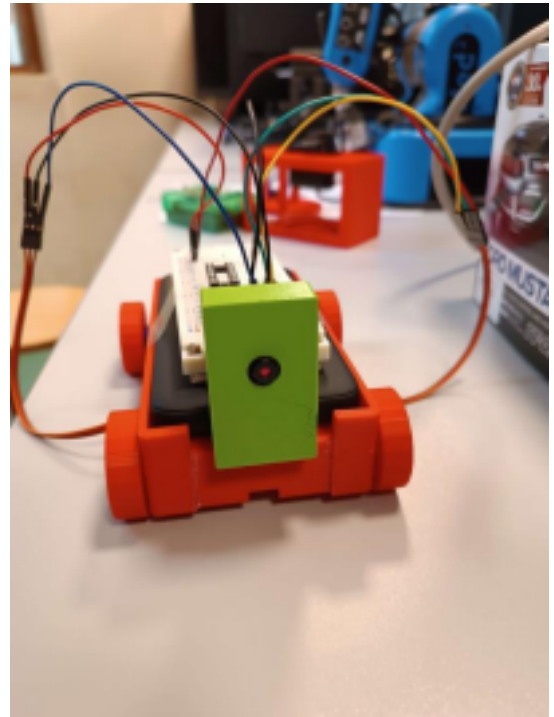
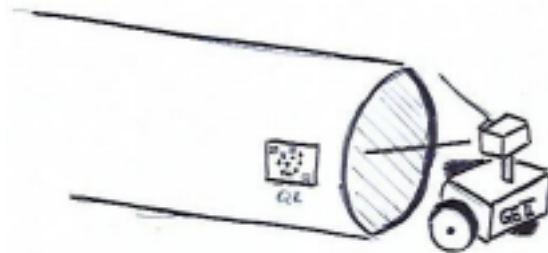


21 MARS 2023

Dossier de conception : Robot « Tortue Ninja »

BENSEBAH Kheireddine



KUZDOWICZ Théo

PROJET TUTE CONCEPTION D'UN
ROBOT TORTUE NINJA

PT

[Table des matières](#)

:	2 I) Cahier
des charges:	3 A)
Cahier des charges	3 II)
Planning:	4
B) Tâches et rôles (Gantt prévisionnel)	4 C)
Gantt réel	5 III)
Schéma fonctionnel :	6
A) Schéma fonctionnel	6 B)
Schéma électrique	7 IV)
Devis	8
A) Devis	8 B)
Justification choix batterie	10 C)
Justification choix moteurs	11 V)
Conception 3D	12
A) Architecture générale	12 B)
Architecture 3D	14 VI)
Test des composants	17 A)
Servomoteurs	17 B)
Capteur à ultrasons	19 C) La
caméra (ESP32-CAM)	21 VII)
Améliorations	21
	VIII)
Conclusion	22

Introduction :

Lors de ce projet tutoré (PT), nous avons pour objectif la conception et la mise en œuvre d'un robot « tortue ninja ». La conception de ce robot se fera en totalité de notre part à l'aide de capteurs, d'un système de communication, d'un microcontrôleur et de nos talents en impression 3D. Durant ce projet, nous sommes donc presque totalement libres, nous pourrons utiliser nos connaissances acquises lors de ce BUT GEII et notre imagination, mais en faisant en sorte de respecter le cahier des charges.

L'objectif principal de projet réside dans la conception intégrale d'un robot « ninja » dont la capacité principale (et mission) consiste à s'infiltrer dans des canalisations. Pour parvenir à nos objectifs, nous avons établi un cahier des charges, définit les étapes clés. Ce dossier de conception retrace et témoigne de notre parcours de réalisation de notre robot « tortue ninja ».

1) Cahier des charges :

Pour garantir une avancée méthodique et efficace de notre projet, nous avons mis en place un cahier des charges qui fixera les contraintes et objectifs clés sur le sujet. Ce document nous permet de définir les spécificités de notre robot, tout en tenant compte des limitations imposées. Grâce à cette démarche, nous serons en mesure d'établir des objectifs clairs pour notre planning.

A) Cahier des charges

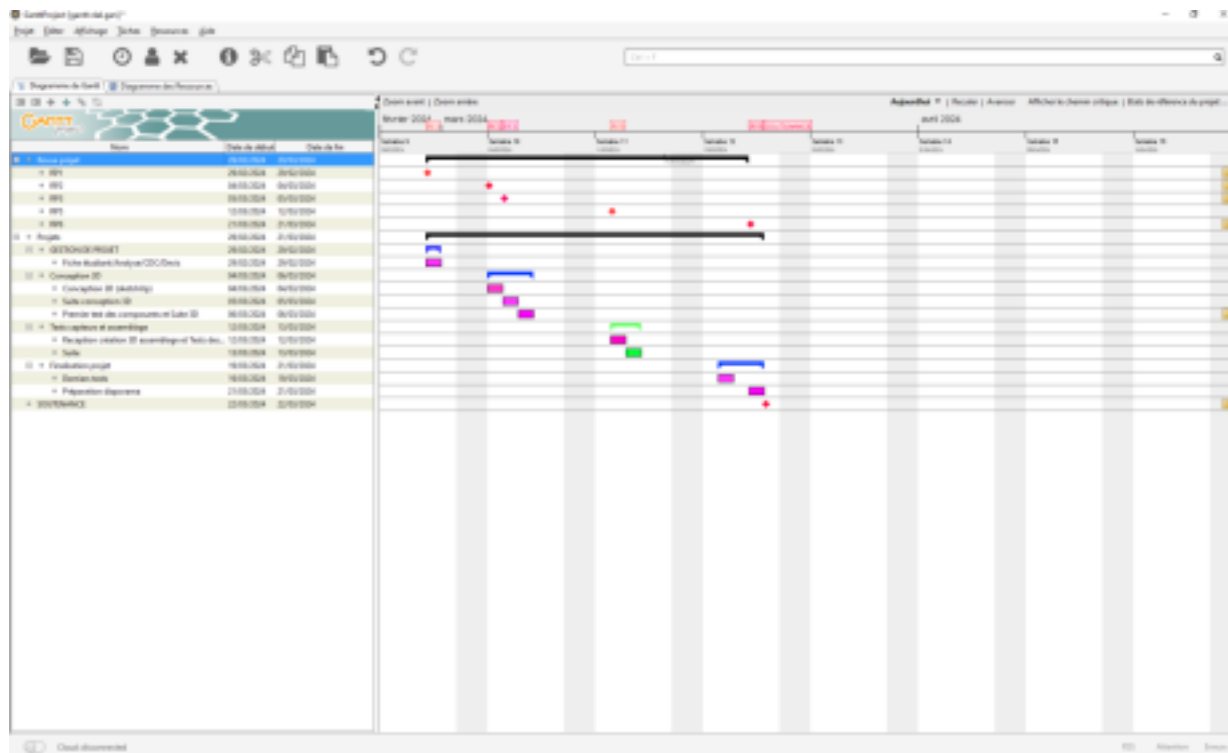
- On souhaite un petit véhicule qui mesure environ 15cm de longueur voir moins si possible
- Un robot de faible consommation d'énergie
- Une longue durée d'utilisation dans le cas d'une canalisation de plusieurs mètres
- Une caméra miniature de bonne qualité
- Une connexion sans fil wifi pour recevoir les images de la caméra
- Une vision nocturne via une LED
- Une détection de fin de parcours via un capteur ultrasons
- Un robot capable de se déplacer de manière autonome
- Deux moteurs

II) Planning :

Afin d'assurer une progression structurée tout au long de ce projet, nous avons élaboré un planning détaillé. Pour cela, nous avons défini les différentes tâches cruciales au bon déroulement de ce projet et aussi les participants et leurs rôles respectifs. Afin d'aboutir à un planning convenable, nous avons donc opté pour l'utilisation d'un outil adapté à la conception de ce genre de planning. C'est pourquoi nous avons choisi Gantt, outil que nous avons utilisé dans un projet passé et qui nous avait plutôt réussi !

B) Tâches et rôles (Gantt prévisionnel)

Pour commencer, nous avons donc mis en place un planning prévisionnel, prenant en considération notre emploi du temps respectif pour ce projet. Ce planning a été structuré de manière à intégrer les dates cruciales associées à nos petites présentations orales périodiques, permettant ainsi un suivi régulier de chaque amélioration et avancement de nos tâches.

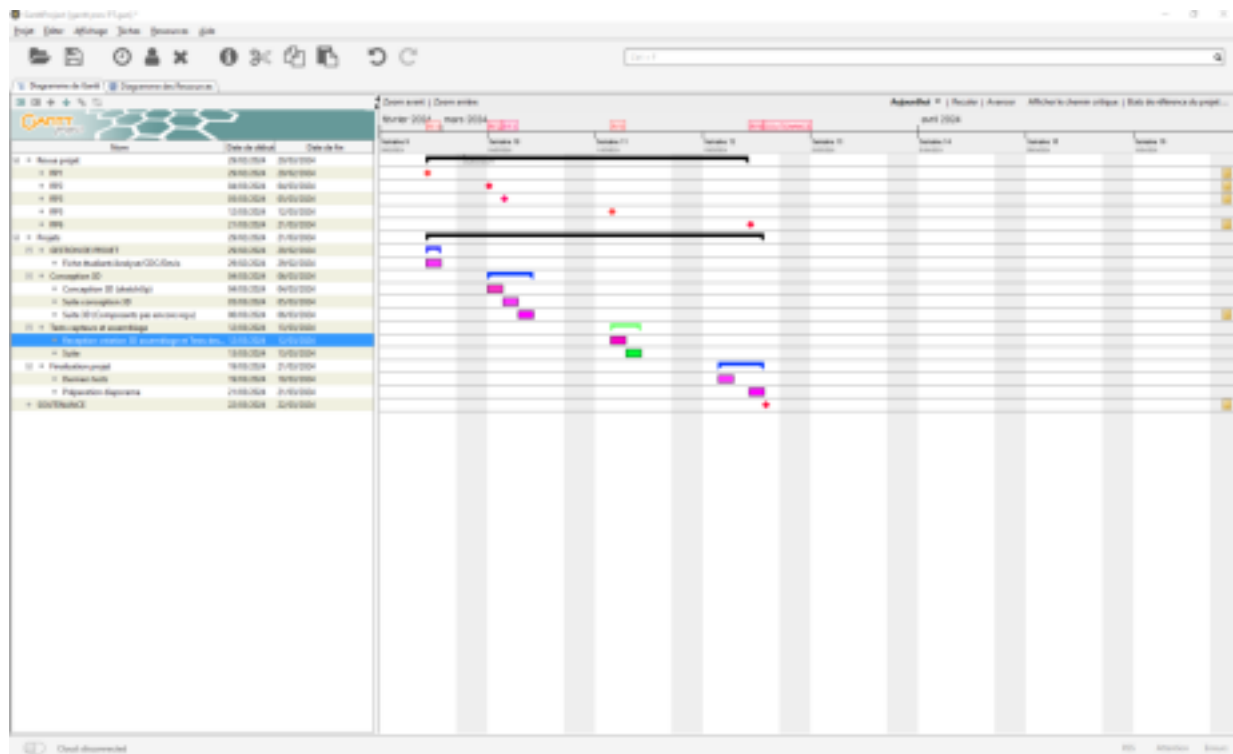


Gantt prévisionnel

C) Gantt réel

Suite à la mise en œuvre du projet, voici désormais le planning réel, mettant en lumière les différents écarts entre notre planning prévisionnel et ce qui a été réellement réalisé.

Les différents écarts notables sont notamment dus à notre conception 3D, car nous ne sommes pas encore assez familiers avec ces différents outils de conception 3D et aussi par rapport à la mise en place de la caméra ESP32-CAM qui nous a causé quelques soucis que nous détaillerons un peu plus tard.



Gantt réel après projet

Bensebah Kheireddine	Kuzdowicz Théo
Dimensionnement du robot et devis	
Tests composants	Modélisation 3D
Programmation fonctionnement du robot	Montage du véhicule

Test du véhicule
Amélioration à prévoir

III) Schéma fonctionnel :

Pour assurer une conception cohérente et fonctionnelle de notre système, nous avons établi un schéma fonctionnel détaillé, nous avons défini les différents composants et leurs interactions.

Nous avons donc pensé à un système composé d'une batterie d'alimentation dédié à l'optimisation et à la performance du circuit. Elle alimentera d'une part le microcontrôleur et de l'autre les servomoteurs de notre robot, assurant ainsi le fonctionnement de la caméra, élément centrale de notre projet, ainsi que le capteur à ultrasons.

En adoptant cette architecture, nous sommes en mesure d'optimiser le circuit et la place que prend notre batterie sur le robot.

A) Schéma fonctionnel

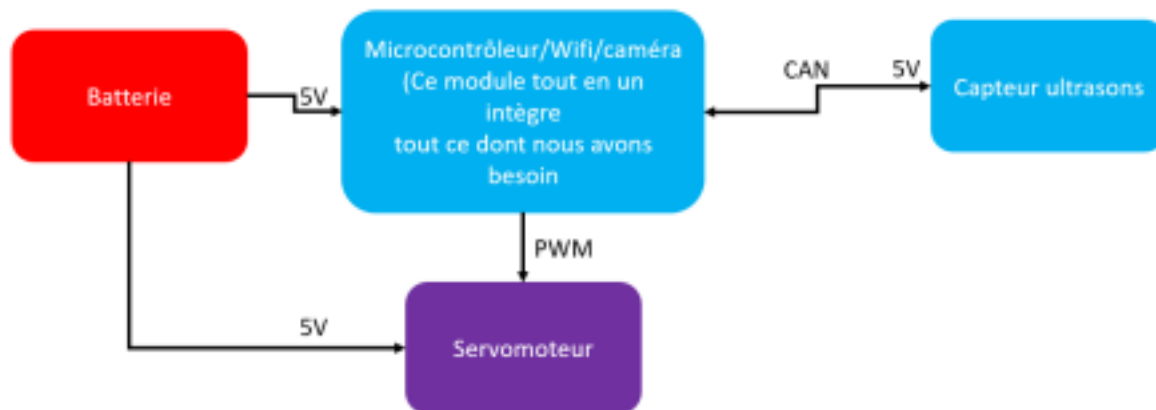


Schéma fonctionnel du robot

B) Schéma électrique

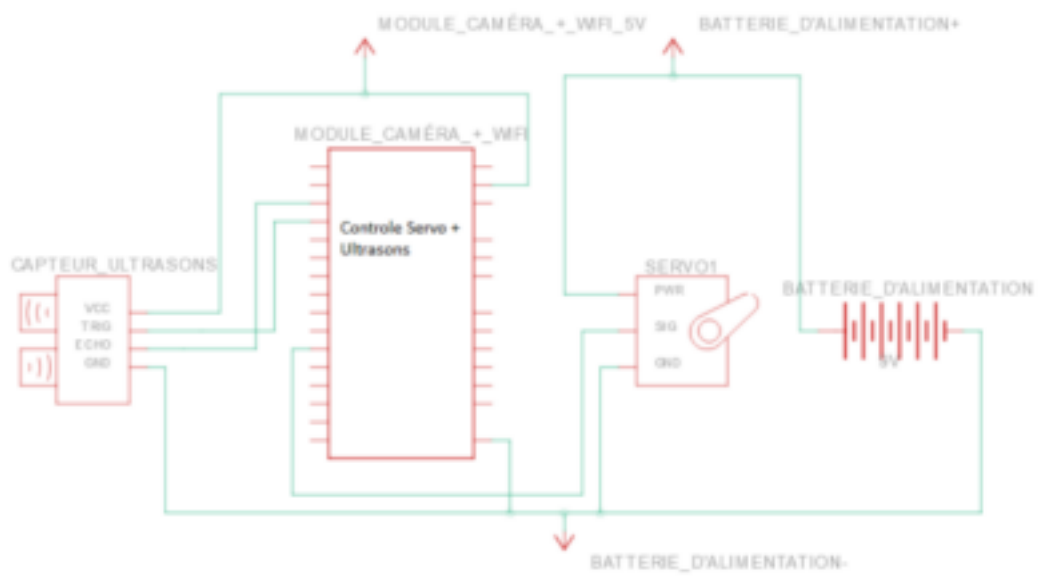



Schéma électrique du robot

IV) Devis

A) Devis

Composant	Référence	Prix	Liens
	Module ESP32CAM	14.30€	https://tinyurl.com/3m5rzsfsz

	Module de détection US HC-SR04	3.90€	https://tinyurl.com/4kdb37kr
	Servomoteur FS90R	5.90€	https://tinyurl.com/4krru2zd

	Câble USB 2.0 à 2 fils	8.49€ (6pcs)	https://tinyurl.com/5833scub
	22.5W 10500mAh Power Bank	22.78€	https://tinyurl.com/4cdytjzb

Pour justifier nos choix de composants, nous avons principalement pris en compte le cahier des charges. En ce qui concerne le capteur à ultrasons, nous avons choisi de prendre un module que nous avons l'habitude d'utiliser dans nos projets, car il fonctionne en 5V mais aussi en 3.3V selon les sorties que nous pourrions utiliser.

Le module esp32-cam, la pièce maîtresse de ce projet est un module tout-en-un. Il est composé d'une caméra, d'un système de communication wifi/Bluetooth, mais aussi d'un microcontrôleur doté de port d'entrées et sorties idéal pour le capteur à ultrasons et le contrôle des servomoteurs.

Pour terminer, en prenant compte du poids, de la puissance et le temps d'utilisation du robot, nous avons choisi notre moteur et nos batteries :

B) Justification choix batterie

Caractéristiques techniques des composants :

Pour cette première étape, on prendra donc les données nécessaires aux calculs de puissances fournis dans les documentations techniques des différents composants listés précédemment.

Microcontrôleur :

- 5V
- 2000mA
- 10g
- $10 \times 10^{-3} \text{ kg} \times 10 \text{ m/s}^2 = 0.1 \text{ N}$

Capteur à ultrasons :

- 5V
- 10mA

- $\frac{0.75}{10} \rightarrow 5 \times 0.015 = 0.075$

- 4.8V

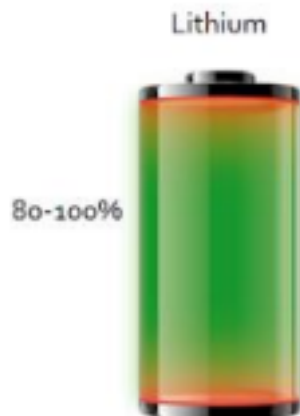
- 9g

- [illegible]

$(10 + 0.075 + 18) = 28.075$ Pour suivre ce résultat sera transformé pour une consommation voulue d'environ 1h d'utilisation du robot :

$$\diamond\diamond\diamond\diamond\diamond\diamond\diamond\diamond\diamond\diamond\diamond\diamond\diamond\diamond\diamond\diamond^{\diamond\diamond} = (10 + 0.0075 + 18) * 1 = 28.075 \diamond\diamond h$$

Cette donnée nous est utile afin de dimensionner la batterie car il faut que sa puissance fournisse soit équivalente à ce que demande notre système sinon le robot ne sera pas fonctionnel. Cette puissance nécessaire est calculée grâce à la profondeur de charge une mesure qui permet de rallonger la durée de vie d'une batterie (pour cela, on ajoutera donc 20% de charge supplémentaire pour garder notre batterie toujours entre 20 et 80% de charge qui était initialement notre 100% de charge :



$$\text{XX}h\text{XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX}=28.075$$

$$\frac{e^{\frac{1}{2} \pi i}}{h^{1/2}} = \frac{1}{\sqrt{2\pi h}}$$

Cette valeur sera donc la capacité minimale nécessaire au bon fonctionnement et à la bonne alimentation de notre système.

C) Justification choix moteurs

Par ailleurs, on souhaite un robot qui avance à 0.25m/s. (chiffre pris de manière arbitraire si on voulait un robot plus rapide on aurait pu aller au-dessus !)

$$\frac{1}{5} = 0.2$$

Ensuite la force motrice qui sert à choisir le toujours le type de moteur souhaité, ici on additionnera au préalable la masse et/ou une approximation comme celle du châssis qui n'est pas encore imprimé que l'on multipliera par l'accélération : $4000 = 10 \times 0.05 = (10 + 5 + 9 + 9 + 300) \times 0.05 = 0.01685$

Pour suivre ici nous avons la puissance motrice qui représente la quantité d'énergie que nous demandera nos moteurs.

$$\text{◆◆◆◆} = \text{◆◆◆◆} \times 0.25 = 0.0042125\text{◆◆} = 4.2125\text{◆◆◆◆}$$

$\Omega=0.25$

$$0.002 = 10 \text{ } \text{ } \text{ } \text{ } \text{ } \text{ } \cdot \text{ } \text{ }^{-1} = 100 \text{ } \text{ } \text{ } \text{ } / \text{ } \text{ } \text{ } \text{ } \text{ } \text{ } \text{ }$$

On a donc choisi notre servo moteur d'environ 110tr/min

V) Conception 3D

Lors des premiers jours de brainstorming et dimensionnement de notre robot, nous avons dû établir au préalable un premier plan « d'attaque » dans le but d'imaginer au mieux notre robot. Pour cela rien de mieux qu'un premier jet en terme de croquis/dessin afin de pouvoir représenter nos pensées.

A) Architecture générale

Le robot ninja sera composé en deux parties :

- Le châssis : Ce sera le corps principal de notre voiture. Il abritera la batterie, le microcontrôleur avec la caméra, un emplacement pour les servos moteur et enfin le capteur à ultrasons.
- Roues : Ils seront fixés au niveau du servomoteur pour permettre la mobilité dans les canalisations

Partie avant du robot
emplacement du capteur
à ultrasons
Tige de fixation des roues
« folles »

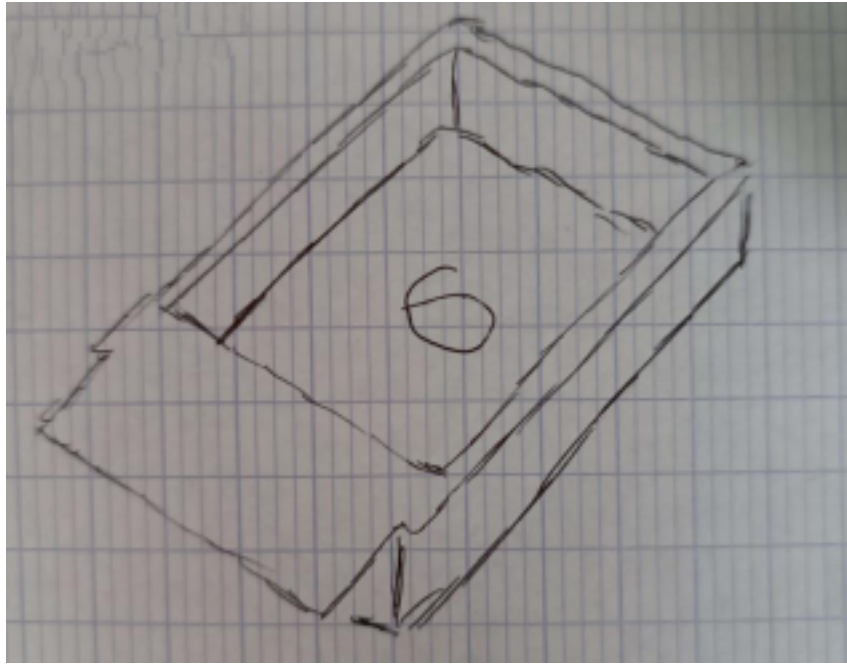
Trou afin de faire passer
les câbles

Concept de roue « folle »
pour le robot

Emplacement des
servomoteurs

Schématisation du robot (vue du dessous)

Pour suivre, en ce qui concerne la caméra la pièce maîtresse de notre conception, nous avons prévu un module qui se place à l'avant du robot qui permet le maintien à la verticale de notre esp32-cam afin d'obtenir les meilleures images possibles !



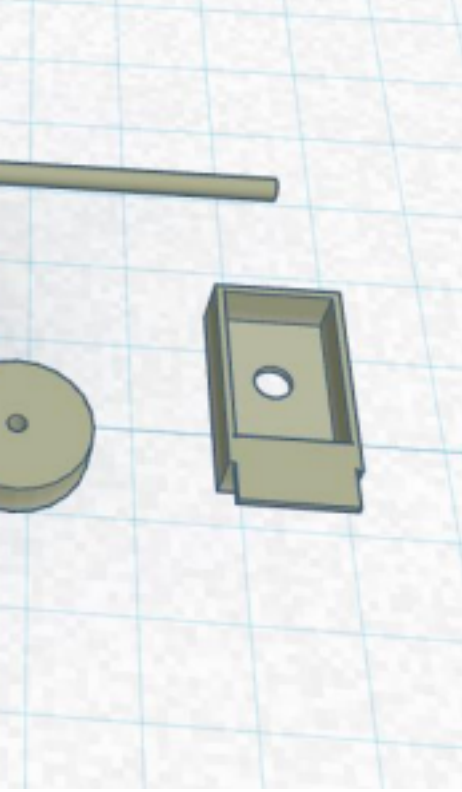
Socle de maintien de l'esp32-cam

De manière général afin d'avoir une petite idée de conception de notre robot en 3D, nous avons commencé par réfléchir à l'architecture et à l'emplacement de chaque composant.

Le châssis sera conçu de manière à optimiser l'espace disponible pour s'introduire dans des canalisations tout en maintenant une structure légère et compacte.

Pour cela, nous avons donc un emplacement qui prévoit la fixation via des vis des deux servomoteurs. A leurs extrémités, les roues y seront fixées. Ensuite nous avons les différents espaces permettant la facilitation du câble management qui permettra d'éviter au maximum de laisser les différents câbles trainer ! De surcroît, nous avons à l'avant de notre châssis l'emplacement prévu pour le capteur à ultrasons et juste au-dessus la place qui accueillera le compartiment de l'esp32-cam. Et pour finir la batterie qui se placera au-dessus du robot pour permettre une meilleure stabilité globale du système.

B) Architecture 3D



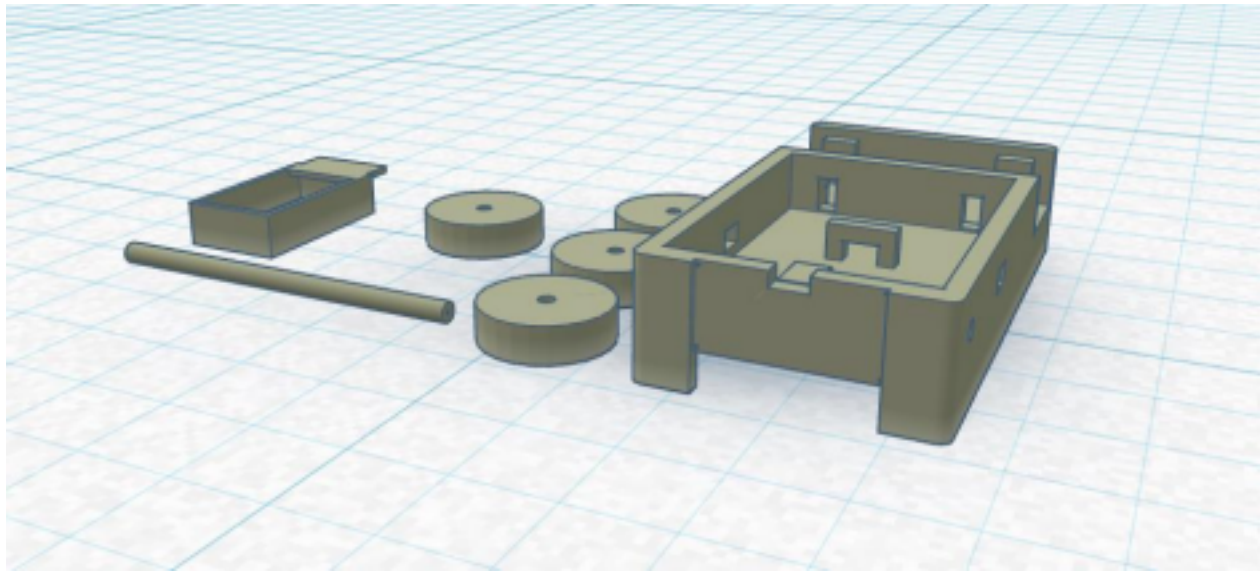
préliminaire, nous avons pu atteindre nos idées en une conception 3D. En
ncipale du châssis qui accueillera les différents composants.

Roue

Emplacement servomoteur

Socle esp32-cam

Design 3D du robot vue du dessous

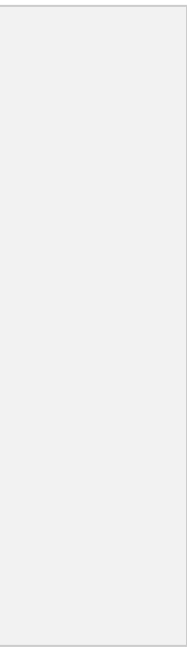


Vue de face (toujours en dessous)

Design 3D du robot vue du dessus à l'avant

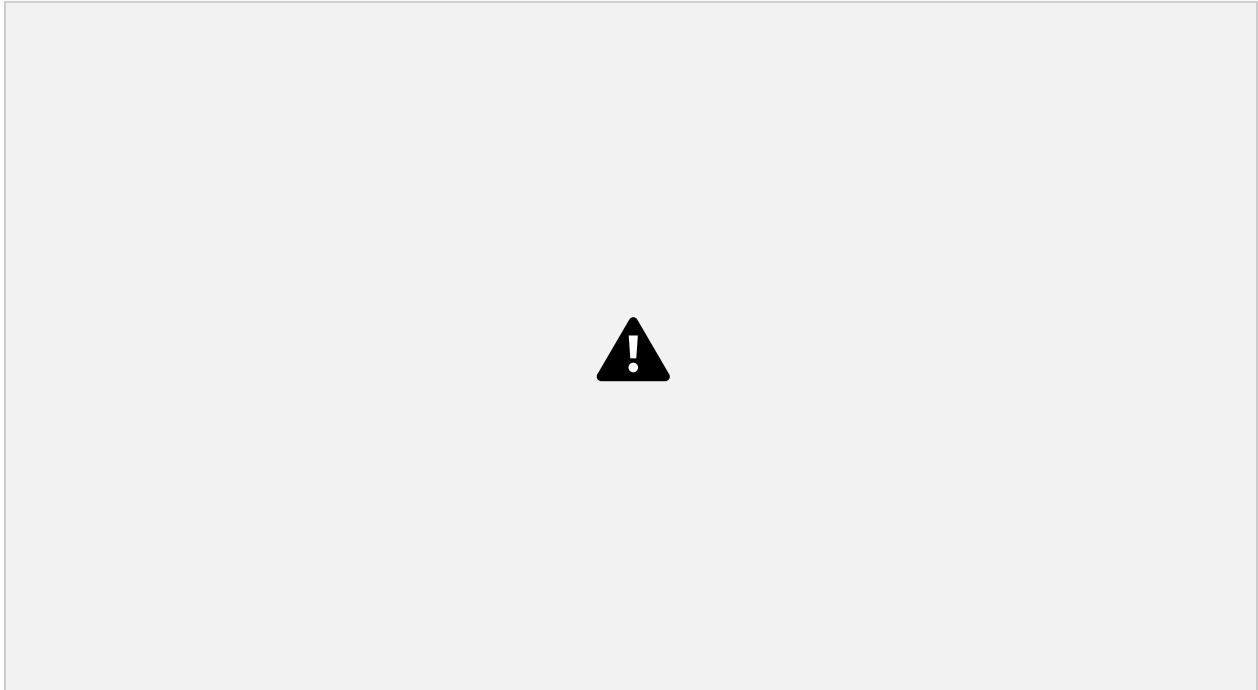
Emplacement capteur ultrasons





Vue avec
servomoteur placé

Vue arrière du socle pour l'esp et vue de côté avec emplacement servomoteur



Sans le servomoteur

VI) Test des composants

Après réception des composants nécessaires à notre projet, nous avons entrepris des tests approfondis pour assurer leur bon fonctionnement. Initialement, nous avons effectué des tests avec l'Arduino pour évaluer la compatibilité et la performance des composants. Ensuite, nous avons procédé à des tests spécifiques avec le microcontrôleur ESP32-CAM afin de confirmer son intégration et sa fonctionnalité dans notre système.

Nous allons donc retenir ici principalement le câblage et le code test directement sur l'esp32-cam avec les servomoteurs et le capteur à ultrasons.

A) Servomoteurs



Schéma de câblage pour les deux servomoteurs

Pour ce premier test, nous avons donc pris soin d'alimenter les deux servomoteurs via la batterie (aussi le microcontrôleur, mais afin de ne pas embrouiller les esprits il n'apparaît pas sur le schéma, car la batterie à deux sorties différentes !) et leur sortie de contrôle sur l'esp32-cam. Grâce au signal pwm

envoyé par celui-ci nos servomoteurs peuvent soit avancer dans un sens et dans l'autre à 360° soit s'arrêter complètement.

```
#include <ESP32_Servo.h>
```

```
Servo monServo1; // Créez une instance de la classe Servo pour le premier  
servomoteur
```

```
Servo monServo2; // Créez une instance de la classe Servo pour le deuxième  
servomoteur
```

```

void setup() {
    monServo1.attach(2); // Spécifiez le numéro de la broche GPIO à laquelle le
    premier servomoteur est connecté
    monServo2.attach(4); // Spécifiez le numéro de la broche GPIO à laquelle le
void loop( ) {
    deuxième servomoteur est connecté
}

    monServo1.write(0); // Avancer
    monServo1.write(90); // Avancer
    monServo2.write(0); // Avancer
    monServo2.write(90); // Arrêt
    delay(1000); // Attendez 1 seconde

    delay(1000); // Attendez 1 seconde
}

```

La bibliothèque ESP32Servo.h ici sert donc au contrôle des servomoteurs grâce au microcontrôleur ESP32. Ce code permet donc grâce à l'esp32 d'envoyer des signaux de commande spécifique à travers une plage de pulsation de largeur d'impulsion (PWM).

B) Capteur à ultrasons



Schéma de câblage du capteur à ultrasons

Même chose pour le capteur à ultrasons sans besoin de détailler sa manipulation avec l'Arduino Uno

qui est classique et identique au câblage réalisé sur l'esp32, nous sommes passés à son test qui s'avère aussi une réussite.

```
#define TRIGGER_PIN 1 2
#include <NewPing.h>

#define ECHO_PIN 14
#define MAX_DISTANCE 200 // Distance maximale à mesurer en centimètres

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);

void setup() {
void loop( ) {
    Serial.begin(115200);
}

    unsigned int distance = sonar.ping_cm( ); // Mesurez la distance en
    delay(500); // Attendez un court instant entre les mesures

centimètres

    Serial.print("Distance : ");
    Serial.print(distance);
    Serial.println(" cm");
}
}
```

Ici l'utilisation de la bibliothèque NewPing.h permet de faciliter l'utilisation du capteur à ultrasons. Celui ci envoie des impulsions sonores à haute fréquence et en mesurant le temps nécessaire pour que ces impulsions rebondissent sur un objet et reviennent au capteur.

C) La caméra (ESP32-CAM)

L'intégration de l'ESP32-CAM dans notre projet a été marquée par plusieurs défis, notamment en raison de problèmes rencontrés avec la caméra. Initialement, nous avons rencontré des difficultés lors de la détection de la caméra, malgré plusieurs tentatives et le suivi de divers tutoriels en ligne. Ces efforts n'ont pas abouti, et nous avons conclu que la caméra ne fonctionnait pas correctement. Après avoir contacté notre fournisseur, nous avons obtenu un nouveau capteur, ce qui a résolu le problème initial de détection de la caméra. Cependant, un nouveau problème est survenu lors de la dernière étape, où l'écran est resté noir lors de la visualisation des données renvoyées par la caméra. Ce problème est particulièrement préoccupant car la caméra constitue l'élément central de notre système. Pour améliorer la situation, nous envisageons d'explorer la possibilité d'utiliser un autre microcontrôleur du même type, doté d'une caméra intégrée, afin de garantir le bon fonctionnement et la fiabilité de notre système.

VII) Améliorations

Face aux défis rencontrés avec l'ESP32-CAM, l'envisagement d'un nouveau microcontrôleur intégrant une caméra offre des perspectives d'amélioration significatives pour notre projet. En adoptant cette approche, nous pourrions potentiellement résoudre les problèmes de compatibilité et de fonctionnement rencontrés avec la caméra externe. De plus, plusieurs améliorations matérielles sont envisageables pour renforcer les performances et la robustesse de notre robot. Tout d'abord, l'installation de roues dotées de caoutchouc pourrait prévenir les glissements et améliorer l'adhérence, garantissant ainsi une meilleure traction dans les environnements difficiles des canalisations. Parallèlement, l'intégration d'une carte micro SD offrirait la possibilité d'enregistrer les événements même en l'absence de connexion Wi-Fi, assurant ainsi une continuité de surveillance sans interruption de la communication. En outre, l'ajout d'une protection hydrophobe empêcherait les dommages potentiels causés par l'eau des canalisations, renforçant la durabilité du robot dans des conditions humides. Enfin, l'exploration de solutions de batterie plus compactes permettrait de réduire l'épaisseur globale du système, offrant ainsi une plus grande maniabilité et une meilleure adaptabilité aux espaces restreints des canalisations. Ces améliorations potentielles visent à optimiser les performances et la résilience de notre robot, garantissant ainsi son efficacité et sa fiabilité dans des environnements complexes et exigeants.

VIII) Conclusion

Fin de projet : Nous arrivons à la conclusion de notre projet tutoré. Notre objectif initial était de concevoir un robot autonome capable de s'infiltrer dans des canalisations, répondant à un cahier des charges précis. Celui-ci a été notre boussole (le cahier des charges), nous guidant dans nos choix et décisions, par ailleurs, la mise en place du planning nous a permis de suivre cet acheminement.

Cependant, notre parcours n'aura pas été sans embuche, notamment avec l'intégration de la caméra ESP32-CAM. Cette composante nous a causé des difficultés techniques. La nécessité de remplacer le capteur devient donc une des options les plus probables. Malgré cela, nous sommes en constante amélioration vis-à-vis du projet, il a permis d'explorer et d'approfondir nos connaissances en conception robotique et en électronique. Les améliorations envisagées offrent des perspectives prometteuses pour optimiser les performances de notre robot.

