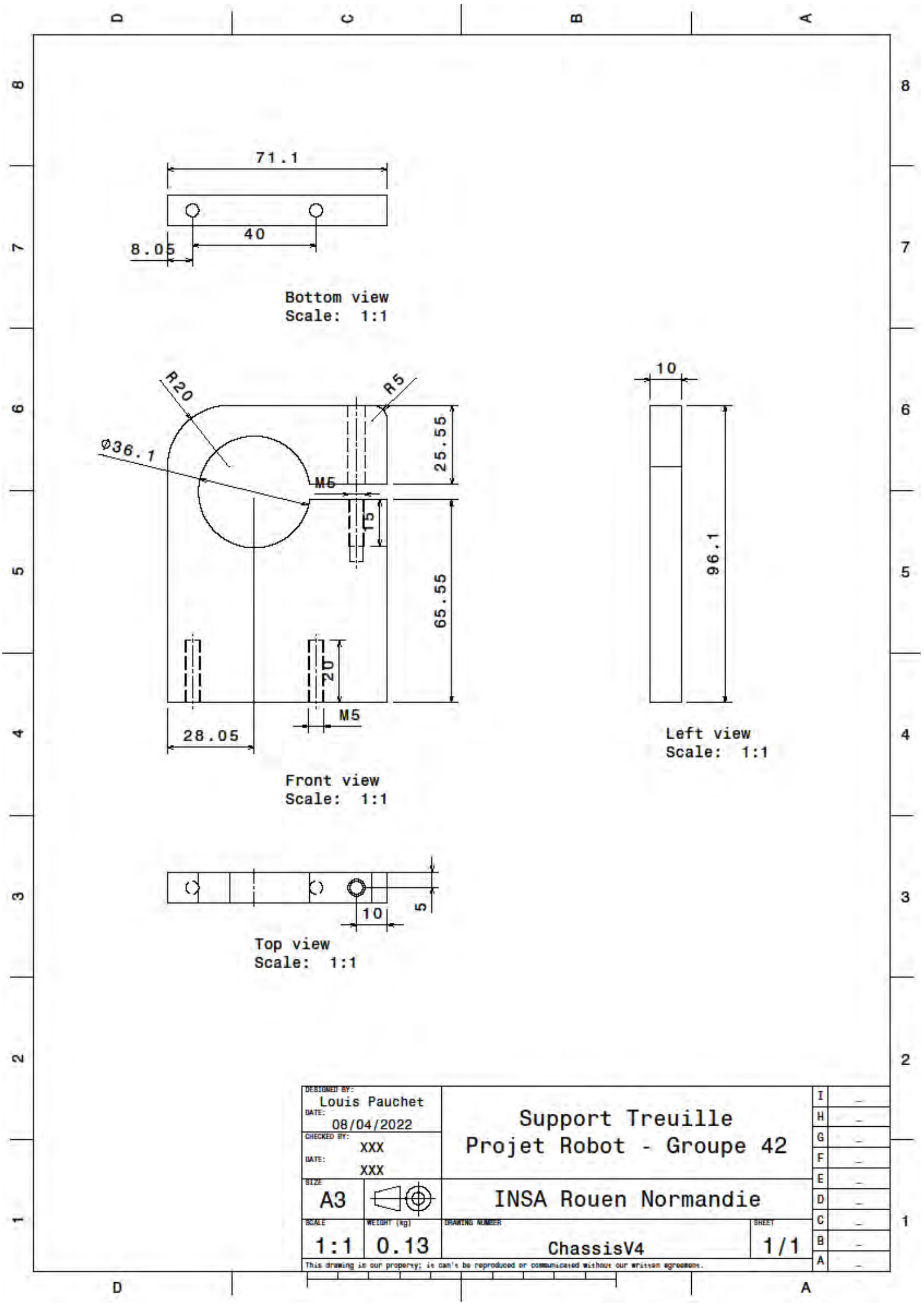
Objectifs de séance	- Choisir l'équipe de projet robot - Choisir le projet de robot et définir les grandes lignes du cahier des charges - Début de la préconception et de l'analyse des solutions techniques à adopter
Bilan de la séance	➢ Réflexion sur le design et les fonctionnalités du robot o Premier design arrêté ➢ Réflexion sur les systèmes mécaniques en jeux et analyse fonctionnelle o Première approximation sur la faisabilité des différentes fonctions souhaité et première réflexion sur les solutions choisis ➢ Découvertes des équipements possibles ➢ Répartition des taches pour les deux prochaines semaines.
Suivi des tâches	
Objectifs de la prochaine séance	- Validation du cahier des charges - Validation du design et dessin du proto - Analyse des besoins en équipement pour chaque fonction technique - CAO des pièces à imprimer en 3D

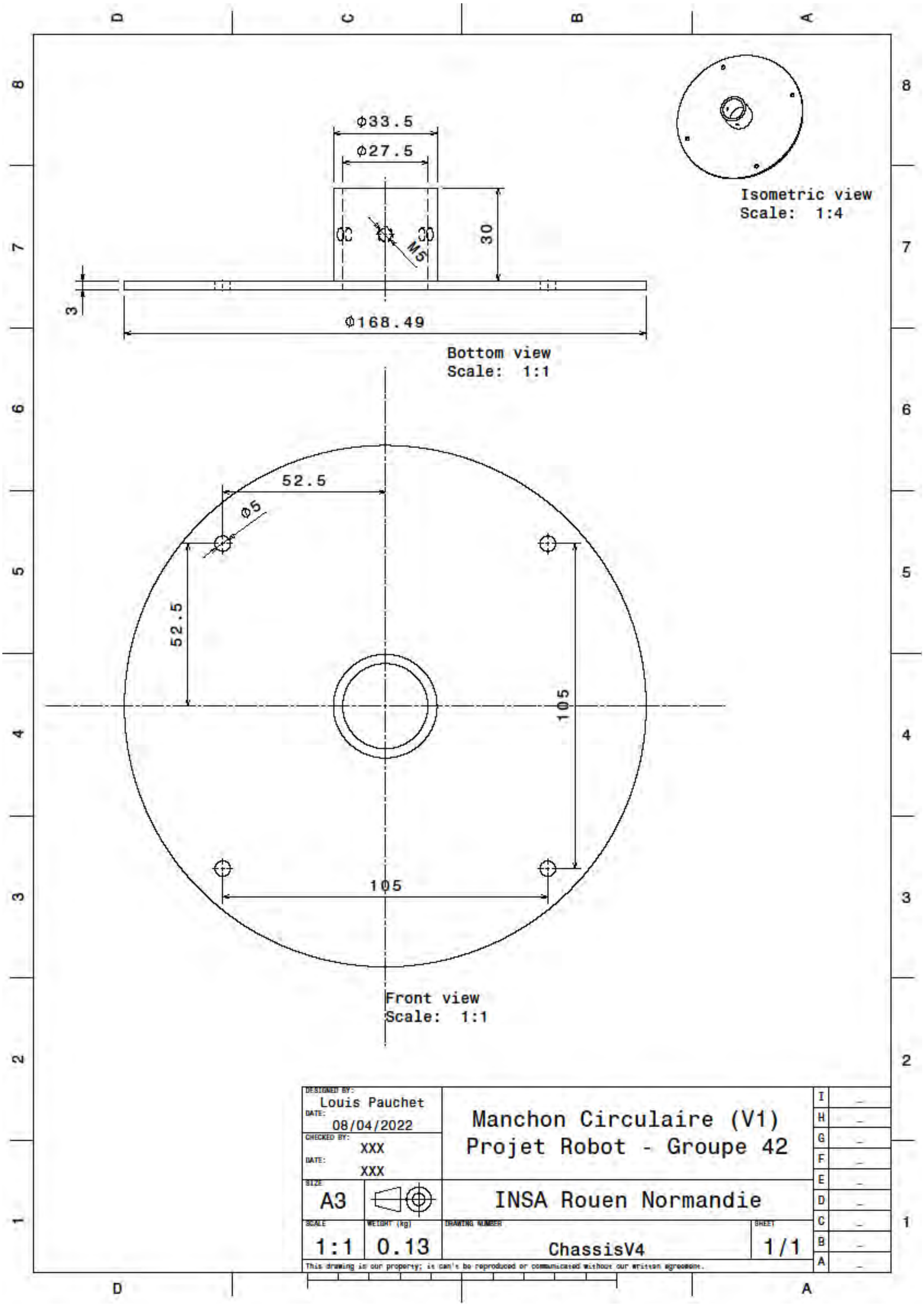
ANNEXE 5 Plans châssis



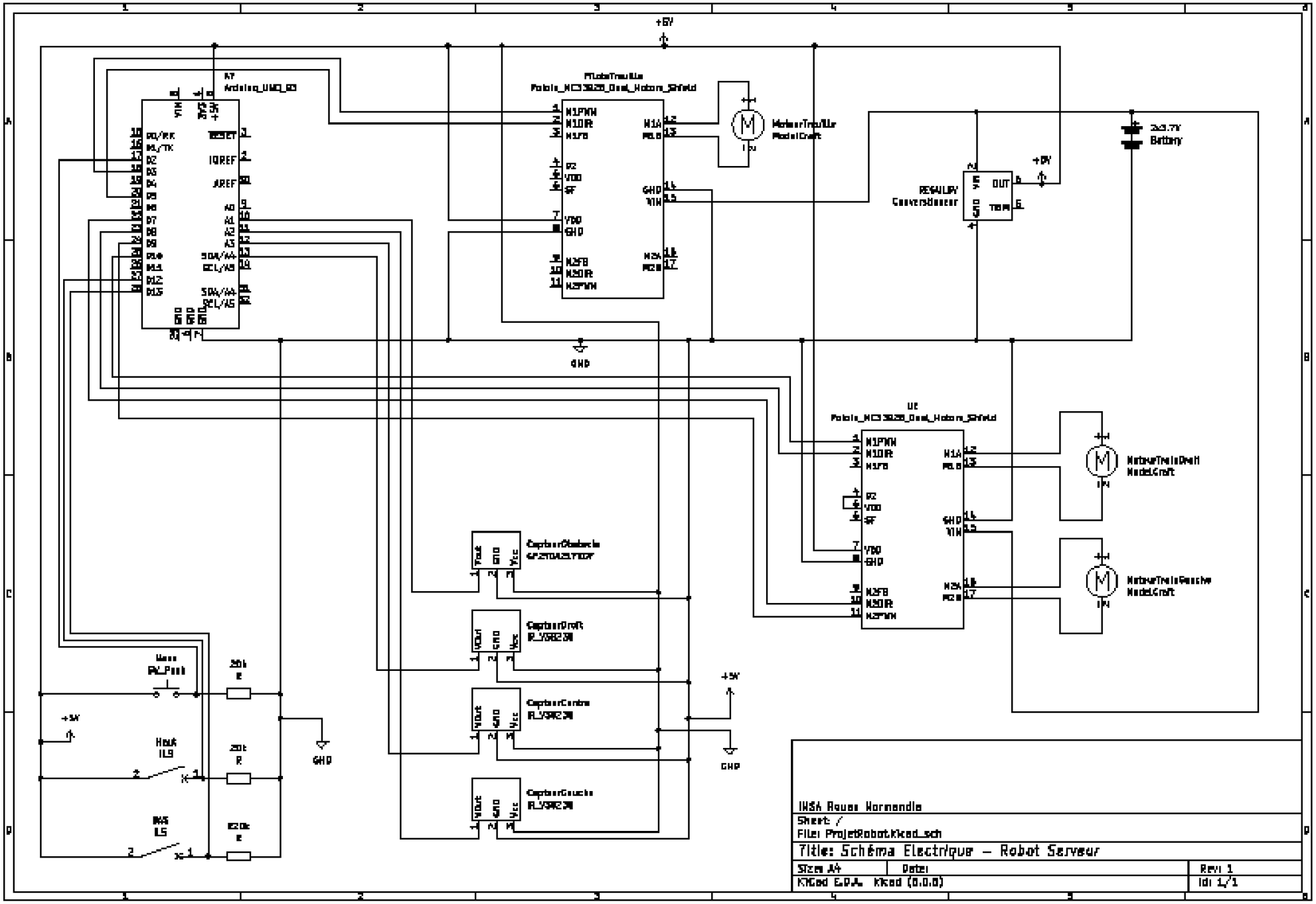

105
52.5
Φ183.73 +0.5 +0.5
Φ5
20
50
52.5
10
20
105
R4
20
50
40
Front view
Scale: 1:1
5
A
Top view
Scale: 1:1
0.04 A
DESIGNED BY:
Louis Pauchet
DATE:
08/04/2022
CHECKED BY:
XXX
DATE:
XXX
Chassis
Projet Robot - Groupe 42
SIZE
A3
INSA Rouen Normandie
SCALE
1:1
WEIGHT (kg)
0.13
DRAWING NUMBER
ChassisV4
SHEET
1/1
This drawing is our property; it can't be reproduced or communicated without our written agreement.


71.1
40
8.05
Bottom view
Scale: 1:1
R20
R5
Ø36.1
25.55
M5
15
65.55
20
M5
28.05
Front view
Scale: 1:1
10
96.1
Left view
Scale: 1:1
10
5
Top view
Scale: 1:1
DESIGNED BY:
Louis Pauchet
DATE:
08/04/2022
CHECKED BY:
XXX
DATE:
XXX
Support Treuille
Projet Robot - Groupe 42
SIZE
A3
INSA Rouen Normandie
SCALE
1:1
WEIGHT (kg)
0.13
DRAWING NUMBER
ChassisV4
SHEET
1/1
This drawing is our property; it can't be reproduced or communicated without our written agreement.


71.1
40
8.05
Bottom view
Scale: 1:1
R20
R5
Ø36.1
M5
25.55
15
65.55
20
M5
28.05
Front view
Scale: 1:1
10
96.1
Left view
Scale: 1:1
10
5
Top view
Scale: 1:1
DESIGNED BY:
Louis Pauchet
DATE:
08/04/2022
CHECKED BY:
XXX
DATE:
XXX
Support Treuille
Projet Robot - Groupe 42
SIZE
A3
INSA Rouen Normandie
SCALE
1:1
WEIGHT (kg)
0.13
DRAWING NUMBER
ChassisV4
SHEET
1/1
This drawing is our property; it can't be reproduced or communicated without our written agreement.


Isometric view
Scale: 1:4
ϕ33.5
ϕ27.5
30
M2
3
ϕ168.49
Bottom view
Scale: 1:1
52.5
ϕ5
52.5
105
105
Front view
Scale: 1:1
DESIGNED BY:
Louis Pauchet
DATE:
08/04/2022
CHECKED BY:
XXX
DATE:
XXX
SIZE
A3
Manchon Circulaire (V1)
Projet Robot - Groupe 42
INSA Rouen Normandie
SCALE
1:1
WEIGHT (kg)
0.13
DRAWING NUMBER
ChassisV4
SHEET
1/1
This drawing is our property; it can't be reproduced or communicated without our written agreement.

ANNEXE 6 Schéma Electrique du Robot



ANNEXE 7 Code du programme fonctionnel et testé

```
//Définition des pins utilisés pour les différentes fonctions

#define BoutonRose 2
#define MonteChargePWM 3
#define MonteChargeDir 5
#define MonteChargeDirMonter 0
#define MonteChargeVitesse 150
#define DetecteurFinDeCourseHaut 13
#define DetecteurFinDeCourseBas 12
#define MoteurDroitPWM 10
#define MoteurDroitDIR 8
#define MoteurDroitAvanceDir 0
#define MoteurGauchePWM 9
#define MoteurGaucheDir 7
#define MoteurGaucheAvanceDir 1
#define VitesseAvance 80
#define RelatifRotation 20
#define CapteurObstacle A1
#define DectionObstacleTension 150
#define CapteurIRCentre A3
#define IRSeuil 3


//Définition des variables
const int IRLenth = 100;
byte IRCentre[IRLenth] ;
byte rangIRCentre = 0;


bool ModeFonctionnement = 0;

//Définition des signatures des fonctions

void MonteCharge();
void ActionnerMoteurMonteCharge(bool, int);
int DetectionObstacle();
int DetectionIR();
void ActionnerMoteurDroit(bool, int);
void ActionnerMoteurGauche (bool, int);

//Définitions des boucles principales

void setup() {
    pinMode(BoutonRose, INPUT);
    attachInterrupt(0, changeMode, RISING);

    pinMode(CapteurObstacle, INPUT);
    pinMode(CapteurIRCentre, INPUT);
    pinMode(DetecteurFinDeCourseBas, INPUT);
    pinMode(DetecteurFinDeCourseHaut, INPUT);

    pinMode(MoteurDroitDIR, OUTPUT);
    pinMode(MoteurDroitPWM, OUTPUT);
    pinMode(MoteurGauchePWM, OUTPUT);
    pinMode(MoteurGaucheDir, OUTPUT);
    pinMode(MonteChargeDir, OUTPUT);
    pinMode(MonteChargePWM, OUTPUT);
}
```

```
void loop() {

ActionnerMoteurMonteCharge(MonteChargeDirMonter, 0);
    switch (DetectionObstacle())
    {
    case 1: //Cas de la détection d'un obstacle - On détecte un obstacle
donc on s'arrète
        ActionnerMoteurDroit(MoteurDroitAvanceDir, 0);
        ActionnerMoteurGauche(MoteurGaucheAvanceDir, 0);
        break;

    default:
        switch (DetectionIR())
        {
        case -1:
            /* code */
            break;

        case 0: //Le capteur centrale détecte un signal donc on avance
            ActionnerMoteurDroit(MoteurDroitAvanceDir, VitesseAvance);
            ActionnerMoteurGauche(MoteurGaucheAvanceDir, VitesseAvance);
            break;

        case 1:
            /* code */
            break;

        default:
            //ActionnerMoteurDroit(MoteurDroitAvanceDir, 0);
            //ActionnerMoteurGauche(MoteurGaucheAvanceDir, 0);
            break;
        }

        break;
    }

}

//Définitions des autres fonctions


void ActionnerMoteurMonteCharge(bool sens, int vitesse) {
    digitalWrite(MonteChargeDir, sens);
    analogWrite(MonteChargePWM, vitesse);
    digitalWrite(MonteChargeDir, sens);
}

void ActionnerMoteurDroit(bool sens, int vitesse) {
    digitalWrite(MoteurDroitDIR, sens);
    analogWrite(MoteurDroitPWM, vitesse);
    digitalWrite(MoteurDroitDIR, sens);
}

void ActionnerMoteurGauche(bool sens, int vitesse) {
    digitalWrite(MoteurGaucheDir, sens);
    analogWrite(MoteurGauchePWM, vitesse);
    digitalWrite(MoteurGaucheDir, sens);
}
```

```
int DetectionObstacle() {
    if (analogRead(CapteurObstacle) > DectionObstacleTension) return 1;
    else return 0;

}

void LectureIRCentre() {
    IRCentre[rangIRCentre] = !digitalRead(CapteurIRCentre);
    if (rangIRCentre < IRLenth) rangIRCentre ++;
    else rangIRCentre = 0;
}

int sum(byte* tab) {
    int somme = 0;
    for(int i = 0; i<IRLenth; i++) {
        somme += tab[i];
    }
    return somme;
}


int DetectionIR() {
    LectureIRCentre();
    if (sum(IRCentre) > IRSeuil) return 0;
    else return 5;
}

void changeMode() {
    ModeFonctionnement = !ModeFonctionnement;
    if (ModeFonctionnement) MonteCharge();
}

void MonteCharge() {
    ActionnerMoteurDroit(MoteurDroitAvanceDir, 0); //On arrete les moteurs
    ActionnerMoteurGauche(MoteurGaucheAvanceDir, 0);
    ActionnerMoteurMonteCharge(MonteChargeDirMonter, MonteChargeVitesse);
//On alume le moteur en monté
    while (!digitalRead(DetecteurFinDeCourseHaut)) //On entre dans une
boucle infinie tant que le détecteur de fin de course n'est pas actionné
    {
    }
    ActionnerMoteurMonteCharge(MonteChargeDirMonter, 0); //On arréte le mo-
teur

    while (!digitalRead(BoutonRose)) //On attend que le bouton soit appuyé
    {
    }


    ActionnerMoteurMonteCharge(!MonteChargeDirMonter, MonteChargeVitesse);
//Actionne le monteur en descente
    while (!digitalRead(DetecteurFinDeCourseBas)) //On attend le fin de
course du bas
    {
    }
    ActionnerMoteurMonteCharge(MonteChargeDirMonter, 0); //On arrête le mo-
teur
    return;
}
```

ANNEXE 8 Code théorique pour les trois capteurs (non testé)

```
//Définition des pins utilisés pour les différentes fonctions

#define BoutonRose 2
#define MonteChargePWM 3
#define MonteChargeDir 5
#define MonteChargeDirMonter 0
#define MonteChargeVitesse 70
#define DetecteurFinDeCourseHaut 13
#define DetecteurFinDeCourseBas 12
#define MoteurDroitPWM 10
#define MoteurDroitDIR 8
#define MoteurDroitAvanceDir 0
#define MoteurGauchePWM 9
#define MoteurGaucheDir 7
#define MoteurGaucheAvanceDir 1
#define VitesseAvance 50
#define RelatifRotation 20
#define CapteurObstacle A1
#define DectionObstacleTension 150
#define CapteurIRCentre A3
#define CapteurIRDroit 20
#define CapteurIRGauche 20
#define IRSeuil 3


//Définition des variables
const int IRLenth = 100;
byte IRCentre[IRLenth] ;
byte rangIRCentre = 0;

byte IRDroit[IRLenth] ;
byte rangIRDroit = 0;

byte IRGauche[IRLenth] ;
byte rangIRGauche = 0;


bool ModeFonctionnement = 0;

//Définition des signatures des fonctions

void MonteCharge();
void ActionnerMoteurMonteCharge(bool, int);
int DetectionObstacle();
int DetectionIR();
void ActionnerMoteurDroit(bool, int);
void ActionnerMoteurGauche (bool, int);

//Définitions des boucles principales

void setup() {
    pinMode(BoutonRose, INPUT);
    attachInterrupt(0, changeMode, RISING);

    pinMode(CapteurObstacle, INPUT);
    pinMode(CapteurIRCentre, INPUT);
```

```
    pinMode(CapteurIRDroit, INPUT);
    pinMode(CapteurIRGauche, INPUT);
    pinMode(DetecteurFinDeCourseBas, INPUT);
    pinMode(DetecteurFinDeCourseHaut, INPUT);

    pinMode(MoteurDroitDIR, OUTPUT);
    pinMode(MoteurDroitPWM, OUTPUT);
    pinMode(MoteurGauchePWM, OUTPUT);
    pinMode(MoteurGaucheDir, OUTPUT);
    pinMode(MonteChargeDir, OUTPUT);
    pinMode(MonteChargePWM, OUTPUT);
}



void loop() {

ActionnerMoteurMonteCharge(MonteChargeDirMonter, 0);
    switch (DetectionObstacle())
    {
    case 1: //Cas de la détection d'un obstacle - On détecte un obstacle
donc on s'arrète
        ActionnerMoteurDroit(MoteurDroitAvanceDir, 0);
        ActionnerMoteurGauche(MoteurGaucheAvanceDir, 0);
        break;

    default:
        switch (DetectionIR())
        {
        case -1: //droit
            ActionnerMoteurDroit(MoteurDroitAvanceDir, VitesseAvance - Re-
latifRotation);
            ActionnerMoteurGauche(MoteurGaucheAvanceDir, VitesseAvance +
RelatifRotation);
            break;

        case 0: //Le capteur centrale détecte un signal donc on avance
            ActionnerMoteurDroit(MoteurDroitAvanceDir, VitesseAvance);
            ActionnerMoteurGauche(MoteurGaucheAvanceDir, VitesseAvance);
            break;

        case 1: //Gauche
            ActionnerMoteurDroit(MoteurDroitAvanceDir, VitesseAvance + Re-
latifRotation);
            ActionnerMoteurGauche(MoteurGaucheAvanceDir, VitesseAvance -
RelatifRotation);
            break;

        default:
            break;
        }

        break;
    }

}



//Définitions des autres fonctions
```

```
void ActionnerMoteurMonteCharge(bool sens, int vitesse) {
    digitalWrite(MonteChargeDir, sens);
    analogWrite(MonteChargePWM, vitesse);
    digitalWrite(MonteChargeDir, sens);
}

void ActionnerMoteurDroit(bool sens, int vitesse) {
    digitalWrite(MoteurDroitDIR, sens);
    analogWrite(MoteurDroitPWM, vitesse);
    digitalWrite(MoteurDroitDIR, sens);
}

void ActionnerMoteurGauche(bool sens, int vitesse) {
    digitalWrite(MoteurGaucheDir, sens);
    analogWrite(MoteurGauchePWM, vitesse);
    digitalWrite(MoteurGaucheDir, sens);
}

int DetectionObstacle() {
    if (analogRead(CapteurObstacle) > DectionObstacleTension) return 1;
    else return 0;

}

void LectureIRCentre() {
    IRCentre[rangIRCentre] = !digitalRead(CapteurIRCentre);
    if (rangIRCentre < IRLenth) rangIRCentre ++;
    else rangIRCentre = 0;
}

void LectureIRDroit() {
    IRDroit[rangIRDroit] = !digitalRead(CapteurIRDroit);
    if (rangIRDroit < IRLenth) rangIRDroit ++;
    else rangIRDroit = 0;
}

void LectureIRGauche() {
    IRDroit[rangIRGauche] = !digitalRead(CapteurIRGauche);
    if (rangIRGauche < IRLenth) rangIRGauche ++;
    else rangIRGauche = 0;
}

int sum(byte* tab) {
    int somme = 0;
    for(int i = 0; i<IRLenth; i++) {
        somme += tab[i];
    }
    return somme;
}


int DetectionIR() {
    LectureIRCentre();
    LectureIRGauche();
    LectureIRDroit();
    int centre = sum(IRCentre);
    int gauche = sum(IRGauche);
    int droit = sum(IRDroit);

    if ((centre > IRSeuil) || (gauche > IRSeuil) || (droit > IRSeuil)) {
```

```
        if ((centre >= gauche) && (centre >= droit)) return 0;
        if ((gauche > centre) && (gauche > droit)) return 1;
        if ((droit > centre) && (droit > gauche)) return -1;

    }

    else return 5;
}

void changeMode() {
    ModeFonctionnement = !ModeFonctionnement;
    if (ModeFonctionnement) MonteCharge();
}

void MonteCharge() {
    ActionnerMoteurDroit(MoteurDroitAvanceDir, 0); //On arrete les moteurs
    ActionnerMoteurGauche(MoteurGaucheAvanceDir, 0);
    ActionnerMoteurMonteCharge(MonteChargeDirMonter, MonteChargeVitesse);
//On alume le moteur en monté
    while (!digitalRead(DetecteurFinDeCourseHaut)) //On entre dans une
boucle infinie tant que le détecteur de fin de course n'est pas actionné
    {
    }
    ActionnerMoteurMonteCharge(MonteChargeDirMonter, 0); //On arréte le mo-
teur

    while (!digitalRead(BoutonRose)) //On attend que le bouton soit appuyé
    {
    }


    ActionnerMoteurMonteCharge(!MonteChargeDirMonter, MonteChargeVitesse);
//Actionne le monteur en descente
    while (!digitalRead(DetecteurFinDeCourseBas)) //On attend le fin de
course du bas
    {
    }
    ActionnerMoteurMonteCharge(MonteChargeDirMonter, 0); //On arrête le mo-
teur
    return;
}
```

ANNEXE 9 Quelques photos de la fabrication

Figure 29 : Visualisation de l'espace

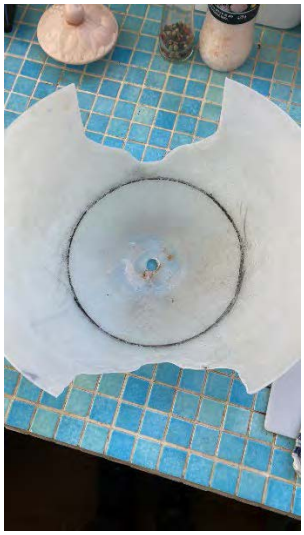


Figure 30 : Marquage du châssis

Figure 31 : Peinture de la demi-sphère

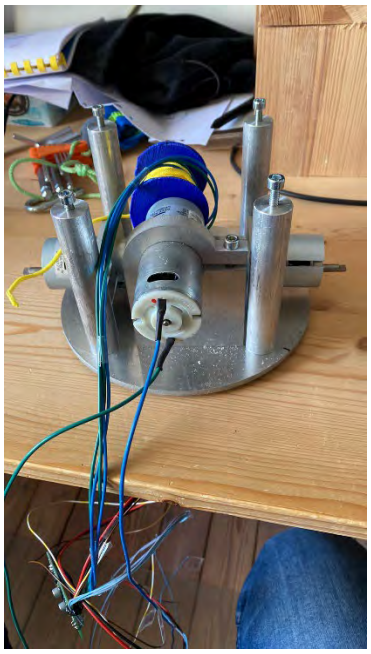


Figure 32 : Assemblage et soudure des composants du châssis

Figure 33 : Montage du nez du robot

Figure 34 : Peinture de la tête

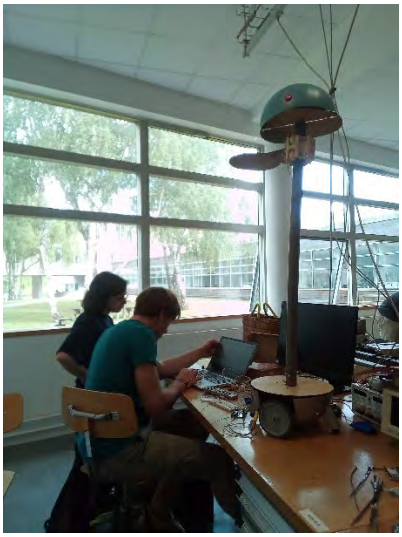


Figure 35 : Séance de travail

Figure 36 : Tournage des axes des roues en dernière minute

Figure 37 : Taraudage des trous pour vis pointaux