



Université
de Lille



POLYTECH[®]
LILLE

Polytech Lille – Avenue Paul Langevin – 59655 Villeneuve d'Ascq cedex

Rapport Projet ingénieur

Robot Centaure

Travail réalisé par :
Mohammed Halaoui

Professeurs encadrants :
REDON Xavier
BOE Alexandre
VANTROYS Thomas

SOMMAIRE

1. Introduction	1
2. cahier de charge	2
3. Diagramme prévisionnel	3
4. Spécifications techniques.....	3
4.1. Partie Puissance.....	4
4.2. Partie Commande.....	9
5.Organisation de la partie Commande	13
5.1. Conception boîtier	14
5.2. Shield Arduino.....	16
6. Montage de la partie commande.....	17
7. Architecture de Commande.....	18
7.1. Commande par Joystick Physique.....	19
7.2. Commande Par Joystick virtuel	19
7.3. Gestion de priorité.....	19
8.Difficultés rencontrées.....	25
8.1.variateur de vitesse.....	26
9. Bilan	32
10. Comparaison entre le diagramme prévisionnel et réel.....	32
11.Développement Robot	33
11.1Hardware.....	34
A. Branchement.....	34
B. Condition de démarrage du variateur.....	34
11.2Software.....	34
12. Conclusion.....	35

INTRODUCTION



Le projet Centaure est un robot repris par différents élèves depuis 2005. Il se déplace sur trois roues et est équipé d'écrans et d'une caméra Kinect. Le robot avance à l'aide de deux moteurs 24 V continu (des moteurs équipant habituellement des fauteuils roulants) qui permettent de faire tourner les deux roues arrière du robot. La roue avant, elle, est une roue folle qui suit le mouvement donné par les roues arrière. Ce projet ayant été repris par de nombreux élèves, il s'appuie donc sur une certaine base existante et le châssis est ainsi déjà monté. Le robot Centaure est un robot d'accueil, il interagit avec les personnes et propose des services au sein de Polytech.

Rapport Projet ingénieur

cahier de charge

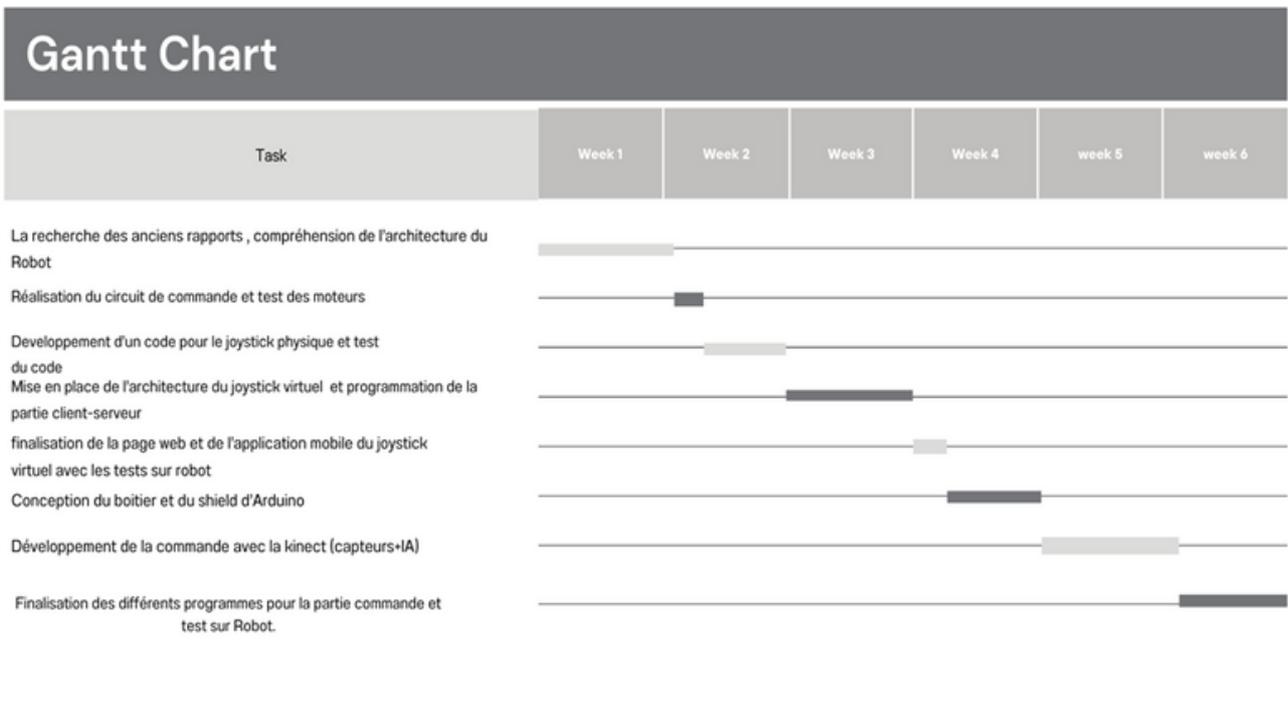
L'objectif général du projet c'est de développer la partie commande du robot.

Je dois réaliser plusieurs tâches. Tout d'abord :

- Suite à l'état des lieux, une mission principale se dessine, développer l'organisation du robot.
- Commande du robot à travers Joystick physique et virtuel

Une fois ces tâches effectuées on peut envisager un contrôle de plus haut niveau du robot par caméra en utilisant le capteur infrarouge et l'IA.

Diagramme prévisionnel



Rapport Projet ingénieur

Spécifications techniques :

Le robot est composé initialement de :

Partie Puissance :

-Deux moteurs DC:

-moteurs SRG0131 24V / 15.5 A / 0.35 kW / 10 km/h

Ces moteurs ont un courant de pic d'approximativement 30-35A. En régime normal, leur consommation est d'environ 8A.

-Un convertisseur DC/DC :



Tension d'entrée : 16,8...31,2V (Recommandée : 24V)

Tension de sortie : 12 VDC

Courant de sortie : 12,5A

Rapport Projet ingénieur

-Un convertisseur DC/DC:



Tension d'entrée : 19...36V (Recommandée : 24V)

Tension de sortie : 12 VDC

Courant de sortie : 12,5A

-Un contacteur:



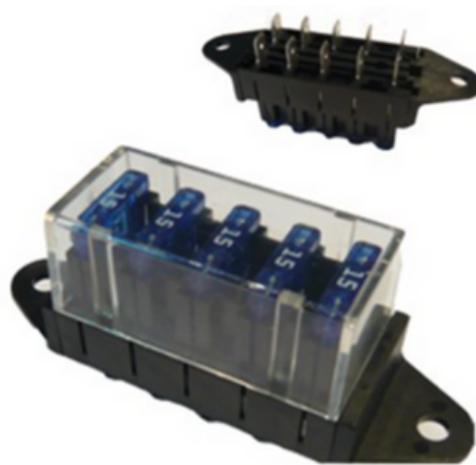
Les contacteurs sont des équipements de sécurité majeurs. Leur fonctionnement est

toujours le même : un circuit de commande en courant faible (<2A) permet, grâce à une

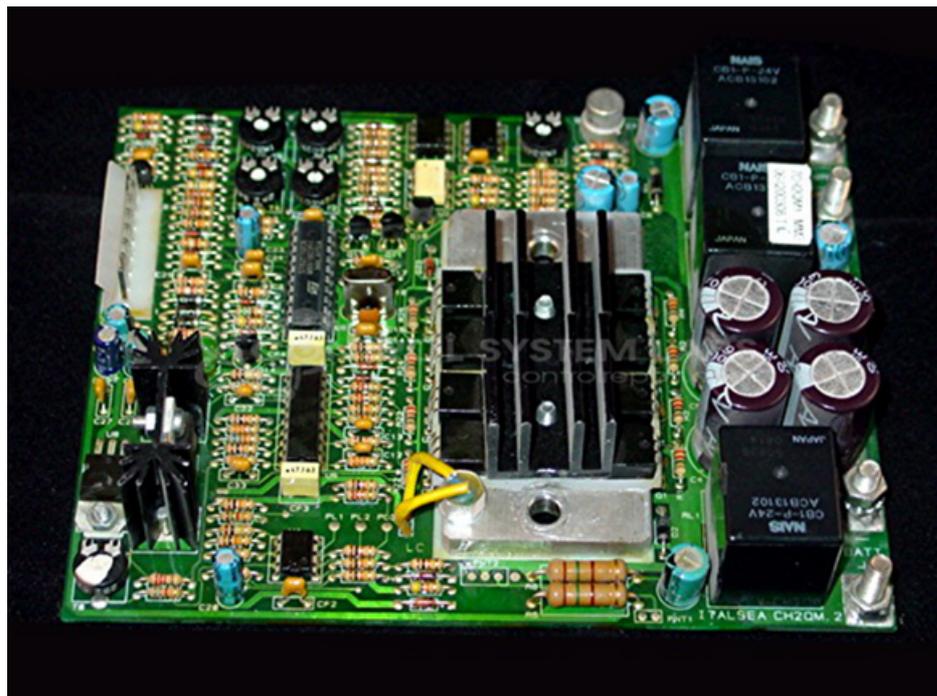
bobine, de fermer les contacts (équivalents à des interrupteurs) du circuit de courant fort (>2A).

Rapport Projet ingénieur

Porte 5 fusibles :



-Deux variateurs de vitesse 8CH2QM :

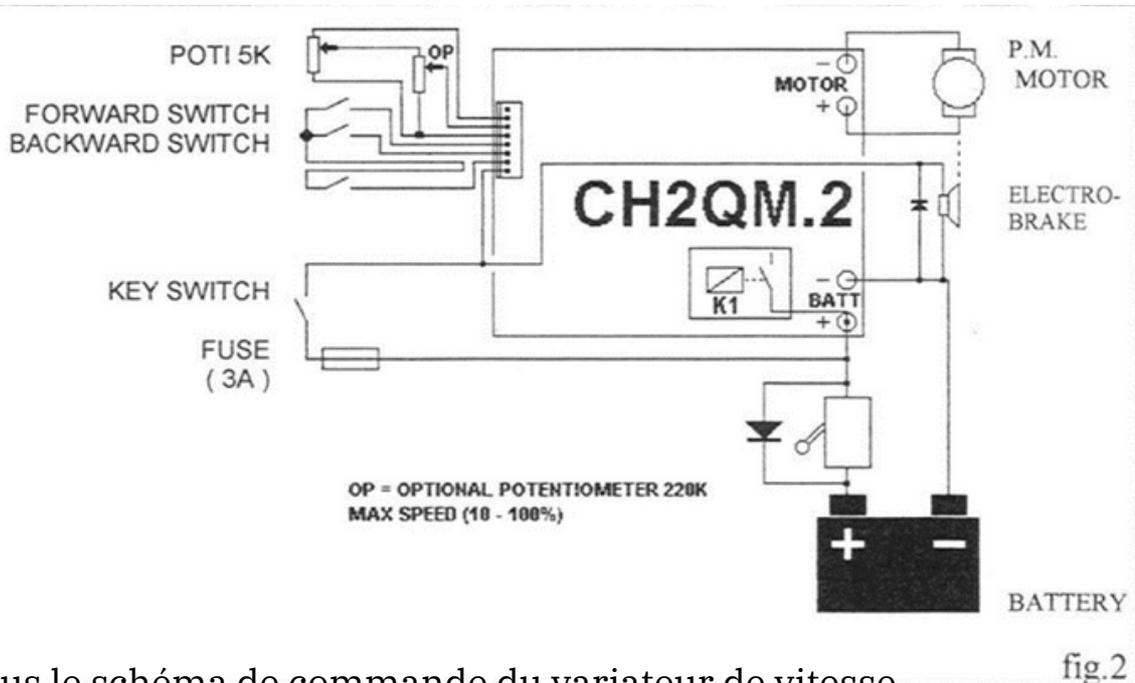


Ces variateurs servent à contrôler la vitesse et le sens de rotation des deux moteurs.

Rapport Projet ingénieur

Ce tableau résume les entrées et les sorties de ce type de variateur qu'il faut respecter pour garantir le bon fonctionnement de ce dernier.

PIN	FUNCTION	DESCRIPTION
1	KEY	KEY-SWITCH INPUT +V BATT
2	COMMON HIGH	OUTPUT COMMON FOR SWITCHES
3	FORWARD SWITCH	INPUT ACTIVE HIGH (+V BATT/PIN2)
4	BACKWARD SWITCH	INPUT ACTIVE HIGH (+V BATT/PIN2)
5	HI -POT. INPUT	POTENTIOMETER SUPPLY
6	C-POT.INPUT	CENTRAL POTENTIOMETER
7	LO-POT.INPUT	REFERENCE GND POTENTIOMETER
8	NOT CONNECTED	



Ci-dessus le schéma de commande du variateur de vitesse.

fig.2

-Deux batteries 12V 45Ah.

On trouve aussi une carte mère avec une mémoire SSD , un écran et une caméra Kinect pour le traitement d'image.

Un châssis à 2 étages :

- premier étage pour la partie puissance
- deuxième étage pour la partie commande.

Rapport Projet ingénieur

Ce tableau résume les entrées et les sorties de ce type de variateur qu'il faut respecter pour garantir le bon fonctionnement de ce dernier.



Partie Commande

La partie qui nous intéresse le plus dans ce projet.

IDE Arduino :

Afin de développer la partie commande j'utilise l'IDE d'Arduino pour la programmation des différentes fonctionnalités.



```

    #include <Arduino.h>
    #include <Servo.h>

    // Servo motor pins
    const int servoPin = 9;
    const int servo2Pin = 10;

    // Motor pins
    const int motorPin1 = 2;
    const int motorPin2 = 3;

    // Servo objects
    Servo servo(servoPin);
    Servo servo2(servo2Pin);

    // Motor speed variables
    int speed1 = 0;
    int speed2 = 0;

    void setup() {
      // Set servo pins as outputs
      pinMode(servoPin, OUTPUT);
      pinMode(servo2Pin, OUTPUT);

      // Attach servos to pins
      servo.attach(servoPin);
      servo2.attach(servo2Pin);

      // Set motor pins as outputs
      pinMode(motorPin1, OUTPUT);
      pinMode(motorPin2, OUTPUT);
    }

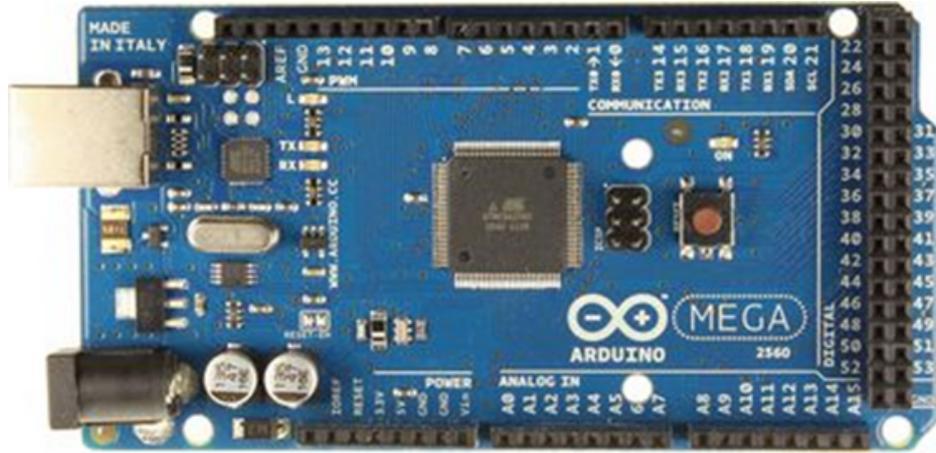
    void loop() {
      // Activate motor 1
      digitalWrite(motorPin1, HIGH);
      // Set speed 1
      analogWrite(motorPin1, 40);
      //delay(1000);
      // Activate motor 2
      digitalWrite(motorPin2, HIGH);
      // Set speed 2
      analogWrite(motorPin2, 40);
      //delay(1000);
    }
  
```

Le compilateur utilise 3474 octets (2%) de l'espace de stockage de programme, le maximum est de 25392 octets.
 Les variables globales utilisent 416 octets (1%) de mémoire dynamique, ce qui laisse 7776 octets pour les variables locales.
 Compiler les bibliothèques pour votre Arduino...

L'environnement de développement intégré Arduino est un environnement facile à utiliser et en même temps contient plusieurs bibliothèques déjà installées .

Rapport Projet ingénieur

- Arduino MEGA :



Arduino est un excellent outil pour développer des objets interactifs. Il existe de nombreuses bibliothèques permettant de faire des choses complexes, comme écrire sur des cartes SD, des écrans LCD, analyser des GPS, ainsi que des bibliothèques pour faire des choses simples, comme des boutons ou allumer une lumière.

L'Arduino Mega offre plus de broches d'E/S numériques (54 contre 14) et analogiques (16 contre 6), ainsi qu'une mémoire Flash plus grande (256 Ko contre 32 Ko) par rapport à l'Arduino Uno.

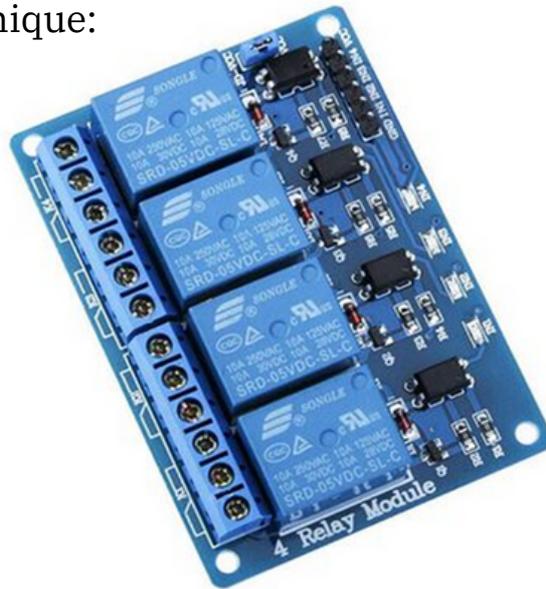
-Shield Arduino:

Carte d'interface permettant de raccorder facilement les modules capteurs et les actionneurs G à une carte Arduino. Développé par les anciennes promotions équipé par deux ADCs pour la commande des variateurs de vitesse .

-Relais électromécanique:

Rapport Projet ingénieur

-Relais électromécanique:



L'utilisation d'un module relais avec Arduino offre de nombreux avantages. Tout d'abord, il permet à Arduino de contrôler des appareils qui nécessitent une tension ou un courant plus élevé que ce que le microcontrôleur peut fournir. Dans notre cas, c'est le contrôle du variateur.

Joystick Physique :

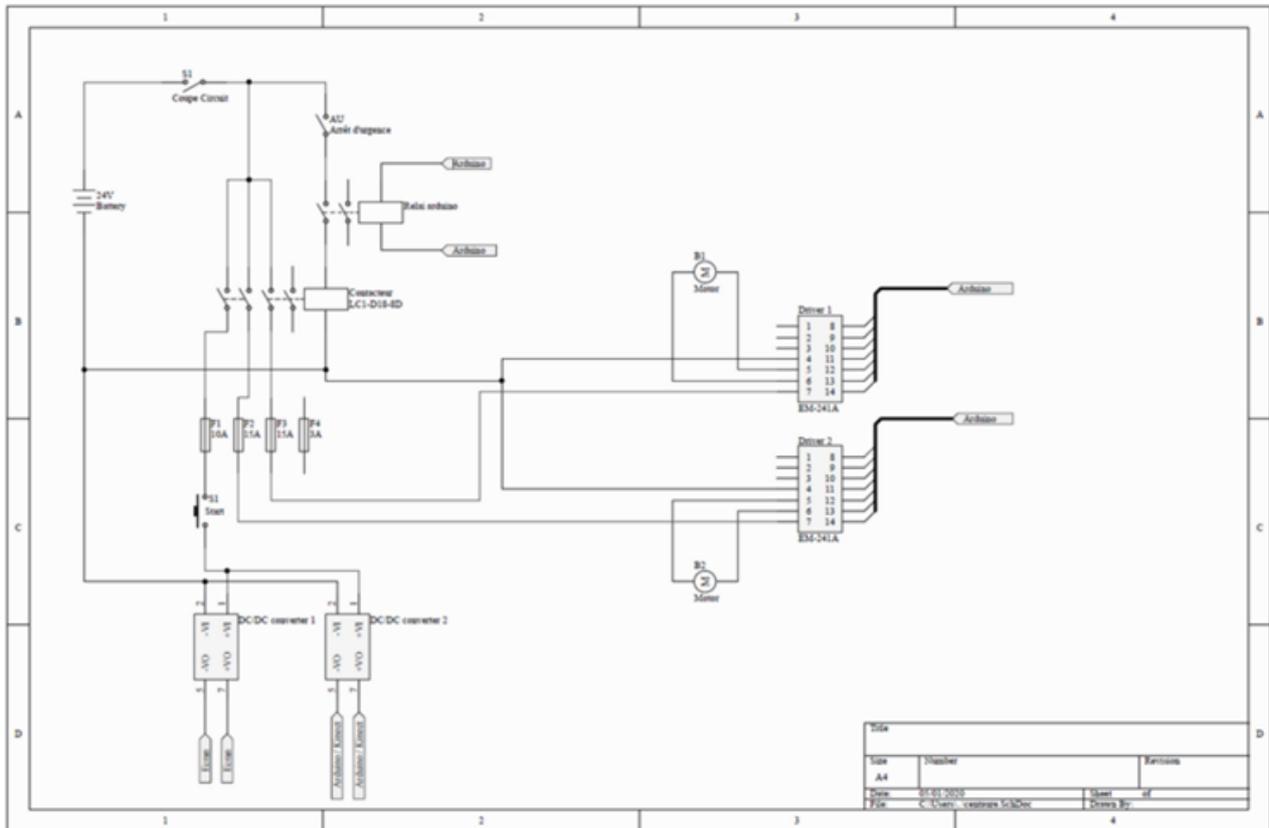
Ce Joystick a été fabriqué par une ancienne promotion dans le but de commander ce Robot.



Rapport Projet ingénieur

Ce joystick est équipé de 2 analogues un pour l'axe des X et l'autre pour l'axe des Y et avec un port RJ11 pour envoyer les coordonnées à l'Arduino à travers les pins analogiques.

Le branchement de la partie commande et la partie puissance forme notre Robot.



Ce schéma est réalisé par une ancienne promotion qui explique le schéma général du Robot.

Langages de programmation :

Le joystick virtuel nécessite l'utilisation de plusieurs langages de Programmation soit pour la Backend ou le Frontend.

Le joystick virtuel est sous forme de page Web et Application mobile

Frontend : Pour cette partie j'utilise :

HTML: Organiser le contenu en un ensemble de paragraphes, une liste d'éléments, utiliser des images ou des tableaux de données.

HTML



CSS: utilisé pour la mise en forme d'une page web codée en HTML. Il s'agit ainsi d'un langage descriptif qui se complète aux langages de balisage (HTML, XML).

CSS



Rapport Projet ingénieur

JavaScript: permet de créer du contenu mis à jour de façon dynamique, de contrôler le contenu multimédia, d'animer des images, et tout ce à quoi on peut penser.



Ces trois langages de programmations sont complémentaires dans le but de créer la partie Frontend du Joystick virtuel. Ils nous permettent d'avoir une page web (application mobile) organisée, personnalisée et interactive.

Backend :

Pour le Backend j'utilise :

NodeJs: une plateforme logicielle libre en JavaScript, orientée vers les applications réseau événementielles hautement concurrentes. Le choix d'utilisation de NodeJS est basé sur le besoin d'avoir un serveur temps réel.

Rapport Projet ingénieur



Application Mobile :

Pour la réalisation de l'application mobile , j'utilise Cordova.

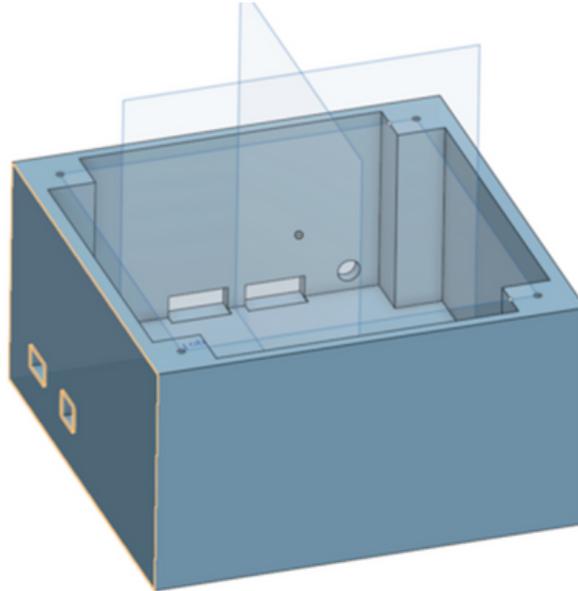


Apache Cordova ou plus anciennement Apache Callback ou PhoneGap², est un framework open-source développé par la Fondation Apache. Il permet de créer des applications pour différentes plateformes (Android, Firefox OS, iOS, Ubuntu, Windows 8...) en HTML, CSS et JavaScript. Le choix d'utilisation de Cordova est basé sur le fait que la page Web est codée en HTML,CSS,JavaScript au lieu de refaire la programmation from Scratch de l'application mobile ,j'ai utilisé Cordova qui permet de faire le lien directement

Organisation de la partie Commande :

Conception boîtier

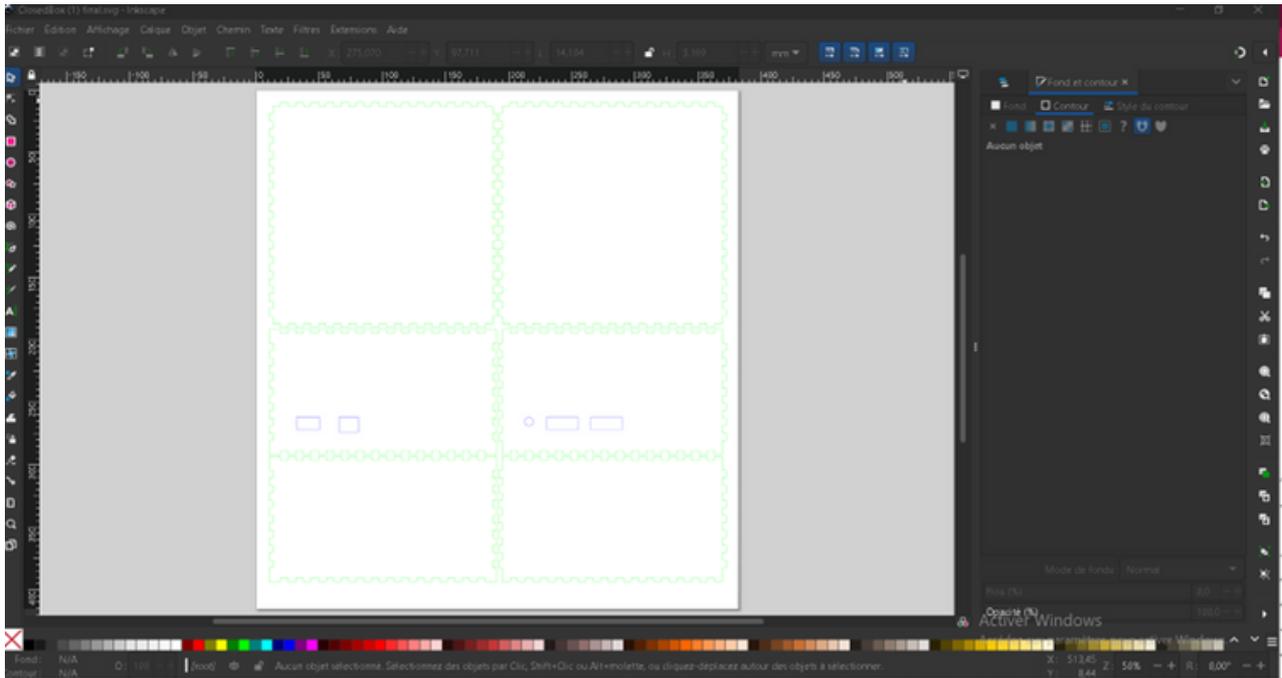
Conception d'un boîtier avec le logiciel Onshape pour organiser la partie Commande du Robot:



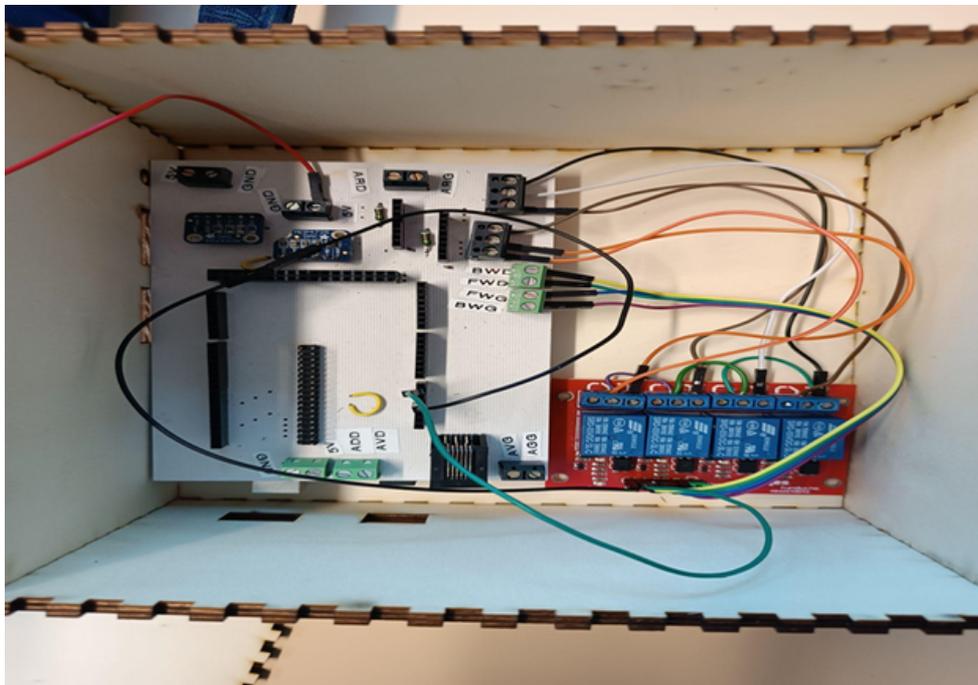
Ce boîtier a été conçu en se basant sur les mesures effectuées sur le Robot et les composants qui vont être mis dans le boîtier. Ces composants sont l'arduino avec le shield et le relais électromécanique. Dans ce boîtier, j'ai pris en considération les différents câbles qui seront connectés à l'arduino et au shield (alimentation, RJ11(joystick physique)...).

Réalisation : En premier lieu, j'ai essayé d'utiliser l'imprimante 3D mais malheureusement l'imprimante est tombée en panne au cours de l'impression et donc j'ai switché au découpage par laser. Pour utiliser la découpe laser, il faut utiliser le logiciel Inkscape.

Rapport Projet ingénieur

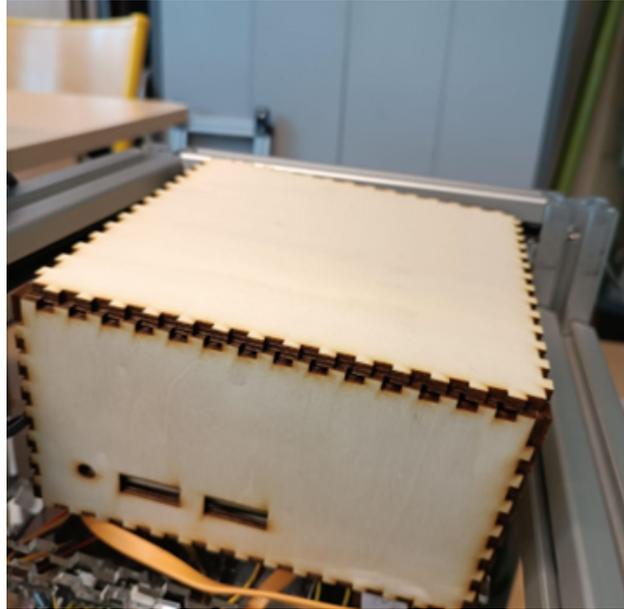


Résultat final :
l'intérieur:



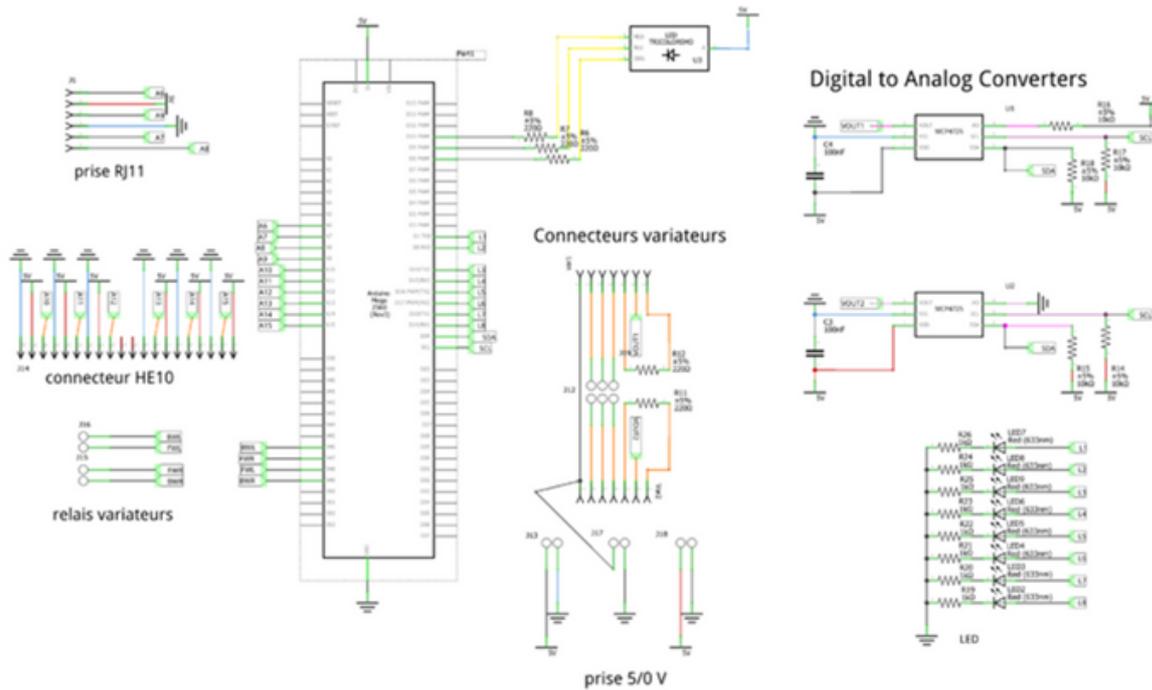
Rapport Projet ingénieur

l'extérieur:



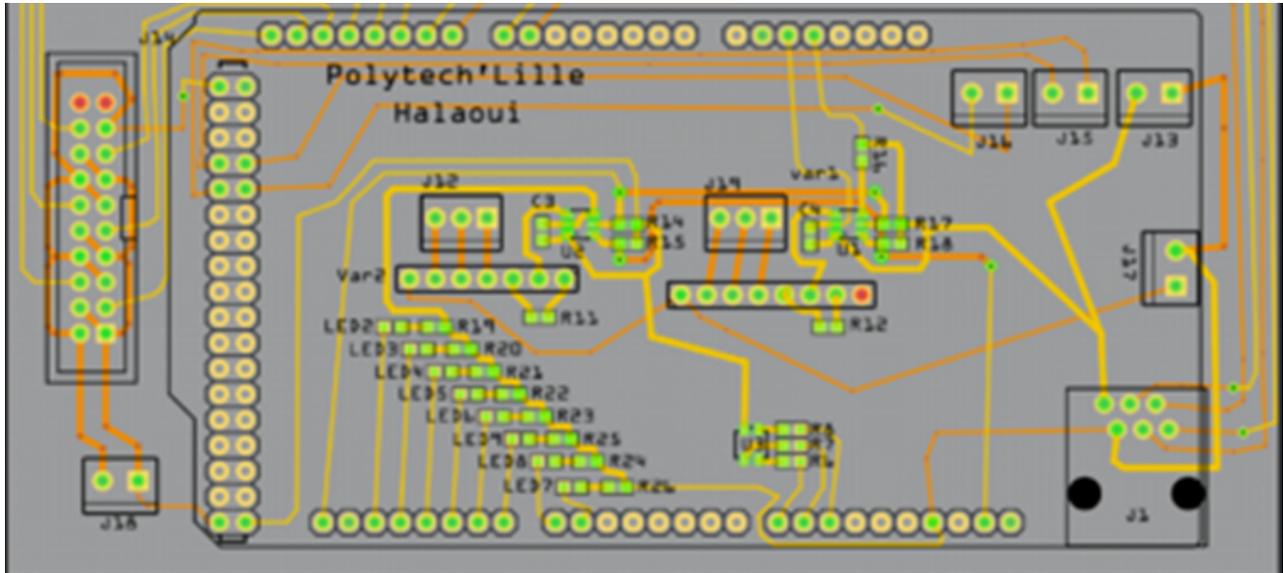
Shield Arduino :

Le shield réalisé par l'ancienne promotion n'est pas fonctionnelle. Je me suis basé sur leur schématique pour corriger les fautes. J'ai utilisé le logiciel Fritzing pour la conception et le routage.



Rapport Projet ingénieur

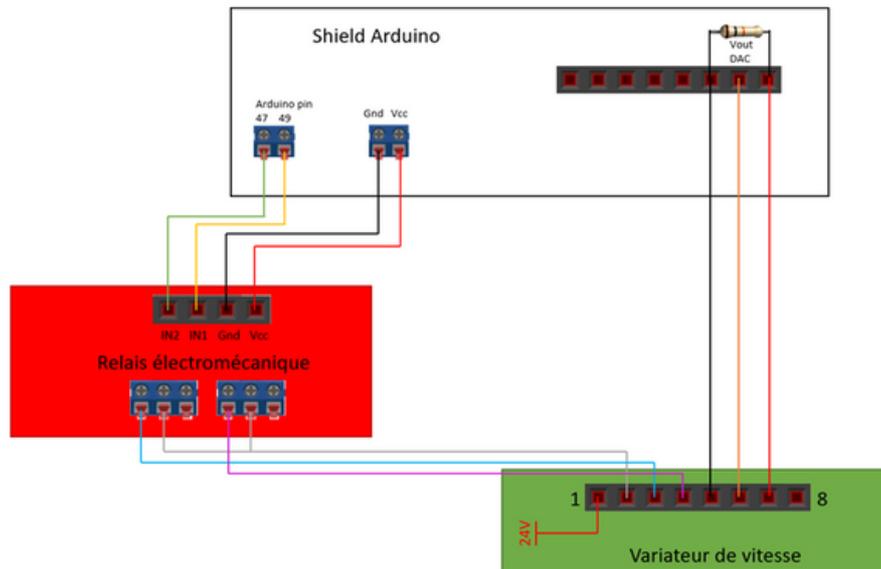
Routage :



Le shield a été conçu principalement pour avoir deux convertisseurs analogique numérique analogique pour la commande du variateur de vitesse. Ce shield est aussi équipé par un connecteur HE10 pour ajouter des capteurs et un port RJ11 pour la commande du joystick physique.

Le DAC joue le rôle du potentiomètre (voir la schématique de commande du variateur) dans le but de varier la vitesse du moteur.

Montage de la partie commande :



Ce montage contient trois composants :

-Shield Arduino qui reçoit des commandes externes soit du Joystick physique , soit du joystick virtuel.Ce shield (Arduino) commande à son tour le relais et le variateur à travers son DAC qui joue le rôle du potentiomètre.

Le relais : Comme mentionné dans la partie spécification , il permet d'avoir une tension et un courant plus important pour la commande du variateur .

Ce relais commande trois I/O du variateur :

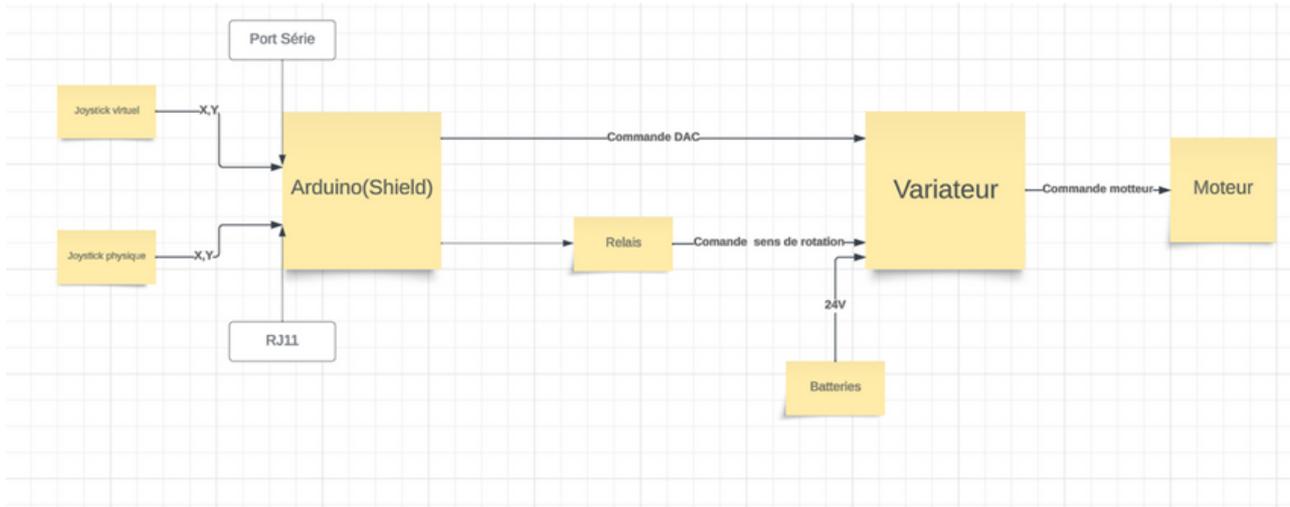
- L'entrée pour avancer(Forward)
- L'entrée pour reculer (Backward)

-La sortie commune

En résumé ,ce relais nous permet de commander la direction.

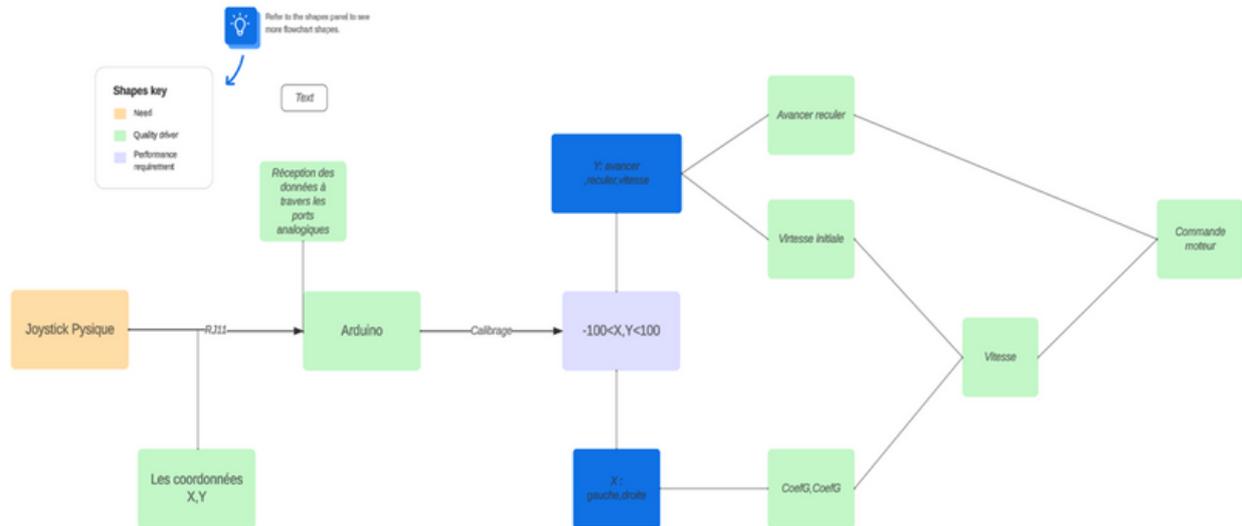
Pour résumer la partie Commande , j'ai modélisé cette partie par un schéma bloc.

Rapport Projet ingénieur



Architecture de Commande

Commande par Joystick Physique:



Rapport Projet ingénieur

Explication :

Cette architecture est divisé en plusieurs étapes :

- 1.L'arduino reçoit les données (les coordonnées X,Y) du joystick physique à travers le cable RJ11.
- 2.On procède au calibrage des coordonnées pour avoir des valeurs comprises entre -100 et 100
3. Y permet de définir la direction (avancer ou reculer) et la vitesse initiale et X permet de définir les coefficients gauches et droits
- 4- la multiplication des coefficients avec la vitesse initiale permet de définir la vitesse finale du moteur droit et du moteur gauche. Cette vitesse permet de définir les sens de rotations des deux moteurs(tourner à droite ou à gauche).

Explication :

Si on veut tourner à droite , on augmente la vitesse du moteur droit et on diminue la vitesse du moteur gauche et vice-versa

5-Une fois on récupère la vitesse finale avec la direction , on peut commander les moteurs

Cette architecture résume le fonctionnement général du programme téléversé dans la carte Arduino.

Test Joystick physique vidéo :

<https://projets->

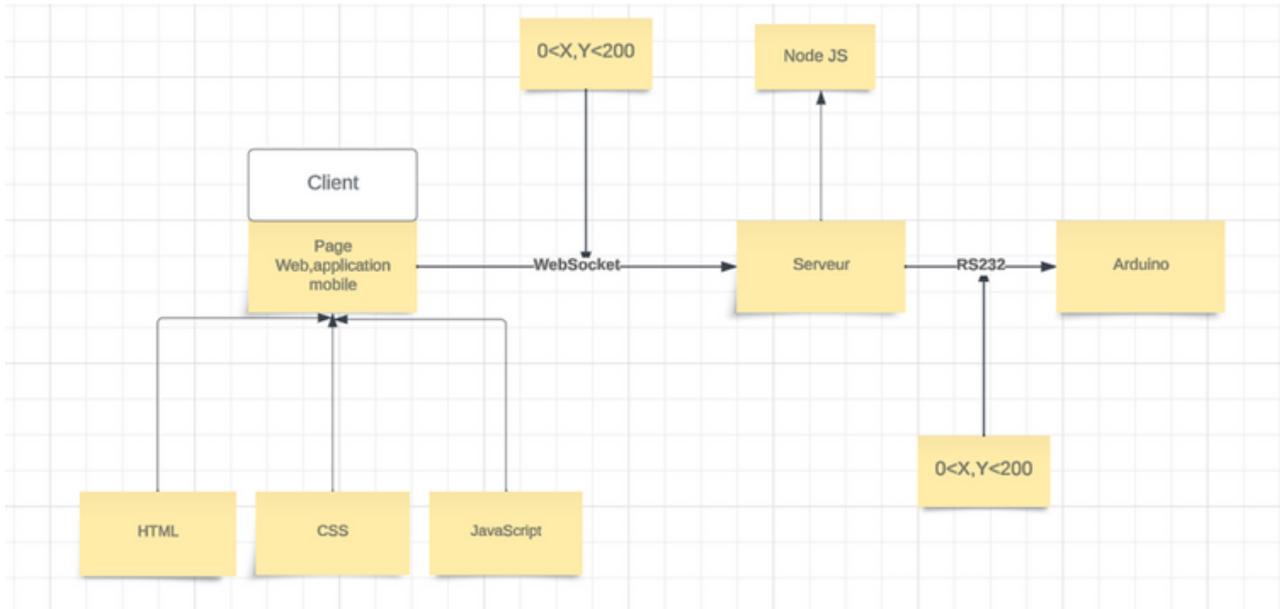
[se.plil.fr/mediawiki/images/d/d5/VID_20231114_162117.mp4](https://projets-se.plil.fr/mediawiki/images/d/d5/VID_20231114_162117.mp4)

Commande Par Joystick virtuel :

Dans cette partie ,j'ai développé deux types de commande :

- Commande par une page Web
- Commande par une application Mobile

1-Architecture générale :



Explication :

Cette architecture est divisé en plusieurs étapes :

Avant d'expliquer les étapes , il faut rappeler que notre client (Page Web ou application mobile) est programmé en utilisant trois langages de programmation :

- HTML
- CSS
- Javascript

1.Le client envoie les coordonnées X,Y à travers Websocket au serveur
 Le protocole WebSocket est un protocole réseau basé sur le protocole TCP. Celui-ci définit la façon dont les données sont échangées entre les réseaux. En raison de sa fiabilité et de son efficacité, il est utilisé par presque tous les clients.

Rapport Projet ingénieur

TCP établit une connexion entre deux points finaux de communication qu'on appelle des sockets. Ainsi, une communication bidirectionnelle s'établit entre les données.

Avantages: le chargement des données est beaucoup plus rapide. L'utilisation d'un WebSocket permet la consultation dynamique en temps réel d'un site Web. Avec le protocole WebSocket, il suffit au client d'établir la connexion avec un serveur Web. La connexion entre le client et le serveur s'établit grâce à la phase de handshake du protocole WebSocket. Dans ce cas, le client envoie toutes les identifications nécessaires à l'échange de données au serveur.

2. Le serveur reçoit les coordonnées. Ce serveur est en NodeJS comme mentionné dans la partie spécification technique ce dernier est basé sur JavaScript et très utile et adéquat avec les applications temps réel.

3- Le serveur envoie les coordonnées à l'Arduino à travers une communication série en utilisant le Protocole RS232

on envoie les coordonnées sous la forme X,Y avec une virgule de séparation

4- L'arduino reçoit l'information sous la forme mentionné ci-dessus puis il effectue la séparation de X et Y. Une fois la séparation est effectuée, on réalise la même procédure mentionnée dans la partie Joystick physique en faisant un calibrage et définition de la direction et la vitesse à travers les coordonnées X et Y récupérées du serveur.

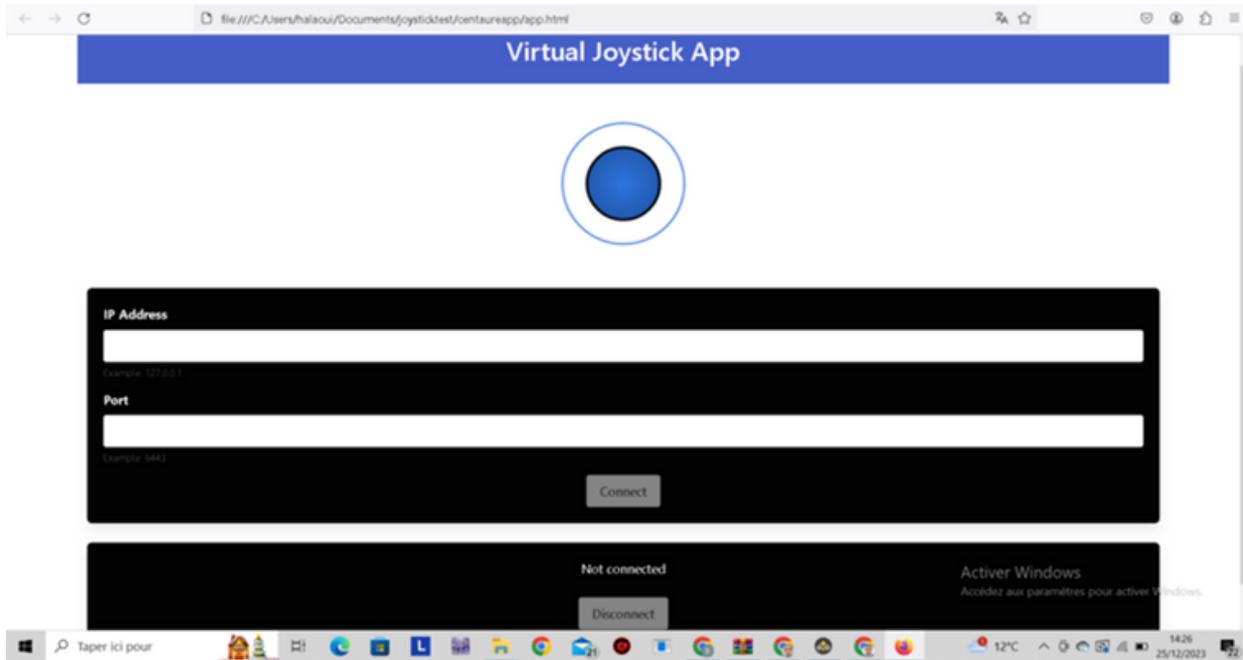
Framework Cordova :

Pour la création de l'application, j'utilise Cordova Apache comme mentionné dans la partie spécification technique car ce Framework permet la création des applications mobiles en utilisant HTML, CSS, JavaScript. Pour créer l'application du Joystick virtuel, il fallait juste installer les packages de Cordova et faire quelques modifications dans le code du serveur NodeJS.

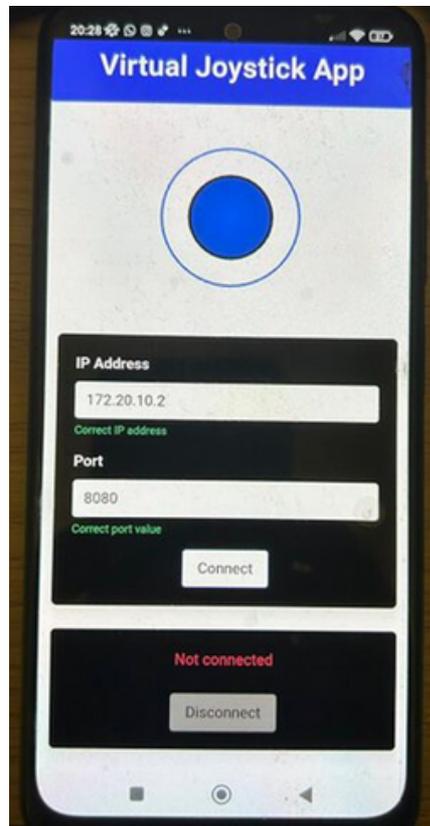
Rapport Projet ingénieur

Architecture Frontend :

Page Web:



Application mobile :



Rapport Projet ingénieur

Comme vous l'avez remarqué , j'ai gardé le même design pour la page Web et l'application mobile.

Pour se connecter , l'utilisateur doit faire entrer l'adresse IP du Robot avec le numéro de Port qui est toujours 8080 car le protocole utilisé c'est WebSocket qui est basé sur HTTP.

Une fois que les entrées sont bonnes ,l'utilisateur peut cliquer sur le bouton "connect" .Ce dernier nous permet de se connecter au serveur.

Le bouton "Disconnect" nous permet la déconnexion du serveur et donc du robot.

Test Joystick virtuel client et serveur avec affichage des coordonnées :

<https://projets-se.plil.fr/mediawiki/images/a/ae/Videotestappm.mp4>

Test Joystick virtuel avec commande d'Arduino :

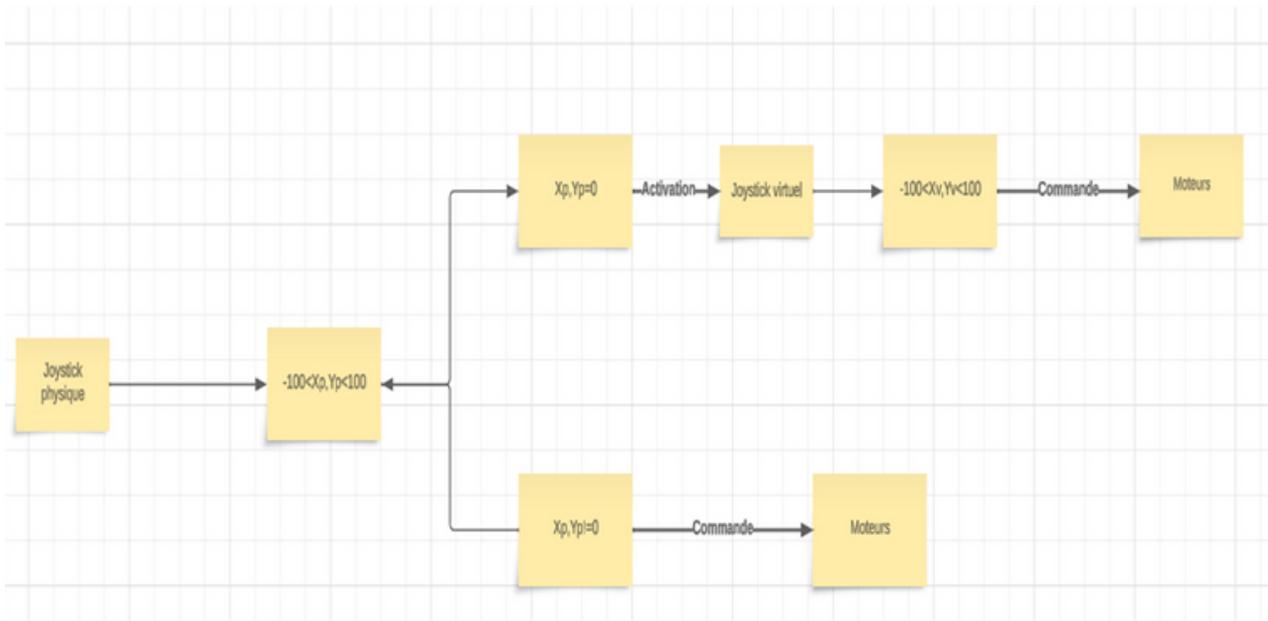
<https://projets-se.plil.fr/mediawiki/images/e/ed/Testcode22.mp4>

Test vitesse joystick virtuel :

[VID_20240120_140137.mp4](#)

Gestion de priorité : Comme nous savons très bien que dans un microcontrôleur, nous pouvons seulement faire tourner un seul code .Afin de gérer la priorité entre le Joystick Physique et le Joystick virtuel , j'ai défini une règle très simple dans le code Arduino globale(gestion des deux types de Joystick).

Rapport Projet ingénieur



La règle est la suivante : Le joystick physique est prioritaire et donc il y'a deux cas possible :

- $X_p, Y_p \neq 0$: le joystick physique est activé et le joystick virtuel bloqué
- $X_p, Y_p = 0$: Le joystick virtuel est activé

J'ai choisi le joystick physique comme prioritaire car il est très utile en cas de panne

réseaux (par exemple l'application mobile est bloquée ou problème de réseaux) et donc ce n'est pas possible de commander par Joystick virtuel.

Difficultés rencontrées:

variateur de vitesse :

Avant de commencer d'explicitier les problèmes rencontrés , ci-dessous un aperçu sur les différents types d'erreurs possible pour le variateur 8CH2QM.

- ALARMS -
(RED LED BLINKING)

CH2QM WIG-WAG CONNECTION: CH2QMW

N° OF BLINKINGS	ALARMS	WHAT TO DO
1	MOSFET SHORT CIRCUIT OR RELAIS ON AT POWER ON OR IN STOP POSITION	CHANGE THE CARD
4	POTI > 2,7V OR < 2,4V AT POWER ON	VERIFY THE MIDDLE POSITION OF THE POTI
5	BREAKING POTI > 4,8V OR < 0,2 V	VERIFY THE WIRING OF THE POTI
ALWAYS ON	THERMAL PROTECTION	WAIT SOME MINUTES AND RESTART

CH2QM SINGLE ENDED CONNECTIONS: CH2QM

N° OF BLINKINGS	ALARMS	WHAT TO DO
1	MOSFET SHORT CIRCUIT OR RELAIS ON AT POWER ON OR IN STOP POSITION	CHANGE THE CARD
2	FORWARD SWITCH ON AT POWER-ON	VERIFY THE WIRING OF THE SWITCHES
3	BACKWARD SWITCH ON AT POWER-ON	VERIFY THE WIRING OF THE SWITCHES
4	POTI > 170mV AT POWER ON	VERIFY THE WIRING OF THE POTI
5	BREAKING POTI > 4,8V OR < 40 mV	VERIFY THE WIRING OF THE POTI
6	FORWARD AND BACKWARD SWITCHES ON	VERIFY THE WIRING OF THE SWITCHES
ALWAYS ON	THERMAL PROTECTION	WAIT SOME MINUTES AND RESTART

Le nombre de clignotements de la led rouge indique un type d'erreur spécifique. Si le variateur est en mode erreur, ce dernier est bloqué et donc les moteurs ne tournent pas.

Rapport Projet ingénieur

Problème 1:

Lors de mes premiers tests sur Robot , j'ai remarqué que les batteries se déchargent une fois que je lance la commande des variateurs. Après vérification des freins

moteurs , j'ai trouvé qu'ils sont bien alimentés par 24V ,sachant que les moteurs tournaient bien.

Après plusieurs commandes de variateur , ce dernier se mettait en mode erreur avec 5 clignotements de la led rouge ce qui indique que la tension de POTI(potentiomètre) n'est pas bonne (soit $>4.78V$ soit $<40mV$).

Afin de vérifier la tension du DAC qui joue le rôle du potentiomètre , j'ai injecté plusieurs valeurs d'entrée dans le DAC et j'ai mesuré la tension de sortie du DAC.

Pourcentage	tension de sortie du DAC /batt(V)
0	0.11
10	0.14
20	0.17
25	0.85
40	1.55
50	1.99
60	2.25
70	3.02
80	3.24
90	3.57
100	3.75

Rapport Projet ingénieur

Comme vous l'avez remarqué que les tensions respectent bien la Datasheet et donc l'erreur ne provient pas de la tension de sortie du DAC.

Solution : Après plusieurs jours de recherche dans les anciens rapports , j'ai trouvé que pour résoudre ce problème il faut que le variateur et l'arduino aient la même masse et donc la même référence de tension ce qui est tout à fait normal.

Problème 2 :

Après la résolution du premier problème , un autre type d'erreur apparaît avec 2 ou 3 clignotements qui indique d'après la datasheet que la tension du forward est activé ou la tension Backward est activé.

Explication :

-Si je commande le moteur pour avancer (Forward) , j'ai 3 clignotements de la led rouge qui indiquent la présence d'une tension dans le pin Backward.

-Si je commande le moteur pour reculer (backward), j'ai 2 clignotements de la led rouge qui indiquent la présence de la tension dans le pin Forward.

Démarche :

Étape 1 : j'ai commandé le moteur pour avancer (Forward) ensuite j'ai mesuré la tension du Backward dans le variateur et j'ai trouvé 0V.

Afin de vérifier cette tension ,j'ai mesuré la tension du Backward au niveau du shield et j'ai trouvé de 0V.On peut conclure que le source du problème ne provient pas du relais connecté à Arduino mais provient surement du variateur.

Afin de confirmer cette hypothèse :

- j'ai débranché le variateur du relais (Arduino) .

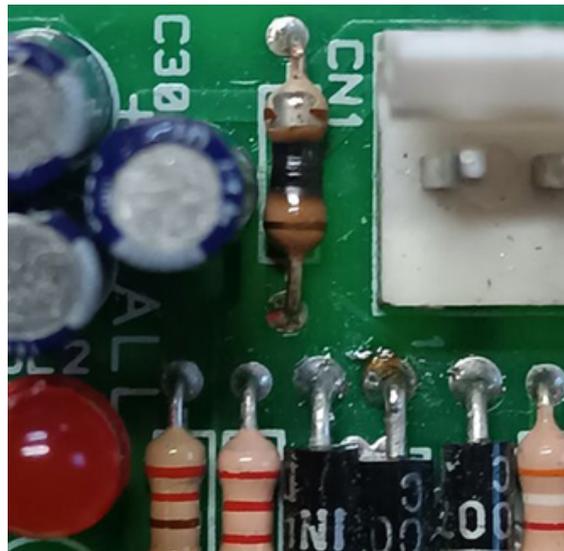
- J'ai juste laissé la commande du DAC pour le potentiomètre.

-J'ai injecté directement 24 V dans le pin du Forward et 0V dans le pin du Backward.

Rapport Projet ingénieur

Résultat : le mode erreur persiste ce qui confirme l'hypothèse.

Après plusieurs tests , une des résistances dans les deux variateurs s'est brûlée à cause d'une surintensité.



RRéparation des variateurs :

Le problème rencontré dans la réparation c'est que la valeur de la résistance est inconnue et l'absence d'informations sur la valeur dans la Datasheet, donc pour identifier la valeur de la résistance il fallait suivre une démarche :

- 1-Dessouder la résistance brûlée du variateur
- 2-Gratter avec un cutter la couche isolante



Rapport Projet ingénieur

3-Sachant que la couche du carbone qui a été coupée, il faut juste identifié la coupure et de considérer qu'on a deux résistance en série , après il faut mesurer la première résistance et la deuxième résistance et de faire la somme des deux résistances pour obtenir la valeur de la résistance grillée.

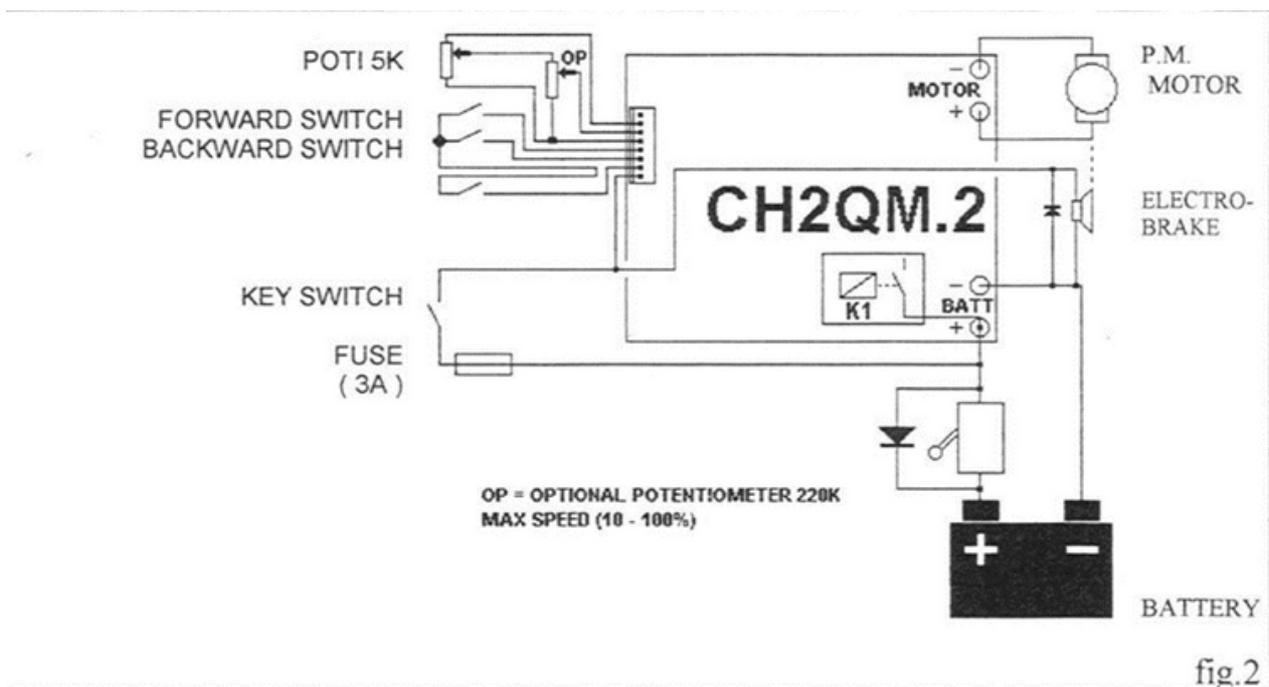


En suivant cette démarche , j'ai pu mesurer la valeur de la résistance qui est égale à 100 Ohms

Après l'identification de la valeur de la résistance , j'ai procédé au changement de la résistance des deux variateurs.

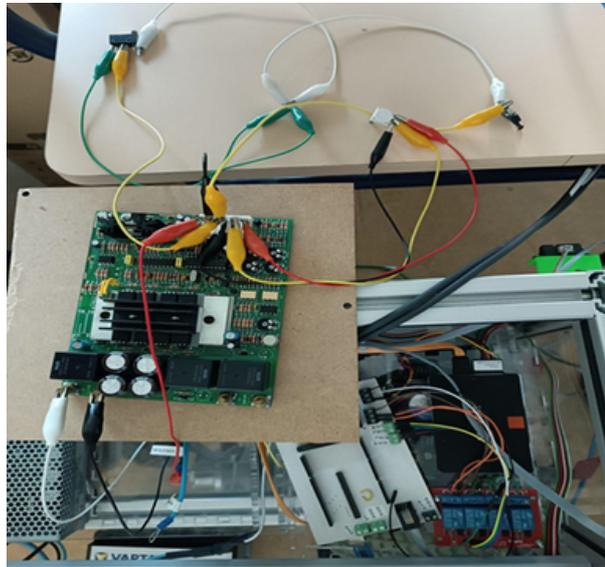
Test et Résultats : Une fois j'ai changé les deux résistances,j'ai réalisé le test de chaque variateur dans le robot et apparemment la résistance branchée marche bien mais le mode erreur persiste.

Schéma de base :



Rapport Projet ingénieur

Afin de bien comprendre le fonctionnement de ce variateur , j'ai réalisé ce montage ci-dessus .



Résultats : le premier variateur marche bien mais le deuxième variateur se met en mode erreur avec un seul clignotement de la led rouge qui indique qu'il faut changer la carte d'après la datasheet. En fin de compte un seul variateur est fonctionnel. .

Variateur fonctionnel vidéo : :

https://drive.google.com/file/d/1Cqmu_UszMJS02WtYoV2dtNeS-ZGuFR4E/view?usp=sharing

Variateur défaillant :

<https://drive.google.com/file/d/1CsSFIiRHgZMyQJ4N1Bt7C7XgF3gv3RJB/view?usp=sharing>

En se basant sur le variateur fonctionnel , j'ai réalisé plusieurs mesures de la tension au borne dupin du potentiomètre.

Résultat : le variateur ne démarre que si la tension au borne du potentiomètre est au alentours de 165mV. Une donnée importante à prendre en considération dans la fonction Setup de l'Arduino.

NB: Pour avoir une tension de 165mv , il faut fournir un rapport de 170 au DAC

Rapport Projet ingénieur

Solution : En prenant en considération cette donnée , j'ai repris le montage de commande précédent et en téléversant un programme simple qui commande un des deux moteurs pour avancer ou reculer et par suite le variateur fonctionne en commandant un des deux moteurs.

Commande moteur vidéo :

https://drive.google.com/drive/folders/1-xRE9jO8qAzk_n6qTiF9zGfg8RL0CMcP?usp=sharing

Bilan :

Après 6 semaines de travail , j'ai pu atteindre plusieurs objectifs :

Organisation étage de commande :

- Conception d'un boîtier pour la partie commande
- Conception du Shield avec plusieurs fonctionnalités mais sans le fabriquer.

Développement de la commande :

- Commande par Joystick physique
- Commande par joystick virtuel (page web ,application mobile)

Objectif non réalisés :

- Commande par la caméra Kinect en utilisant :
 - Le capteurs infrarouge.
 - L'IA.

Le problème que j'ai eu avec le variateur m'a pris plus que 50% de mon temps de travail et le fait que j'ai commencé le projet un mois en retard m'a empêché de réaliser tous les objectifs.

Comparaison entre le diagramme prévisionnel et réel :

Comme je l'ai déjà mentionné , le problème que j'ai eu dans le variateur m'a empêché de fabriquer le shield et d'implémenter la commande par kinect et par suite le diagramme réel est décalé par rapport au diagramme prévisionnel.

Diagramme Réel :



Développement Robot :

Afin de développer le robot dans le futur , il faut :

Hardware :

-Avoir deux variateurs de vitesse fonctionnels , il faut faire attention à deux points :

1.Branchement :

en prenant deux variateurs de vitesse avec la même référence que les ancien et il faut faire attention à quelque points lors du branchement :

- le variateur et l'Arduino doivent avoir la même masse.
- Faire attention au court-circuit donc une bonne isolation des câbles.
- Respecter le branchement avec le shield .

Rapport Projet ingénieur

.Condition de démarrage du variateur :

Pour démarrer le variateur et pour éviter les modes erreurs :

-Il faut garantir une tension d'initialisation du potentiomètre dans les alentours de 165mV.

-Il faut garantir une tension nulle dans les deux pins Forward et Backward lors du démarrage du variateur.

Sans ces deux conditions ,le variateur va toujours se mettre en mode erreur.

Software :

-Développement d'Algorithme pour la caméra Kinect soit en utilisant le capteur infrarouge soit en utilisant l'IA.

Capteur Infrarouge :

L'avantage de la Kinect par rapport à une caméra classique est qu'elle dispose d'un capteur infrarouge. Ainsi, pour chaque pixel de la vidéo, on dispose des taux de rouge-vert-bleu (RGB) de celui-ci ainsi que de sa distance à la caméra. On utilise la libfreenect, qui permet à un système UNIX de contrôler et d'exploiter un capteur Kinect.

Intelligence artificielle :

Nous pouvons utiliser Google Colab comme plateforme et Tensor Flow comme logiciel. Grâce à cette plateforme, nous pouvons réaliser l'apprentissage du circuit et coder facilement sans avoir d'installations à faire.

CONCLUSION

En revisitant le processus de débogage, je ne peux m'empêcher de reconnaître l'impact significatif que la réalisation du montage de base aurait eu sur la compréhension du fonctionnement du variateur, en particulier en ce qui concerne les conditions de démarrage. Avec le recul, il est évident que débiter par cette étape aurait pu fournir des informations cruciales pour orienter de manière plus éclairée mes actions ultérieures.

La décision de se baser sur les anciens rapports a été un choix qui, bien que motivé par le désir d'économiser du temps, a finalement compliqué davantage la résolution du problème. La réalisation du montage de base aurait non seulement permis une meilleure appréhension des spécifications techniques, mais aurait également mis en lumière des aspects pratiques du fonctionnement du variateur qui auraient échappé à une approche purement théorique.

Cette réflexion souligne l'importance cruciale d'une compréhension approfondie du système dès le départ. Mon engagement à apprendre de cette expérience se traduit par la ferme résolution de débiter tout futur projet de débogage en établissant d'abord les fondations pratiques nécessaires pour une analyse approfondie.

En fin de compte, chaque défi rencontré dans ce processus de débogage contribue à ma croissance professionnelle, et je suis reconnaissant pour ces enseignements qui m'orientent vers une approche plus robuste et éclairée du débogage technique.