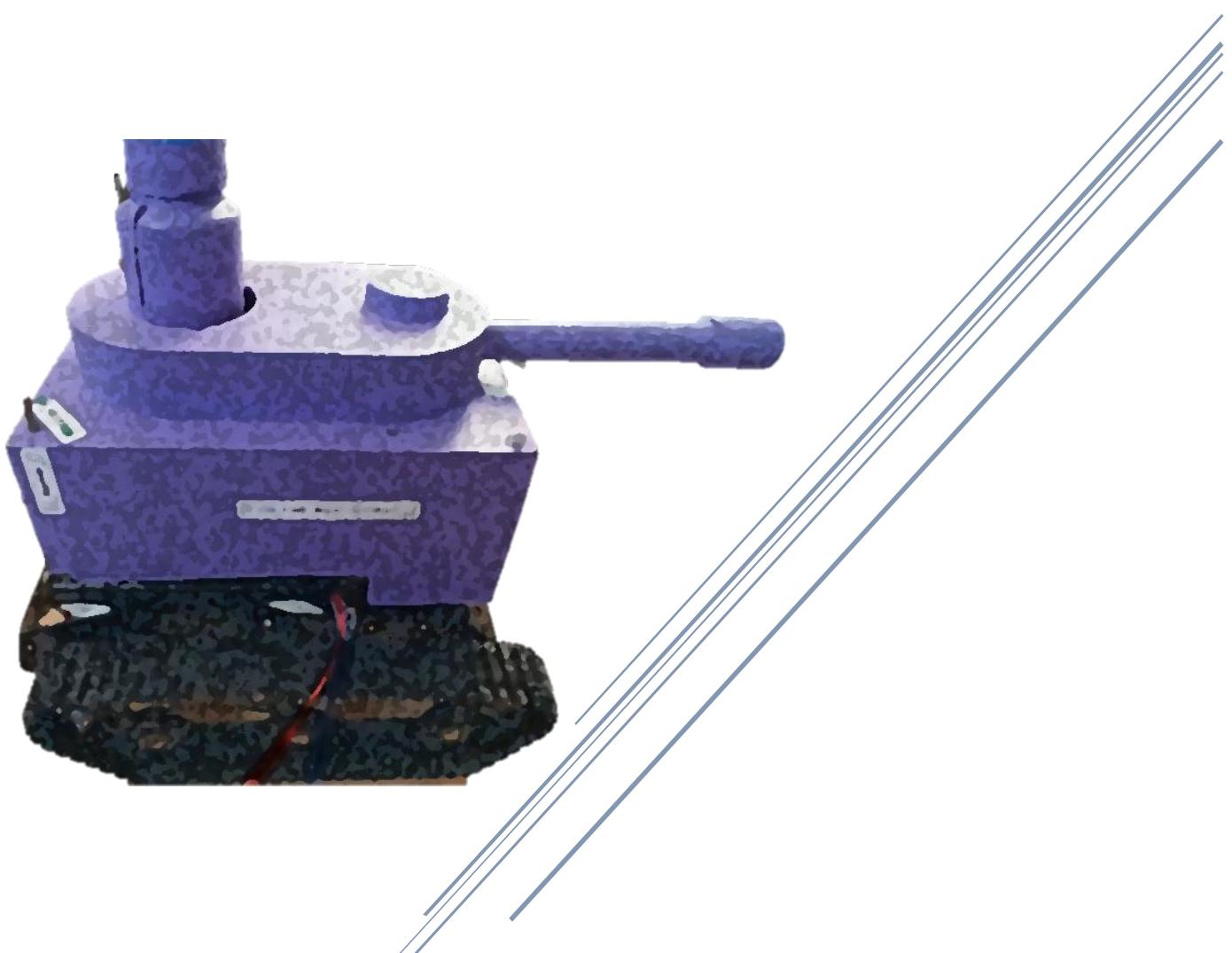


RAPPORT DE PROJET TUTORÉ

Étude, conception et réalisation d'un char bi-mode à jet d'eau



MAMILONNE Didier

VINCENT Isaac

2020-2021

Formation : Licence Pro Mécatronique en alternance

SOMMAIRE

SOMMAIRE	
I. PRÉSENTATION	1
1. Contexte et objectifs du projet	1
2. Problématique	1
3. Solution apportée	1
II. GESTION DU PROJET	2
1. Ressources humaines	2
2. Planification et répartition prévisionnelles des tâches	2
3. Planification et répartition réelles des tâches	3
III. ANALYSE DU PROJET	5
1. Les besoins et les contraintes	5
2. L' analyse des risques	7
3. L'analyse environnementale, les fonctions et les liens inter-projets	8
IV. ÉTUDE DU PROJET	9
1. Études matérielles, logicielles et choix retenu(s)	9
2. Le cahier des charges	13
V. CONCEPTION ET RÉALISATION DU CHAR BI-MODE À JET D'EAU	16
1. Conceptions mécaniques de la carrosserie du char, du boîtier de la manette, du bac de rétention d'eau et des languettes de fixation	17
2. Conceptions électriques/électroniques du char, de la manette et des PCB	22
3. Conceptions et réalisations des programmes pour les Arduino de la manette et du char	26
4. Réalisation du réservoir d'eau, du montage électrique, et mécanique de la manette et du char	30
1. Tests, essais	41
VI. DIFFICULTÉS RENCONTRÉES	42
1. Retard et modification de la planification initiale	42
2. Difficultés organisationnelles	43
VII. CONCLUSION	44
1. Évolutions possibles du projet	44
2. Apports personnels et/ou professionnels	44
VIII. ANNEXES	45

I. PRÉSENTATION

Pour débuter ce rapport, nous parlerons dans un premier temps du contexte et des objectifs qui nous ont poussés à réaliser ce projet, puis deuxièmement nous introduirons la problématique, et enfin nous expliquerons la solution que nous avons choisi pour répondre à cette problématique.

1. Contexte et objectifs du projet

Au cours de notre année en licence professionnelle mécatronique à l'IUT de Cergy-Neuville, nous devons réaliser un projet durant à peu près 6 mois allant du mois de janvier 2020 jusqu'au mois de juin 2021. Celui-ci devait être réalisé par groupe de deux élèves, afin que chacun du groupe puisse apporter son savoir-faire théorique et pratique.

Ce projet a pour objectif principal de nous apporter les compétences exigées par le référentiel national de ce diplôme. Par conséquent, il était nécessaire de réaliser un projet alliant différentes disciplines tel que la mécanique, en passant par l'électronique et l'informatique ; il devait avoir aussi une finalité ludique pour que nous prenions plaisir à le réaliser.

Pour nous, ce projet reste un moyen d'obtenir notre diplôme, mais aussi de nous développer personnellement et professionnellement, car il nous apporte l'ambition et l'envie de finaliser une tâche entamée, ainsi que le goût du travail en équipe, l'ambition, l'organisation et la confiance en soi.

2. Problématique

Pour réaliser ce projet nous devons faire face à la problématique qu'elle soulève et que doit répondre bon nombre de mécatronicien, c'est-à-dire l'étude, la conception, la réalisation et la mise en fonctionnement d'un dispositif regroupant des parties électriques, mécaniques et informatiques, en maîtrisant les phénomènes physiques imprévus que cette union peut engendrer, et le tout en respectant les délais.

3. Solution apportée

Après plusieurs réflexions nous nous sommes orientés sur l'étude, la conception et la réalisation d'un char bi-mode. Plus précisément, il fonctionnera selon deux modes : un mode télécommandé (grâce à une manette), et un mode asservi (suivant une ligne). Il devra être léger et de petite envergure, comme la plupart des mini-char que l'on voit dans les hypermarchés par exemple. Pour le côté ludique, nous nous sommes penchés dans un premier temps sur un char à jet de feu, mais vu les dangers apportés par cette idée (possibilités de brûlures ou d'explosion), nous nous sommes rétractés pour nous tourner sur le jet d'eau.

Nous trouvions cette idée de projet intéressante, car elle reste complexe à concevoir et à réaliser tant sur le plan technique (pluridisciplinarité), quel sur le plan organisationnel (gestion du temps et des tâches). Sa réalisation nécessite une part importante des compétences exigées par le secteur de la mécatronique tel que la contrôle-commande (ex : par la manette), l'électrotechnique (ex : par les moteurs ou la pompe), et la programmation informatique des Arduino.

La gestion du projet reste la suite logique que nous devons suivre.

II. GESTION DU PROJET

La gestion est une étape cruciale dans un projet, elle permet en effet un gain de temps et d'efficacité. Pour notre projet, elle s'étend sur trois parties que nous présenterons ci-dessous : les ressources humaines disponibles, la planification et répartition prévisionnelles des tâches et la planification et répartition réelle des tâches.

1. Ressources humaines

Pour entreprendre et mener à terme ce projet nous serons un groupe de deux élèves :

- Didier
- Isaac

2. Planification et répartition prévisionnelles des tâches

Avant de commencer la réalisation de notre planning GANTT sur le logiciel Mindview7.0, nous avons réfléchi, puis fait la liste des différentes tâches à effectuer que nous avons ensuite classés dans un ordre logique et chronologique. Ces tâches sont :

01	- L'étude préalable du système
02	- La commande et la réception du matériel
03	- La conception du char et de la télécommande
04	1 - La conception mécanique de la carrosserie du char
05	2 - La conception mécanique de la télécommande
06	- La conception des circuits électroniques
07	1 - La conception du mode char « télécommandé »
08	2 - La conception du mode char « asservi »
09	- L'impression 3D de la carrosserie du char et de la télécommande
10	- La réalisation du char
11	1 - Le montage mécanique

12	2 - Le montage électrique/électronique
13	3 - La programmation d'asservissement des Arduino
14	- Les tests et essais de fonctionnement
15	- Les rédactions et présentations
16	1 - La rédaction du manuel utilisateur
17	2 - La rédaction du rapport de projet
18	3 - Réalisation de la présentation Powerpoint
19	Jalon 1 : La remise du cahier des charges
20	Jalon 2 : Le point d'avancement des conceptions (livrable #1)
21	Jalon 3 : Le point d'avancement de la finalisation des conceptions (livrable #2)
22	Jalon 4 : Le point d'avancement d'exécution du projet (livrable #3)

Puis à l'aide de Mindview7.0 nous avons élaboré la planification GANTT , en estimant une durée approximative pour chaque tâche, mais aussi une priorité et les ressources allouées.

En ce qui concerne la répartition prévisionnelle des tâches, nous avons estimé que pour répartir le plus équitablement les tâches, elles seront allouées théoriquement à 50 % par ressource.

Voir [ANNEXE 1](#) : Diagramme de GANTT prévisionnel et diagramme des ressources.

3. Planification et répartition réelles des tâches

Le diagramme de GANTT et diagramme des ressources en [ANNEXE 2](#), représentent la planification et la répartition des tâches réelles et finales, que nous comparons par la suite à la planification et répartition prévisionnelles précédente. On remarque sur la nouvelle liste des tâches ci-dessous que de nombreux changements ont été apportés (que j'ai surligné en gris pour une meilleure visibilité) :

01	- L'étude préalable du système – cahier des charges
02	- La commande du matériel
03	- La réception du matériel
04	- La conception mécanique du char, de la télécommande, du boitier de rétention d'eau et des supports de fixation
05	1 - La conception mécanique de la carrosserie du char
06	2 - La conception mécanique de la télécommande
07	2 - La conception du boitier de rétention d'eau et des supports de fixation
08	- La conception électriques du char, de la manette, des PCB et conception des programmes Arduino
09	1 - La conception de la manette et des PCB
10	2 - La conception du char
11	3 - La conception des programmes Arduino
12	- L'impression 3D de la carrosserie du char, de la télécommande, du boitier de rétention d'eau et des supports de fixation
13	- La réalisation du char
14	1 - Création des programmes Arduino
15	2 - Le montage mécanique
16	3 - Le montage électrique/électronique

17	- Les finalisations, tests et essais de fonctionnement
18	- Les rédactions et présentations
19	1 - La rédaction du manuel utilisateur
20	2 - La rédaction du rapport de projet
21	3 - Réalisation de la présentation Powerpoint
22	Jalon 1 : La remise du cahier des charges
01	Jalon 2 : Le point d'avancement des conceptions (livrable #1)
02	Jalon 3 : Le point d'avancement de la finalisation des conceptions (livrable #2)
03	Jalon 4 : Le point d'avancement d'exécution du projet (livrable #3)

Ligne 1 : le terme « cahier des charges » a été rajouté, pour plus de précision et pour éviter des ambiguïtés.

Ligne 3 : nous avons séparé l'étape de « commande et de réception du matériel » en deux parties distinctes, une partie « commande du matériel » et une partie « réception du matériel », pour souligner le fait que la commande est une étape finalisée à 100%, mais quel est impacté par les retards de réception de certains composants.

Ligne 4 : durant le déroulement de la conception de la carrosserie du char, nous nous sommes rendu compte nous aurions des difficultés à trouver un moyen de le fixer au châssis, de ce fait nous avons décidé de créer nous-mêmes un moyen de fixation châssis-carrosserie. Nous avons aussi décidé de concevoir un bac de rétention d'eau dans le cas où il y aurait une fuite du réservoir, pour la localiser mais aussi pour protéger l'ensemble des circuits électroniques.

Ligne 8 : dans la partie conception électrique, nous avons décidé de rajouter la conception des PCB la conception des programmes Arduino. Pour les PCB, nous cherchions un moyen de pouvoir distribuer les alimentations aux différents circuits du char ; sachant que l'utilisation de borniers de connexion électrique de type WAGO ne fonctionne pas avec des conducteurs d'âme conductrice de section de 0,2 mm², l'utilisation de PCB pour souder les conducteurs dessus est une solution viable pour répondre à notre besoin. Le « repiquage » d'alimentation par la soudure de plusieurs conducteurs entre eux en revanche n'était pas une solution fiable à long terme, et reste moins pratique.

Ligne 9 : la conception électrique de la manette s'étant déroulé rapidement, nous avons « pioché » du temps dans cette tâche pour réaliser la conception des PCB.

Ligne 11 : nous ne disposions pas encore du matériel nécessaire pour commencer les différentes étapes de réalisation, donc nous en avons profité pour réaliser l'étape de la conception des programmes Arduino.

Ligne 12 : on tenait à préciser les différentes impressions 3D réalisées.

Ligne 17 : le terme « finalisation » a été rajouté et elle comprend différentes étapes (collages ou fixations de cartes ou de composants, résolutions des imprévus, téléchargement du programme final sur chaque Arduino...).

À noter qu'il reste difficile de quantifier en pourcentage la quantité de travail réelle réalisée, cependant nous avons pu y attribuer une valeur approximative (voir répartitions réelles des tâches en [ANNEXE 2](#)).

À la fin de cette étape, nous pouvons poursuivre sur l'analyse du projet.

III. ANALYSE DU PROJET

Pour mener à bien notre projet, nous avons utilisé différents outils d'analyse comme le SWOT, le QQCOQPC, la BÊTE À CORNES ou encore le diagramme d'HISHIKAWA.

L'analyse c'est quoi ? L'analyse est l'examen de l'information (trier, additionner, comparer) pour mieux comprendre les relations entre le "tout" et les "parties".

Qu'est-ce qu'un outil d'analyse ? C'est l'ensemble des documents facilitant la planification et l'exécution d'un travail. Pourquoi et comment les choisir ?

Il existe aujourd'hui un nombre important d'outils d'analyse pour faire les bons choix ; il faut se poser les bonnes questions dès le début du projet. Dans notre cas, nous nous sommes demandé par exemple « si nous devions commercialiser notre char, quelles clientèles viserions-nous ? ». À la suite de notre choix qui se tourne (en cas de vente) vers une clientèle plutôt jeune, nous avons commencé à réfléchir à comment atteindre nos objectifs et c'est à ce moment que les outils d'analyse rentrent en jeu.

1. Les besoins et les contraintes

- **Le QQCOQPC**

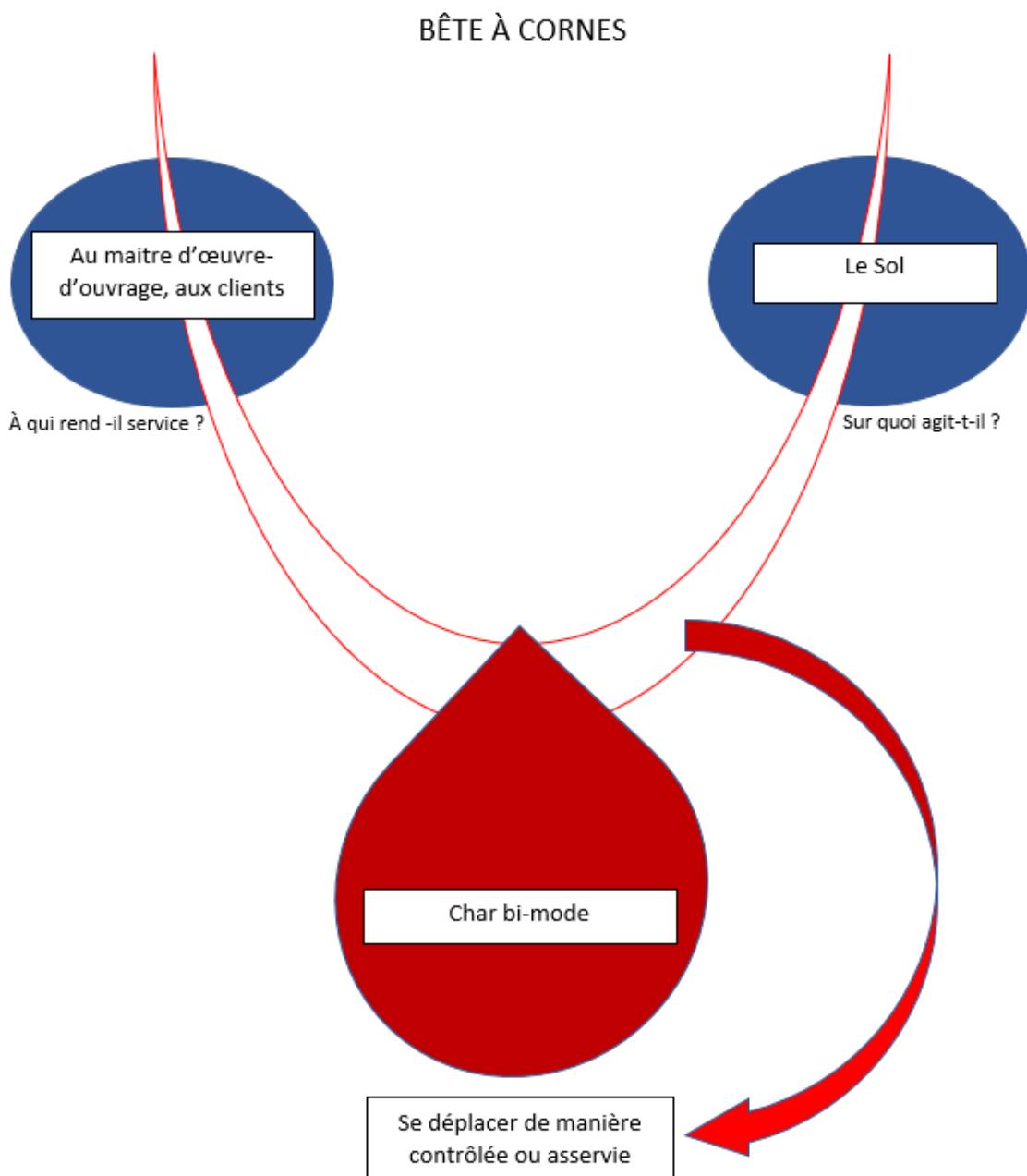
Le QQCOQPC (Qui, Quoi, Combien, Ou, Quand, Pourquoi, Comment), appelé aussi méthode du questionnement est un outil d'aide à la résolution de problèmes comportant une liste quasi exhaustive d'informations sur la situation. Elle nous aide aussi à définir les trois principales contraintes de notre projet :

Situation : Réalisation d'un Char Bi-mode à jet d'eau, dans le cadre du projet de fin d'année de notre licence professionnelle mécatronique		
	Qui ?	MOE (Didier et Isaac)
	Quoi ?	Réalisation d'un Char Bi-mode à jet d'eau
Contrainte de budget	Combien ?	Ressources à définir (budget allouer par l'IUT)
	Où ?	IUT
Contrainte de temps (délai)	Quand ?	Du 13 janvier au 11 juin 2021
	Pourquoi ?	Pour compléter notre formation, acquérir un savoir et des compétences dans le domaine technique et en gestion de projet ; en cas de commercialisation future du projet, faire du profit

Contrainte de qualité	Comment ?	Faire l'étude, la conception, et la réalisation du char bi-mode à jet d'eau, en utilisant du matériel adapté ayant un bon rapport qualité-prix
-----------------------	------------------	--

- **La bête à cornes**

Nous avons aussi utilisé l'outils « bête à cornes » qui est un outil d'analyse fonctionnelle du besoin. En matière d'innovation, il est tout d'abord nécessaire de formuler le besoin sous forme de fonctions simples (dans le sens de « fonctions de bases ») que doit remplir le produit ou le service innovant.



Après l'étude des besoins et contraintes, nous passons à l'analyse des risques.

2. L' analyse des risques

Lors de l'élaboration du projet, il est très important d'analyser les risques pour pouvoir pallier toute éventualité et pour éviter l'échec de celui-ci. Nous utiliserons dans un premier temps l'outil S.W.O.T pour synthétiser les forces et faiblesses liée à ce projet au regard des menaces et opportunités qui y sont liées, puis dans un second temps nous afficherons sous forme de tableau, les difficultés et opportunités majeures du projet et les solutions possibles, et pour terminer on présentera le diagramme d'ISHIKAWA pour notre projet.

- **Le S.W.O.T**

Le SWOT va nous servir à lister les forces, les faiblesses, les opportunités et les menaces de notre projet. Nous avons choisi de le présenter sous la forme du tableau ci-dessous :

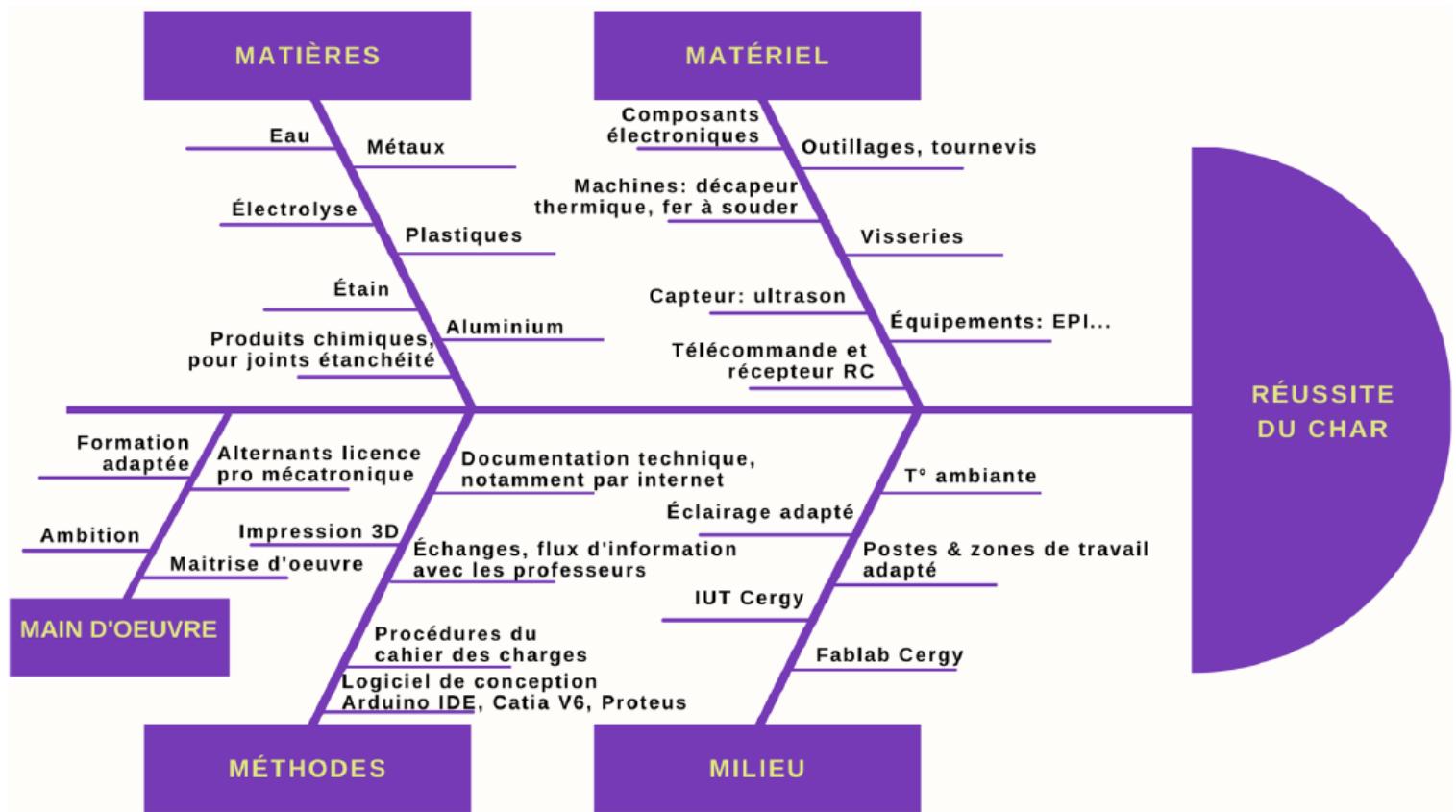
DIAGRAMME S.W.O.T			
<ul style="list-style-type: none"> - Accessible au plus grand nombre (si mise en commercialisation) - idée novatrice du mode asservi - Léger et compacte - Ludique 	 <p>INTERNAL • STRENGTH • WEAKNESS</p> <p>EXTERNAL • OPPORTUNITY • THREAT • POSITIVE • NEGATIVE</p>	<ul style="list-style-type: none"> - Distance de fonctionnement limitée - Adapté à une clientèle qui a un besoin bien précis. - Fonctions limitées par la taille - Ce que les concurrents font mieux que nous 	<ul style="list-style-type: none"> - Présence de concurrents sur le marché - Consommateurs : risque de lassitude - Environnement réglementaire si mise en vente

- **Les risques et parades**

Le tableau en [ANNEXE 3](#) représente les dangers et opportunités liées à ce projet, les risques potentiels qui s'en suivent, et les parades de prévention/protection à utiliser, permettant d'améliorer le degré d'acceptabilité de ces risques et permettre ainsi la continuité du projet.

- **Les causes et effets**

Nous avons aussi eu recours au diagramme de causes et effets, ou diagramme d'Ishikawa, (ou encore dénommé diagramme en arêtes de poisson ou encore 5M). C'est un outil développé par Kaoru Ishikawa en 1962 et servant dans la gestion de la qualité. Il nous permet de visualiser globalement l'environnement de notre projet.



On poursuit notre analyse sur le plan environnemental, mais aussi sur les fonctions et les liens inter-projets.

3. L'analyse environnementale, les fonctions et les liens inter-projets

- **L'analyse environnementale**

L'ISO 14001 définit l'analyse environnementale comme l'analyse des activités, des produits et services d'un organisme susceptibles d'avoir un impact sur l'environnement. À partir de cela, on peut établir notre analyse, et plus esthétiquement sous la forme de tableau située à l'[ANNEXE 4](#).

- **Les différentes fonctions du char**

Le diagramme FAST est une méthode qui va nous permettre de décrire les différentes fonctions (à différentes échelles) de la réalisation de notre char bi-mode.

- **Liens inter-projets**

L'image en [ANNEXE 5](#) représente les liens inter-projets ; plus précisément les liens entre les différentes tâches nécessaires à l'accomplissement du projet, ainsi que les sous-branches de niveau 1 et 2 associées leur tâche respective.

Après cette analyse poussée, on peut désormais passer à l'étude du projet, étape elle aussi très importante.

IV. ÉTUDE DU PROJET

Pour nous permettre de réaliser notre char bi-mode, nous allons devoir étudier et connaître le matériel que nous allons utiliser et faire un choix parmi les différentes possibilités qui s'offre à nous, avant d'archiver tout ce que nous avons vu jusqu'à présent dans le cahier des charges.

1. Études matérielles, logicielles et choix retenu(s)

Pour faire le choix de notre matériel, nous avons décidé, pour nous aider, de séparer notre projet par groupe et par fonction (pour chacun des groupes) et que l'on représente dans le tableau ci-dessous (pour l'étude des consommables et des éléments de protection tels que les composants électroniques de base, voir [ANNEXE 6](#)) :

Groupe 1 : MANETTE (ou télécommande)		Groupe 2 : CHAR (ou tank)	
Fonction 1	Alimentation et régulation (commande)	Fonction 1	Alimentation et régulation + signalisation (puissance + commande)
Fonction 2	Émission et transmission du signal (de l'information)	Fonction 2	Réception et transmission du signal (de l'information)
Fonction 3	Choix et signalisation du mode (télécommandé ou asservi)	Fonction 3	Choix et signalisation du mode (télécommandé ou asservi)
Fonction 4	Unité de traitement	Fonction 4	Unité de traitement
Fonction 5	Boîtier de la manette (dimensions correctement choisies)	Fonction 5	Jet d'eau (réservoir, pompe...)
		Fonction 6	Suiveur de ligne et télécommandé
		Fonction 7	Détection d'obstacle + signalisation (sonore + lumineuse)
		Fonction 9	Châssis, chenilles, et entraînement du char
		Fonction 10	Carrosserie du char

Après avoir conçu ce tableau, il nous faut dès à présent faire le listing pour chaque fonction de groupe, des différents choix d'éléments dont nous aurons besoin pour accomplir cette ou ces tâche(s). Pour savoir et choisir quelle fonction est prioritaire parmi toutes les autres, on peut se baser sur le corps

humain. Sachant que l'on peut comparer un système automatisé à un être vivant, on peut donc affirmer que le cerveau est l'élément à prendre en compte en priorité, et de ce fait on débutera par la fonction 4 qui est l'unité de traitement et de calcul pour chaque groupe.

- Étude des choix matériels de l' « unité de traitement » pour le groupe 1 et 2 :**

Pour les deux groupes, nous hésitions entre trois unités de traitements très connues que sont l'Arduino, le Raspberry et l'API (Automate Programmable industrielle). Nous avons donc décidé de faire une comparaison entre ces éléments dans le tableau ci-dessous :

	ARDUINO	RASPBERRY	AUTOMATE
Avantage(s)	<ul style="list-style-type: none"> - Type : Microcontrôleur - Simplicité - Prêt à l'emploi - Léger - Pas cher 	<ul style="list-style-type: none"> - Type : Ordinateur monocarte - Convient pour les tâches simples et compliquées - Léger - Pas cher 	<ul style="list-style-type: none"> - Type: A.P.I - Robuste - Facilité de programmation
Inconvénient(s)	<ul style="list-style-type: none"> - Niveau d'apprentissage faible en électronique et en programmation - Fragile 	<ul style="list-style-type: none"> - Le logiciel doit être télécharger et installé par l'utilisateur - Fragile 	<ul style="list-style-type: none"> - Cher - Lourd

Sources : <https://www.ionos.fr/digitalguide/serveur/know-how/arduino-vs-raspberry-pi/> et <https://www.automation-sense.com/blog/automatisme/differences-entre-microcontroleur-et-automate.html>

L'élément le plus important qui a pesé sur notre décision reste le poids, le char devais être léger et de petite envergure, donc l'automate n'est pas la bonne décision. Nous avons finalement choisi l'Arduino pour sa simplicité globale. Sa fragilité peut être réduite par une coque de protection.

- Étude des choix matériels de l' « alimentation et de la régulation » pour le groupe 1 et 2 :**

Ils seront résumés sous forme de tableau comme précédemment :

	Groupe 1 (Manette)		Groupe 2 (Char)	
	Commande	Puissance	Commande	Puissance
Carte Arduino (choix initial)	Uno	N/A	Uno	N/A
Carte Arduino (choix final par maître d'ouvrage)	Nano	N/A	Méga	N/A
Alimentation (choix initial et final)	10 – 12 V	N/A	10 – 12 V	10 – 12 V
Régulation (choix initial)	10 – 12 V → 8V	N/A	10 – 12 V → 8V	N/A
Régulation (choix final)	10 – 12 V → 5V	N/A	10 – 12 V → 8V	N/A

Nous soulignons le fait qu'au départ nous avons opté pour des piles AA (Le format AA possède une bonne quantité de charges électriques $\geq 2000 \text{ mAh}$), mais pour des raisons d'écoconception nous nous sommes finalement penchés vers des accumulateurs rechargeables (de même dimension que les piles AA LR06).

- Étude des choix matériels de l' « émission et la transmission du signal » pour le groupe 1 et 2 :**

Pour cette étude plusieurs choix s'offraient à nous dont un choix filaire et deux choix sans fil :

Types de connexion :	Filaire (liaison série)	Sans fil (liaison série)	Sans fil (liaison série)
Protocole/forme :	N/A	Bluetooth	Wifi
Avantages :	<ul style="list-style-type: none"> Pas de pertes de signal Signal fluide et rapide, pas de latence 	<ul style="list-style-type: none"> Bonne mobilité (de 10 à 200 mètres, dépend de la version et des conditions : murs...) 	<ul style="list-style-type: none"> Bonne mobilité (de 35 en intérieur à 250 mètres en extérieur)
Inconvénients :	<ul style="list-style-type: none"> Mobilité très limitée (longueur des conducteurs) Les conducteurs ou la soudure peuvent se casser (mouvements) 	<ul style="list-style-type: none"> Perte de signal possible Signal pas toujours fluide et rapide, latence Consommation électrique < Wifi 	<ul style="list-style-type: none"> Perte de signal possible Signal pas toujours fluide et rapide, latence Consommation électrique > Bluetooth
Choix final :	<p>Nous avons finalement choisi la liaison série sans fil Bluetooth, car elle nous permet de contrôler le char sur une distance assez conséquente, tout en consommant peu d'énergie. Il se trouve être le bon compromis entre la liaison sans-fil Wi-Fi et la liaison filaire.</p>		
Sources des données :	https://www.logiquetechno.com/distance-signal-bluetooth/ https://pro.orange.fr/actualites/5-chooses-que-vous-ignorez-a-propos-du-wifi-CNT000000JCIFq.html		

Les données envoyées et reçues par ces modules Bluetooth, sont les informations ou signaux liés au déplacements du char, qui sont réalisés grâce à un joystick situé sur la manette. Le choix de ce joystick se porte sur un module joystick adapté à l'Arduino.

- Étude des choix matériels permettant la « commutation des modes » pour le groupe 1 et 2 :**

Dans cette partie, une étude approfondie n'est pas nécessaire, il suffit juste d'utiliser un commutateur deux positions : on sera donc soit en mode télécommandé (position 1), soit en mode asservi (position deux). La signalisation se fera par deux LED, dont le courant sera limité par des résistances.

- Étude des choix matériels du « boîtier de la manette » pour le groupe 1 :**

Pour réduire les coûts de commande, nous avons décidé de produire nous-mêmes notre boîtier. Nous l'avons étudié puis conçu, cependant, nous avons pu récupérer bien plus tard un boîtier existant à l'IUT, dont les dimensions sont correctes et assez importantes pour intégrer nos différents dispositifs.

- Étude des choix de la fonction « jet d'eau » (ou GDO en raccourci) pour le groupe 2 :**

Pour étudier la fonction « jet d'eau » on peut s'aider du principe de la chaîne fonctionnelle de ce système.



Pour avoir un circuit entièrement fonctionnel, nous devons avoir à notre disposition ou commander les différents éléments de cette chaîne ; après étude nous pouvons remplacer la chaîne ci-dessus par la suivante :

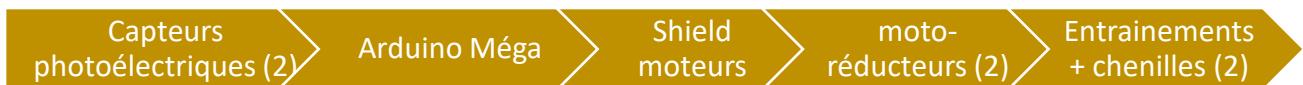


Il faut donc rajouter dans notre liste de commande, le relais qui sera commandé en 5V, la pompe à eau, qui sera alimenté par le circuit de puissance, et le tuyau d'eau qui doit avoir sa dimension adaptée à l'orifice de sortie de la pompe.

- Étude des choix de la fonction « suiveur de ligne » (mode asservi) et le mode télécommandé pour le groupe 2 :**

Le mode asservi va dépendre de deux capteurs photoélectriques, permettant de capter la ligne noire de la zone blanche, et lorsque le char détecte cette ligne noire il doit activer les moteurs pour tourner dans la direction opposé à cette ligne.

On se base encore sur la chaîne fonctionnelle précédemment citée, on obtient :



Sera donc rajouté à notre liste : les deux capteurs « suiveur de ligne », le Shield moteur (qui permet l'alimentation et le contrôle des deux moteurs), les deux motoréducteurs, les chenilles et les mécanismes d'entraînement.

Pour le mode télécommandé la chaîne fonctionnelle sera la suivante :



Le joystick permettra le déplacement du char (avant, arrière, gauche, droite). La partie communication correspond à l'émission et à la réception des signaux provenant du joystick entre les deux Arduino (Arduino de la manette et Arduino du char) préférentiellement par leurs ports série (pins RX / TX).

- **Étude des choix de la fonction « détection d'obstacle » (DO en raccourci) pour le groupe 2 :**

La chaîne fonctionnelle de détection d'obstacle sur laquelle nous nous sommes basés pour effectuer la liste et la commande des pièces liées à cette fonction est la suivante :



À partir de cette frise, on remarque qu'il nous faut aussi le capteurs à ultrason ainsi que le haut-parleur. En effet pour détecter l'obstacle il nous faut un capteur prévu à cet effet, l'information sera traitée par l'Arduino Méga qui activera le MOSFET du circuit, et donc le haut-parleur. Les LED de signalisation d'obstacle seront alimentées directement sur une sortie de l'Arduino. Certains composants ou consommables tels que les MOSFET, les LED, le haut-parleur et les résistances, sont récupérables à l'IUT, donc il n'est pas nécessaire de les commander.

- **Étude des choix de la fonction « châssis, chenilles, et entraînement du char » pour le groupe 2 :**

Cette partie est une étape de grande envergure nécessitant du temps pour le dimensionnement, la conception, l'impression et le montage de ses différents éléments mécaniques et permettant le bon fonctionnement de l'entraînement du char. Notre temps étant très limité, nous nous sommes donc orientés sur l'achat d'un kit comprenant tous ces éléments, un « package tout-en-un », nous permettant ainsi de nous focaliser en priorité sur les autres fonctions du char.

- **Étude des choix de la fonction « carrosserie du char » pour le groupe 2 :**

Nous avons choisi de concevoir (sur Catia) et de réaliser (impression 3D) la carrosserie, car il n'existe pas de modèle de carrosserie en vente adapté au châssis de notre char.

2. Le cahier des charges

L'élaboration de notre cahier des charges s'appuie notamment sur les prérequis et les besoins exigés par le maître d'ouvrage, ainsi que sur les périmètres auxquels nous allons devoir faire face.

- **Les prestations demandées :**

- Le client demande : un système pluridisciplinaire (mécanique, électronique, informatique) et ludique.

L'objectif principal de ce projet est l'étude, la conception, la réalisation et la finition du char bi-mode à jet d'eau dans un délai de 21 semaines par la maitrise d'ouvrage (MOE : MAMILONNE Didier et VINCENT Isaac) et devra être complété au moins à 90%.

Les différentes composantes de cet objectif se répartissent en fonctions des critères suivants :

F	Faisabilité	Oui	L'objectif peut être atteint dans le temps imparti au projet.
O	Organisé dans le temps	Oui	Dans le cas d'une bonne gestion/planification et que les fonds et/ou matériels nécessaires sont disponibles dans les temps
R	Réaliste	Oui	Le projet reste faisable à 90% - 100% (écart de 10% rajouté dû aux conséquences du COVID pour la réalisation du projet)
M	Mesurable	Oui	Quantité matérielle et budgétaire mesurable (budget de 200 € environ maximum pour chaque groupe)
E	Explicite	Oui	La finalité et la MOE sont ciblées

Nous souhaitons créer ce char dans les délais, en utilisant des composants fiables, solides, pour une meilleure durabilité de l'ensemble, en favorisant un bon rapport qualité-prix. Les prestations demandées pour réaliser ce projet sont des activités de natures industrielles, car nous entreprenons des activités de production ou de transformation grâce à l'utilisation d'outils, de techniques et de composants industriels.

- Les périmètres du projet :**

- L'objectif de la réalisation du projet pour le maître d'œuvre reste l'obtention de la licence professionnelle, ainsi que l'acquisition de compétences techniques, de gestion de projet et professionnelles.

- Les limites du projet se résument par son fonctionnement bi-mode (changeable par la télécommande):

- **Mode radiocommandé :** Char commandé et dirigé par la manette radiocommandée, capable d'envoyer des jets d'eau par son canon (espace du réservoir limité).

- **Mode Asservi :** Char suiveur de ligne.

Dans les deux cas, il s'arrête en cas d'obstacle détecté en face de lui, et l'indique à l'utilisateur grâce à un signal sonore et lumineux.

Le tableau suivant énumère les différentes parties concernées par le déroulement du projet :

Maitrise d'ouvrage - demandeur	Andres ARCIENIEGAS – IUT Cergy
Clients finaux – utilisateurs finaux	- L'IUT (étudiants et/ou professeurs) - Dans le cas d'une mise en vente du produit par la suite : Enfants et adultes (6 ans et +)
Maitrise d'œuvre	MAMILONNE Didier et VINCENT Isaac
Service internes	Licence mécatronique - IUT de Cergy
Prestataire(s) externe(s) - sous-traitant(s)	Non envisagé

Listing des composants

À partir de l'étude du projet c'est à dire des différents composants nécessaires à la réalisation projet, et de l'élaboration du cahier des charges, on peut dorénavant établir la liste des matériels à commander.

- Le matériel à commander :

Manette / Char	Matériel à commander (Choix initial)	Matériel pouvant être commandé, et ayant été commandé et reçu (choix final)	Fonction/ utilité	Image
M	Module joystick		Diriger le char et activer le jet d'eau	
C	Shield moteur L293D	Shield moteur Adafruit V2 (le fournisseur n'avait plus de Shield moteur L293D en stock)	Alimenter et gérer le sens de rotation et la vitesse des moteurs	
M & C	Module Bluetooth HC-05 (x2)		Reçoit les informations envoyées par le joystick puis le transmet au char	
M & C	Supports de piles (2x8 et 1x6)		Mise en série des piles ou accumulateurs	
C	Module suiveur de ligne TCRT5000 (x2) (Plage de fonctionnement: 1mm à 25mm)		Détection de la couleur noire/blanche, (mode suiveur de ligne)	
C	Kit tank (Moteurs + mécanismes d'entraînement + chenilles) 300mm x 230mm x 124mm (full aluminium) 1,2Kg (moto-réducteurs : 160 tr/min)		Multiple	

- Le matériel disponible et fourni par l'IUT**

Manette / Char	Matériel de l'IUT	Fonction/ utilité	Image
C	Arduino Méga	Unité de traitement du char	
M	Arduino Nano	Unité de traitement de la manette	
M & C	Résistances, condensateurs, transistors, LED, commutateurs, conducteurs, étain, régulateurs de tension LM7808 ou LM7805, diodes, accumulateurs AA LR06...	Multiple	
C	Capteur à ultrason HC-SR-04	Permet la détection d'un obstacle (calcul d'une distance)	
C	Haut-parleur 8 Ω	Signal la présence d'un obstacle	
C	Pompe à eau submersible + tuyau	Permet la fonction « jet d'eau »	

L'analyse et l'étude du projet étant finalisés, nous pouvons passer à l'étape tant attendue de la conception/réalisation.

V. CONCEPTION ET RÉALISATION DU CHAR BI-MODE À JET D'EAU

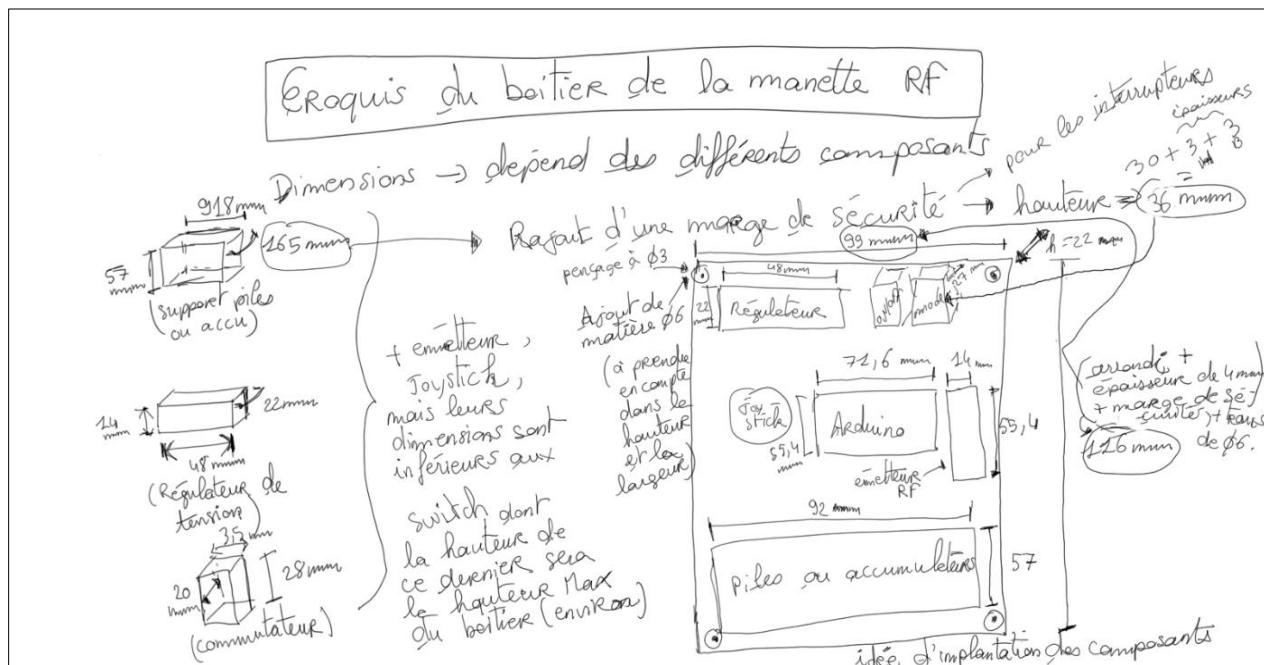
Avant de pouvoir réaliser notre char bi-mode, il nous faut avant tout concevoir toute la partie mécanique comprenant la carrosserie, le boîtier de la manette et des languettes de fixation (même si le boîtier de la manette n'a pas été imprimé en 3D, je parlerai quand même de sa conception vu qu'elle a été faite), mais aussi la partie électrique du char, de la manette et des PCB. Par la suite, la conception et la réalisation des programmes Arduino seront réalisées, pour pouvoir ainsi enchainer sur la réalisation du montage électrique et mécanique de l'ensemble. Aussi, le projet pourra être clos par des tests et essais de fonctionnement.

1. Conceptions mécaniques de la carrosserie du char, du boîtier de la manette, du bac de rétention d'eau et des languettes de fixation

Nous avons décidé de réaliser au moyen d'un outil CAO (CatiaV5 de Dassault System), la conception (pour une finalité d'impression 3D) d'un boîtier pour la manette lui permettant d'être plus résistant au choc et pouvant aussi maintenir de manière stable les différents circuits électroniques. C'est aussi le cas pour la carrosserie du char qui permettra d'avoir en plus une bonne ergonomie visuelle.

- Le boîtier de la manette**

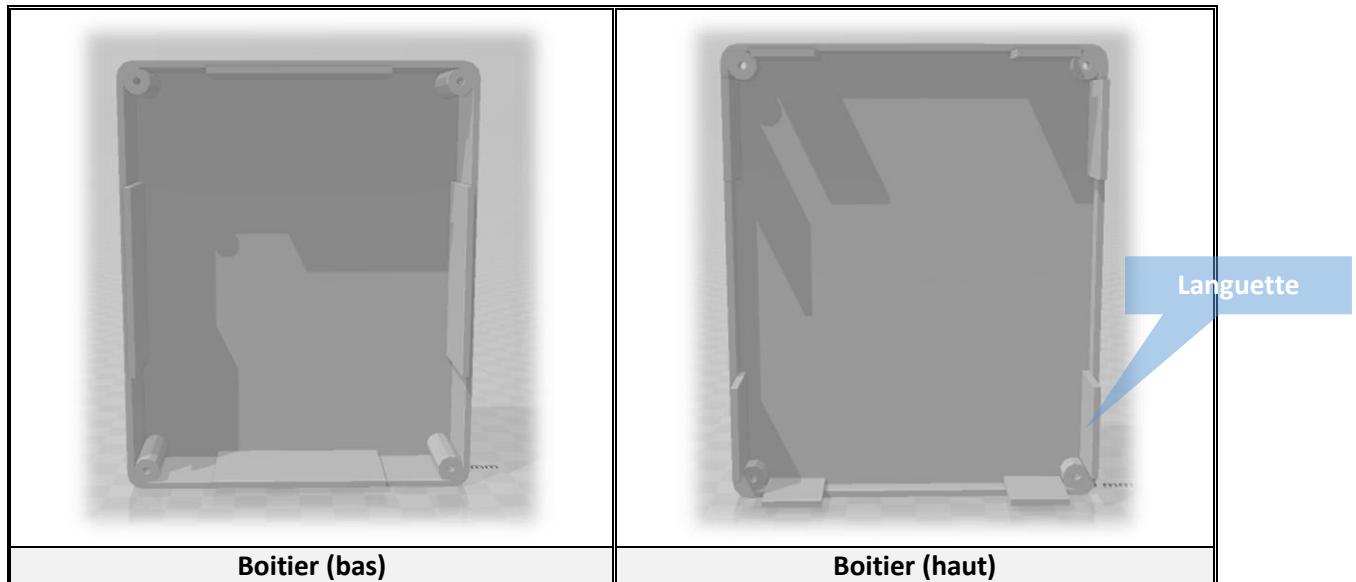
Avant de réaliser la conception de la manette, nous nous sommes penchés sur la réalisation d'un croquis nous permettant de placer les différents éléments qui y seront intégrés avec leurs dimensions (hauteur, longueur, largeur), pour nous permettre d'avoir un aperçu global du placement de ces éléments ainsi que du dimensionnement que nous devons apporter au boîtier.



En se basant sur ce croquis nous avons calculer des dimensions probables pour le boîtier, auxquels nous avons rajouté une marge en cas d'imprévu, marge qui pourra aussi faciliter l'espacement et donc le refroidissement des différents circuits.

Après quelques heures de conception sur Catia, nous avons été assez satisfaits du résultat. Malgré une bonne résistance mécanique permise par une épaisseur de deux millimètres, des contours et du maintien en position permise par les quatre vis de fixation de Ø3, nous avons voulu améliorer le

produit en rajoutant une mise en position autorisée par différentes languettes dans l'intérieur du boîtier, en tenant compte bien sûr de la surface requise par les différents composants. L'écart de ces languettes multiplie ainsi par deux fois la résistance mécanique du boîtier.



VOIR DESSINS D'ENSEMBLE ET DE DEFINITION EN [ANNEXE 7](#)

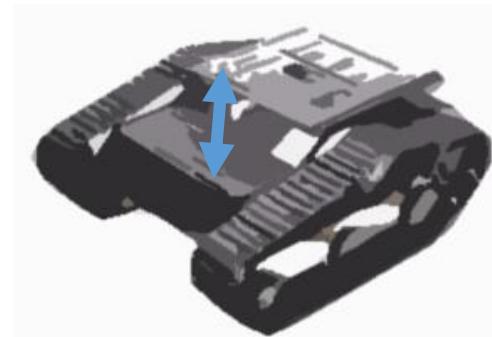
Le boîtier étant terminée nous avons suivi la même méthodologie pour réaliser la carrosserie du char. Du fait de la disponibilité de boîtier à l'IUT, notre boîtier n'a pas été imprimer en 3D et il ne sera donc pas utilisé, mais nous gardons son fichier sous le coude pour de futurs projets.

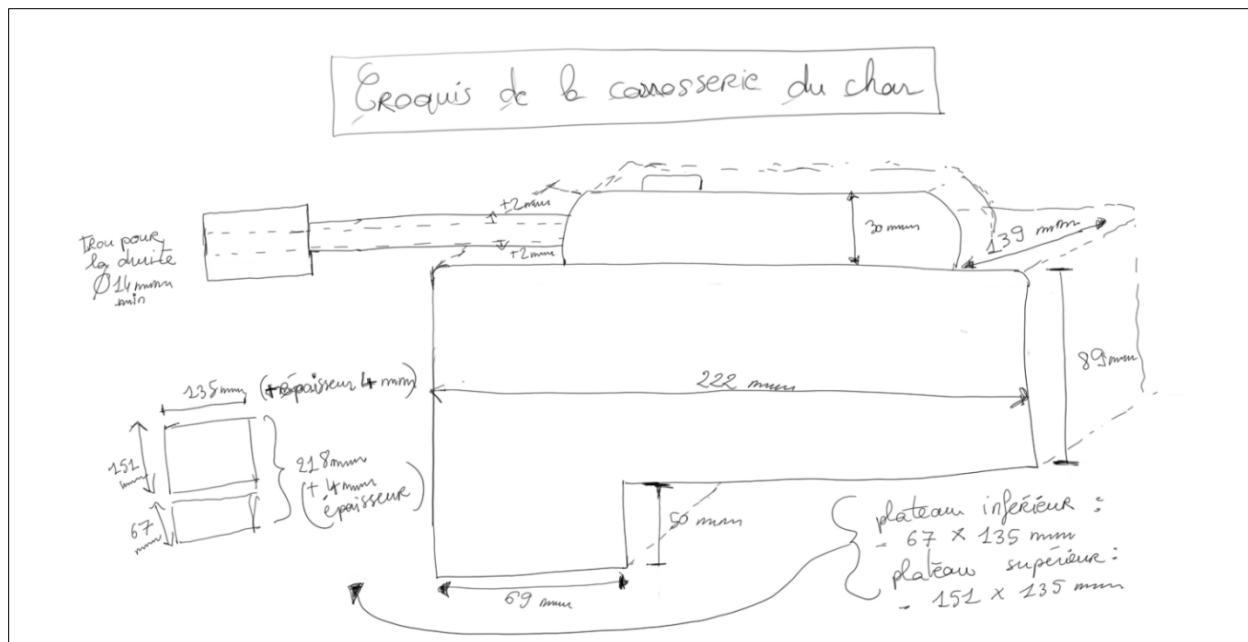
- La carrosserie du char**

Comme développé ci-dessus, nous entamons la réalisation de notre carrosserie par un croquis. Pour nous aider dans cette tâche, nous avons récupéré des informations sur le futur châssis du char, tels que les dimensions de la plaque supérieure et de la plaque inférieure, pour faciliter son dimensionnement. La plaque supérieure nous permettra de coller ou fixer le bac de rétention d'eau que je développerai par la suite. Après réalisation des différents calculs nous obtenons une dimension probable auquel nous rajoutons une marge de sécurité. Sachant la carrosserie malléable du fait de sa constitution en plastique, nous nous aiderons de languettes de fixation pour faciliter son positionnement sur le châssis.

Le plateau inférieur du châssis étant abaissé par rapport au plateau supérieur, nous avons rajouté une longueur supplémentaire de 50 millimètres (n'ayant pas trouvé de renseignements sur la distance de l'abaissement), que nous pourrons facilement réduire en cas de mauvaises surprises (enlèvement de matière avec la scie à métaux ou la Dremel).

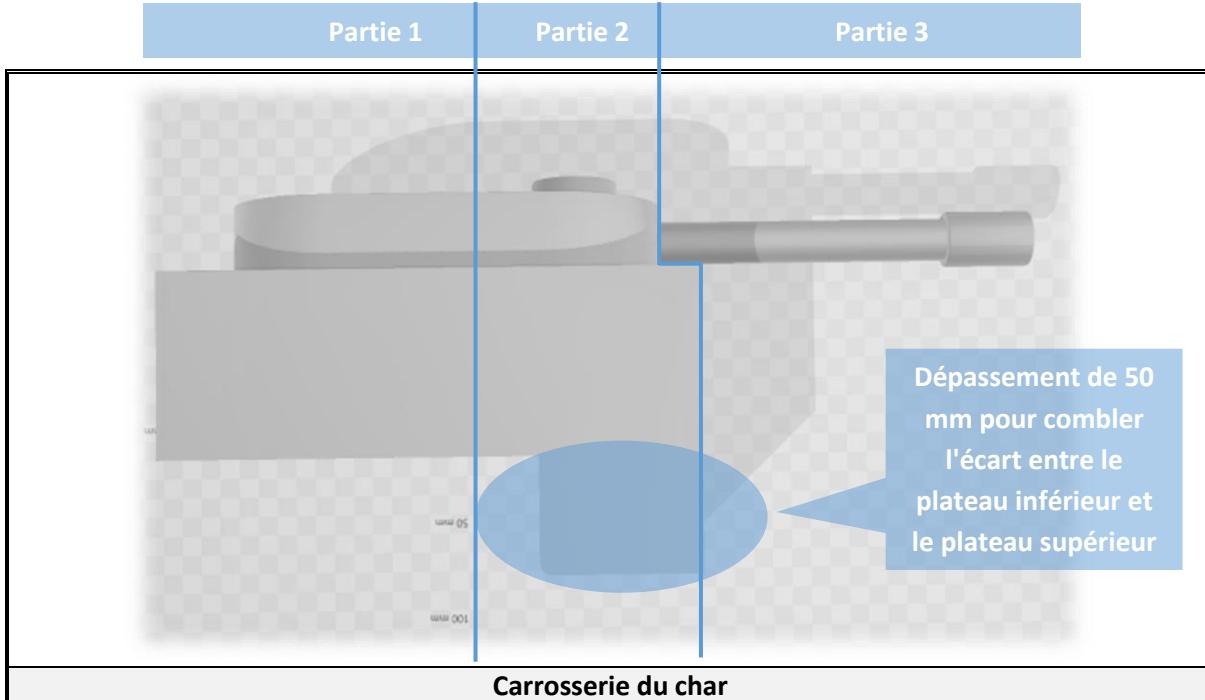
Nous avons aussi pris en compte le diamètre du creux du canon pour permettre le passage du tuyau d'eau pour le jet d'eau.





Après la réalisation en CAO, nous obtenons donc notre carrosserie au format numérique. Nous avons pu l'imprimer et nous l'avons récupéré ; nous pourrons l'utiliser lors du montage complet du char.

À noter que l'imprimante 3D à l'IUT ayant une surface ou zone d'impression limitée, il a fallu sectionnez notre char en 3 parties, avant de pouvoir passer à son impression (on utilisera les languettes de fixation pour les fixer).



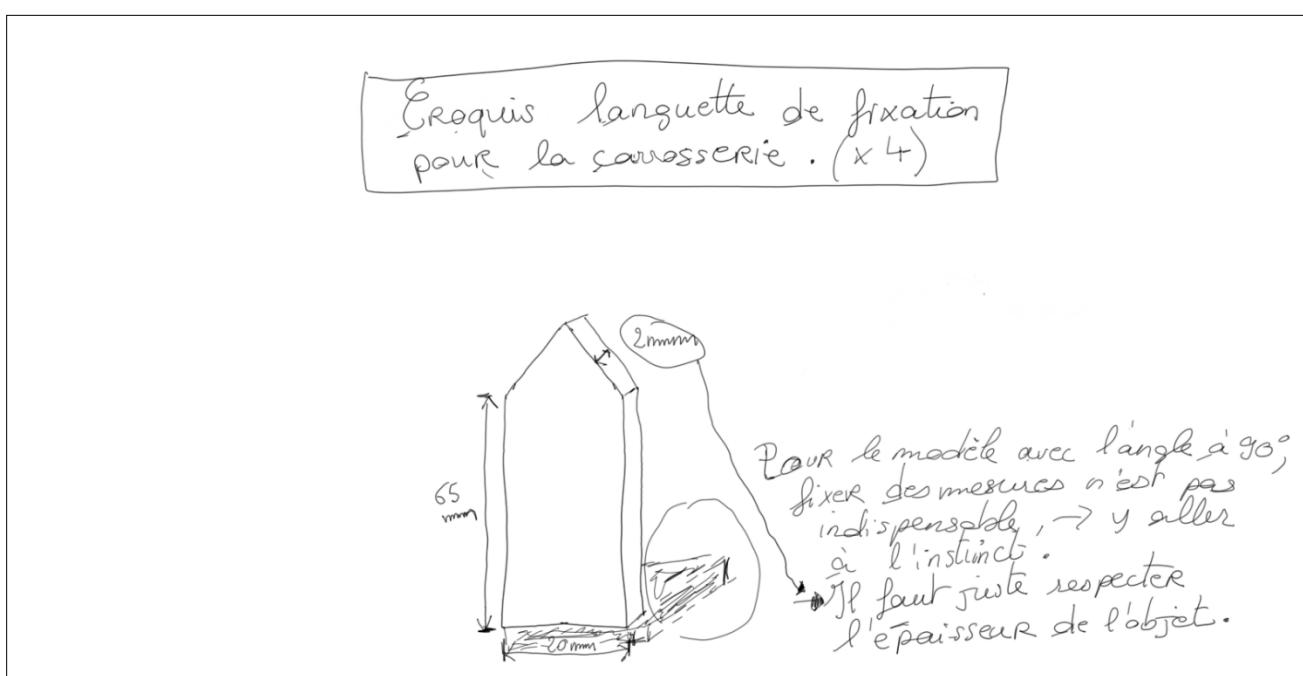
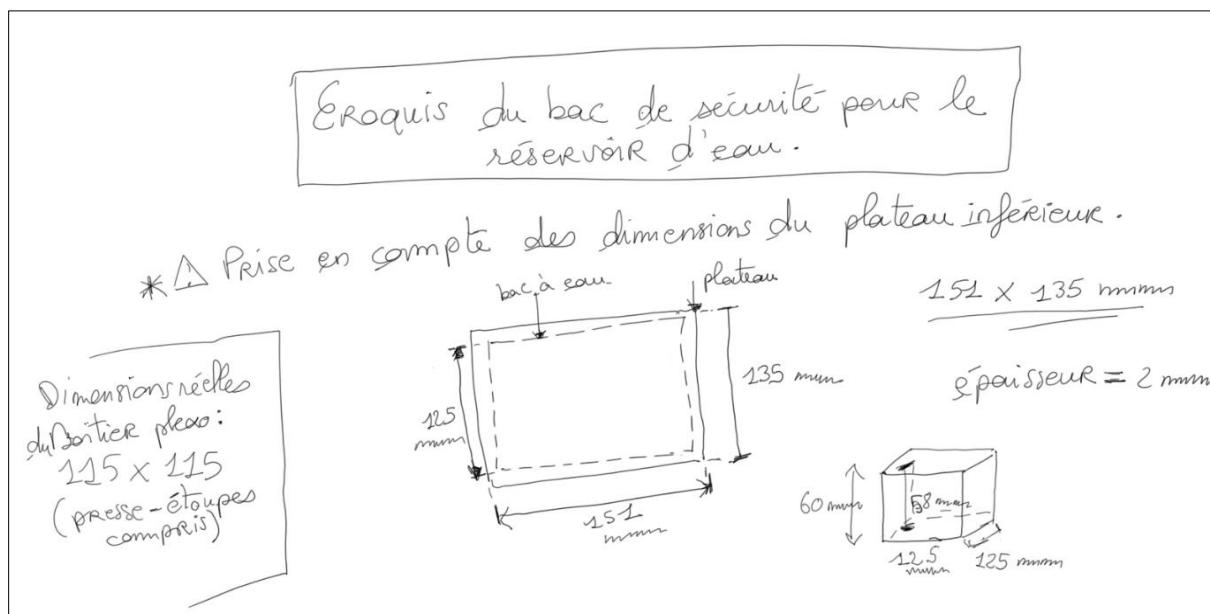
VOIR DESSIN DE DEFINITION EN [ANNEXE 8](#)

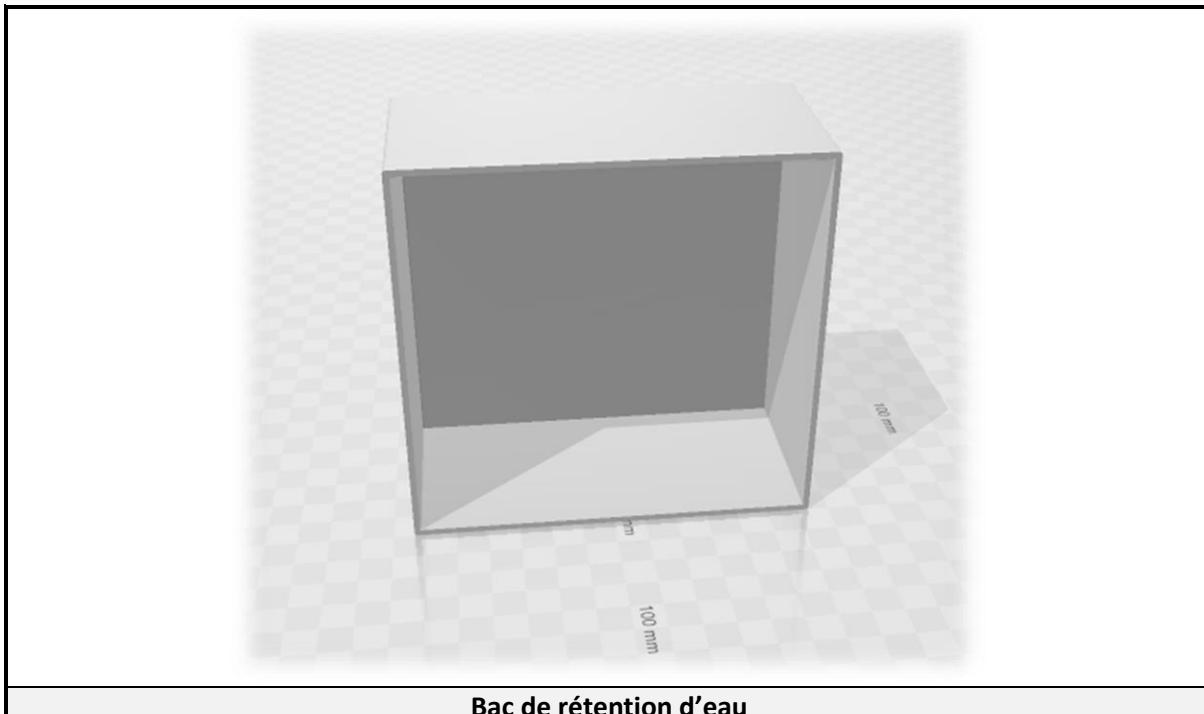
- Du matériel utile : le bac de rétention d'eau et les languettes de fixation**

Ces deux éléments peuvent paraître anodin mais sont d'une utilité cruciale pour la réalisation de notre char. Nous savons tous que l'eau et l'électricité ne font pas bon ménage, et c'est pour cela que nous avons décidé de réaliser un bac de rétention d'eau. En cas de fuite du réservoir on pourra réagir rapidement. On pourra facilement remarquer la présence d'eau dans le bac nous informant ainsi d'un défaut au niveau du réservoir d'eau.

Niveau conception on reste sur la même logique, à savoir la réalisation d'un croquis basé sur les dimensions du réservoir d'eau puis la réalisation sur notre logiciel de CAO.

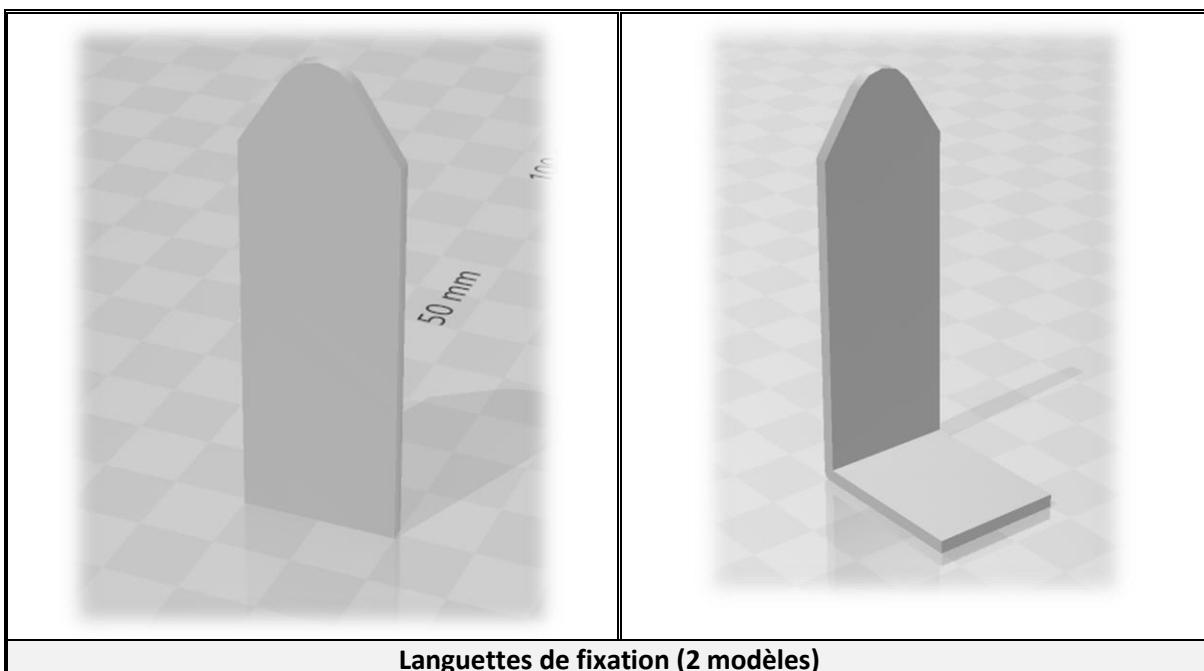
Même chose pour les languettes de fixation (un modèle droit et un autre modèle avec angle à 90°) qui nous permettra de fixer la carrosserie au châssis ainsi que les différentes parties de la carrosserie.





Bac de rétention d'eau

VOIR DESSIN DE DEFINITION EN [ANNEXE 9](#)



Languettes de fixation (2 modèles)

VOIR DESSIN DE DEFINITION EN [ANNEXE 10](#)

Après la conception mécanique nous pouvons passer à l'étape suivante, c'est à dire la conception électrique de la manette, des PCB et du char.

2. Conceptions électriques/électroniques du char, de la manette et des PCB

Ayant terminé la conception et la réalisation des différents éléments mécaniques, nous nous sommes orientés sur la partie électrique des deux différents circuits : la manette et le char. Une idée d'optimisation et de simplification du câblage nous est parvenue, à savoir la réalisation de PCB dont nous développerons les spécificités dans la troisième partie.

- **Schéma de câblage de la manette**

Ayant déjà pensé aux différents circuits nécessaires au bon fonctionnement de la manette, on peut dorénavant passer par la schématisation électrique (logiciel utilisé : Autodesk Eagle).

On doit quand même réfléchir et s'attarder sur des points importants, tel que le courant maximum pouvant fournir l'Arduino (en totalité 250 mA et par Pin 40 mA, source : www.forum.arduino.cc), les différents moyens de protection des éléments, quelle valeur de résistance pour limiter le courant dans un récepteur de type LED (courant maximum d'une LED : souvent 20 mA, source : www.forum.arduino.cc) par exemple, en passant par la diode de protection d'inversion de l'alimentation ...

Après s'être renseigné, on décide de commencer nos schémas électriques.

Dans le cadre de l'écoconstruction, nous avons pour objectif de diminuer au mieux le courant absorbé (diminuant ainsi la décharge des piles) par chaque récepteur à une valeur correct, nous essayerons d'utiliser pour cela des résistors à forte valeur pour les récepteurs qui consomme beaucoup (ex : LED...). Pour l'alimentation des accumulateurs seront utilisés à la place des piles jetables.

VOIR SCHEMA ELECTRIQUE DE LA MANETTE EN ANNEXE 11

- **Schéma de câblage du char**

En suivant les règles précédemment citées, on poursuit par la réalisation du schéma électrique du char. On fait aussi attention à ce que le courant absorbée par chaque moteur ne dépasse pas les 1,2A maximum que peuvent fournir notre Shield moteur Adafruit V2 (nos deux moteurs consomment 100 mA chacun à vide), puis on entame la conception de notre circuit électrique.

VOIR SCHEMA ELECTRIQUE DU CHAR EN [ANNEXE 12](#)

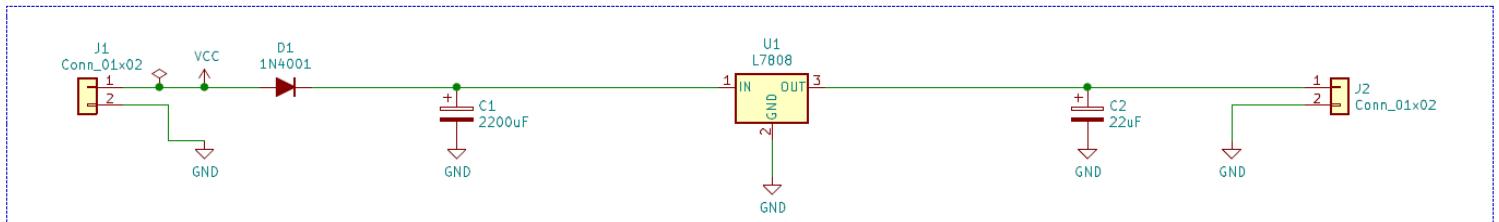
À la fin de cette conception nous nous sommes penchées sur une nouvelle problématique, comment allons-nous repiquer ou dispatcher l'alimentation 5V que peut fournir l'Arduino ? Devrions-nous faire plusieurs soudure sur un même point (efficace ou non) ? Devrions-nous utiliser des connecteurs du style « Wago » ? Comment réduire au maximum l'utilisation de conducteurs ? Ces questions ont été éclaircies par l'idée d'utilisation de circuits électroniques communément appelé PCB.

- Une idée d'optimisation : Les PCB**

L'utilisation de PCB reste la solution que nous avons trouvée pour répondre aux problématiques de câblage permettant d'améliorer efficacement l'optimisation du montage, du câblage et du temps.

Ayant remarqué que de nombreux éléments nécessitent l'alimentation 5 V provenant de l'Arduino, et que de nombreux « petits » composants peuvent être rassemblés sur une carte électronique, on a décidé de concevoir ou d'utiliser des PCB : nous allons donc concevoir par la suite, un PCB d'alimentation 5V (tension provenant de l'Arduino du char), un PCB d'alimentation 12V (commande) et 12V (puissance) pour le char, un PCB de régulation de tension (9V -> 7V ou 8V) pour la manette et un PCB de régulation de tension (12V -> 8V, 9V ou 10V) pour le char. On réalise ainsi 3 circuits en tout, le PCB de régulation de tension étant doublé (on utilisera le logiciel Kicad pour la conception des PCB).

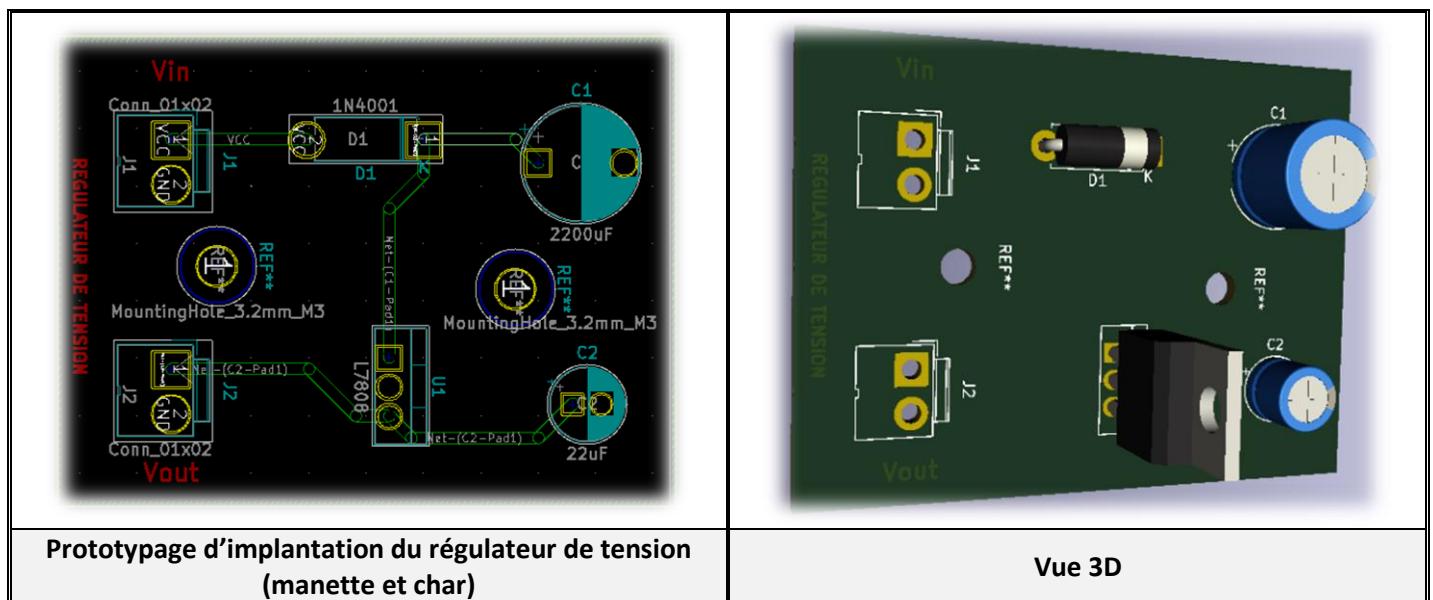
Commençons tout d'abord par le PCB de régulation de tension. Avant d'arriver à l'étape de conception propre du PCB, il faut tout d'abord créer le schéma électrique associé.



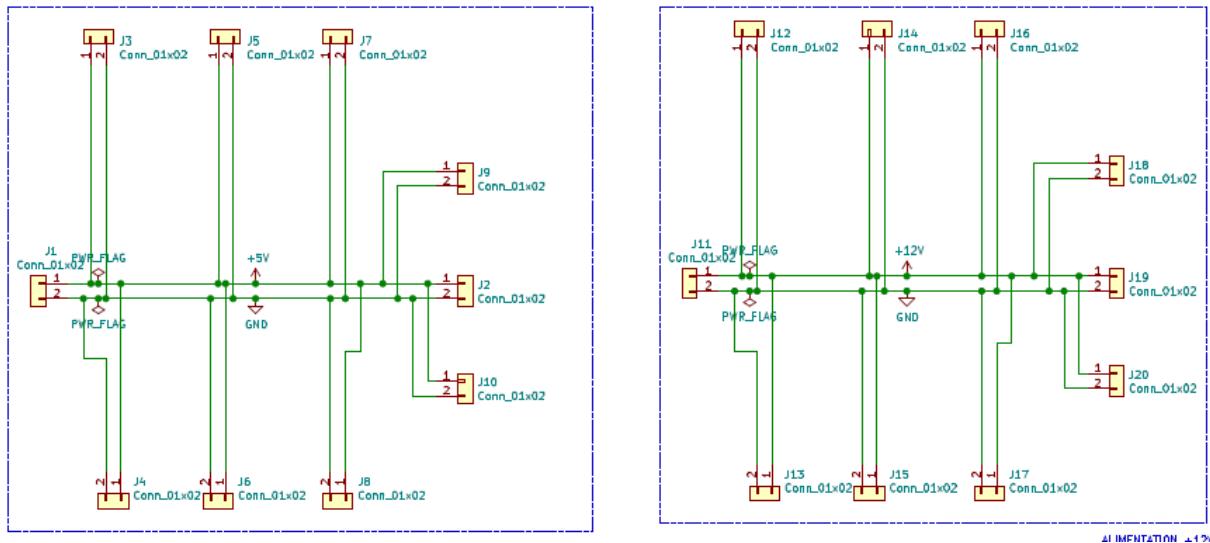
Régulateur de tension

Après avoir créé le schéma on peut passer en mode PCB pour commencer à réaliser les contours, les trous et les pistes de la carte. J'ai pris large en choisissant une largeur de piste de 1 mm, car elle permet la circulation d'un courant de plus de 2 A, selon l'outil de calcul et de dimensionnement de Kicad et permettra en retour une meilleure conductibilité du courant.

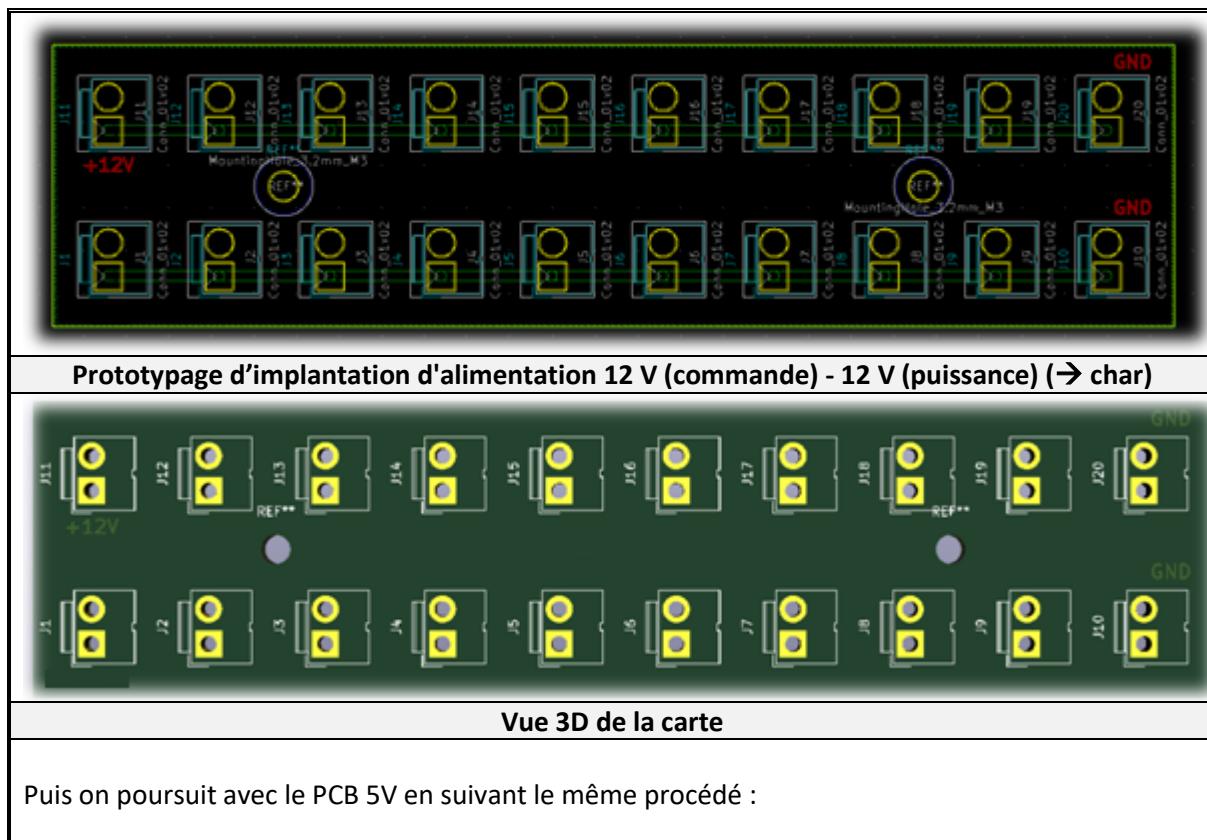
On retrouve sur la visualisation 3D de conception ci-dessous les différents éléments du schéma électrique, à savoir les connecteurs d'alimentation d'entrée et de sortie (qui permettent uniquement d'avoir l'empreinte et les trous pour réaliser la soudure), la diode, les deux condensateurs et le régulateur de tension.



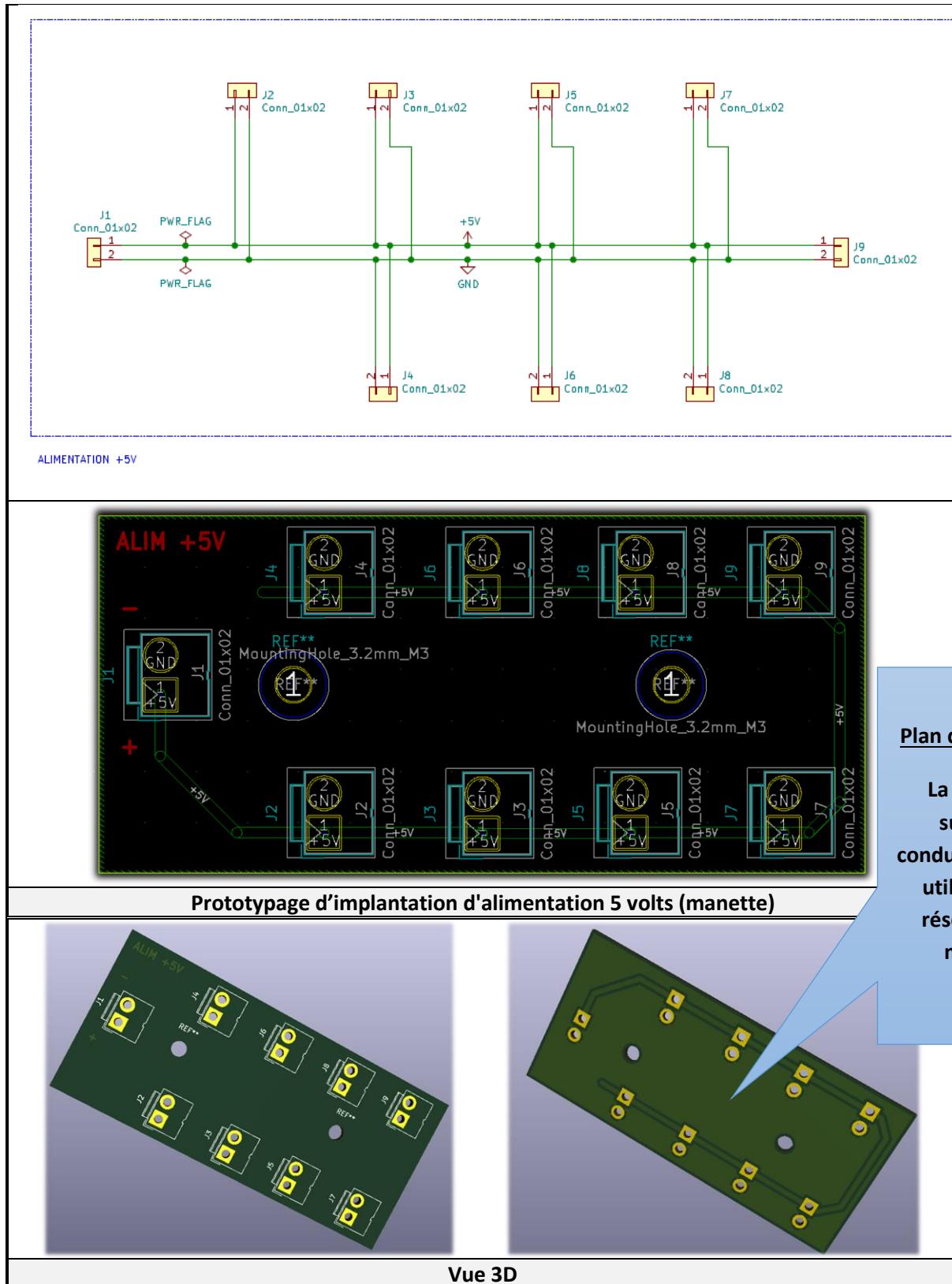
Puis on enchaîne avec la schématisation du PCB d'alimentation 12 volts (commande) - 12 volts (puissance).



On passe ensuite en mode de conception PCB et on obtient la visualisation du prototypage d'implantation ainsi que la vue 3D de la futur carte.



Puis on poursuit avec le PCB 5V en suivant le même procédé :



À noter que pour ces trois circuits j'ai utilisé un plan de masse, me permettant ainsi d'avoir une masse commune pour la carte d'alimentation 12V (commande) – 12V (puissance).

Après avoir terminé la conception le PCB, on peut passer la partie logicielle c'est à dire à la programmation en langage C des deux Arduino.

3. Conceptions et réalisations des programmes pour les Arduino de la manette et du char

Au niveau software, on doit réaliser deux programmes pour permettre le bon fonctionnement du système de manière télécommandée ou asservi.

Après préparation au brouillon, on édite (sur le logiciel CodeBlocks::Arduino) le code de l'Arduino pour la manette et pour le char. Voir les codes en [ANNEXE 13](#) et [ANNEXE 14](#).

- Programmation de l'Arduino de la manette**

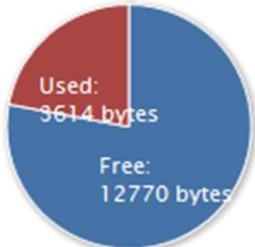
Avant de réaliser le programme on crée une représentation logique qui nous aide et facilite la compréhension du fonctionnement du système :

SI	
Il'entrée 7 est en position haute et que l'entrée 4 est en position basse	Il'entrée 4 est en position haute et que l'entrée 7 est en position basse
Alors on entre dans le mode télécommandé → Allumer la LED à la sortie 13 et éteindre la LED à la sortie 12	Alors on entre dans le mode asservi → Allumer la LED à la sortie 12 et éteindre la LED à la sortie 13
Mapper l'entrée analogique 0 de 1 à 100 (axe Y Joystick) Mapper l'entrée analogique 1 de 101 à 200 (axe X Joystick)	Envoyer la lettre « Y » sur la liaison série (nous informera que nous sommes bien en mode asservi)
SI	
L'entrée analogique 0 est > 55 et < 45	L'entrée analogique 1 est > 155 et < 145
- Envoyer la lettre « A », « B », ou « C » en fonction de la valeur d'entrée analogique pour avancer - Envoyer la lettre « R », « S », ou « T » en fonction de la valeur d'entrée analogique pour reculer	- Envoyer la lettre « D », « E », ou « F » en fonction de la valeur d'entrée analogique pour tourner à droite - Envoyer la lettre « G », « H », ou « I » en fonction de la valeur d'entrée analogique pour tourner à gauche
SINON SI	
L'entrée 2 est en position haute	
Envoyer la lettre « O » sur la liaison série (active le mode jet d'eau)	
SINON	
Envoyer la valeur « Z » sur la liaison série (valeur utile pour le calibrage du système)	
Récapitulatif des valeurs de la liaison série : ‘O’ → pour activer le jet d'eau ‘Y’ → mode asservi De 1 à 45 et de 55 à 100 → mode télécommandé (axe Y Joystick)	
De 101 à 145 et de 155 à 200 → mode télécommandé (axe X Joystick) ‘Z’ → position centrale neutre des joysticks, utile pour le calibrage de la manette.	

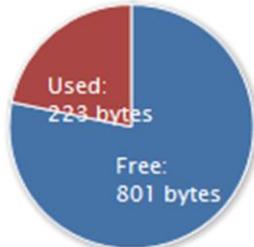
VOIR LE CODE EN C ARDUINO DE LA MANETTE EN [ANNEXE 13](#)

En compilant, Codeblocks::Arduino nous informe sur les mémoires utilisées par notre code en fonction des différentes mémoires disponibles pour le modèle de notre carte Arduino ; on sait donc si on doit optimiser notre code ou non.

FLASH Utilization



SRAM Utilization



EEPROM Utilization



- Programmation de l'Arduino du char**

On se base toujours sur le même principe cité précédemment :

Détection d'un obstacle			
Oui		Non	
SI		SI	
L'entrée analogique 1 et en position haute	L'entrée analogique 0 et en position haute	L'entrée analogique 1 et en position haute	L'entrée analogique 0 et en position haute
Active la fonction mode_telecommande_2() (ce mode télécommandé active uniquement la marche arrière)	En mode asservi, on désactive les 2 moteurs et on active la fonction Alarme(), l'obstacle doit être dégagé	Activer la fonction mode_telecommande1(), l'engin est contrôlé par le joystick (fonctionnement normal)	Engin en marche continue suivant une ligne

Il est nécessaire de créer différentes blocs de codes qui seront utilisées dans le programme ; en créant différentes fonctions on a une meilleure lisibilité, une flexibilité non négligeable et une meilleure vision du fonctionnement contrairement à un gros bloc de codes.

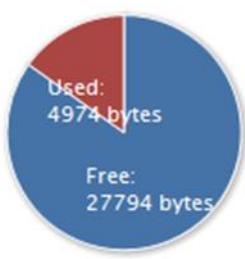
Fonctions	Description
Detection_obstacle()	Cette fonction utilise le capteur à ultrason pour calculer et connaître la distance de celui-ci face à un obstacle ; il renvoie une valeur 1 ou 0, s'il détecte respectivement un obstacle ou non.
Mode_asservi()	Ce mode terminer un contrôle du chat uniquement par le code. Dans ce mode, le contrôle du char par la télécommande est impossible.
Mode_telecommande1()	Cette fonction permet la marche avant et arrière du char par la télécommande.
GDO()	Active le jet d'eau par la lettre « O » est envoyé.

Moteur_avancer1()	Ces fonctions permettent l'activation des 2 moteurs dans le même sens (avancer le char) pour le mode télécommandé (trois niveaux de vitesse).
Moteur_reculer1()	Ces fonctions permettent l'activation des 2 moteurs dans le même sens (reculer le char) pour le mode télécommandé (trois niveaux de vitesse).
Moteur_gauche1()	Cette fonction autorise l'activation d'un moteur, l'autre moteur étant à l'arrêt (virage à gauche du char) pour le mode télécommandé (trois niveaux de vitesse).
Moteur_droite1()	Cette fonction autorise l'activation d'un moteur, l'autre moteur étant à l'arrêt (virage à droite du char) pour le mode télécommandé (trois niveaux de vitesse).
Desactivation_moteur()	Désactive les moteurs immédiatement.
Mode_telecommande2()	Cette fonction permet uniquement la marche arrière du char par la télécommande en cas de détection d'obstacle (mode télécommandé trois niveaux de vitesse).
Moteur_avancer2()	Cette fonction gère la marche avant du char en mode asservi (vitesse réduite).
Moteur_gauche2()	Cette fonction permet le virage à gauche du char en mode asservi (vitesse réduite).
Moteur_droite2()	Cette fonction permet le virage à droite du char en mode asservi (vitesse réduite).
Alarm()	Ce code active l'indicateur sonore et lumineux (un haut-parleur génère des tonalités et une LED clignote) en cas de détection d'un obstacle que ce soit en mode télécommandé ou en mode asservi.
Data_manage()	Récupère les données reçus sur la liaison série

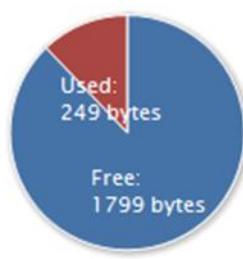
VOIR LE CODE EN C ARDUINO DU CHAR EN [ANNEXE 14](#)

Mémoires utilisées par notre code :

FLASH Utilization



SRAM Utilization



EEPROM Utilization



Mémoire EEPROM : mémoire non volatile ; conserve les données du système en absence d'alimentation (ex : configuration des différents composants d'Arduino)

Mémoire SRAM : mémoire volatile; perd toute ses données lorsqu'on coupe l'alimentation ; stocke des données qui sont en cours de traitement (ex : code en cours d'exécution, variables...)

Mémoire FLASH : mémoire non volatile ; elle ne perd pas ses données après avoir coupé l'alimentation ; support de stockage (ex : le code en C reste stocké sur la carte et peut être réutilisable si non changé)

Le tableau ci-dessous montre les différentes spécificités du code que nous avons utilisé :

Spécificité du code	
Variables (des pins)	On utilise des variables globales pour qu'elles soient utilisables et/ou modifiables par n'importe quelle fonction du code.
Fonction <code>Serial.println()</code>	Cette fonction permet de débuguer facilement notre code.
Fonction <code>map()</code>	La fonction <code>map()</code> permet de faire un produit en croix, ainsi il réalise une mise à l'échelle entre deux plages de valeurs.
Fonction <code>tone()</code>	Cette fonction produit un signal carré de la tension qu'il reçoit, avec une fréquence créant un son ou une tonalité par le haut-parleur.
Fonction <code>noTone()</code>	Désactive une fonction <code>tone()</code> précédemment utilisée.
#include <Adafruit_MotorShield.h> //----- Adafruit_MotorShield AFMS = Adafruit_MotorShield(); Adafruit_DCMotor *rightMotor = AFMS.getMotor(1); Adafruit_DCMotor *leftMotor = AFMS.getMotor(3); //----- rightMotor->setSpeed(85); rightMotor->run(FORWARD); leftMotor->run(RELEASE);	<p>1 - On inclut la bibliothèque du shield moteur Adafruit.</p> <p>-----</p> <p>2 - On indique à la bibliothèque sur quelle branche est raccordée le moteur de droite (1) et le moteur de gauche (2).</p> <p>-----</p> <p>3 - On fixe la vitesse du moteur de droite à 85/255, puis on l'active dans le sens d'avancement</p> <p>Le moteur de gauche reste désactivé</p>

Après avoir finalisé la conception on peut désormais passer à l'étape de la réalisation.

4. Réalisation du réservoir d'eau, du montage électrique, et mécanique de la manette et du char

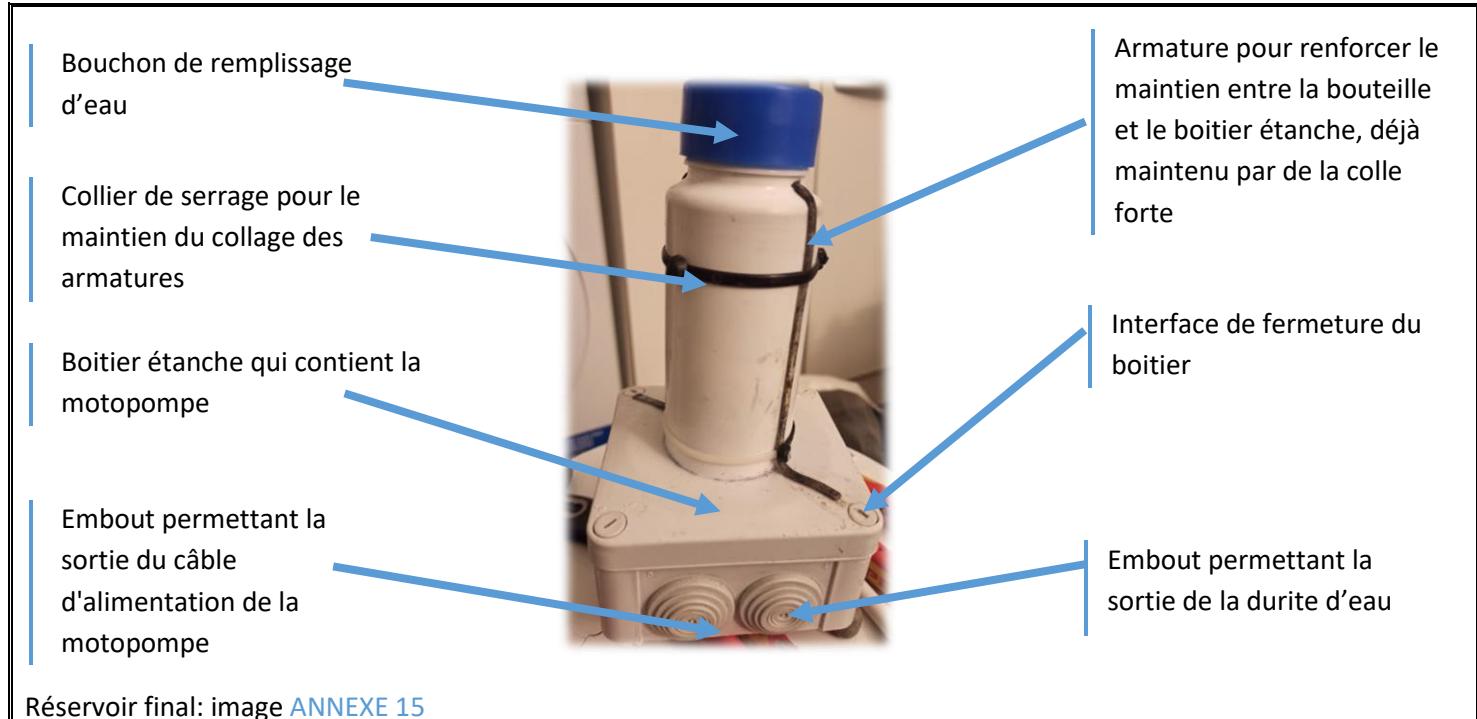
Dans cette partie nous allons développer les différentes étapes qui nous ont menés à la réalisation du char. Nous parlerons dans un premier temps de la réalisation du réservoir d'eau, puis dans un second temps nous développerons la partie montage électrique de la manette, puis du char.

1) Réalisation du réservoir d'eau

Voyant qu'aucun réservoir ne répondait à nos attentes, tant au niveau des dimensions qu'au niveau des possibilités, nous avons décidé de réaliser le nôtre avec du matériel facilement accessible.

<p>Nous avons tout d'abord récupéré une bouteille de 149 mm Ø que nous avons coupé, pour le fusionner avec un boîtier Plexo de dérivation étanche IP 65 de 7 embouts. Cette bouteille permettra de faciliter le remplissage du réservoir par le bouchon du haut.</p> <p>On a rajouté au niveau du bas de cette bouteille, un morceau de plastique que nous avons trouvé à plusieurs endroits pour qu'il serve de filtre « gros ».</p>	
	<p>Quant au boîtier Plexo, il contiendra la pompe. Au niveau des embouts, on pourra faire ressortir le fil d'alimentation de la pompe ainsi que la durite de refoulement d'eau. On rajoutera du mastic de bond au niveau des ouvertures ce qui permettra une bonne étanchéité de l'ensemble du système.</p> <p>En cas de fuite, le bac à eau sécuritaire va récupérer l'eau et dans le même temps sous signaler le problème, à condition de déposer la carrosserie.</p>

L'image ci-dessous représente le format final du réservoir d'eau :



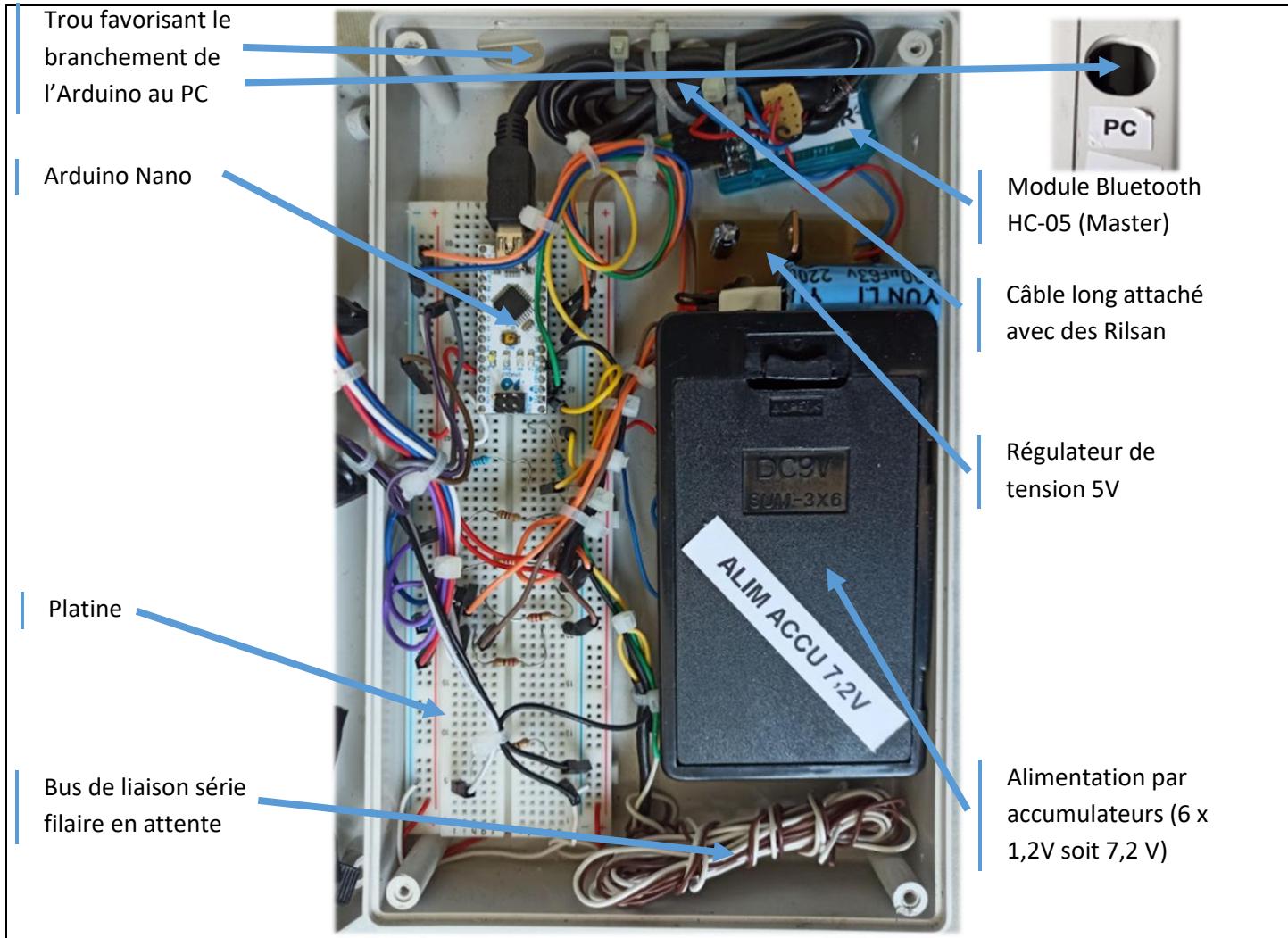
2) Réalisation du câblage électrique et du montage mécanique de la manette

Ne disposant pas tout de suite du kit contenant le châssis et les chenilles, nous nous sommes donc orientés sur la réalisation mécanique et électrique de la manette. Nous nous sommes aidés du schéma électrique de la manette pour arriver à se repérer parmi tous les fils et tous les composants à câbler.

A - Câblage électrique de la manette :

Nous avons utilisé des conducteurs à âme uni-brin pour faciliter la soudure, et des jumper pour relier les différents modules à la platine. Cette dernière nous a été très utile pour câbler rapidement les différents éléments à l'Arduino Nano. De plus, n'ayant pas reçu tout de suite le module Bluetooth permettant la communication entre les deux Arduino, nous avons laissé en attente un bus filaire composé de deux conducteurs (un blanc et un marron). Voir « Réalisation de la manette : image 1 ».

Pour limiter les risques de futurs défauts, nous avons ajouté du scotch isolant sur les parties conductrice du couvercle. Voir « Réalisation de la manette : image 2 ».



Réalisation de la manette : image 1

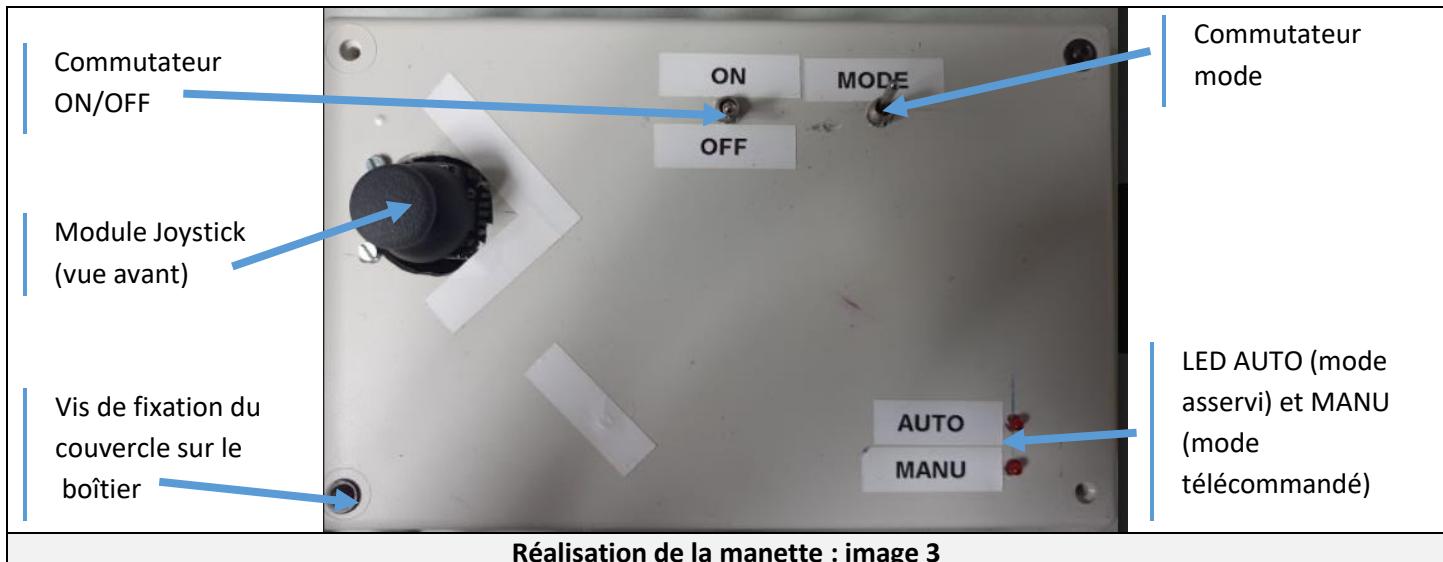


Réalisation de la manette : image 2

B - Montage mécanique de la manette

Pour la fixation des composants nous avons utilisé du ruban double face dans la majorité des cas. Certains autres composants, tel que les cartes électroniques (face bombée par l'étain), ou les boucles de fil trop long, ont eu plus de difficultés à rester collé avec le ruban nous avons du soigner les fixer en plus avec une vis ou soit l'attacher avec des Rilsan. Voir « Réalisation de la manette : image 2 ».

Le joystick a bien été fixé mais uniquement avec deux vis côté gauche ; les vices gênants pour la rotation maximale du joystick, nous avons décidé de rajouter un morceau de Rilsan noir que nous avons fixé de chaque côté avec des vis sur le couvercle du boîtier. Quant aux LED et aux commutateurs deux positions, ils sont maintenus à la colle forte sur le couvercle. Voir « Réalisation de la manette : image 3 ».



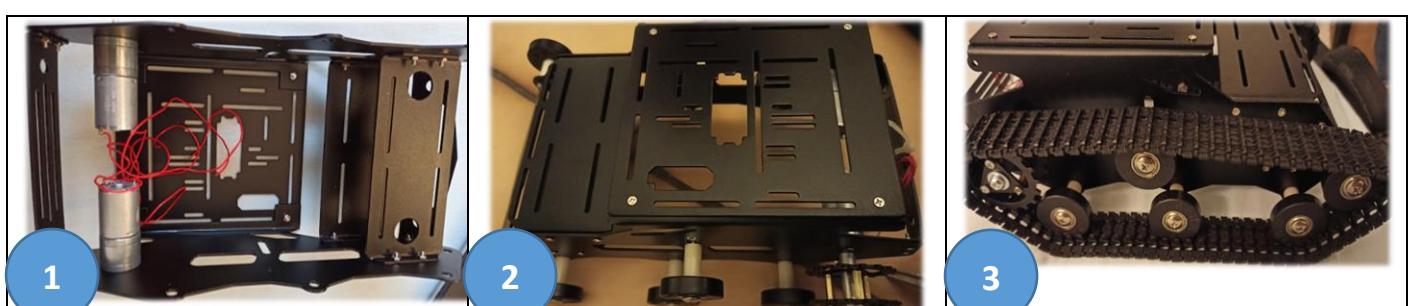
À la fin du montage électrique et mécanique de la manette, les deux modules Bluetooth ont été synchronisé pour permettre leur communication (procédures de configuration à l'[ANNEXE 16](#)), et le programme précédemment conçu de la manette a été téléchargé dans l'Arduino Nano.

3) Réalisation du câblage électrique et du montage mécanique du char

Ayant reçu le kit de montage du châssis du char, nous passons désormais au montage, à l'assemblage mécanique et au câblage électrique du char.

A - Montage du châssis du char

La première étape de ce parcours consiste à monter le char avec les différents éléments du kit :



1

2

3

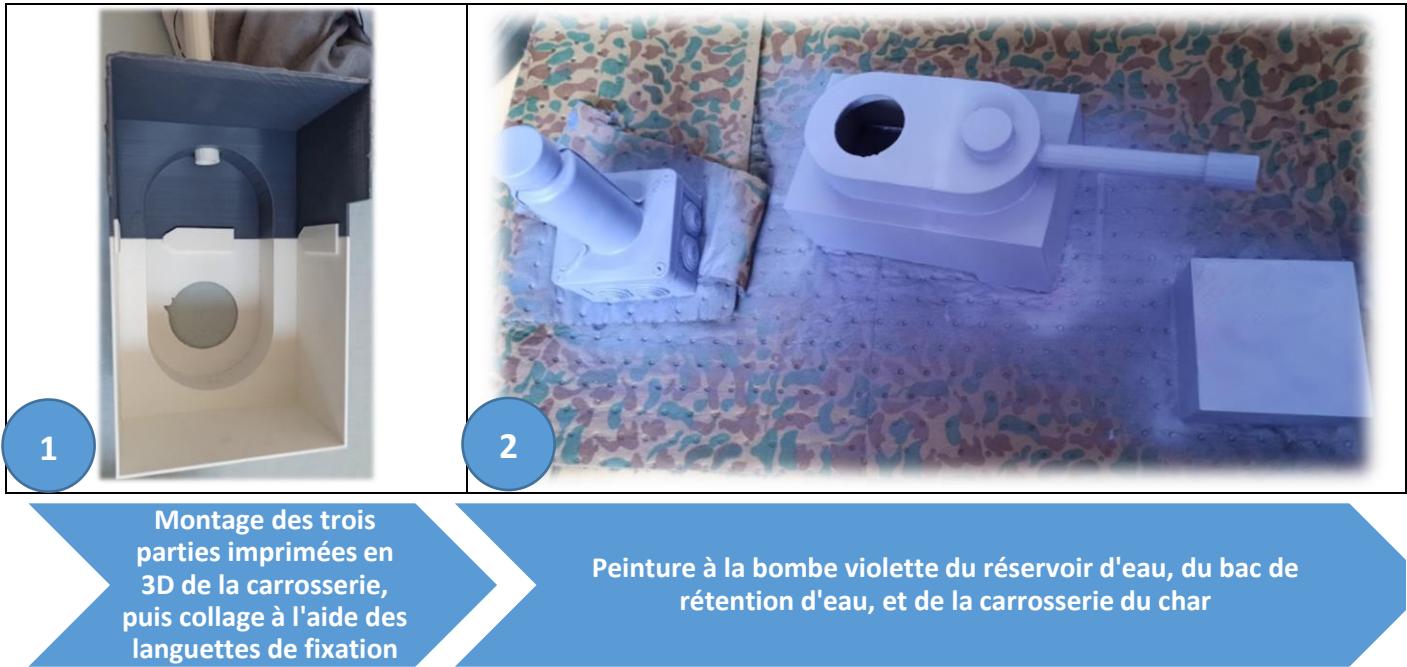
Montage des deux moteurs et des différentes plaques

Mise en place des roues et des deux pignons d'entraînement

Placement des chenilles

B - Assemblage de la carrosserie du char

Pour une meilleure apparence et ergonomie de l'engin, nous décidons ensuite de peindre les éléments imprimés en 3D (à savoir la carrosserie du char dont les trois parties sont préalablement fixées, le réservoir d'eau et le bac de rétention d'eau) en une couleur violette :



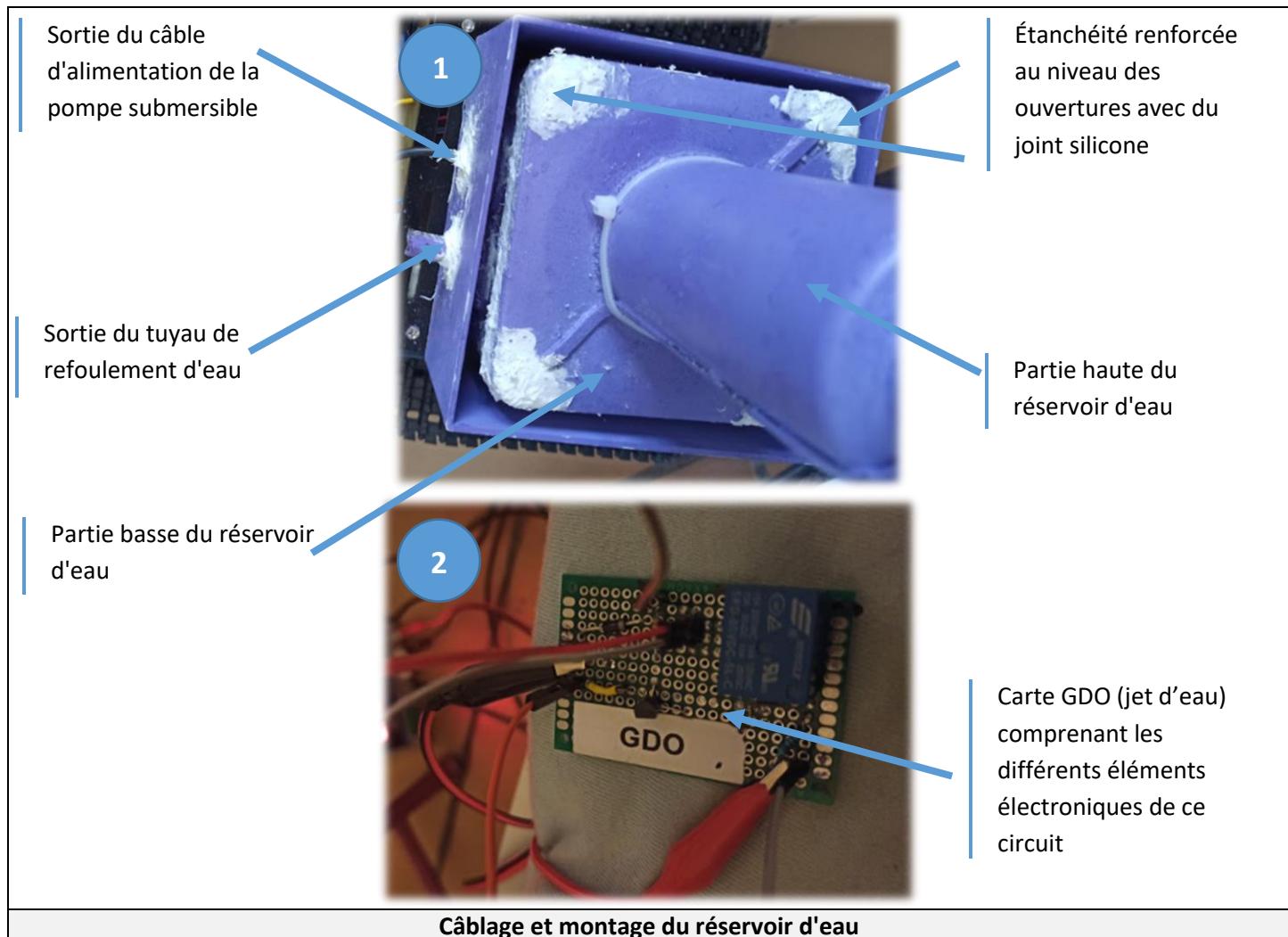
C - Câblage des différents circuits électriques du char

Nous passons à une partie importante de la réalisation, à savoir le passage de conducteurs pour alimenter les différents parties et circuit de notre char. Nous développerons cette partie en trois : nous parlerons dans un premier temps, du câblage et montage du réservoir d'eau, puis des travaux effectués sur la carrosserie, et ensuite nous terminerons par le montage mécanique et électrique les différents éléments sur le châssis.

Partie 1 : Câblage et montage du réservoir d'eau

Le montage et câblage du réservoir d'eau dans cette étape, consiste à insérer la pompe dans la partie basse du réservoir, tout en faisant ressortir son câble d'alimentation et son tuyau de refoulement d'eau, au niveau des embouts du boîtier Plexo comme précisé à la page 26.

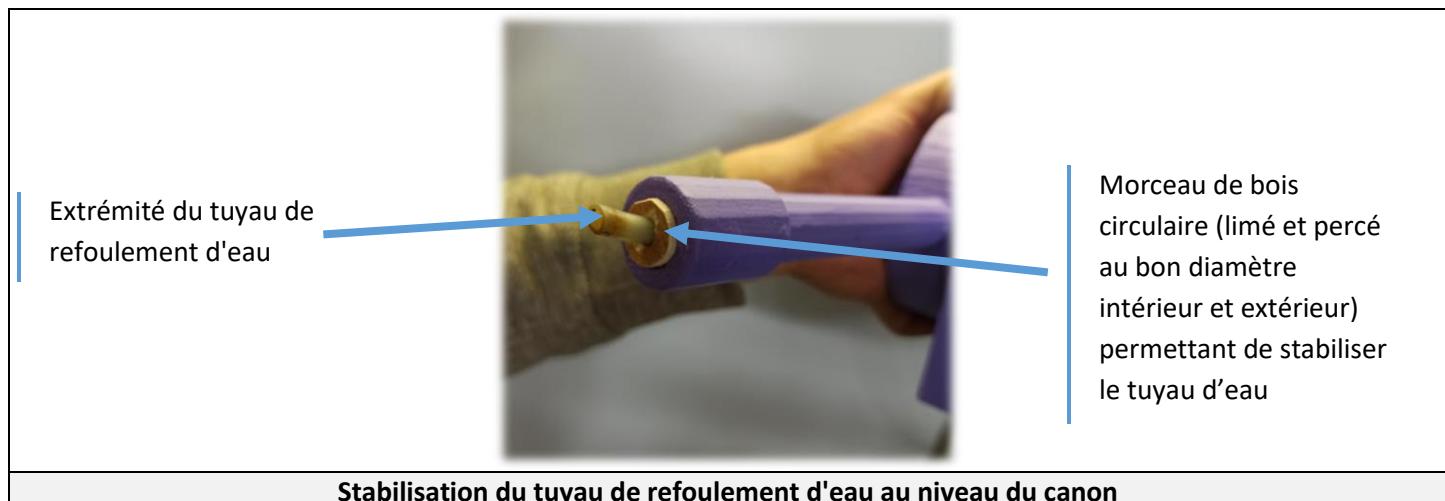
On en profite pour réaliser aussi son PCB (un PCB unique par circuit : cette information est développée dans la partie 3) s'appuyant sur le schéma électrique, et en utilisant les différents éléments requis pour son bon fonctionnement.



Partie 2 : Travaux effectués sur la carrosserie du char

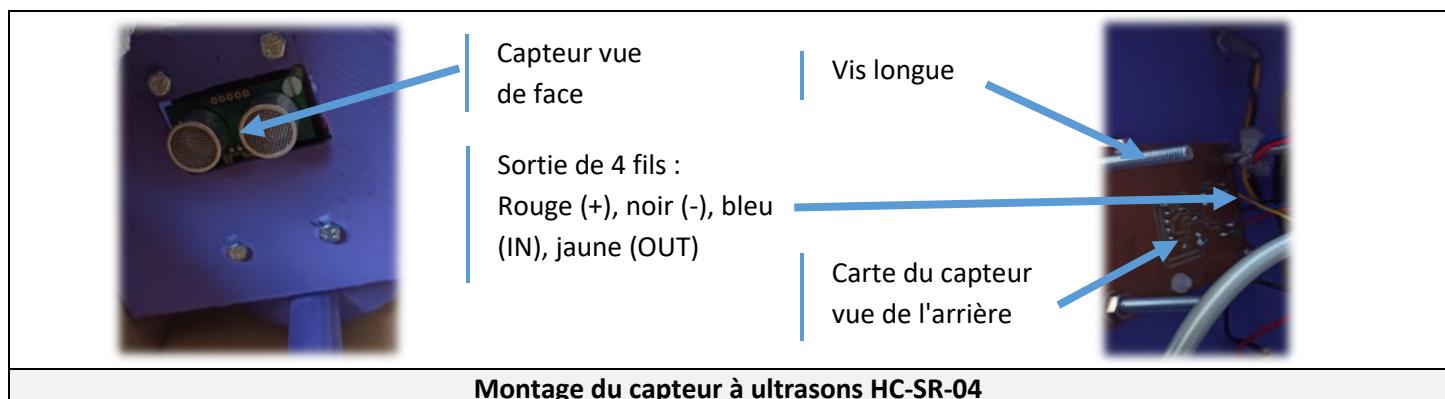
Au niveau de la carrosserie, nous avons tout d'abord cherché à réduire le diamètre intérieur du canon pour maintenir le tuyau de refoulement de la pompe qui était réduit à l'extrémité permettant ainsi une sortie du fluide à une plus grande vitesse (objectif : augmenter la longueur du jet d'eau), à cela s'ajoute aussi la perçage des trous pour fixer le module de détection d'obstacles (capteur à ultrason HC-SR-04), le positionnement du haut-parleur ainsi que les LED et les commutateurs.

Pour stabiliser le tuyau de refoulement d'eau au niveau du canon, nous avons rajouté un morceau de bois de forme circulaire et troué au diamètre idéal (le même que le bout du tuyau) et que nous avons ensuite inséré à l'extrémité du canon.

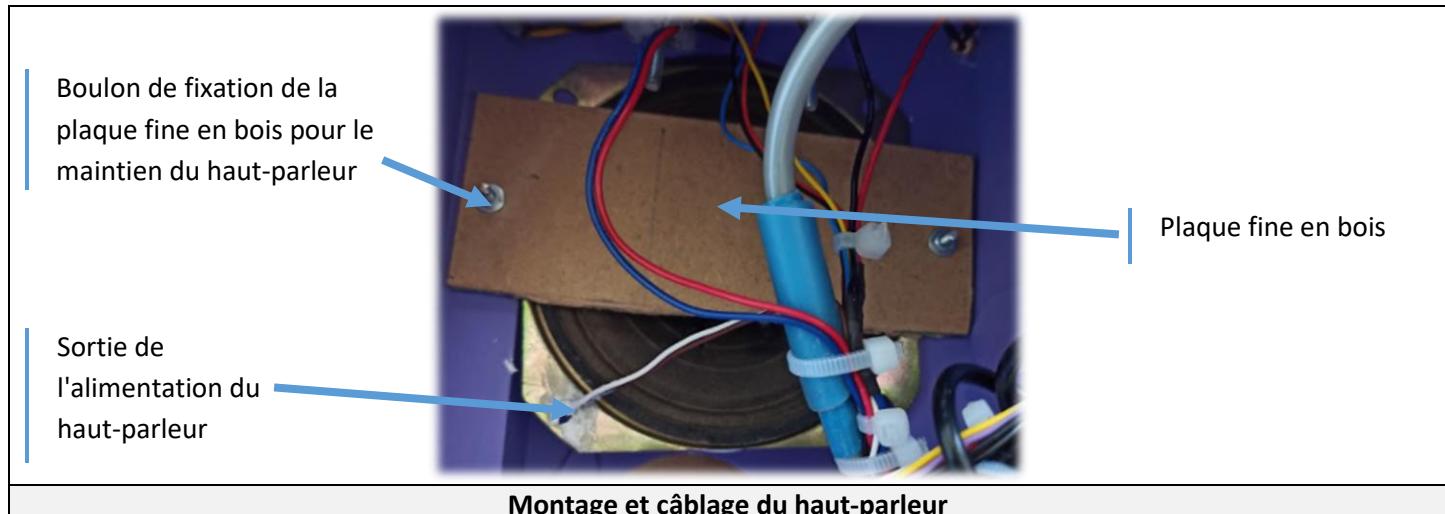


Puis, on enchaîne avec le montage du capteur à ultrasons permettant la fonction détection d'obstacle. Le module étant déjà fixé sur une carte électronique, il nous fallait percer au niveau des trous situés aux quatre côtés de cette carte. Nous avons en effet choisi de fixer cette carte sur la face avant de la carrosserie, car c'était la zone la plus adaptée. Le souci principal résidait dans le fait que la face avant de la carrosserie est assez haute par rapport au sol ; on a donc décidé de fixer cette carte avec des longues vis en rajoutant un écrou et un compte écrou, permettant d'orienter la carte légèrement vers l'avant en direction du bas (fixation du capteur : voir l'image ci-dessous).

Niveau câblage, on ressort les quatre fils en laissant assez de longueur pour faciliter le montage/démontage.

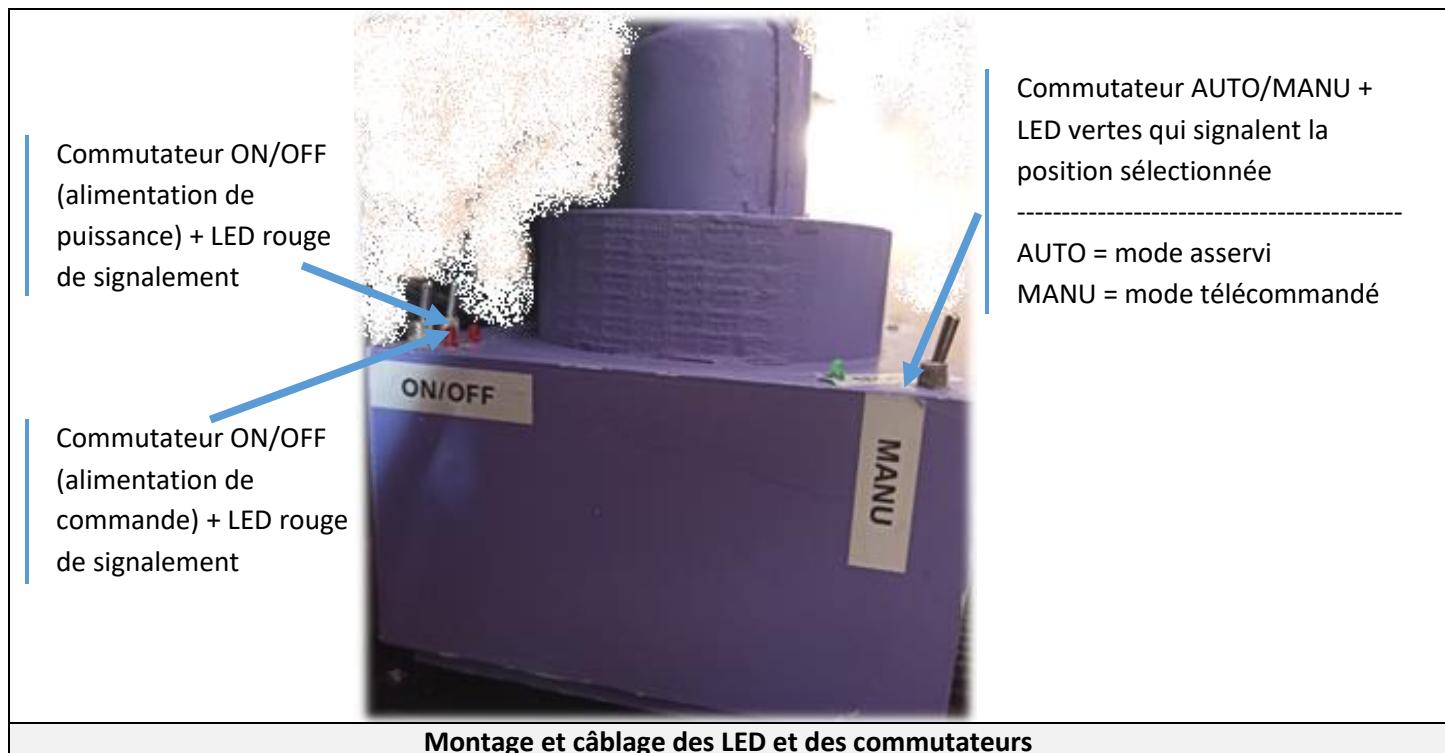


Le haut-parleur ayant une envergure assez imposante, nous avons choisi de mettre la partie aimantée dans le creux situé en haut de la carrosserie. Il est maintenu par une plaque de bois fine fixée avec deux boulons de chaque côté.



Par la suite, on enchaîne sur les fixations des LED et des commutateurs. Sur le char, il y a trois commutateurs et six LED en tout. Après les percements des trous de leur emplacement , ils ont été placés puis collés à la colle forte. Chaque LED est bien entendu protégée par une résistance pour limiter le courant y circulant. Deux circuits PCB, dont l'un étiqueté « LED » contiennent les différents éléments tel que les résistance de protection, l'alimentation... ce qui est très pratique pour se repérer lorsque l'on souhaite dépanner cette partie du circuit. L'image ci-dessous représente l'implantation de ces éléments sur la carrosserie.



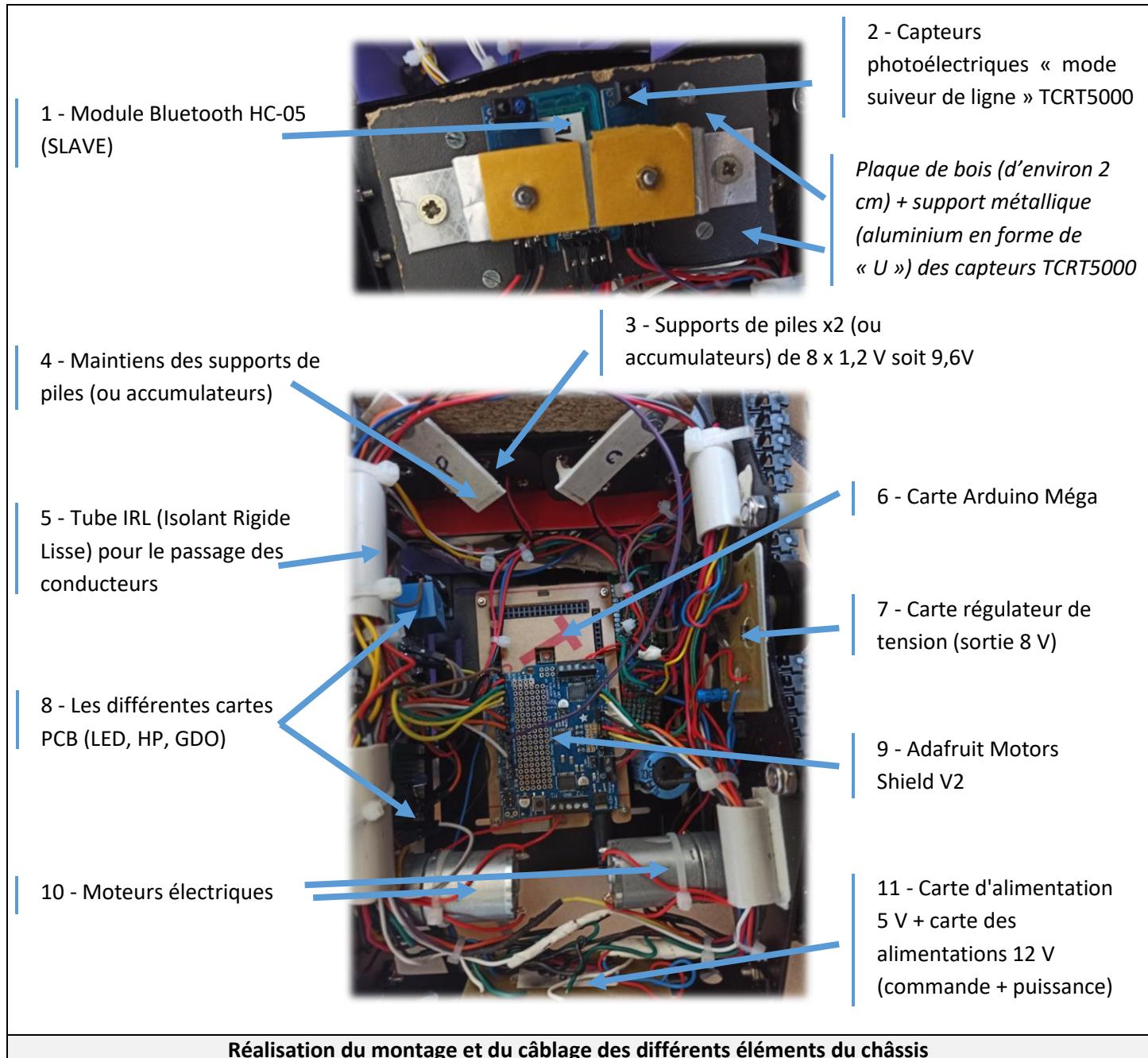


Partie 3 : montage mécanique et électrique les différents éléments du châssis

Les PCB ont été d'une aide très précieuse, facilitant ainsi le câblage les essais et le dépannage des différents circuits. Nous avons donc fait un bon choix de partir sur la réalisation de circuits séparés par PCB ; nous avons réalisé différentes cartes tel que la carte jet d'eau (raccourci : GDO), la carte haut-parleur (raccourci HP) , une carte LED...

Cette dernière partie de réalisation correspond au montage mécanique et électrique des différents composants qui ont été câblé et fixé sur le châssis, à savoir les deux moteurs (moto-réducteurs), la carte Arduino Méga, le Shield moteur Adafruit V2, les supports de batteries (ou d'accumulateurs) ... pour ne citer qu'eux.

L'image ci-dessous représente le montage et le câblage final des différents éléments fixés sur le châssis du char.



Réalisation du montage et du câblage des différents éléments du châssis

Nous allons énumérer dans le tableau ci-dessous les différents principaux composants qui y sont représentés, et expliquer succinctement leur utilité, leur montage et leur câblage (plus de détails sur le câblage des éléments dans le schéma électrique à l'[ANNEXE 12](#)) :

N° - élément	Utilité - fonction	Montage	Câblage																		
1 - Module Bluetooth HC-05 (SLAVE)	Récupère les données provenant du module maître et transmet ces données sur la liaison série de l'Arduino du char	Fixation à l'avant du char avec du scotch double face	<table border="1"> <tr> <td>STATE</td> <td>VCC -> +5V</td> </tr> <tr> <td>LEVEL: 3.3V</td> <td>GND -> 0V</td> </tr> <tr> <td>RXD</td> <td>RX -> TX</td> </tr> <tr> <td>TXD</td> <td>TX -> RX</td> </tr> <tr> <td>POWER: 3.6V-6V</td> <td></td> </tr> <tr> <td>GND</td> <td></td> </tr> <tr> <td>VCC</td> <td></td> </tr> <tr> <td>FC-114</td> <td></td> </tr> <tr> <td>EN</td> <td></td> </tr> </table>	STATE	VCC -> +5V	LEVEL: 3.3V	GND -> 0V	RXD	RX -> TX	TXD	TX -> RX	POWER: 3.6V-6V		GND		VCC		FC-114		EN	
STATE	VCC -> +5V																				
LEVEL: 3.3V	GND -> 0V																				
RXD	RX -> TX																				
TXD	TX -> RX																				
POWER: 3.6V-6V																					
GND																					
VCC																					
FC-114																					
EN																					

2 - Capteurs photoélectriques « mode suiveur de ligne » TCRT5000	La broche numérique (DO) aura une valeur de 0 ou 1 indiquant si un objet est ou non détecté	Les deux capteurs sont fixés sur le plaque métallique en forme de U lui-même fixé par des vis sur la plaque en bois	VCC -> +5V GND -> 0V DO -> A2 et A3 
3 - Supports de piles x2 (ou accumulateurs) de 8 x 1,2 V soit 9,6V	Permet la mise en série des piles (ou accumulateurs)	Les accumulateurs dans leur support sont maintenus par un élément en plastique visser sur la plaque de bois (P-> puissance , C-> commande)	 Fil noir raccordé à la masse Fil rouge vers le régulateur de tension
4 - Maintien des supports de piles (ou accumulateurs)	Maintient le support de piles (ou accumulateurs) pour éviter qu'il bouge ou qu'il sort du châssis	Les deux sont fixés par une vis sur la plaque de bois, ils font pression sur le support de piles (ou accumulateurs)	 Pas de câblage nécessaire pour cet élément
5 - Tube IRL (Isolant Rigide Lisse) pour le passage des conducteurs	Permet de réaliser un câblage propre, aide aussi à pouvoir retirer les supports de piles facilement vers l'extérieur sans être gêné par les conducteurs	Fixation réalisée par colliers de serrage, et/ou par scotch double face	 Pas de câblage nécessaire pour cet élément
6 - Carte Arduino Méga	Unité de traitement « et cerveau » du char	La carte est maintenue dans un cache de protection ; ce cache est fixé sur le châssis avec du scotch double face	 Il fournit l'alimentation 5 V, 3,3 V et est alimenté par le régulateur de tension 8 V
7 - Carte régulateur de tension (sortie 8 V)	Régule la tension provenant des piles ou accumulateurs pour alimenter l'Arduino sous 8V	Le régulateur de tension est maintenu par du scotch double face sur le châssis (sur le côté)	 Il reçoit environ 10 V en entrée et restitue 8 V à l'Arduino
8 - Les différentes cartes PCB (LED, HP, GDO)	Attribuer une carte PCB par circuit, permet un meilleur repérage et aide au dépannage des différentes parties ou fonctions du circuit	Les différentes cartes sont généralement fixées sur le châssis avec du scotch double face	 Les conducteurs et les autres éléments ont été soudés sur la carte suivant le schéma électrique
9 - Adafruit Motors Shield V2	Alimente et contrôle les deux moteurs	Le Shield est monté directement sur la carte Arduino	 Repiquage des pins Arduino et communication en série par les ports SDA et SCL (les communications séries utilisées sur l'Arduino, voir ANNEXE 17)

10 - Moteurs électriques	Permet le déplacement du char	ils sont montés dans les trous du châssis prévu à cet effet		Niveau câblage, ils sont montés sur les borniers M1 et M3 du Shield moteur
11 - Carte d'alimentation 5 V + carte des alimentations 10-12 V (commande + puissance)	Alimente les différents récepteurs du char en fonction de la tension de ce dernier (l'alimentation 5 V provient de l'Arduino et l'alimentation 10 - 12 V provient des piles ou accumulateurs)	Les cartes d'alimentation sont collées sur le châssis avec du scotch double face		Les conducteurs des différents circuits sont soudés dessus.

On peut ensuite passer au test et essai de notre projet.

1. Tests, essais

Les tests et essais restent une priorité absolue car ils permettent la vérification du bon fonctionnement des différents éléments répondant aux exigences prévus dans le cahier des charges.

Dans notre cas nous avons eu des modifications à réaliser plutôt du côté du code Arduino (nous avons téléverser du code modifié régulièrement sur l'Arduino de notre char car le fonctionnement ne correspondait pas entièrement à nos besoins), le câblage étant généralement correct, et la conception des différents éléments mécaniques aussi (nous avons quand même trouvé des soucis dans ces domaines). Nous allons récapituler sous forme de tableau, différents tests réalisés sur notre char.

Fonction	Test(s) effectué(s)	Problème(s) rencontré(s)	Solution(s) apportée(s)
Jet d'eau	Activation du jet d'eau par la manette	Le jet d'eau ne s'active pas et après quelques tests au multimètre, il s'avère que le MOSFET ne s'active pas	Shuntage de la résistance de $1\text{k}\Omega$ située en amont de la GATE du MOSFET du circuit jet d'eau (GDO) 
« Multiple »	Test de mise sous tension des circuits de puissance et de commande du char	Les LED de présence d'alimentation ne s'allument pas ou scintillent	Problème de faux contacts ou de soudures ou conducteurs cassés, donc conducteurs ou soudures à refaire. 
Détection d'obstacle	Signalisation par le haut-parleur lors de la détection d'obstacle	Le volume reste trop élevé	On avait prévu d'alimenter le haut-parleur en 5 V, on a décidé de l'alimenter avec le 3,3 V de l'Arduino 
Jet d'eau	Activation du jet d'eau par la manette	La distance du jet n'est pas très grande	Réduction du diamètre du tuyau pour augmenter la vitesse de sortie du fluide 

Communication	Diriger le char avec la manette	Il y a une certaine latence entre le moment où on active le joystick et le moment où le char se met à se déplacer	Pas de solution apportée pour le moment, mais nous pensons que pour pallier ce problème, il faudrait utiliser des modules Bluetooth ou Wifi plus efficaces ou tout simplement utiliser la communication série en filaire
Détection d'obstacle	Réaction(s) du char lorsqu'un obstacle est détecté et lorsque le joystick est activé	Le char ne s'arrête pas lorsque l'obstacle est détecté et il ne recule pas lorsque le joystick est activé vers le bas	Nous avons modifié le code en rajoutant la fonction <code>data_manage()</code> en amont de la condition « if », dans la fonction <code>detection_obstacle()</code> 
Déplacement général du char (peu importe le mode)	Déplacement du char avec le joystick ou déplacement du char en mode « suiveur de ligne »	Perte de puissance plus ou moins rapide du char	8 piles en série fournissent généralement 12 V, tandis que 8 accumulateurs 9,2 V. Les deux moteurs ayant une tension de fonctionnement nominale fixée à 12 V, la perte de puissance se ressent très tôt dans notre cas.

Parmi les difficultés techniques récentes développées précédemment, il en existe d'autres qui nous ont aussi impacté tout au long de notre projet ; nous en parlerons dans la partie suivante.

VI. DIFFICULTÉS RENCONTRÉES

Les retards associés aux modifications de la planification initiale, auxquels s'ajoute des soucis organisationnels, sont autant de difficultés diverses rencontrées lors de la réalisation de notre char.

1. Retard et modification de la planification initiale

Bien heureux d'avoir pu finaliser notre projet, nous avons quand même eu du retard dans la planification initiale du fait de retard de livraison de nos commandes. Nous nous sommes organisés pour ne pas perdre de temps.

En effet ne disposant pas des pièces nécessaires pour passer à la réalisation du char, nous avons réalisé l'étape de conception informatique (programmes Arduino) entre temps.

De plus, la ressource du Maitre d'œuvre ou MOE (Didier), pour combler le retard d'obtention de certains éléments commandés ou non, a dû utiliser provisoirement son matériel personnel pour pouvoir avancer dans le projet (émetteur / récepteur Bluetooth et Arduino Méga reçu tardivement).

2. Difficultés organisationnelles

Nous en étions conscients au début du projet, mais du fait de l'alternance école-entreprise, on a eu peu de temps pour travailler ensemble sur la réalisation du projet (possible grâce aux cours de projet à l'IUT). De ce fait, Didier a dû avancer sur la réalisation du char chez lui, sachant qu'il dispose du matériel requis. Ainsi, nous nous étions fixés à chacun 50% de charge de travail tout au long du projet, mais ce fut trop ambitieux.

VII. CONCLUSION

1. Évolutions possibles du projet

À la suite de ce projet, nous avons quelques idées de modification ou d'amélioration comme par exemple :

- La carrosserie personnalisable : dans l'idée d'une mise en vente futur, on pourra imprimer à l'imprimante 3D la carrosserie du char avec le nom du futur propriétaire et à la couleur qu'il souhaite.
- L'amélioration du canon par la mise en place d'un moteur électrique permettant sa rotation latérale (grâce un autre joystick).
- Un système affichant le niveau de charge des piles et fixé sur la carrosserie du char (afficheur à segments par exemple...)

2. Apports personnels et/ou professionnels

Durant ce projet complet qui regroupait des phases de réflexion, d'étude, de conception, et de réalisation, nous avons pu apprendre à :

- Travailler en équipe : au niveau de l'ambiance mais aussi dans la répartition des tâches pour mener à bien le projet, car sans cohésion le projet n'aurait pas pu aboutir.
- Respecter des délais : Pour respecter au mieux le planning et pouvoir finaliser notre projet dans le temps imparti
- Aboutir au résultat : même si l'objectif principal des projets tutorés est d'utiliser nos connaissances et/ou compétences et d'en acquérir d'autres, nous nous devions de fournir un char fonctionnelle répondant au cahier des charges.

VIII. ANNEXES

ANNEXE 01	Diagramme de GANTT prévisionnel et ressources
ANNEXE 02	Diagramme de GANTT final et ressources
ANNEXE 03	Tableau des risques et parades du char bi-mode
ANNEXE 04	Analyse environnementale du char bi mode
ANNEXE 05	Liens inter-projets (Mind Mapping final)
ANNEXE 06	Choix des composants électroniques de base
ANNEXE 07	Dessins d'ensemble et de définition du boîtier de la manette
ANNEXE 08	Dessin de définition de la carrosserie du char
ANNEXE 09	Dessin de définition du bac de rétention d'eau
ANNEXE 10	Dessins de définition des languettes de fixation
ANNEXE 11	Schéma électrique de la manette
ANNEXE 12	Schéma électrique du char
ANNEXE 13	Code en langage C de l'Arduino de la manette
ANNEXE 14	Code en langage C de l'Arduino du char
ANNEXE 15	Image du réservoir d'eau (réservoir finalisé)
ANNEXE 16	Procédure de configuration des modules Bluetooth
ANNEXE 17	Les communications séries utilisées sur les Arduino
ANNEXE 18	Reporting mensuels
ANNEXE 19	Schéma d'implantation simplifié des composants

ANNEXE 01

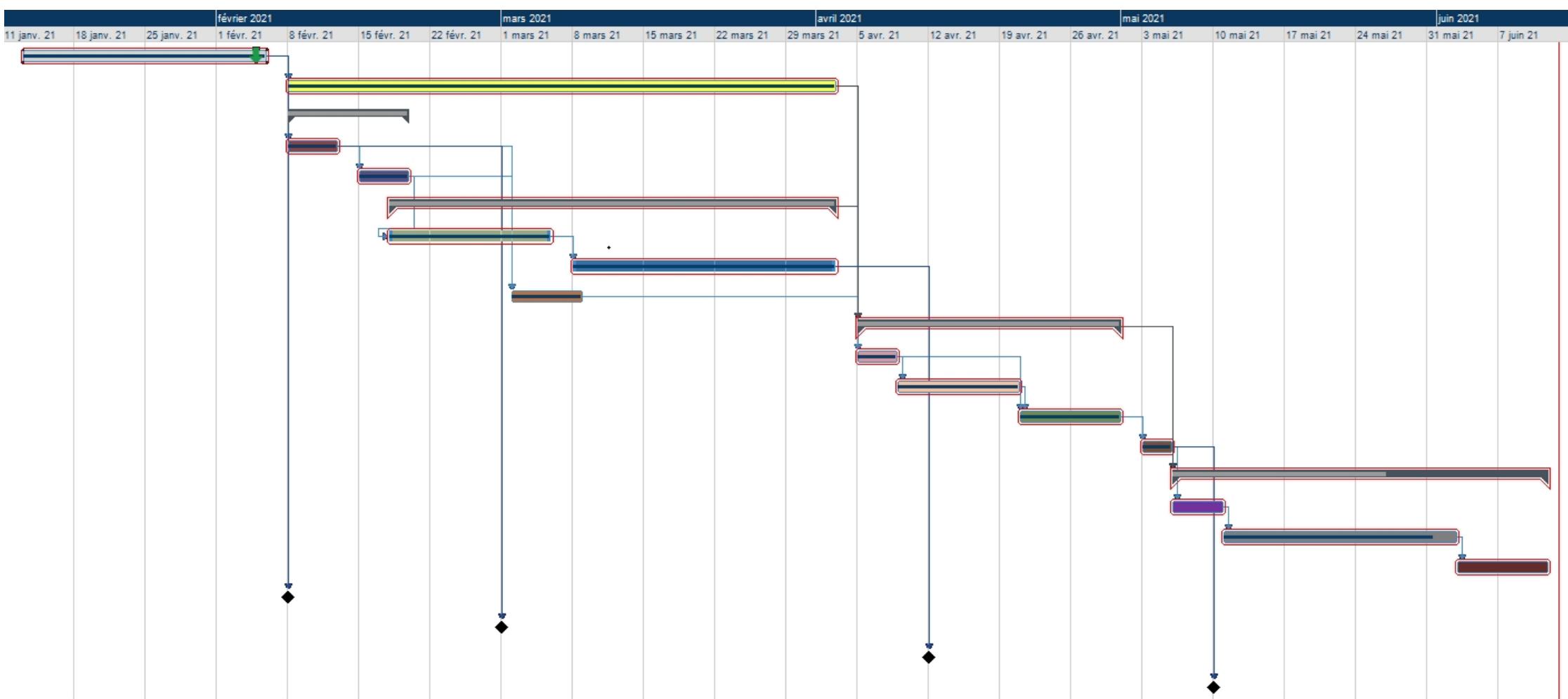
Diagramme de GANTT prévisionnel et ressources

LOGICIEL UTILISÉ :
MINDVIEW 7

- Tâches et ressources

		Nom de tâche	Durée	Début	Fin	Préd...	Pro...	Prio...	Ressources	Travail	Coût
1	0	✓ Étude préalable du système - Cahier des charges	18 jours	13/01/2021	05/02/2021		100%	500	Didier[60%];Isaac[40%]	144 h	0,00 €
2	0	✓ Commande du matériel	1 jour	08/02/2021	08/02/2021	1	100%	500	Didier[50%];Isaac[50%]	8 h	200,00 €
3		Réception du matériel	58 jours?	08/03/2021	26/05/2021	2	90%	500	Didier[50%];Isaac[50%]	464 h	0,00 €
4	0	✓ Conceptions mécaniques du char, de la télécommande, du boîtier de rétention d'eau et des supports de fixation	12 jours	08/02/2021	23/02/2021		100%	500		96 h	0,00 €
5		✓ 1. Conception de la carrosserie du char	5 jours	08/02/2021	12/02/2021	1	100%	500	Didier[70%];Isaac[30%]	40 h	0,00 €
6		✓ 2. Conception de la télécommande	5 jours	15/02/2021	19/02/2021	5	100%	500	Didier[70%];Isaac[30%]	40 h	0,00 €
7		✓ 3. Conception du boîtier de rétention d'eau et des supports de fixation	2 jours	22/02/2021	23/02/2021	6	100%	500	Didier[80%];Isaac[20%]	16 h	0,00 €
8		✓ Conceptions électriques du char, de la manette, des PCB et conception des deux programmes Arduino	32 jours	18/02/2021	02/04/2021		100%	500		292 h	0,00 €
9		✓ 1. Conception de la manette et des PCB	12 jours	18/02/2021	05/03/2021	6	100%	500	Didier[80%];Isaac[20%]	96 h	0,00 €
10		✓ 2. Conception du char	20 jours	08/03/2021	02/04/2021	9	100%	500	Didier[70%];Isaac[30%]	160 h	0,00 €
11		✓ 3. Conception des programmes Arduino	5 jours	24/02/2021	02/03/2021	4	100%	500	Didier[90%];Isaac[10%]	36 h	0,00 €
12	0	✓ Impression 3D de la carrosserie du char, de la télécommande, du boîtier de rétention d'eau et des supports de fixation	5 jours	08/03/2021	12/03/2021	5;6	100%	500	Isaac[50%];Didier[50%]	40 h	1,00 €
13	0	✓ Réalisation du char	46 jours	05/04/2021	07/06/2021	8;2	100%	500		1103,98 h	0,00 €
14		✓ 1. Création des programmes Arduino	46 jours	05/04/2021	07/06/2021		100%	500	Didier[50%];Isaac[50%]	368 h	0,00 €
15		✓ 2. Montage mécanique	46 jours	05/04/2021	07/06/2021	12;10;7	100%	500	Isaac[50%];Didier[50%]	368 h	0,00 €
16		✓ 3. Montage électrique/électronique	46 jours	05/04/2021	07/06/2021		100%	500	Didier[70%];Isaac[30%]	367,98 h	0,00 €
17	0	✓ Finalisation, tests et essais de fonctionnement	5 jours	08/06/2021	14/06/2021	14	100%	500	Didier[70%];Isaac[30%]	40 h	0,00 €
18		✓ Rédaction et présentation	17 jours	24/05/2021	15/06/2021		58%	500		144 h	0,00 €
19		✓ 1. Rédaction du manuel utilisateur	1 jour	15/06/2021	15/06/2021	17	0%	500	Didier[50%];Isaac[50%]	8 h	0,00 €
20		✓ 2. Rédaction du rapport de projet	13 jours	24/05/2021	09/06/2021		80%	500	Didier[60%];Isaac[40%]	104 h	0,00 €
21		✓ 3. Rédaction et réalisation de la présentation numérique du projet (powerpoint)	4 jours	10/06/2021	15/06/2021	20	0%	500	Didier[50%];Isaac[50%]	32 h	0,00 €
22		✓ Jalon 1 : Remise du cahier des charges	0 jours	08/02/2021	08/02/2021	1	100%	500		0 h	0,00 €
23		✓ Jalon 2 : Point d'avancement conceptions (livrable #1)	0 jours	01/03/2021	01/03/2021	5	100%	500		0 h	0,00 €
24		✓ Jalon 3 : Point d'avancement finalisation des conceptions - début de réalisation (livrable #2)	0 jours	12/04/2021	12/04/2021	10	100%	500		0 h	0,00 €
25		✓ Jalon 4 : Point d'avancement d'exécution du projet (livrable #3)	0 jours?	14/06/2021	14/06/2021	17	100%	500		0 h	0,00 €
26		✓ Examen final : Soutenance du Projet	0 jours	17/06/2021	17/06/2021		0%	500		0 h	0,00 €

- GANTT



ANNEXE 02

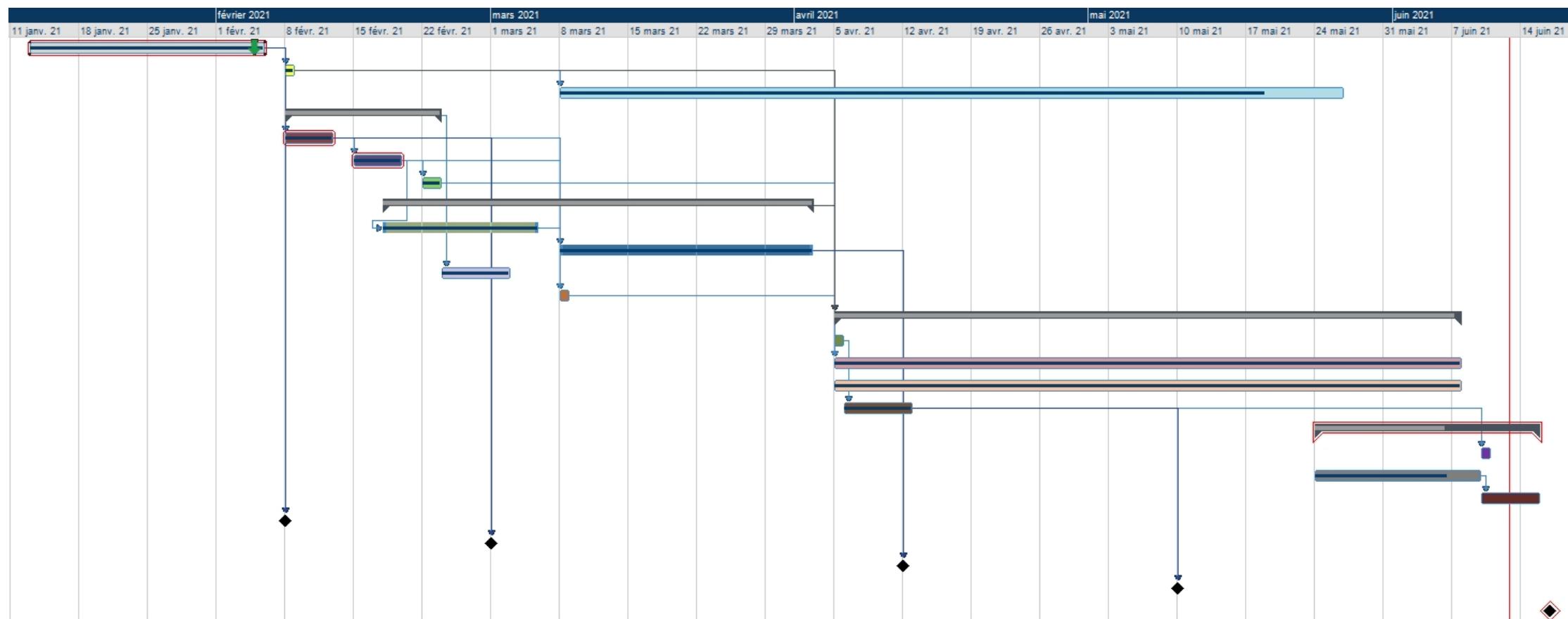
Diagramme de GANTT final et ressources

LOGICIEL UTILISÉ :
MINDVIEW 7

- Tâches et ressources

		Nom de tâche	Durée	Début	Fin	Préd...	Pro...	Prio...	Ressources	Travail	Coût
1	0	Étude préalable du système - Cahier des charges	18 jours	13/01/2021	05/02/2021		100%	500	Didier[60%];Isaac[40%]	144 h	0,00 €
2	0	Commande du matériel	1 jour	08/02/2021	08/02/2021	1	100%	500	Didier[50%];Isaac[50%]	8 h	200,00 €
3		Réception du matériel	58 jours?	08/03/2021	26/05/2021	2	90%	500	Didier[50%];Isaac[50%]	464 h	0,00 €
4	0	Conceptions mécaniques du char, de la télécommande, du boîtier de rétention d'eau et des supports de fixation	12 jours	08/02/2021	23/02/2021		100%	500		96 h	0,00 €
5		1. Conception de la carrosserie du char	5 jours	08/02/2021	12/02/2021	1	100%	500	Didier[70%];Isaac[30%]	40 h	0,00 €
6		2. Conception de la télécommande	5 jours	15/02/2021	19/02/2021	5	100%	500	Didier[70%];Isaac[30%]	40 h	0,00 €
7		3. Conception du boîtier de rétention d'eau et des supports de fixation	2 jours	22/02/2021	23/02/2021	6	100%	500	Didier[80%];Isaac[20%]	16 h	0,00 €
8		Conceptions électriques du char, de la manette, des PCB et conception des deux programmes Arduino	32 jours	18/02/2021	02/04/2021		100%	500		292 h	0,00 €
9		1. Conception de la manette et des PCB	12 jours	18/02/2021	05/03/2021	6	100%	500	Didier[80%];Isaac[20%]	96 h	0,00 €
10		2. Conception du char	20 jours	08/03/2021	02/04/2021	9	100%	500	Didier[70%];Isaac[30%]	160 h	0,00 €
11		3. Conception des programmes Arduino	5 jours	24/02/2021	02/03/2021	4	100%	500	Didier[90%];Isaac[10%]	36 h	0,00 €
12	0	Impression 3D de la carrosserie du char, de la télécommande, du boîtier de rétention d'eau et des supports de fixation	5 jours	08/03/2021	12/03/2021	5;6	100%	500	Isaac[50%];Didier[50%]	40 h	1,00 €
13	0	Réalisation du char	46 jours	05/04/2021	07/06/2021	8;2	100%	500		1103,98 h	0,00 €
14		1. Création des programmes Arduino	46 jours	05/04/2021	07/06/2021		100%	500	Didier[50%];Isaac[50%]	368 h	0,00 €
15		2. Montage mécanique	46 jours	05/04/2021	07/06/2021	12;10;7	100%	500	Isaac[50%];Didier[50%]	368 h	0,00 €
16		3. Montage électrique/électronique	46 jours	05/04/2021	07/06/2021		100%	500	Didier[70%];Isaac[30%]	367,98 h	0,00 €
17	0	Finalisation, tests et essais de fonctionnement	5 jours	08/06/2021	14/06/2021	14	100%	500	Didier[70%];Isaac[30%]	40 h	0,00 €
18		Rédaction et présentation	17 jours	24/05/2021	15/06/2021		58%	500		144 h	0,00 €
19		1. Rédaction du manuel utilisateur	1 jour	15/06/2021	15/06/2021	17	0%	500	Didier[50%];Isaac[50%]	8 h	0,00 €
20		2. Rédaction du rapport de projet	13 jours	24/05/2021	09/06/2021		80%	500	Didier[60%];Isaac[40%]	104 h	0,00 €
21		3. Rédaction et réalisation de la présentation numérique du projet (powerpoint)	4 jours	10/06/2021	15/06/2021	20	0%	500	Didier[50%];Isaac[50%]	32 h	0,00 €
22		Jalon 1 : Remise du cahier des charges	0 jours	08/02/2021	08/02/2021	1	100%	500		0 h	0,00 €
23		Jalon 2 : Point d'avancement conceptions (livrable #1)	0 jours	01/03/2021	01/03/2021	5	100%	500		0 h	0,00 €
24		Jalon 3 : Point d'avancement finalisation des conceptions - début de réalisation (livrable #2)	0 jours	12/04/2021	12/04/2021	10	100%	500		0 h	0,00 €
25		Jalon 4 : Point d'avancement d'exécution du projet (livrable #3)	0 jours?	14/06/2021	14/06/2021	17	100%	500		0 h	0,00 €
26		Examen final : Soutenance du Projet	0 jours	17/06/2021	17/06/2021		0%	500		0 h	0,00 €

- GANTT



ANNEXE 03

Tableau des risques et parades du char-mode

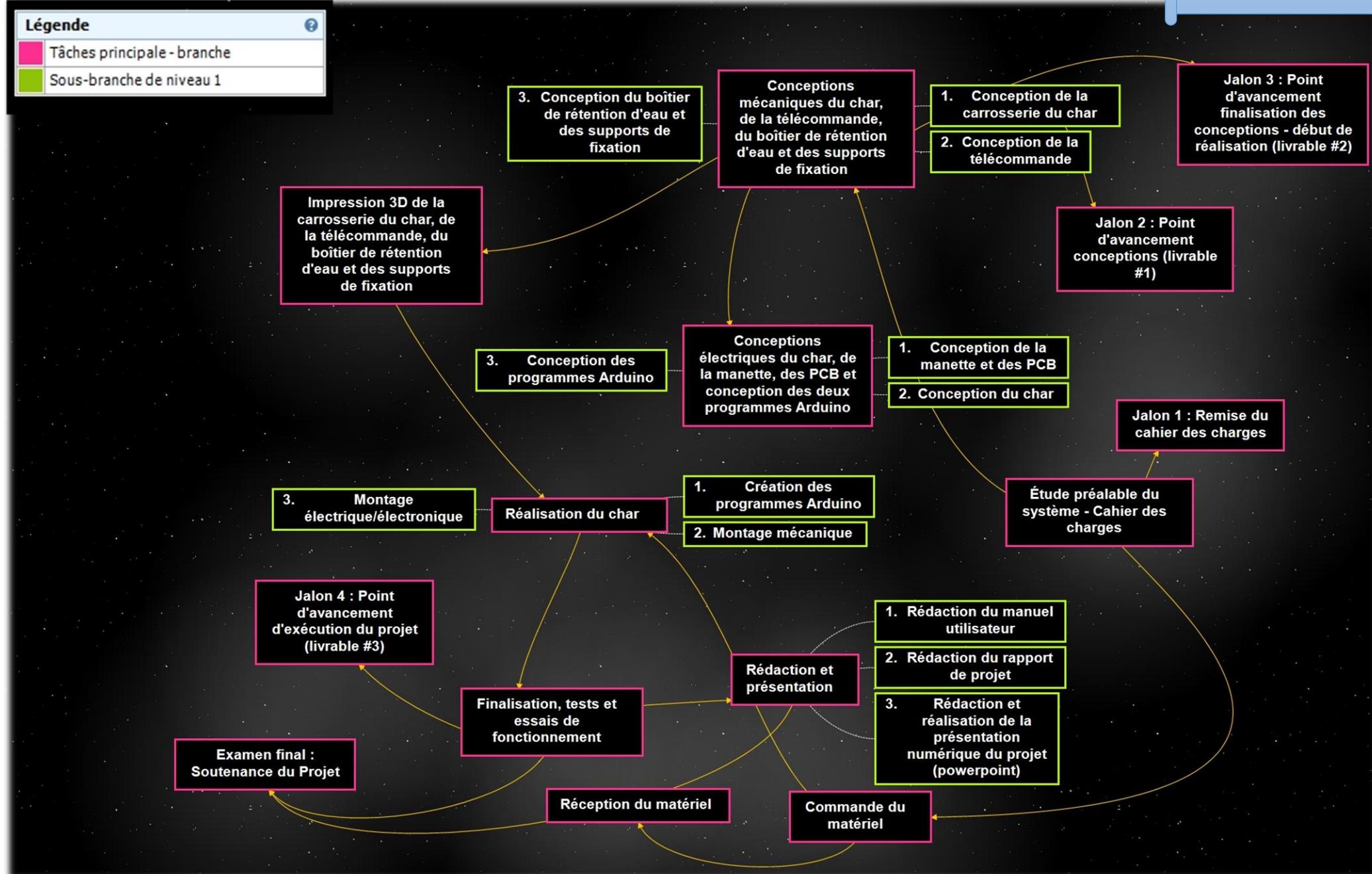
Danger / opportunité	Risque / potentiel positif	Occurrence (1 à 4)	Fréquence d'exposition (1 à 4)	Gravité (1 à 4)	Produit	Parades de prévention / moyens de valorisation	Parades de protection / moyens de sauvegarde	Nouvelle occurrence	Nouvelle fréquence d'exposition	Nouvelle gravité	Nouveau produit	Degré d'acceptabilité / de chance
Ex : Le jet d'eau du char est utilisé à côté d'une armoire électrique sous tension de 230 volts	Électrisation, électrocution	1	1	2	2	Bien connaître son environnement avant d'utiliser la pompe à eau	EPI	1	1	1	1	99%
Acquérir une expérience	Richesse intellectuelle	3	3	3	9	CV	Rapport de projet, photos, intellect	4	4	4	4	99%
Apprendre à anticiper	Autonomie	2	2	2	6	Prime salariale	Prise de notes, photos, intellect	3	3	3	3	99%
Recyclage de pièces	Recyclage	2	2	2	6	Économies	Prise de notes, photos, intellect, stockage	3	3	3	3	99%
Complexité du projet	Projet non finalisé, démorale de la MOE	2	3	4	24	Documentation régulière, conseil des professeurs	Prise de notes, photos, intellect	2	3	2	12	99%
Retard ou non livraison de produits commandés	Retard sur l'avancement du projet	3	3	4	36	Vérifier régulièrement l'état des commandes, utiliser des sites de commandes fiables	Prise de notes, photos, intellect	2	2	2	8	99%

Budget d'achat du matériel mal évalué	Retard sur l'avancement du projet, manque de pièces	2	3	3	12	Prévoir une marge d'imprévus de la ressource budgétaire de 3 à 10%, par rapport au budget du devis	Prise de notes, photos, intellect	1	2	2	4	99%
Composants de mauvaise qualité	Dysfonctionnement du système, rachat donc pertes budgétaires	3	3	3	27	Acheter des composants fiables (site fiable et bon rapport qualité/prix)	Prise de notes, photos, intellect	2	2	2	8	99%
Planning mal aménagé	Retard sur l'avancement du projet	2	2	3	12	Mettre à jour régulièrement le planning en fonction de l'avancement	Prise de notes, photos, intellect	1	2	2	4	99%
Coactivité mal gérée	Retard sur l'avancement du projet	2	2	2	8	Meilleure répartition des tâches entre les différents membres du MOE	Prise de notes, photos, intellect	1	1	1	1	99%

ANNEXE 04

Analyse environnementale du char bi-mode

Activités	Eléments consommés	Eléments rejetés	Aspect environnemental	Impact environnemental	Aspect environnemental significatif	Occurrence (1 à 4)	Fréquence d'exposition (1 à 4)	Gravité (1 à 4)	Produit (probabilité)
1 - Déplacer le char	Énergie		Consommation d'énergie	Pollution sonore	Santé humaine	1	1	1	1
2 - Arroser (jet d'eau)		Eau	Consommation d'eau	Pollution du sol	Locaux IUT	1	3	2	6
3 - Consommer l'énergie électrique	Piles		Consommation d'énergie	Épuisement des ressources naturelles	Écologie	3	1	2	6
4 - Jeter le char (fin de vie ou HS)		Caoutchouc, produits chimiques, châssis en métal, composants électroniques et carrosserie plastique	Production de déchets	Impacts liés au stockage, à la collecte et à l'incinération des déchets	Lieu de traitement sol/air	3	2	3	18
5 - Réparer le char		Composant(s) HS	Production de déchets	Impacts liés au stockage, à la collecte et à l'incinération des déchets	Lieu de traitement sol/air	3	2	2	12
6 - Soudure		Fumée toxique	Pollution sur corps humain et sol/air	Impacts sur le corps humain et sur sol/air	Pollution sol/air	3	1	2	6



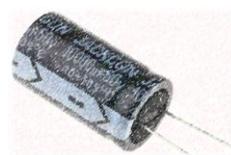
Résistances

- Protéger les LED (en limitant le courant)
- Protéger les pins Arduino (éviter les court-circuits)



Condensateurs

- Stabiliser et éliminer les variations de tension en entrée et en sortie du régulateur de tension



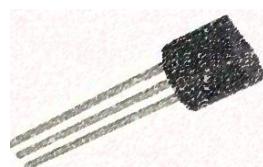
Diodes

- Protéger le circuit en cas d'inversion de l'alimentation
- Mode diode de roue libre : protège les circuits contre les surtensions lors de l'ouverture de la charge inductive



Transistors

- À la même fonction qu'un interrupteur : commute le courant par la commande en tension sur la "Gate"



D
C
B
A

4

4

3

3

2

2

1

1

D

A

Vue de face
Echelle : 4:5

Vue isométrique
Echelle : 7:10

Quantité	Référence	Type
4	Vis M3x16	Acier
1	Boitier_manette_rad iocommandée (bas)	PLA
1	Boitier_manette_rad iocommandée (haut)	PLA

DESIGNED BY: MAMILONNE Didier DATE: 13/02/2021	CHAR BI-MODE À JETS D'EAU	
CHECKED BY: XXX DATE: XXX		
SIZE A4		LP MÉCATRONIQUE
SCALE 9:20	WEIGHT (kg) 0,35	DRAWING NUMBER Boitier de la manette
		SHEET 1 / 7

This drawing is our property; it can't be reproduced or communicated without our written agreement.

I
–

H
–

G
–

F
–

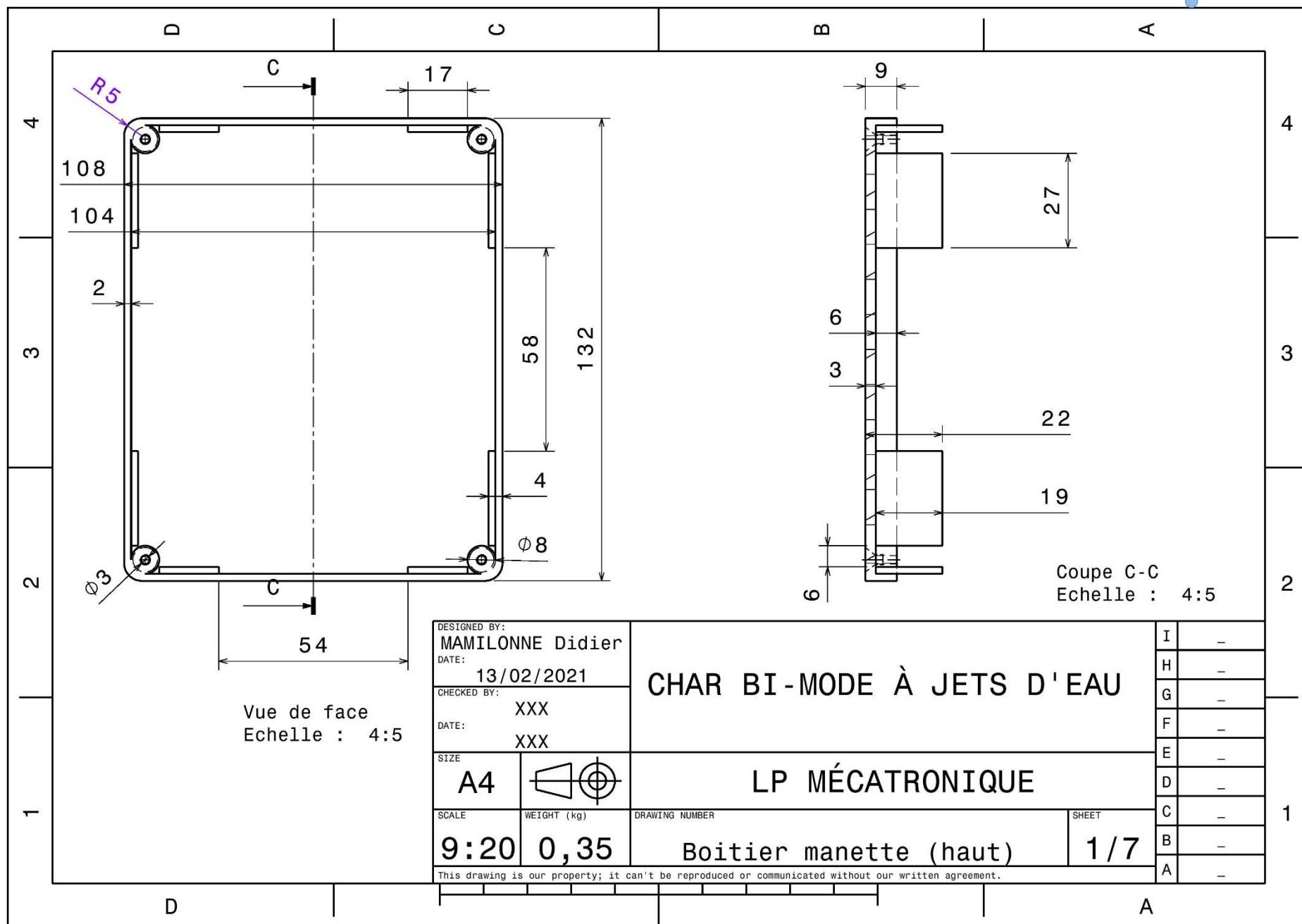
E
–

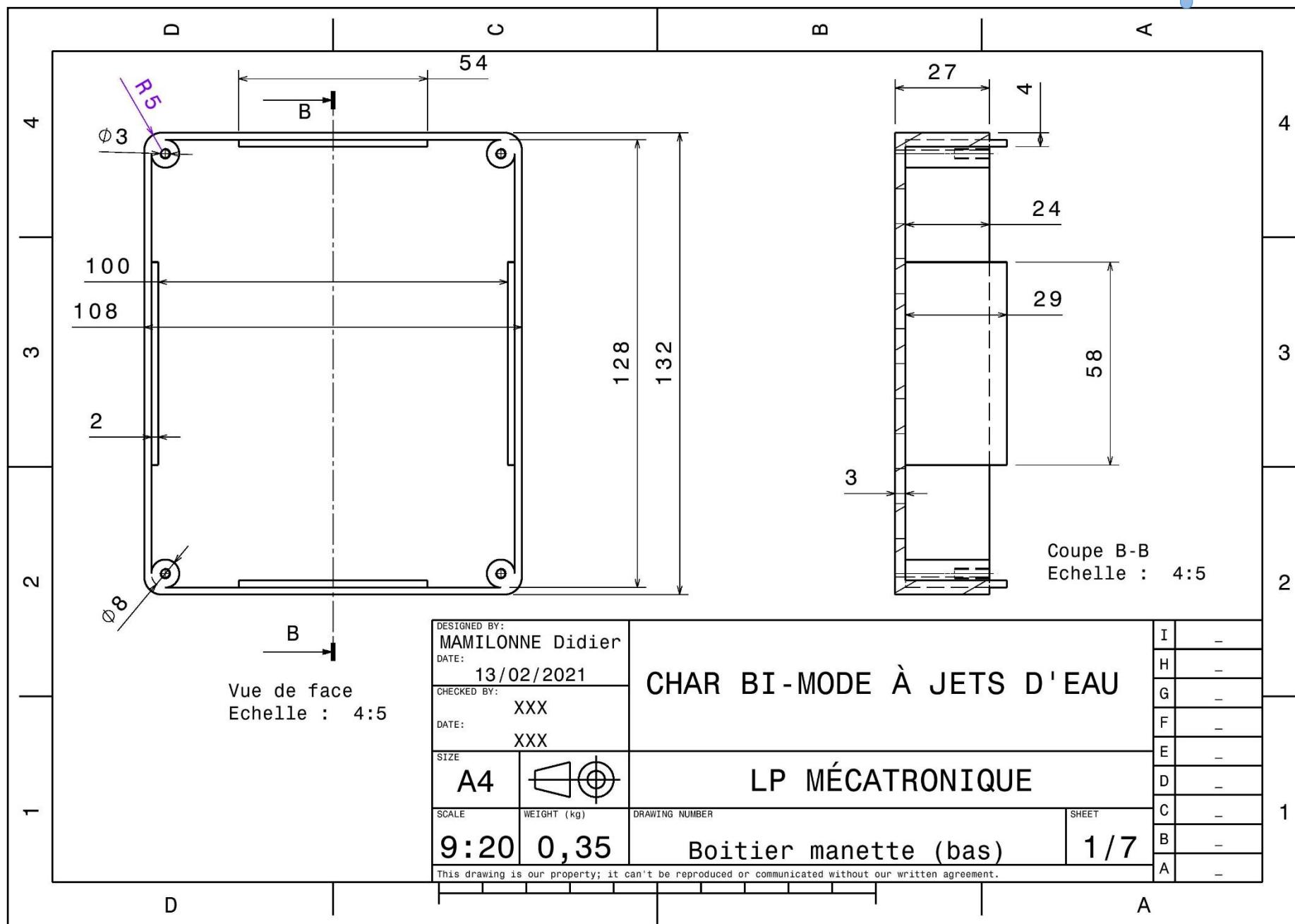
D
–

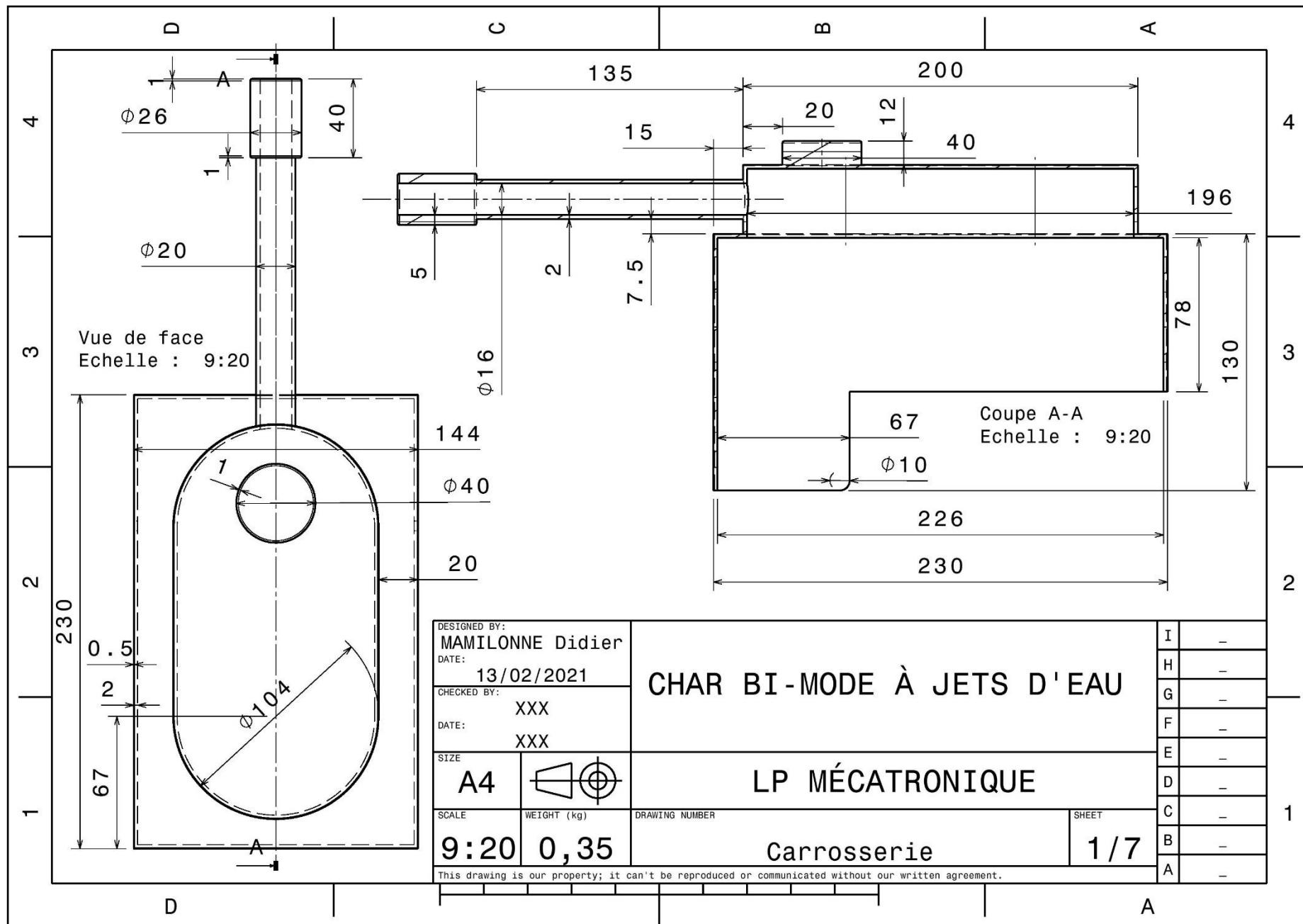
C
–

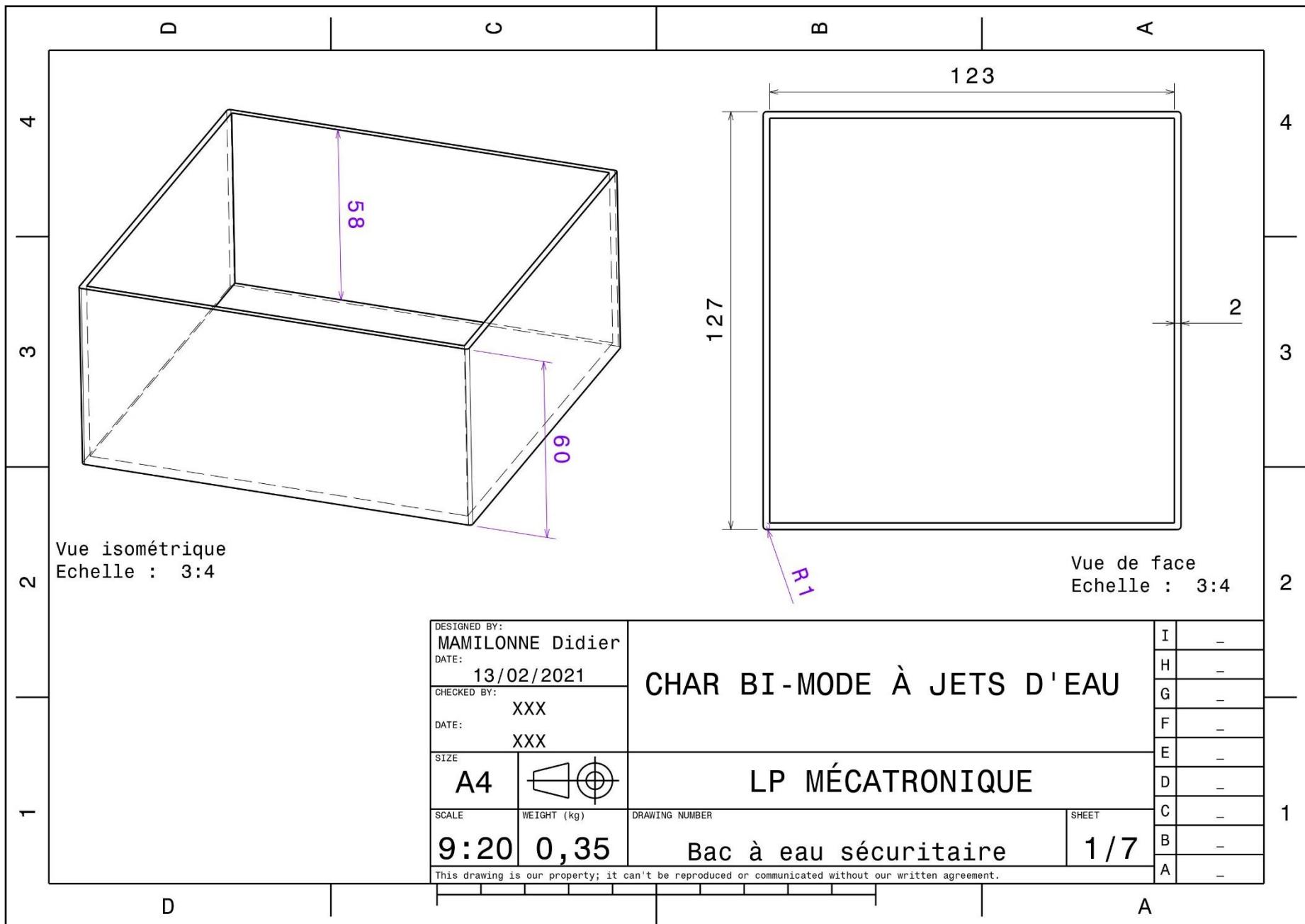
B
–

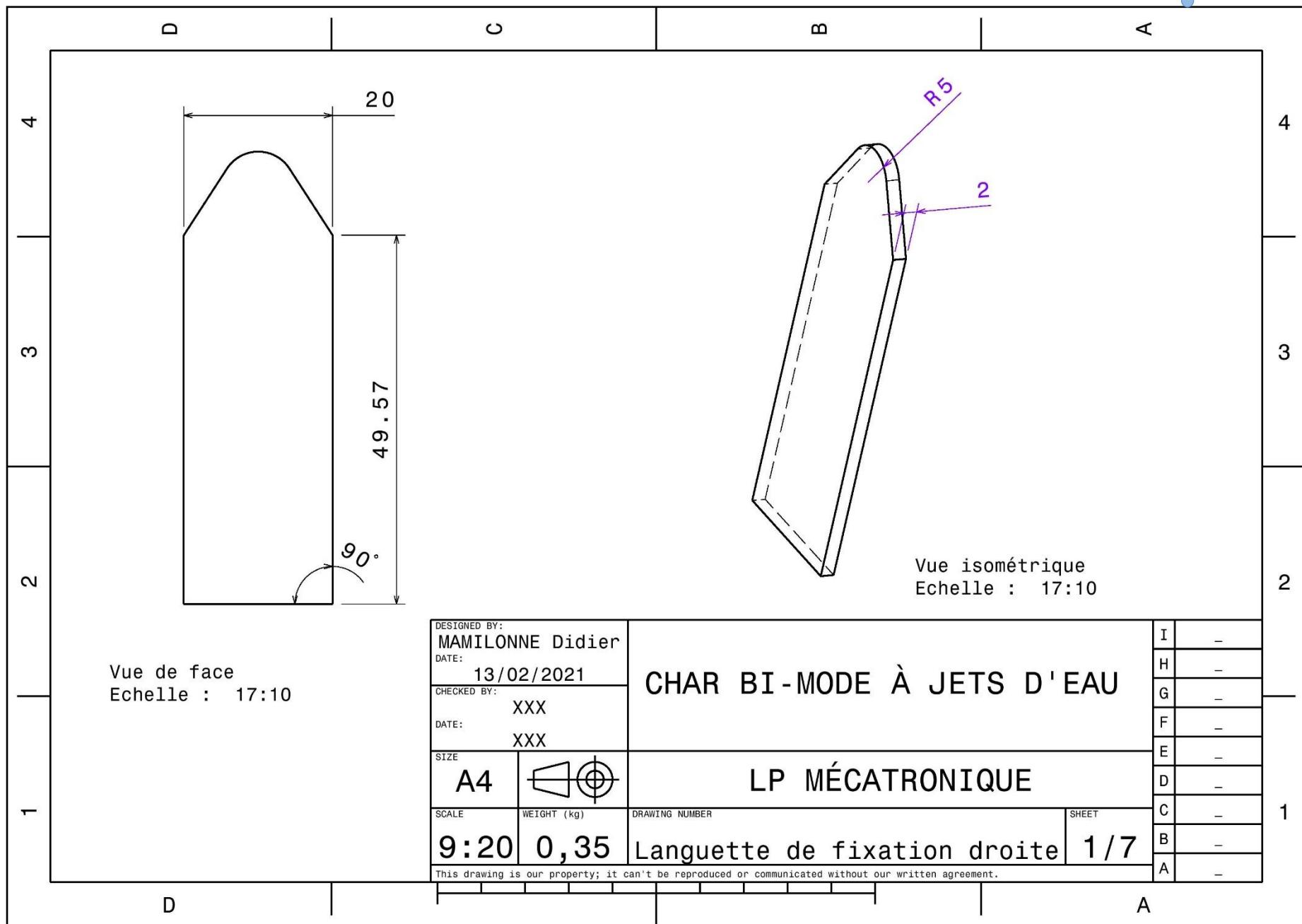
A
–

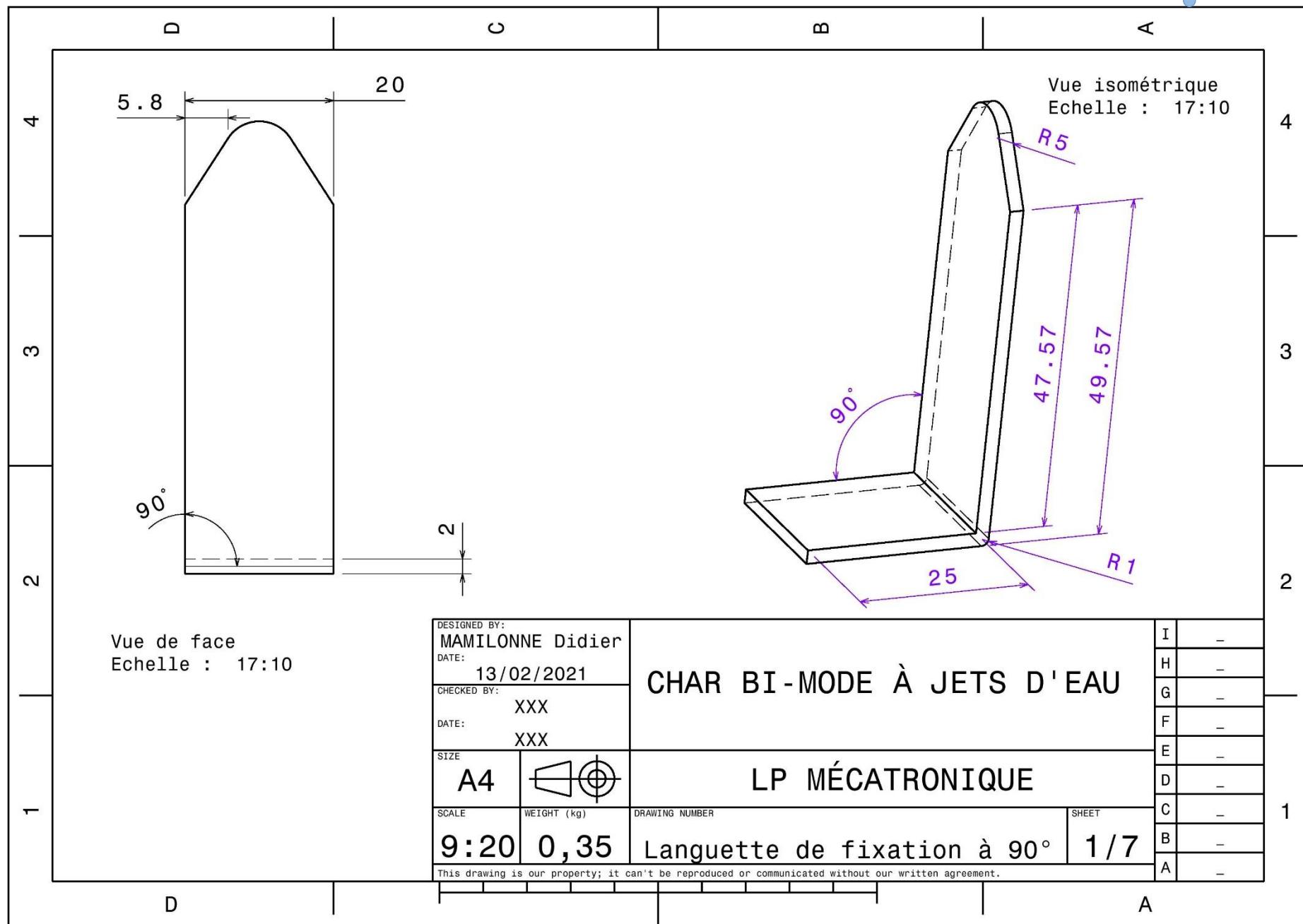






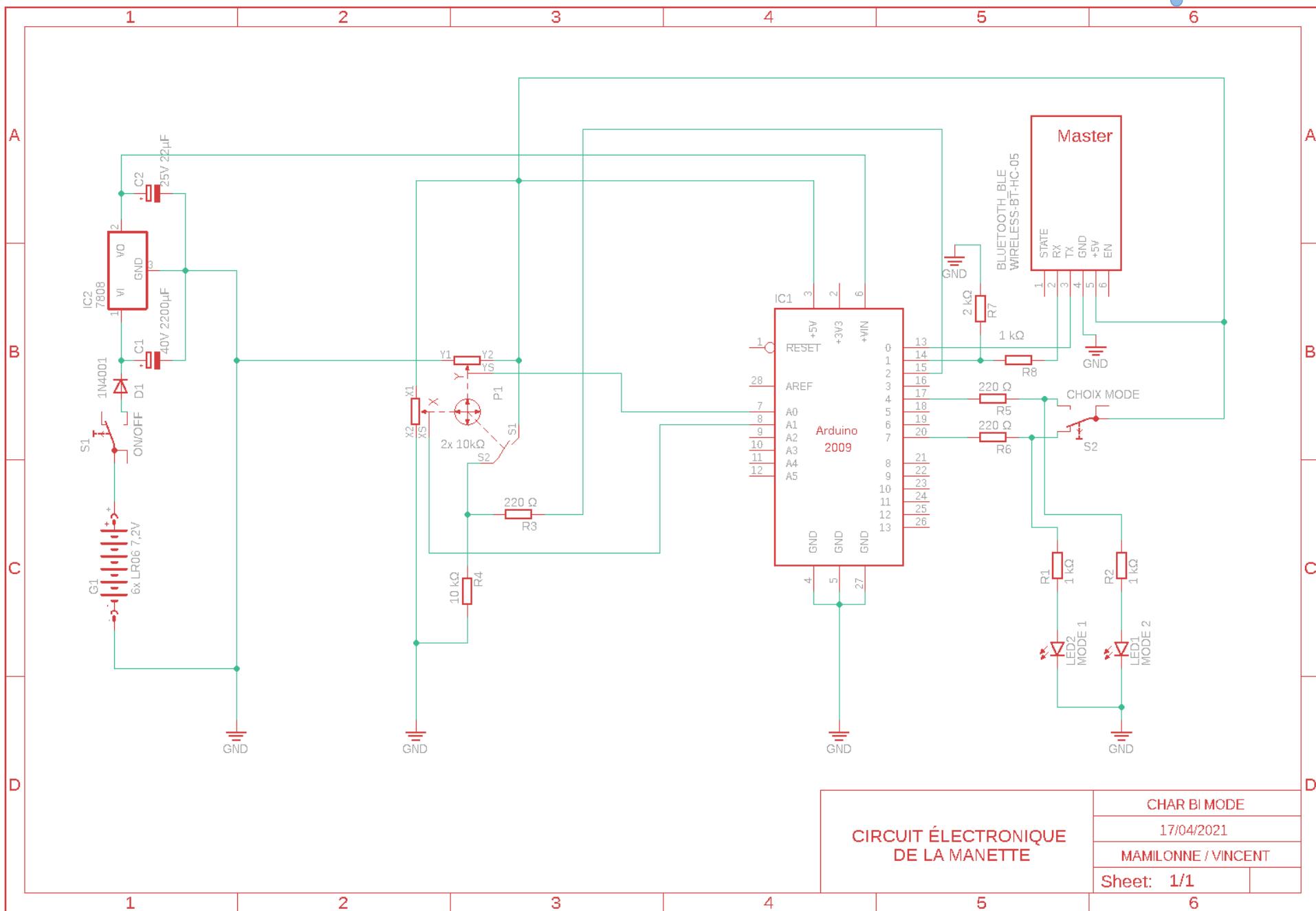






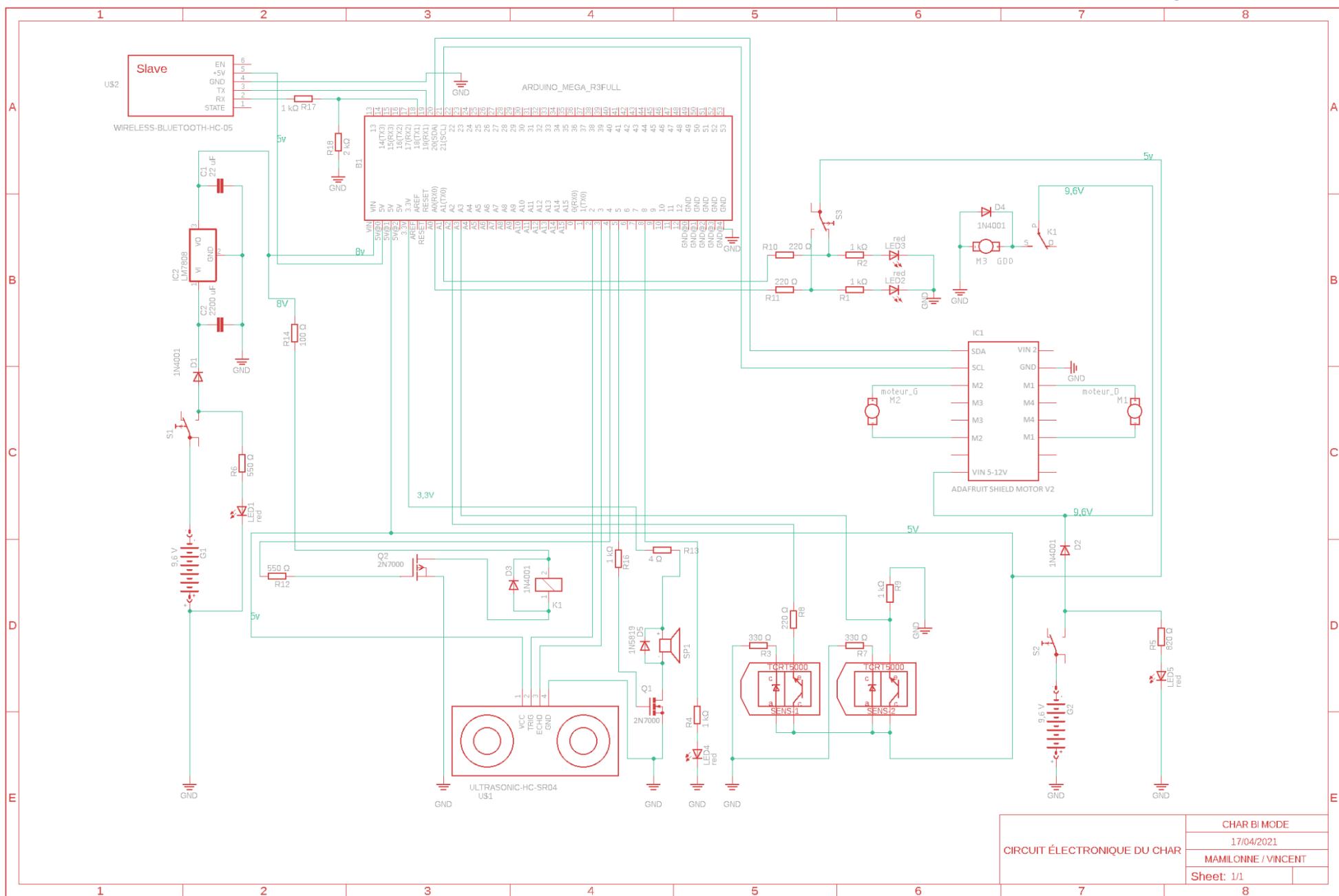
ANNEXE 11

Schéma électrique de la manette



ANNEXE 12

Schéma électrique du char



COMMENTAIRES DU CODE SURLIGNÉS EN GRIS

ANNEXE 13

Code en langage C de l'Arduino de la manette

LOGICIEL UTILISÉ : CODEBLOCKS::ARDUINO

```
//Déclaration des variables
unsigned int AO = A0;
unsigned int AI = A1;
unsigned int D2 = 2;
unsigned int D4 = 4;
unsigned int D7 = 7;
unsigned int D12 = 12;
unsigned int D10 = 10;
unsigned int mode1 = 0;
unsigned int mode2 = 0;
unsigned int analog0 = 0;
unsigned int analog1 = 0;
unsigned int digital0 = 0;
unsigned int digital1 = 0;
unsigned int etatSW = 0;

//-----
//-----send_data (int digit0,int digit1);
//Fonction permettant la comparaison et le transfert des données par le port série
//-----
//-----void setup()
{
    Serial.begin(9600);
//début de la transmission des données séries à 9600 bauds
    pinMode(AO, INPUT);
    pinMode(AI, INPUT);
```

```
pinMode(D2, INPUT_PULLUP);
//configuration du pin 2 en tant qu'entrée
    pinMode(D4, INPUT);
    pinMode(D7, INPUT);
}

//-----
//-----void loop()
{
    mode1 = digitalRead(D7);
    mode2 = digitalRead(D4);
    etatSW = digitalRead(D2);
    analog0 = analogRead(AO);
//On affecte à la variable analog0 les valeurs analogiques reçues sur le pin A0
    analog1 = analogRead(AI);
    digital0 = map(analog0, 0, 1023, 1, 100);
//On convertit les valeurs analogiques en valeurs numériques [0,1023] puis mise à l'échelle [1,100]
    digital1 = map(analog1, 0, 1023, 101, 200);

    if (mode1 == HIGH && mode2 == LOW)
    {
        send_data(digital0,digital1);
    }
    else if (mode2 == HIGH && mode1 == LOW)
    {
```

```
        Serial.println('Y');
//Telecommande : mode asservi
        delay (500);
    }
}

//-----
//-----void send_data (int digit0, int digit1)
{
    if (digit0 <= 10 && digit0 >= 1 && digit1 < 155 && digit1 > 145)
    {
        Serial.println('T');
//Reculer : grande vitesse
        delay (500);
    }
    else if (digit0 <= 25 && digit0 >= 11 && digit1 < 155 && digit1 > 145)
    {
        Serial.println('S');
//Reculer : moyenne vitesse
        delay (500);
    }
    else if (digit0 <= 45 && digit0 >= 26 && digit1 < 155 && digit1 > 145)
    {
        Serial.println('R');
//Reculer : petite vitesse
        delay (500);
    }
}
```

```

-----  

    else if (digit0 <= 65 &&  
digit0 >= 55 && digit1 < 155 &&  
digit1 > 145)  
{  
    Serial.println('A');  
    //Avancer : petite vitesse  
    delay (500);  
}  
else if (digit0 <= 80 &&  
digit0 >= 66 && digit1 < 155 &&  
digit1 > 145)  
{  
    Serial.println('B');  
    //Avancer : moyenne vitesse  
    delay (500);  
}  
else if (digit0 <= 100 &&  
digit0 >= 81 && digit1 < 155 &&  
digit1 > 145)  
{  
    Serial.println('C');  
    //Avancer : grande vitesse  
    delay (500);  
}  
  
-----  

    else if (digit1 <= 110 &&  
digit1 >= 101 && digit0 < 55 &&  
digit0 > 45)  
{  
    Serial.println('F');  
    //Droite : grande vitesse  
    delay (500);  
}

```

```

-----  

    else if (digit1 <= 125 &&  
digit1 >= 111 && digit0 < 55 &&  
digit0 > 45)  
{  
    Serial.println('E');  
    //Droite : moyenne vitesse  
    delay (500);  
}  
else if (digit1 <= 145 &&  
digit1 >= 126 && digit0 < 55 &&  
digit0 > 45)  
{  
    Serial.println('D');  
    //Droite : petite vitesse  
    delay (500);  
}  
  
-----  

    else if (digit1 <= 165 &&  
digit1 >= 155 && digit0 < 55 &&  
digit0 > 45)  
{  
    Serial.println('G');  
    //Gauche : petite vitesse  
    delay (500);  
}  
else if (digit1 <= 180 &&  
digit1 >= 166 && digit0 < 55 &&  
digit0 > 45)  
{  
    Serial.println('H');  
    //Gauche : moyenne vitesse  
    delay (500);  
}  
else if (digit1 <= 200 &&  
digit1 >= 181 && digit0 < 55 &&  
digit0 > 45)

```

```

{  
    Serial.println('I');  
    //Gauche : grande vitesse  
    delay (500);  
}  
  
-----  

else if (etatSW == LOW)  
{  
    Serial.println('O');  
    //GDO  
    delay (500);  
}  
else  
{  
    Serial.println('Z');  
    //Pour calibrage  
    delay (500);  
}

```

**COMMENTAIRES DU
CODE SURLIGNÉS EN GRIS**

ANNEXE 14

Code en langage C de l'Arduino du char

```
#include <Adafruit_MotorShield.h>

Adafruit_MotorShield AFMS =
Adafruit_MotorShield();
Adafruit_DCMotor *rightMotor =
AFMS.getMotor(1);
Adafruit_DCMotor *leftMotor =
AFMS.getMotor(3);

char data = '0';
//Déclaration des variables
unsigned int DR0A = 0;
unsigned int DR1A = 0;
unsigned int DR2A = 0;
unsigned int DR3A = 0;
unsigned int DR3 = 0;
unsigned int DR9 = 0;
unsigned int DR10 = 0;
unsigned int AOO = 0;
unsigned int AIO = 0;
unsigned int D2 = 2;
unsigned int D3 = 3;
unsigned int D4 = 4;
unsigned int D5 = 5;
//unsigned int D6 = 6;
unsigned int D7 = 7;
unsigned int D8 = 8;
unsigned int D9 = 9;
unsigned int D10 = 10;
//unsigned int D11 = 11;
//unsigned int D12 = 12;
//unsigned int D13 = 13;
unsigned int AO = A0;
unsigned int AI = A1;
unsigned int AII = A2;
unsigned int AIII = A3;
```

```
unsigned int distance_min = 18;
// détection d'obstacle : En cm
unsigned int DO = 0;

//-----
//-----
```

/* DÉBUT DES DIFFÉRENTES FONCTIONS */

```
//-----
//-----
```

```
void Data_manage()
//Fonction qui gère les données reçues par la liaison série
{
    if (Serial.available() > 0)
    {
        data = Serial.read();
        Serial.print(data);
    }
}

//-----
//-----
```

```
void Alarm(void)
//Alarme du Haut-parleur
{
    Data_manage();
    digitalWrite(D8, HIGH);
    tone(D5,300);
    delay(1000);
    tone(D5,600);
    digitalWrite(D8, LOW);
```

```
delay(1000);
tone(D5,900);
delay(1000);
digitalWrite(D8, HIGH);
tone(D5,1200);
delay(2000);
}

//-----
//-----
```

```
void GDO(void)
//Fonction qui active ou non le jet d'eau
{
    if (data == 'O' && DR0A == HIGH)
    {
        digitalWrite(D4, HIGH);
        Serial.println("GDO");
    }
    else
    {
        digitalWrite(D4, LOW);
    }
}

//-----
//-----
```

```
void DetectObstacle (void)
//Fonction qui détecte un obstacle en face
{
    long duree = 0;
    long distance = 0;
    digitalWrite(D2, LOW);
```

**LOGICIEL UTILISÉ :
CODEBLOCKS::ARDUINO**

```

        delayMicroseconds(2);
// 02 Millisecondes
        digitalWrite(D2, HIGH);
        delayMicroseconds(10);
// 10 Millisecondes
        digitalWrite(D2, LOW);
        duree = pulseIn(D3, HIGH);
        distance = duree * 340 / (2 *
10000);
        if (distance > distance_min)
        {
            Data_manage();
            DO = 0;
            digitalWrite(D8, LOW);
            noTone(D5);
        }
        else if (distance <=
distance_min)
        {
            Data_manage();
            DO = 1;
            Alarm();
            if (DROA == HIGH)
            {
                Mode_telecommande2();
            }
        }
        else
        {
            Data_manage();
        }
    }

//-----

```

```

void Desactivation_moteur(void)
//Fonction qui désactive tous les
moteurs
{
    digitalWrite(D4, LOW);
    leftMotor->run(RELEASE);
    rightMotor->run(RELEASE);
    Data_manage();
}

//-----
void Moteur_avancer1(void)
//Fonction qui fait avancer le
char en petite vitesse (mode
télécommandé)
{
    //analogWrite(D9, 85);
    //analogWrite(D10, 85);
    //
    rightMotor->setSpeed(85);
    rightMotor->run(FORWARD);
    leftMotor->setSpeed(85);
    leftMotor->run(FORWARD);
    //
}

void Moteur_avancer2(void)
//Fonction qui fait avancer le
char en moyenne vitesse (mode
télécommandé)
{
    //analogWrite(D9, 170);
    //analogWrite(D10, 170);
    //
    rightMotor->setSpeed(170);
    rightMotor->run(FORWARD);
    leftMotor->setSpeed(170);
    //
}

```

```

leftMotor->run(FORWARD);
//_____
}

void Moteur_avancer3(void)
//Fonction qui fait avancer le
char en grande vitesse (mode
télécommandé)
{
    //analogWrite(D9, 255);
    //analogWrite(D10, 255);
    //
    rightMotor->setSpeed(255);
    rightMotor->run(FORWARD);
    leftMotor->setSpeed(255);
    leftMotor->run(FORWARD);
    //
}

//-----
void Moteur_reculer1(void)
//Fonction qui fait reculer le
char en petite vitesse (mode
télécommandé)
{
    //analogWrite(D11, 85);
    //analogWrite(D6, 85);
    //
    rightMotor->setSpeed(85);
    rightMotor->run(BACKWARD);
    leftMotor->setSpeed(85);
    leftMotor->run(BACKWARD);
    //
}

void Moteur_reculer2(void)
//Fonction qui fait reculer le

```

```
char en moyenne vitesse (mode  
télécommandé)  
{  
    //analogWrite(D11, 170);  
    //analogWrite(D6, 170);  
    //  
    rightMotor->setSpeed(170);  
    rightMotor->run(BACKWARD);  
    leftMotor->setSpeed(170);  
    leftMotor->run(BACKWARD);  
    //  
}  
  
void Moteur_reculer3(void)  
//Fonction qui fait reculer le  
char en grande vitesse (mode  
télécommandé)  
{  
    //analogWrite(D11, 255);  
    //analogWrite(D6, 255);  
    //  
    rightMotor->setSpeed(255);  
    rightMotor->run(BACKWARD);  
    leftMotor->setSpeed(255);  
    leftMotor->run(BACKWARD);  
    //  
}  
  
-----  
  
void Moteur_gauche1(void)  
//Fonction qui fait tourner le  
char à gauche en petite vitesse  
(mode télécommandé)  
{  
    //analogWrite(D9, 85);  
    //analogWrite(D10, 0);  
    //
```

```
rightMotor->setSpeed(85);  
rightMotor->run(FORWARD);  
leftMotor->run(RELEASE);  
//  
}  
  
void Moteur_gauche2(void)  
//Fonction qui fait tourner le  
char à gauche en moyenne vitesse  
(mode télécommandé)  
{  
    //analogWrite(D9, 170);  
    //analogWrite(D10, 0);  
    //  
    rightMotor->setSpeed(170);  
    rightMotor->run(FORWARD);  
    leftMotor->run(RELEASE);  
    //  
}  
  
void Moteur_gauche3(void)  
//Fonction qui fait tourner le  
char à gauche en grande vitesse  
(mode télécommandé)  
{  
    //analogWrite(D9, 255);  
    //analogWrite(D10, 0);  
    //  
    rightMotor->setSpeed(255);  
    rightMotor->run(FORWARD);  
    leftMotor->run(RELEASE);  
    //  
}  
  
-----  
  
void Moteur_droit1(void)  
//Fonction qui fait tourner le
```

```
char à droite en petite vitesse  
(mode télécommandé)  
{  
    //analogWrite(D10, 85);  
    //analogWrite(D9, 0);  
    //  
    leftMotor->setSpeed(85);  
    leftMotor->run(FORWARD);  
    rightMotor->run(RELEASE);  
    //  
}  
  
void Moteur_droit2(void)  
//Fonction qui fait tourner le  
char à droite en moyenne vitesse  
(mode télécommandé)  
{  
    //analogWrite(D10, 170);  
    //analogWrite(D9, 0);  
    //  
    leftMotor->setSpeed(170);  
    leftMotor->run(FORWARD);  
    rightMotor->run(RELEASE);  
    //  
}  
  
void Moteur_droit3(void)  
//Fonction qui fait tourner le  
char à droite en grande vitesse  
(mode télécommandé)  
{  
    //analogWrite(D10, 255);  
    //analogWrite(D9, 0);  
    //  
    leftMotor->setSpeed(255);  
    leftMotor->run(FORWARD);  
    rightMotor->run(RELEASE);  
    //  
}
```

```

//-----
//-----



void Moteur_avancer2A(void)
//Fonction qui fait avancer le
char (mode asservi)
{
    //analogWrite(D9, 80);
    //analogWrite(D10, 80);
    //
    rightMotor->setSpeed(80);
    rightMotor->run(FORWARD);
    leftMotor->setSpeed(80);
    leftMotor->run(FORWARD);
    //
}

//-----
//-----



void Moteur_gauche2A(void)
//Fonction qui fait tourner le
char à gauche (mode asservi)
{
    //analogWrite(D9, 80);
    //analogWrite(D10, 0);
    //
    rightMotor->setSpeed(80);
    rightMotor->run(FORWARD);
    leftMotor->run(RELEASE);
    //
}

```

```

void Moteur_droit2A(void)
//Fonction qui fait tourner le
char à droite (mode asservi)
{
    //analogWrite(D10, 80);
    //analogWrite(D9, 0);
    //
    leftMotor->setSpeed(80);
    leftMotor->run(FORWARD);
    rightMotor->run(RELEASE);
    //

}

//-----
//-----



void Mode_telecommande1(void)
//Fonction mode télécommandé du
char
{
    //digitalWrite(D12, HIGH);
    //digitalWrite(D13, HIGH);
    Data_manage();
    //

Faire avancer le char
    if (data == 'A')
    {
        Moteur_avancer1();
    }
    else if (data == 'B')
    {
        Moteur_avancer2();
    }
    else if (data == 'C')
    {
        Moteur_avancer3();
    }
    //

Faire reculer le char

```

```

else if (data == 'R')
{
    Moteur_reculer1();
}
else if (data == 'S')
{
    Moteur_reculer2();
}
else if (data == 'T')
{
    Moteur_reculer3();
}
//-----



Tourner à droite le char
else if (data == 'D')
{
    Moteur_droit1();
}
else if (data == 'E')
{
    Moteur_droit2();
}
else if (data == 'F')
{
    Moteur_droit3();
}
//-----



Tourner à gauche le char
else if (data == 'G')
{
    Moteur_gauche1();
}
else if (data == 'H')
{
    Moteur_gauche2();
}
else if (data == 'I')
{
    Moteur_gauche3();
}

```

```

        }
        //-----
        else if (data == 'O')
        {
            GDO();
        }
        //-----
        else if (data == 'Z')
        {
            Desactivation_moteur();
        }
        else if (DR1A == HIGH &&
data == 'Y' && DO == 0)
        {
            Mode_asservi();
        }
    }

//-----
void Mode_telecommande2(void)
//Fonction mode télécommandé
(reculer tout droit reste
possible en cas d'obstacle)
{
    //digitalWrite(D12, HIGH);
    //digitalWrite(D13, HIGH);
    Data_manage();
    if (data == 'R')
    {
        Moteur_reculer1();
    }
    else if (data == 'S')
    {
        Moteur_reculer2();
    }
    else if (data == 'T')
    {

```

```

        Moteur_reculer3();
    }
    else if (data == 'O')
    {
        GDO();
    }
    else
    {
        Data_manage();
    }
}

//-----
void Mode_asservi(void)
//Fonction mode asservi du char
{
    //digitalWrite(D12, HIGH);
    //digitalWrite(D13, HIGH);
    Data_manage();
    DR9 = digitalRead(D9);
    DR10 = digitalRead(D10);
    Serial.println("Mode
asservi");
    Serial.println("DR9=" &
DR9);
    Serial.println("DR10=" &
DR10);
    if (DR1A == HIGH || data ==
'Y')
    {
        DetectObstacle();
        if (DR9 == LOW && DR10 ==
HIGH && DO == 0)
        {
            Moteur_droit2A();
        }
    }
}

//-----
/* FIN DES DIFFÉRENTES FONCTIONS
*/
//-----

void setup(void)
{
    Serial.begin(9600);
}

```

```

    else if (DR9 == HIGH &&
DR10 == LOW && DO == 0)
    {
        Moteur_gauche2A();
    }
    else if (DR9 == LOW && DR10 ==
LOW && DO == 0)
    {
        Moteur_avancer2A();
    }
    else if (DO == 1)
    {
        Desactivation_moteur();
    }
    else if (DR1A == HIGH || data ==
'Y' && DO == 1)
    {
        Desactivation_moteur();
    }
    else if (DR0A == HIGH && data ==
'Z' && DO == 0)
    {
        Mode_telecommande1();
    }
}

//-----
/* FIN DES DIFFÉRENTES FONCTIONS
*/
//-----
```

```
AFMS.begin();
pinMode(AO, INPUT);
pinMode(AI, INPUT);
pinMode(AII, INPUT);
pinMode(AIII, INPUT);
pinMode(D3, INPUT);
pinMode(D2, OUTPUT);
pinMode(D4, OUTPUT);
pinMode(D5, OUTPUT);
//pinMode(D6, OUTPUT);
pinMode(D8, OUTPUT);
pinMode(D9, INPUT);
pinMode(D10, INPUT);
//pinMode(D11, OUTPUT);
//pinMode(D12, OUTPUT);
//pinMode(D13, OUTPUT);
DR0A = digitalRead(AO);
DR1A = digitalRead(AI);
DR2A = digitalRead(AII);
DR3A = digitalRead(AIII);
DR9 = digitalRead(D9);
DR10 = digitalRead(D10);
}

//-----
-----
```

```
void loop(void)
{
    Data_manage();
    DetectObstacle();
    if (DR0A == HIGH && data != 'Y') //Tant qu'il n'y a pas d'obstacle
    {
```

```
        if (DO == 0)
        {
            Mode_telecommande1();
            DetectObstacle();
        }
        else if (DO == 1)
        {
            Mode_telecommande2();
            DetectObstacle();
        }
        else if (DR1A == HIGH && data == 'Y')
        {
            if (DO == 0)
            {
                Mode_asservi();
            }
            else if (DO == 1)
            {
                Desactivation_moteur();
            }
        }
    }
```

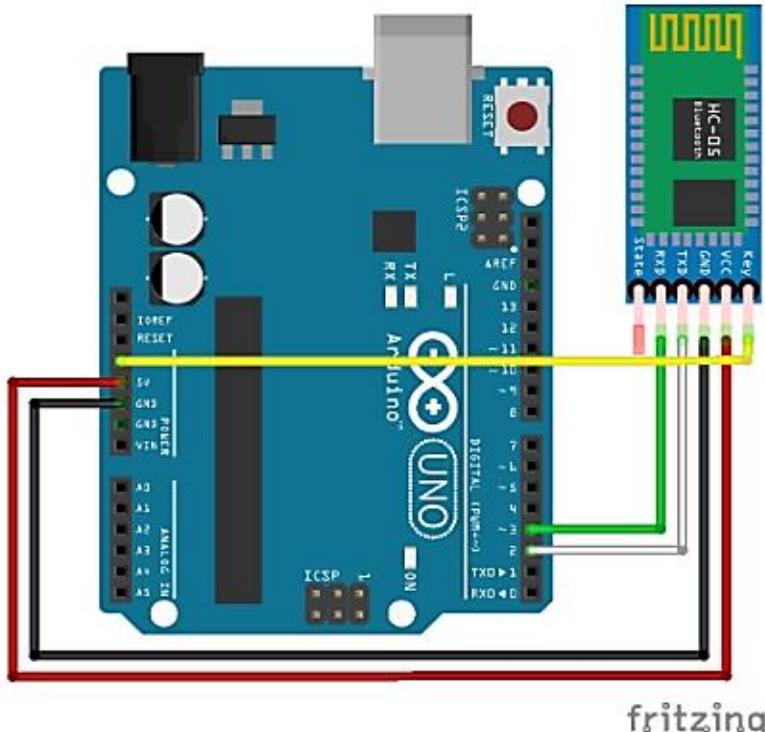


ANNEXE 16

Procédure de configuration des modules Bluetooth

Source pour l'annexe : <https://www.aranacorp.com/fr/votre-arduino-communique-avec-le-module-hc-05/>

Schéma de câblage pour la configuration



code à compiler et à téléverser dans l'Arduino
Avec le logiciel Arduino IDE

```
#include <SoftwareSerial.h>
#define rxPin 2
#define txPin 3
#define baudrate 38400

String msg;

SoftwareSerial hc05(rxPin ,txPin);

void setup(){
  pinMode(rxPin,INPUT);
  pinMode(txPin,OUTPUT);

  Serial.begin(9600);
  Serial.println("ENTER AT Commands:");
  hc05.begin(baudrate);
}

void loop(){
  readSerialPort();
  if(msg!="") hc05.println(msg);

  if (hc05.available()>0){
    Serial.write(hc05.read());
  }
}

void readSerialPort(){
  msg="";
  while (Serial.available()) {
    delay(10);
    if (Serial.available() >0) {
      char c = Serial.read(); //gets one byte from serial buffer
      msg += c; //makes the string readString
    }
  }
}
```

MAITRE (uniquement le modèle HC-05)

ESCLAVE (modèle HC-05 ou HC-06)

Lorsque le code est téléchargé, ouvrir le moniteur série puis entrez les commandes AT suivantes :

- | | |
|---|--|
| <ul style="list-style-type: none"> Pour tester la communication, tapez AT dans le monitor série de l'IDE Arduino. Si tout va bien, le module doit répondre OK. Pour changer le nom du module en MASTER_BT, tapez AT+NAME=MASTER_BT Pour modifier le code PIN du module par 1234, tapez AT+PSWD=1234 . Le module devrait répondre OK. Pour passer le module en maître saisir AT+ROLE=1 | <ul style="list-style-type: none"> Pour tester la communication, tapez AT dans le monitor série de l'IDE Arduino. Si tout va bien, le module doit répondre OK. Pour changer le nom du module en SLAVE_BT, tapez AT+NAME=SLAVE_BT Pour modifier le code PIN du module par 1234, tapez AT+PSWD=1234 . Le module devrait répondre OK. Pour passer le module en esclave saisir AT+ROLE=0 |
|---|--|

- | | |
|--|---|
| <ul style="list-style-type: none"> Pour vérifier la bonne prise en compte de la commande tapez : AT+ROLE ? le module doit répondre 1 Pour modifier la vitesse de communication du module (seulement si nécessaire), tapez AT+UART=<Param1>,<Param2>,<Param3> avec Param1, 2 et 3 les paramètres de communication série: le baud rate, le bit d'arrêt et le bit de parité respectivement. (ex : pour 9600 bauds entrez AT+UART=9600,0,0) Enregistrer l'adresse du module esclave pour que le module maître puisse s'appairer: AT+BIND=98d3,32,21450e (remplacez les deux points « : » par des virgules « , ») | <ul style="list-style-type: none"> Pour vérifier la bonne prise en compte de la commande tapez : AT+ROLE ? le module doit répondre 0 Pour modifier la vitesse de communication du module (seulement si nécessaire), tapez AT+UART=<Param1>,<Param2>,<Param3> avec Param1, 2 et 3 les paramètres de communication série: le baud rate, le bit d'arrêt et le bit de parité respectivement. (ex : pour 9600 bauds entrez AT+UART=9600,0,0) Entrez AT+ADDR pour obtenir l'adresse du module esclave (dans notre cas, +ADDR:98d3:32:21450e), et la noter quelque part |
|--|---|

1 - Pour rentrer en mode de configuration la broche « *Key* » doit être alimenté en 5V, ou si le module dispose d'un bouton poussoir de configuration appuyez dessus.

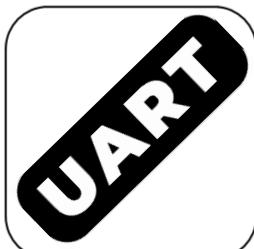
2 - Pour permettre aux deux modules de s'appairer, mettre le module hors tension et débrancher le conducteur de la broche « *Key* », puis remettre le module sous tension.

3 - Pour communiquer entre eux, les deux modules doivent avoir le même mot de passe, la même vitesse de communication (configurée à 9600 bauds par défaut), et il faut obligatoirement un module configuré en maître et l'autre module en esclave.



On utilise deux types de communication série bien distincts dans notre projet :

- 1- **I2C**, pour la communication des données gérant les moteurs par le Shield Moteur Adafruit
- 2- **UART**, pour la communication des données par Bluetooth entre l'Arduino du char et la l'Arduino de la manette

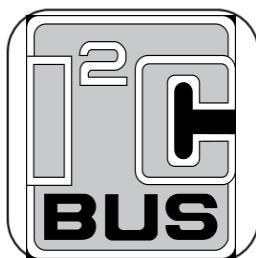


Source image :
<https://esphome.io/components/uart.html>

"Un **UART**, pour *Universal Asynchronous Receiver Transmitter*, est un émetteur-récepteur asynchrone universel.

Afin de faciliter l'interopérabilité entre périphériques (PC, microcontrôleur, modem...) des vitesses de transmission sont normalisées par multiples et sous-multiples de 9600 baud, l'unité baud correspondant à un symbole par seconde."

Source : <https://fr.wikipedia.org/wiki/UART>



"I²C (Inter-Integrated Circuit) est un bus série synchrone bidirectionnel, où plusieurs équipements, maîtres ou esclaves, peuvent être connectés au bus.

La connexion est réalisée par l'intermédiaire de deux lignes :
 SDA (Serial Data Line) : ligne de données bidirectionnelle,
 SCL (Serial Clock Line) : ligne d'horloge de synchronisation bidirectionnelle."

Source : <https://fr.wikipedia.org/wiki/I2C>

PORTS DE COMMUNICATION	
UART	I2C
Rx (Réception) et Tx (Transmission)	SDA (Serial Data) et SCL (Serial Clock)
PRINCIPE DE FONCTIONNEMENT	
A diagram showing two devices, Device-1 and Device-2. Device-1 has a TxD pin connected to Device-2's RxD pin, and Device-2's TxD pin connected back to Device-1's RxD pin, forming a simple point-to-point connection.	A diagram showing a master-slave configuration. Master-1 (top) and Master-2 (bottom) are connected to Slave-1 (top) and Slave-2 (bottom) respectively via SDA and SCL lines. Each master has its own SDA and SCL pins, while each slave shares a common SDA and SCL pair with other slaves.

Source: <https://www.rfwireless-world.com/Terminology/UART-vs-SPI-vs-I2C.html>

	UART	I2C
Caractéristiques	<ul style="list-style-type: none"> - Les bits sont envoyés en série un à la fois - Envoie un octet par caractère - Débit en bauds (9600, 19200, 38400, 57600....) en bits par seconde - Le nombre de bits d'arrêt (1 en général) : sert à aider le récepteur durant sa synchronisation temporelle avec le transmetteur - il est dit asynchrone, car il n'y a pas de signal d'horloge qui synchronise l'émetteur et le récepteur lors de la transmission du signal 	<ul style="list-style-type: none"> - Transmission synchrone grâce au signal d'horloge fourni lors de la communication - Signal d'horloge toujours transmis par le maître sur le bus - Envoie les bits du destinataire à tous les esclaves avant d'envoyer le bit du message. Le bon destinataire répond, et c'est à ce moment que le maître envoie les bits du message.

Source: <https://fr.quora.com/Quelle-est-la-difference-entre-l-I2C-et-le-UART#:~:text=Normalement%202%20conducteurs%20%C3%A9lectriques%20sont,donc%20consid%C3%A9r%C3%A9%20un%20protocole%20synchrone.>

TABLEAU DE REPORTING MENSUEL DU CHAR BI-MODE**FÉVRIER 2021**

Date	Point d'avancement – à faire (avant prochaine échéance)	Risque(s) / Opportunité(s)	Demande d'arbitrage (MOA)	
			Problème(s) rencontré(s)	Solution(s) envisagée(s)
05/02/2021 09/02/2021	- Cahier des charges réalisé - Passer la commande, début la conception mécanique Remise de la liste des composants à commander		Certaines mesures ont dû être prises de manière approximative sachant que celles-ci nécessitent la possession du char monté, donc lors de la conception de la carrosserie nous avons "pris large", ainsi deux choix s'offre à nous:	
11/02/2021	- Cahier des charges amélioré puis finalisé et remis à la MOA par mail		- Attendre l'arrivée du châssis du char, procéder au montage, puis modifier les mesures sur Catia, en fonction de mesures réelles.	
12/02/2021	- Fin de la conception mécanique des éléments suivants (fichiers sauvegardés en .pdf et .stl pour la future impression 3D) : bac à eau sécuritaire, boîtier de manette (parties haute et basse), carrosserie du char, languette de fixation modèle 1 (angle à 90°), et languette de fixation modèle 2 (sans angle)		- Effectuer l'impression 3D des pièces comme prévus et s'adapter	
Du 15/02/2021 à u 28/02/2021	- Début de la conception électrique/électronique du char puis de la manette (semaine 7 et 8)	- Prendre de l'avance sur le projet par rapport au plan établi des activités par semaine (cf. cahier des charges)		



TABLEAU DE REPORTING MENSUEL DU CHAR BI-MODE

MARS 2021

Date	Point d'avancement – à faire (avant prochaine échéance)	Risque(s) / Opportunité(s)	Demande d'arbitrage (MOA)	
			Problème(s) rencontré(s)	Solution(s) envisagée(s)
05/03/2021	- À faire : finaliser les conceptions mécaniques et récupérer les éléments de commande qui sont arrivés	- Le code en langage C n'étant pas encore implémenté dans chaque Arduino et donc n'ayant pas encore été testé le code risque de ne pas correctement fonctionner dès le départ ; des corrections et/ou des réglages du code seront à prévoir Nous souhaitons éviter la simulation du code C sur Tinkercad, car après nos premiers essais nous rencontrons des bugs, ainsi qu'une lenteur de simulation inacceptable, sûrement causée par le manque de puissance de calcul de notre navigateur web et/ou du serveur du site		
08/03/2021	- Le dessin 3D CAO de la carrosserie du char, étant trop volumineux pour la zone d'impression de l'imprimante 3D de l'IUT, on a donc décidé de modifier le dessin en le « coupant » en 3 parties (char divisé en deux + canon)			
15/03/2021	- Réalisation complète du code en langage C pour le mode télécommandé et le mode asservi du char avec CodeBlocks::ARDUINO			
12/04/2021	- Prochainement : jalon du lundi 12 avril 17h30 Livrable #2 : Point d'avancement finalisation des conceptions - début de réalisation (implémentation logicielle et matérielle)			



TABLEAU DE REPORTING MENSUEL DU CHAR BI-MODE

AVRIL 2021

Date	Point d'avancement - à faire (avant prochaine échéance)	Risque(s) / Opportunité(s)	Demande d'arbitrage (MOA)	
			Problème(s) rencontré(s)	Solution(s) envisagée(s)
10/04/2021 Lundi 12 avril 2021	Correction d'erreurs décelées dans le code du char Jalon 3 : livrable #2 Point d'avancement de finalisation des conceptions Fichiers de conception finalisés de mécanique (catiaV6) de la carrosserie du char et de la télécommande, et d'électronique (TinkerCAD ou Proteus) du câblage de la télécommande et du char	Opportunité de pouvoir poursuivre la réalisation et le câblage chez soi en important du matériel nous permettant d'avancer sachant que les délais se rapprochent	Les composants récupérés ne correspondent pas tout à fait au composant dont les valeurs sont inscrites sur les schémas électriques, mais on fera avec	On utilisera quand même ces composants car leur valeur n'impactera pas trop le fonctionnement du système (sauf les résistances de valeur de 2 MΩ)
16/04/2021	Récupération de composant électronique nous permettons de pouvoir débuter la réalisation et le câblage électrique			



TABLEAU DE REPORTING MENSUEL DU CHAR BI-MODE

MAI 2021

Date	Point d'avancement - à faire (avant prochaine échéance)	Risque(s) / Opportunité(s)	Demande d'arbitrage (MOA)	
			Problème(s) rencontré(s)	Solution(s) envisagée(s)
Du 01/05/2021 au 31/05/2021	Pendant notre temps libre et durant les heures de projet fixées dans l'emploi du temps, avancement sur le montage mécanique et sur le câblage électrique du char avec les composants électroniques disponibles et les pièces et composants reçus		La date de la soutenance approche, et on a toujours pas reçu certains composants indispensables pour le bon fonctionnement du char tel que l'émetteur et le récepteur Bluetooth, ainsi que le joystick	la liaison sans fil Bluetooth pourra être remplacé par une liaison série filaire (Rx Tx), en attendant de recevoir les deux modules Bluetooth, dans le cas échéant le joystick pourra être remplacé par deux potentiomètres et un bouton poussoir



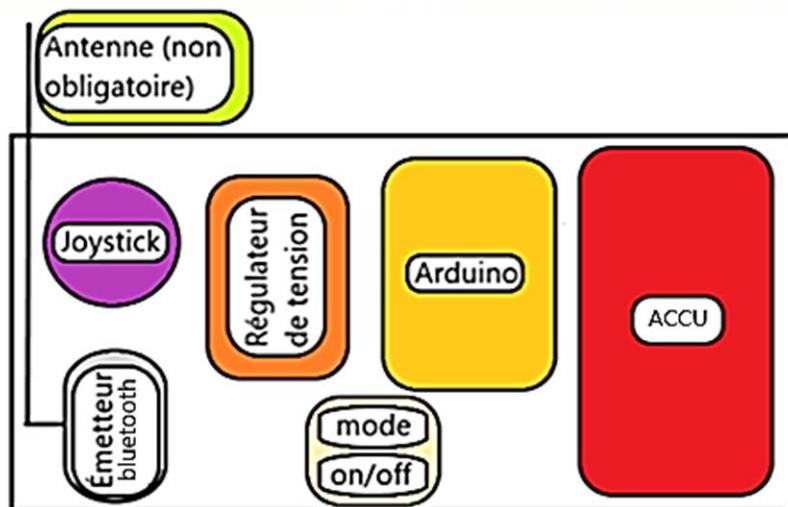
TABLEAU DE REPORTING MENSUEL DU CHAR BI-MODE

JUIN 2021

Date	Point d'avancement - à faire (avant prochaine échéance)	Risque(s) / Opportunité(s)	Demande d'arbitrage (MOA)	
			Problème(s) rencontré(s)	Solution(s) envisagée(s)
Du 01/06/2021 au 16/09/2021 Jeudi 17 juin 2021	<ul style="list-style-type: none"> -Avancement sur la rédaction du rapport de projet puis réalisation du PowerPoint associé, rédaction aussi des différents éléments requis pour la soutenance - Poursuite du montage et de la réalisation du char, puis test(s) de fonctionnement et calibrage(s) si nécessaire <p>Soutenance du projet : Passage à l'oral pour la soutenance du projet</p>			



MANETTE



CHAR

