

31 mai 2023

Projet Ferrari

By Arthur de Sallier Dupin



Sommaire

Table des matières

Introduction :.....	3
Objectif :	3
Cahier des charges :.....	3
Diagramme de Gantt :	3
Défis rencontrés et solutions apportées :	3
Sketch Arduino :.....	4
Fonctionnement du Robot :	4
Prise en main :	4
Etape 1 :.....	4
Etape 2 :.....	4
Etape 3 :.....	4
Fritzing :	5
Fonctionnement Pont H	6
Conclusion :	6
Amélioration :.....	6
Avantages :	7
Annexe :	8
Cahier des charges projet Ferrari :	8
Sketch Arduino du projet Ferrari :	10



Introduction :

Le présent rapport rend compte du projet de conception et de fabrication d'un robot Arduino contrôlé par Bluetooth, intitulé "Projet Ferrari". Ce rapport présente les détails du projet, y compris les différentes phases de développement, les défis rencontrés et les solutions apportées, ainsi que Sketch Arduino utilisé.

Objectif :

Ce projet a pour but de créer un robot puissant, transportable, maniable et réactif capable de se déplacer à distance en Bluetooth. L'objectif était de se lancer un défi technique en utilisant un pont H et en explorant les capacités de communication sans fil offertes par le Bluetooth.

Cahier des charges :

Le cahier des charges a été rédigé pour définir les spécifications et les fonctionnalités clés du robot. Il comprenait des éléments tels que la connexion Bluetooth avec une application Android, la réception des commandes, le déplacement du robot en utilisant les moteurs à courant continu et la roue jockey, ainsi que des spécifications techniques sur l'alimentation et le design compact du robot. Le cahier des Charges est disponible dans l'annexe du rapport.

Diagramme de Gantt :

Le diagramme de Gantt suivant illustre le planning du projet :

Semaines	03-avr	10-avr	17-avr	24-avr	01-mai	08-mai	15-mai	22-mai	29-mai
Choix du projet									
Rédaction du cahier des charges									
Collecte des éléments									
Réalisation du sketch									
Test sur une modélisation avec des leds									
Assemblage des éléments									
Test en condition réelles									
Validation du projet par rapport au cahier des charges									
Ecriture du rapport du projet									
Réalisation du Powerpoint de présentation									

Défis rencontrés et solutions apportées :

L'un des principaux défis rencontrés lors du projet a été la compréhension et le branchement correct du pont H, qui permet de contrôler les moteurs. Des conseils ont été sollicités auprès de mon Père, qui a su m'expliquer comment comprendre le schéma de branchement du pont H. Une partie dans ce rapport est dédiée au fonctionnement du pont H.

Un autre défi a été la surtension de l'alimentation logique (VLS) du pont H qui a entraîné la destruction de plusieurs ponts H. Après avoir identifié la source du problème, une alimentation adaptée a été utilisée, permettant de protéger le pont H et d'éviter de futurs dysfonctionnements.



Par précaution, pour conserver l'avancée du projet, j'ai décidé d'utiliser GitHub pour conserver les modifications du sketch Arduino en cas de problème. Le projet Est disponible sur mon compte GitHub : [GitHub@Adsdworld](#). Le projet peut être facilement cloné pour reproduire et/ou améliorer le projet. L'agent ssh n'a pas été facile à configurer mais maintenant je maîtrise l'outil Git.

Sketch Arduino :

Le sketch développé pour le contrôle du robot Arduino a été basé sur une philosophie de code simple, structuré et commenté pour une compréhension immédiate. Il est disponible dans l'annexe du rapport, offrant ainsi une référence pratique pour les futurs projets similaires.

Fonctionnement du Robot :

Après connexion au module Bluetooth, l'application va écrire une lettre dans le cache de l'Arduino, si aucun bouton n'est pressé par défaut « S » est écrit, les autres lettres possibles sont : F, B, R, L, I, H, J, G. Lorsque l'application n'est pas connectée, aucune lettre n'est écrite dans le cache, lorsque F, B, R, L, I, H, J, G ne sont pas lues, alors le robot s'arrête. Chaque lettre correspond à un déplacement précis, cf. Cahier des charges : Bluetooth RC Controller.

Prise en main :

L'application « Bluetooth RC Controller est conçue pour une prise en main rapide en 3 étapes :

Etape 1 : Se connecter au module Bluetooth via l'application « Bluetooth RC Controller », le récepteur Bluetooth est sécurisé par un code pin, 0000 ou 1234 en général dans mon cas c'était 1234.

Etape 2 : choisir le mode de contrôle : soit par appuie sur des boutons soit par contrôle gyroscopique.

Etape 3 : Hop vous contrôlé le robot !

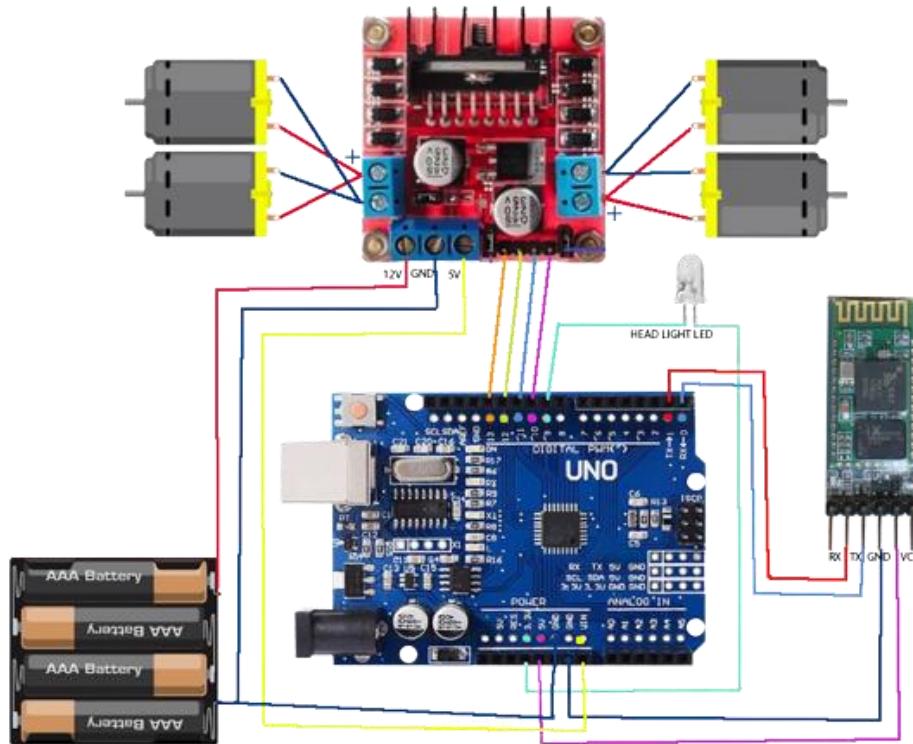


Fritzing :

Fritzing est un logiciel trouvable gratuitement sur internet, téléchargeable contre donation sur le site officiel, Il permet de réaliser des schémas visuels de branchements de composants Arduino.

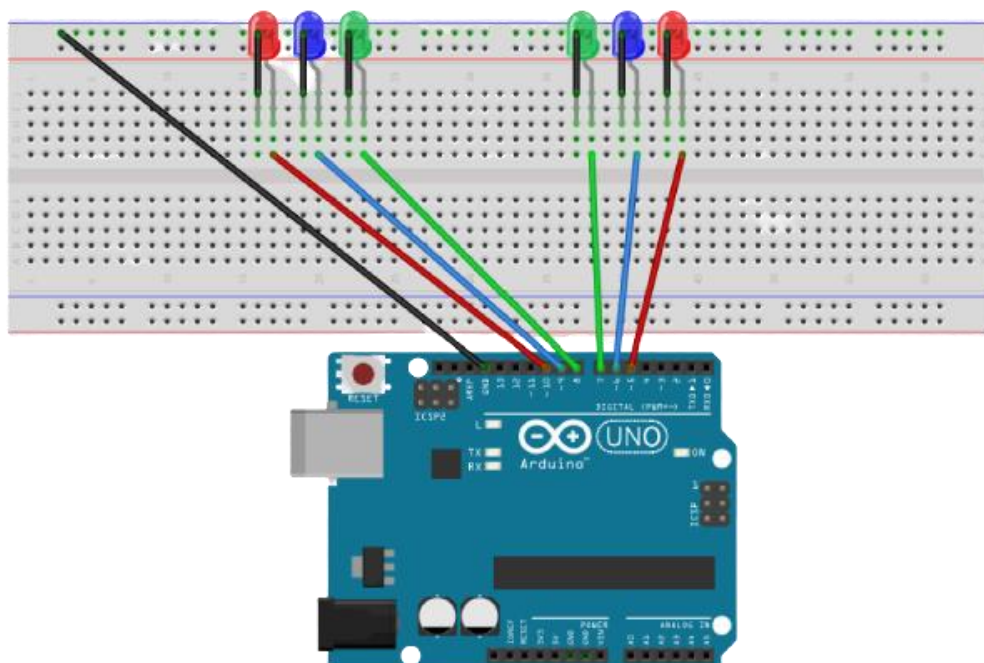
Voici un schéma du montage du robot récupéré sur internet réalisé sur Fritzing
([source](#))

Ce schéma inclus 4 moteurs dans mon projet je n'en utilise que 2 pour mon robot, la led présente sur le schéma n'est pas utile non-plus.



Voici un schéma du montage de la board de test avec les leds réalisé sur Fritzing :

Ce montage a été utilisé pour vérifier le fonctionnement du sketch avant de téléverser le code dans le robot.



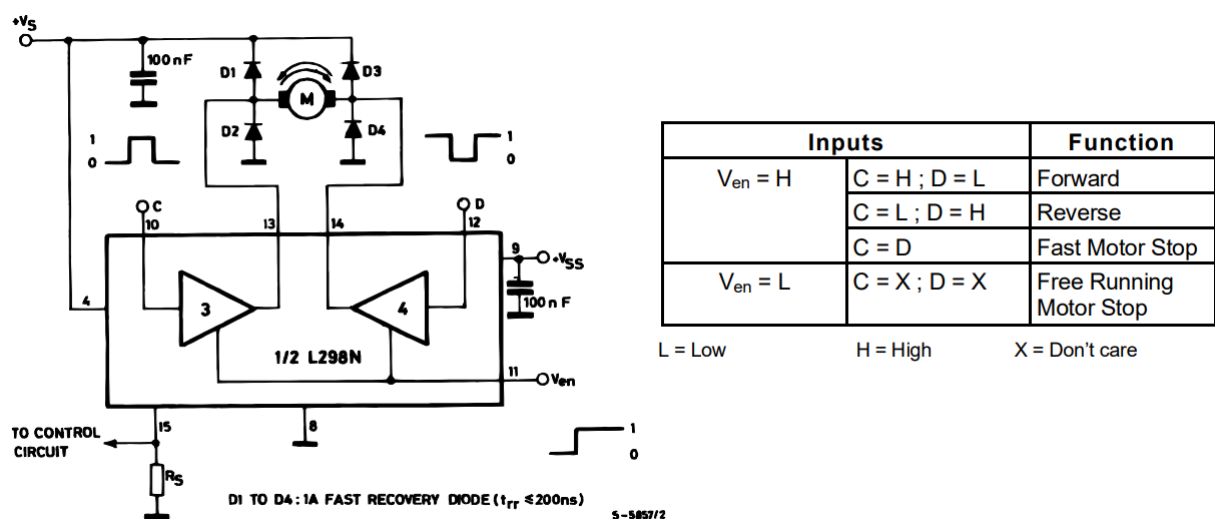


Fonctionnement Pont H

Le pont H permet de contrôler jusqu'à 2 moteurs, il existe bien d'autres utilisations possibles du pont H, Il dispose de 15 broches qui permettent un contrôle total des 2 moteurs. Au minimum, IN 1/2/3/4 doivent être relié à l'Arduino, car on peut laisser EN A/B sur HIGH avec des petits clips.

1	OUT 1	Entré moteur A
2	VS	Entré alimentation moteur
3	EN A	Si HIGH par rapport au VLS, permet d'activer le moteur A
4	GND	Masse
5	CSA	Capteur inutile pour le projet, à relier à la masse
6	OUT 2	Sortie moteur A
7	IN 1	Si IN 1 = HIGH et IN 2 = HIGH, arrêt forcé du moteur
8	IN 2	Si IN 1 = HIGH et IN 2 = LOW, Le moteur tourne dans un sens Si IN 1 = LOW et IN 2 = HIGH, Le moteur tourne dans l'autre sens
9	VLS	Entré alimentation logique
10	EN B	Si HIGH par rapport au VLS, permet d'activer le moteur B
11	OUT 3	Entré moteur B
12	CSB	Capteur inutile pour le projet, à relier à la masse
13	IN 3	Si IN 3 = HIGH et IN 4 = HIGH, arrêt forcé du moteur
14	IN 4	Si IN 3 = HIGH et IN 4 = LOW, Le moteur tourne dans un sens Si IN 3 = LOW et IN 4 = HIGH, Le moteur tourne dans l'autre sens
15	OUT 4	Sortie moteur B

Schéma du branchement du moteur trouver sur la [datasheet](#) :



Conclusion :

Le Projet Ferrari a été une expérience enrichissante et stimulante. La réalisation d'un robot Arduino contrôlé par Bluetooth a permis d'explorer les domaines de l'électronique, de la programmation et de la communication sans fil.

Amélioration :

Tout d'abord, il est possible d'améliorer le robot avec un caméra pour du control sans vision directe. De plus, en modifiant en profondeur le robot, il est envisageable de passer à une communication via wifi pour un control à distance total.



Avantages :

Ce robot est un jouet idéal pour un enfant, il est petit, transportable, les pièces sont changeables facilement et par sa petite taille il permet de s'amuser en intérieur pour un moindre coût.



Annexe :

Cahier des charges projet Ferrari :

By Arthur de Sallier Dupin

Objectif :

Le but de ce projet est de concevoir et de fabriquer un petit robot Arduino à deux roues et une roue jockey contrôlé par Bluetooth via une application Android. Ce robot sera capable de se déplacer en suivant les commandes de l'application Android via Bluetooth.

Fonctionnalités :

- Connexion Bluetooth avec une application Android (app : Bluetooth RC controller).
- Réception de commandes de l'application Android via Bluetooth.
- Déplacement en fonction du signal reçu par Bluetooth en utilisant les deux roues motrices et la roue jockey.

Spécifications techniques :

- Le robot doit être contrôlé par une carte Arduino.
- Le robot doit avoir deux moteurs à courant continu pour les roues.
- Le robot doit être équipé d'un module Bluetooth slave pour la communication sans fil avec l'application Android.
- L'application Android doit être conçue pour communiquer avec le robot via Bluetooth et permettre au robot de suivre les commandes de l'utilisateur.
- Les moteurs doivent être alimentés par une batterie rechargeable.
- La carte Arduino doit être alimentée par une batterie rechargeable.
- Le robot doit pouvoir être contrôlable dans des espaces étroits.
- A pleine vitesse il doit aller aussi vite qu'un humain qui marche.
 - Poids inférieur à 1kg

Design :

- Le robot doit être compact et doit être transportable dans une boîte de 25 cm sur 18 cm sur 7 cm sans être démonté.

Matériel :

- 2 moteurs
- Une roue jockey
- Une carte Arduino
- Un module Bluetooth slave
- Un pont H
- Une batterie pour les moteurs
- Une batterie pour la carte Arduino
- Des fils



- Un châssis

Bluetooth RC controller :

- **Permet d'envoyer des données soit par :**
 - Appuye sur des boutons
 - Détection d'une position gyroscopique
- **Données :**
 - « donnée » → Action(PWN_droite, PWN_gauche)
 - « F » → Avancer(255, 255)
 - « B » → Reculer(255, 255)
 - « R » → Droite(0, 150)
 - « L » → Gauche(150, 0)
 - « I » → Avancer+Droite(100, 33)
 - « G » → Avancer+Gauche(33, 100)
 - « H » → Reculer+Droite(100, 33)
 - « J » → Reculer+Gauche(33, 100)

Délai :

Le 30 mai, date de présentation du projet, le projet devra être terminé.



Sketch Arduino du projet Ferrari :

/*

Description : Contrôler une voiture avec 2 moteurs sur son téléphone

app "Bluetooth RC controller" disponible sur le Play store

*/

//définition des variables

char t; //stocke les actions dans le buffer de l'arduino

int Current_C=0; //stocke la puissance du moteur au cours du temps

int Current_D=0;

int Current_E=0;

int Current_F=0;

int Forward_and_Back_Power=255; //définitions des puissances pour les mouvements

int Right_and_Left_Power=150;

int Other_directions=100;

int Turning_Forward_and_Back_Power=Other_directions/3;

int AC=5; //définition de la pin du moteur droit

int AD=6; //définition de la pin du moteur gauche

int BC=9; //définition de la pin du moteur droit

int BD=10; //définition de la pin du moteur gauche

//définition des fonctions

void f_vitesse(char A, char B, int C, int D, int E, int F){ //cette fonction ajuste la puissance voulue sur les 2 moteurs

```
while (C!=Current_C || D!=Current_D || E!=Current_E || F!=Current_F){
  if (C<Current_C){analogWrite(AC, --Current_C);}
  if (C>Current_C){analogWrite(AC, ++Current_C);}
  if (D<Current_D){analogWrite(AD, --Current_D);}
  if (D>Current_D){analogWrite(AD, ++Current_D);}
  if (E<Current_E){analogWrite(BC, --Current_E);}
  if (E>Current_E){analogWrite(BC, ++Current_E);}
  if (F<Current_F){analogWrite(BD, --Current_F);}
  if (F>Current_F){analogWrite(BD, ++Current_F);}
  delay(1);
}
```

```
void setup() {
  pinMode(AC,OUTPUT);
  pinMode(AD,OUTPUT);
  pinMode(BC,OUTPUT);
  pinMode(BD,OUTPUT);
}
```

```
void loop() {
  if(Serial.available()){
    t = Serial.read();
```



```
}

if(t == 'F'){
f_vitesse(HIGH, HIGH, 0, Forward_and_Back_Power, 0, Forward_and_Back_Power);
}else{
if(t == 'B'){
f_vitesse(HIGH, HIGH, Forward_and_Back_Power, 0, Forward_and_Back_Power, 0);
}else{
if(t == 'L'){
f_vitesse(HIGH, LOW, 0, Right_and_Left_Power, 0, 0);
}else{
if(t == 'R'){
f_vitesse(LOW, HIGH, 0, 0, 0, Right_and_Left_Power);
}else{
if(t == 'H'){
f_vitesse(HIGH, HIGH, Other_directions, 0, Turning_Forward_and_Back_Power, 0);
}else{
if(t == 'J'){
f_vitesse(HIGH, HIGH, Turning_Forward_and_Back_Power, 0, Other_directions, 0);
}else{
if(t == 'I'){
f_vitesse(HIGH, HIGH, 0, Turning_Forward_and_Back_Power, 0, Other_directions);
}else{
if(t == 'G'){
f_vitesse(HIGH, HIGH, 0, Other_directions, 0, Turning_Forward_and_Back_Power);
}else{
f_vitesse(LOW, LOW, 0, 0, 0, 0); //dans ce cas T est égal à S ou null, mais pour éviter tout
problème on ne vérifie pas t
}}}}}}}}
}

}
```