

Projet Arduino
Curious Car
Rapport de conception



Table des matières

1. Objectif du projet.....	2
2. Fonctionnement général.....	2
3. Matériel utilisé.....	2
4. Assemblage.....	3
4.1. Phares.....	3
4.2. Mouvement.....	3
4.3. Caméra.....	4
4.4. Alimentation.....	4
5. Résultat obtenu.....	5
6. Difficultés rencontrées.....	5
7. Retour personnel sur le projet.....	6
8. Sources d'inspiration.....	7
9. Annexe code.....	7

1. Objectif du projet

Au début, l'objectif du projet était de concevoir une petite voiture qu'on pouvait contrôler par Wi-Fi à l'aide d'un ordinateur ou un smartphone. Une caméra IP embarquée nous permettrait de voir où va la voiture et d'observer ce qui se passe autour. 2 capteurs de distance, devant et derrière, assureraient l'évasion d'obstacles par la voiture. Un mini haut-parleur attaché à la voiture jouerait des messages audio prédéfinis à la demande. On avait aussi prévu des phares et un clignotant qu'on pouvait allumer ou éteindre.

Mais à cause des problèmes rencontrés durant l'assemblage, j'ai dû supprimer le clignotant, le haut-parleur et les capteurs de distance du projet. L'objectif du projet est devenu donc de faire une voiture contrôlable par Wi-Fi avec une caméra et des phares.







2. Fonctionnement général

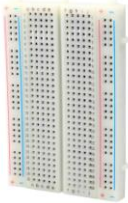


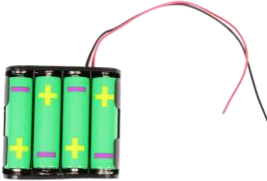


Une carte Wi-Fi programmable se connecte à un réseau Wi-Fi prédéfini dans le code. Elle crée un serveur sur le port 80 (port standard pour le protocole HTTP). Et quand un client va à son adresse IP dans un navigateur Internet, elle lui renvoie une page web interactive. Sur cette page il y a des boutons avec toutes les commandes possibles. Les commandes sont : tourner à droite ou à gauche, aller en avant ou en arrière, allumer ou éteindre les phares. En appuyant sur un bouton, la page se renouvelle avec une nouvelle requête GET dont le paramètre correspond à la commande écrite à côté du bouton. Cette requête est analysée par la carte Wi-Fi, et une action est réalisée en fonction de la commande reçue.

La caméra IP est totalement séparée du système. Elle a sa propre batterie et sa propre adresse IP. On peut accéder l'image à l'aide de l'application fournie avec la caméra.

3. Matériel utilisé

Le matériel utilisé est :

 Carte Node Mcu 1.0 (x1)	 Voiture radiocommandée (x1)	 Caméra IP (x1)
 Cable micro USB (x1)	 Pilote moteur L293D (x1)	 Carte micro SD (x1)

 <p>Carte de test (x1)</p>	 <p>LED blanches (x2)</p>	 <p>Résistances 220 Ohm (x2)</p>
 <p>Piles de 1,5V (x4)</p>	 <p>Mini powerbank (x1)</p>	 <p>Fils Arduino</p>

Cette liste changeait en fonction des problèmes rencontrés, et la liste présentée est la liste finale.

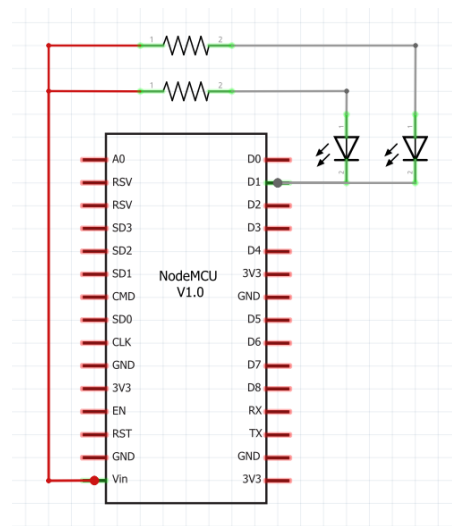
4. Assemblage

4.1. Phares

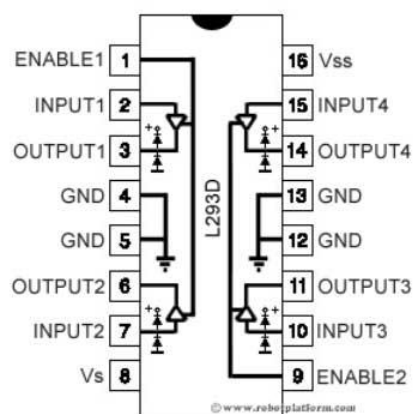
J'ai commencé l'assemblage par le plus facile : les phares. J'ai utilisé 2 LED blanches et 2 résistances de 220 Ohm. Le schéma de branchement est comme ci-contre.

Quand le pins D1 est en mode LOW, les phares sont allumées. Sinon, elles sont éteintes.

J'ai soudé chaque diode avec sa résistance, et encore j'ai prolongé les pattes des composants par des fils pour que les phares puissent atteindre leur place devant.

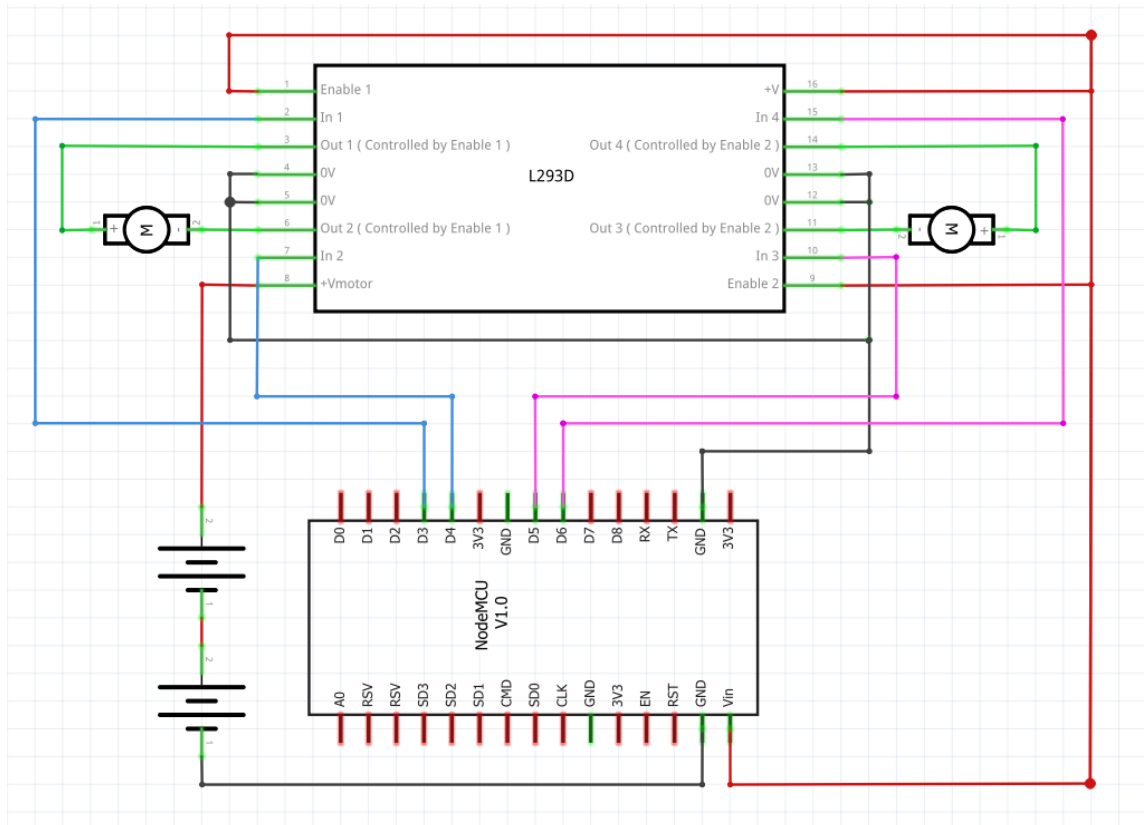


4.2. Mouvement



Ensuite, j'ai réalisé le contrôle du mouvement par la carte Wi-Fi. J'ai commencé par enlever le circuit qui était déjà à l'intérieur de la voiture car je n'allais pas utiliser la communication radio à 27MHz qu'il permettait et j'avais besoin de la place à l'intérieur de la voiture. Pour piloter les 2 moteurs DC, qu'il y avait dans la voiture, j'ai utilisé un microcircuit pilote moteur L293D. Les signaux en sortie des I/O du Node Mcu sont de faible puissance, et L293D les transforme en courants suffisants pour piloter les moteurs. L293D permet de piloter 2 moteurs en même temps. De plus, il permet de changer leur sens de rotation à l'aide des ponts en H.

Voici le schéma des branchements :



J'ai connecté les enable à 5V pour simplifier le câblage car la table de vérité du microcircuit est la suivante :

ENABLE1	INPUT1	INPUT2	OUTPUT1	OUTPUT2
0	X	X	0	0
1	0	0	0	0
1	1	0	1	0
1	0	1	0	1
1	1	1	1	1

Et j'ai utilisé 4 piles de 1,5V pour alimenter les moteurs séparément du système pour ne pas le surcharger. Sinon, si on alimentait les moteurs par la carte Node Mcu, on ne pouvait pas tourner les 2 moteurs en même temps car il n'y avait pas assez de puissance.

4.3. Caméra

Pour rajouter la caméra, il a fallu seulement la configurer pour qu'elle se connecte à notre Wi-Fi et la mettre sur le toit de la voiture. La configuration se fait dans l'application fournie avec la caméra.

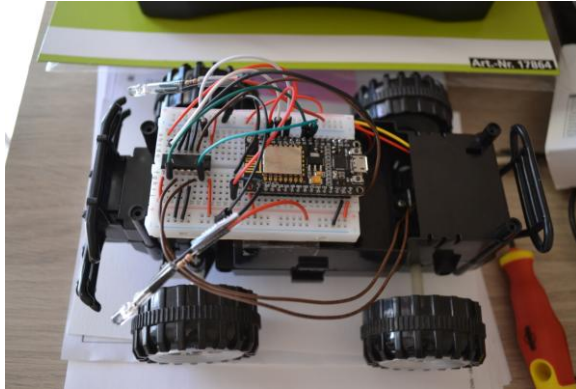
4.4. Alimentation

J'ai alimenté la carte Node Mcu avec un mini powerbank de charge 2200mAh via un petit câble micro USB. Une source d'alimentation comme ça est beaucoup plus durable qu'une pile de 9V, par exemple.

Une 2^{ème} source d'alimentation, 4 piles de 1,5V, servent à alimenter les moteurs séparément du système pour ne pas le surcharger.

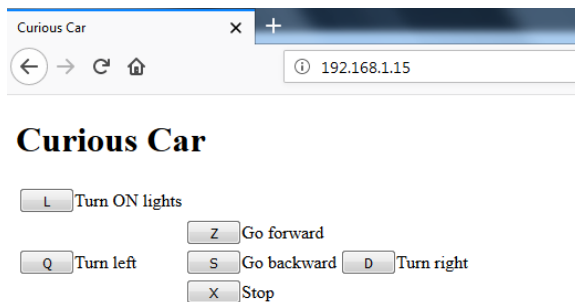
5. Résultat obtenu

On fait tous les branchements. On télécharge le programme sur la carte Node Mcu. On attache la caméra et le powerbank. Finalement, on obtient ce qu'on avait prévu au début ! Plus ou moins.



La voiture se connecte à notre Wi-Fi et on peut lui envoyer des commandes dans le navigateur. Les commandes sont :

- Allumer ou éteindre les phares.
- Tourner à gauche.
- Tourner à droite.
- Aller en avant.
- Aller en arrière.



Le Wi-Fi permet de contrôler la voiture à une distance beaucoup plus grande que sa radiocommande , à 50m au lieu de 5m.

En ce qui concerne la caméra, on peut voir l'image dans l'application fournie avec la caméra.

Curious Car fait un exemple de robot de téléprésence.

6. Difficultés rencontrées

- **Connectivité de la caméra IP.** On pouvait bien se connecter à la caméra IP directement et voir l'image, mais elle ne voulait pas se connecter au Wi-Fi que je lui indiquais pour qu'on puisse se connecter à la caméra sur le réseau. Pour résoudre ce problème, il a fallu changer l'adresse IP de la caméra dans son application pour que la box Wi-Fi et la caméra soient dans le même sous-réseau. Mais parfois la caméra bugue et ne veut toujours pas se connecter au Wi-Fi.

- **Autonomie de la voiture.** Au début je comptais alimenter la voiture avec 4 accumulateurs rechargeables de 1,5V. Ils étaient dans la compléctation initiale de la voiture, la tension de 6V était convenable pour le port V_{in} de l'Arduino qui supporte jusqu'à 12V, et il ne me restait qu'à brancher ces 6V à V_{in} et la masse à GND. Mais je me suis aperçu que le port V_{in} ne marchait pas car rien ne se passait si on alimentait l'Arduino par V_{in} avec les accumulateurs, ni avec un générateur de tension continu 5V. C'était pareil pour les 5 autres cartes Atmega328P que j'ai testées, donc c'était un problème des cartes de ce type. L'alimentation par V_{cc} marchait, mais pour cela il me fallait une source de tension continue 5V que je n'avais pas. Finalement, j'ai décidé d'utiliser un mini powerbank de charge 2200mAh qui alimenterait la carte via un câble micro USB.
- **Commandes sur Internet.** Pour résoudre le problème précédent, j'avais d'autres idées, notamment de commander sur Internet un régulateur 5V ou une carte Arduino Uno (qui pourrait aussi me servir après le projet). J'ai fait la commande sur un site chinois célèbre en début avril, maintenant on est fin mai et je n'ai toujours rien reçu. C'est bien que j'ai pu trouver une autre solution pour alimenter le système.
- **Manque de place sur le support.** La voiture radiocommandée, qui m'a servi de support, n'était pas très grande, et j'ai dû supprimer quelques fonctionnalités initialement prévues au profit des fonctionnalités plus importantes pour le projet. En particulier, j'avais prévu un clignotant et un haut-parleur sur le toit de la voiture. Je les ai fait, mais j'ai dû les enlever pour avoir de la place où accrocher la caméra et le powerbank.
- **Messages audio.** Le son que j'ai obtenu était assez bon, mais il y avait du bruit derrière qui restait après que l'audio était fini. Je pense que c'était à cause d'une mauvaise connexion entre le jack audio et les fils. Je l'ai faite avec de la colle en silicone pour ne pas abîmer le haut-parleur, il valait mieux peut-être utiliser un adaptateur jack audio. Le bruit dans le haut-parleur était la 2^{ème} raison de le supprimer du projet.
- **Delai de réaction.** Au début, le mouvement de la voiture était sous contrôle de la carte Arduino qui recevait des commandes de la carte Wi-Fi via les ports RX TX. Mais il y avait un certain delai de réaction qui n'était pas très agréable. J'ai décidé donc de contrôler le mouvement de la voiture directement par la carte Wi-Fi. Cette décision a conduit à la suppression des capteurs de distance du projet car avec Node Mcu on ne pouvait pas mesurer la distance en temps réel à cause du fait que sa boucle tourne seulement quand on renouvelle la page du serveur.
- **Moteurs.** D'abord, j'alimentais les moteurs par les 5V de la carte Wi-Fi. Mais je me suis aperçu qu'on ne pouvait pas tourner 2 moteurs en même temps, il n'y avait pas assez de puissance. J'ai donc alimenté les moteurs séparément du système avec 4 piles de 1,5V qui étaient dans la compléctation initiale de la voiture.
- **Soudage.** J'avais besoin de faire quelques soudures, et le métal fondu ne voulait pas trop se coller sur les pattes des composants. J'ai résolu ce problème à l'aide du colophane.

7. Retour personnel sur le projet

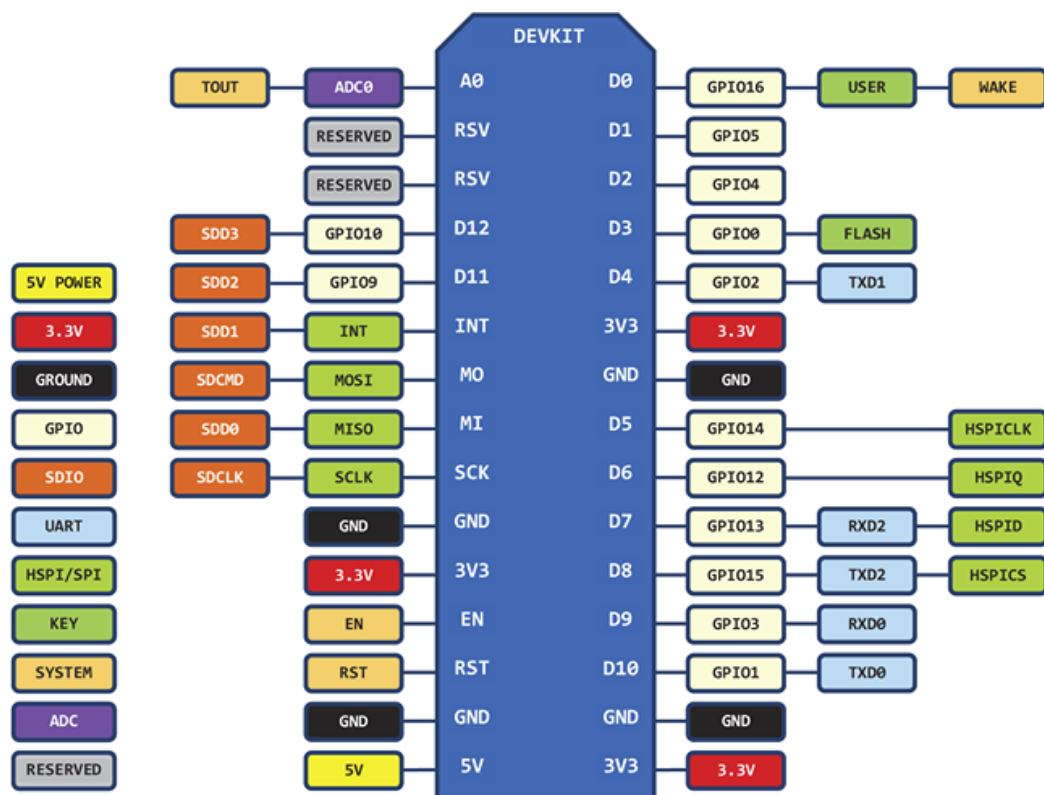
C'était intéressant de réaliser de quelque chose à partir d'un concept, chercher et trouver des solutions. J'étais déçue et stressée par la partie électronique car il y avait trop de choses qui ne marchaient pas comme attendu. La partie algorithmique et programmation était bien.

8. Sources d'inspiration

- **TD d'élec. au S3.** J'ai repris quelques bouts de code, notamment pour les LED et pour les capteurs de distance (qui au final n'ont pas fait partie du projet).
- **Cours d'Introduction Au Web au S3.** Etait utile pour comprendre la communication serveur-client.
- **Cours d'Applications Du Web au S4.** Etait utile pour écrire un script JavaScript qui a permis d'utiliser les touches du clavier pour contrôler la voiture.
- **Internet.**
Un projet similaire, mais pas exactement le même :
<https://www.youtube.com/watch?v=Ck1rouoTh0o>
Commencer à travailler avec Node Mcu :
<https://www.youtube.com/watch?v=G6CqvhXpBKM>
Un projet simple avec Node Mcu :
<https://www.youtube.com/watch?v=ZPUg4Uw3A0E>
Arduino qui parle :
<https://www.youtube.com/watch?v=gi9mqIha8n0>

9. Annexe code

Remarque : sur Node Mcu les numéros des pins écrits sur la carte ne correspondent pas à leur numéros réels. Par exemple, le pin D1 c'est GPIO5, et son numéro réel c'est 5.



D0(GPIO16) can only be used as gpio read/write, no interrupt supported, no pwm/i2c/ow supported.

```

// pour la connexion Wi-Fi et la création d'un serveur
#include <ESP8266WiFi.h>
const char* ssid = "your wifi name"; // nom du Wi-Fi
const char* password = "your password"; // mot de passe
WiFiServer server(80);

// pour les phares
const int led = 5; // D1
boolean lights = false; // indique si les phares sont allumées

// pour le moteur de devant qui fait la voiture tourner ses roues de devant à gauche ou à droite
const int left = 0; // D3
const int right = 2; // D4
// pour le moteur de derrière qui fait la voiture aller en avant ou en arrière
const int forward = 14; // D5
const int backward = 12; // D6

void setup(){
  Serial.begin(115200);
  // initialisation des pins
  pinMode(led,OUTPUT);
  pinMode(left,OUTPUT);
  pinMode(right,OUTPUT);
  pinMode(forward,OUTPUT);
  pinMode(backward,OUTPUT);
  // les phares sont éteintes au début
  digitalWrite(led,HIGH);
  // la voiture ne bouge pas au début
  stopCar();
  // on connecte la carte Wi-Fi au Wi-Fi prédéfini
  WiFi.begin(ssid,password);
  // on attend que la connexion se fait
  while(WiFi.status() != WL_CONNECTED) delay(500);
  Serial.println("WiFi connected");
  // on récupère l'adresse IP de la carte
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
  // on lance le serveur
  server.begin();
}

void loop(){
  // on vérifie si un client s'est connecté
  WiFiClient client = server.available();
  if(!client) return;
  // on attend que le client envoie des données
  while(!client.available()) delay(1);
  // on lit la 1ère ligne de la requête du client
  String request = client.readStringUntil('\r');
  client.flush();

  // on trouve quelle commande a été envoyée (si elle a été envoyée) et on met à jour le système

```



```

if(request.indexOf("cmd=turnOnLights") != -1){ lights = true; digitalWrite(led,LOW); }
else if(request.indexOf("cmd=turnOffLights") != -1){ lights = false; digitalWrite(led,HIGH); }
else if(request.indexOf("cmd=turnLeft") != -1) turnLeft();
else if(request.indexOf("cmd=turnRight") != -1) turnRight();
else if(request.indexOf("cmd=forward") != -1) goForward();
else if(request.indexOf("cmd=backward") != -1) goBackward();
else if(request.indexOf("cmd=stopCar") != -1) stopCar();

// on retourne la réponse du serveur
// l'en-tête
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println(""); // il ne faut pas oublier la ligne vide après la fin de l'en-tête
// la page HTML
client.println("<!DOCTYPE html>");
client.println("<html>");
client.println("<head>");
client.println("<meta charset=\"utf-8\">");
client.println("<title>Curious Car</title>");
// ce script JavaScript reconnaît l'appui sur un bouton et renouvelle la page avec la
requête qui correspond au bouton appuyé
// remarque : le code JavaScript s'exécute dans le navigateur Internet du client
client.println("<script type=\"text/javascript\">");
// cette fonction est appelée lorsque l'un des boutons prédéfinis est pressé
client.println("document.addEventListener(\"keypress\", function onEvent(event){");
client.println("switch(event.key){");
// par exemple, si le bouton pressé est "l", la commande transmise sera de changer
l'état des phares (allumer ou éteindre)
if(lights)
client.println("case \"l\": window.location.replace('?cmd=turnOffLights'); break;");
else
client.println("case \"l\": window.location.replace('?cmd=turnOnLights'); break;");
client.println("case \"q\": window.location.replace('?cmd=turnLeft'); break;");
client.println("case \"d\": window.location.replace('?cmd=turnRight'); break;");
client.println("case \"z\": window.location.replace('?cmd=forward'); break;");
client.println("case \"s\": window.location.replace('?cmd=backward'); break;");
client.println("case \"x\": window.location.replace('?cmd=stopCar'); break;");
client.println("default: return;");
client.println("}");
client.println("});");
client.println("</script>");
// fin du script JavaScript
client.println("</head>");
client.println("<body>");
client.println("<h1>Curious Car</h1>");
client.println("<table style=\"border: 0;\">");
if(lights)
client.println("<tr><td><button style=\"width: 50px;\"");
onclick=\"window.location.replace('?cmd=turnOffLights');\">L</button>Turn OFF
lights</td></tr>");
else

```

```

    client.println("<tr><td><button style=\"width: 50px;\"
onclick=\"window.location.replace('?cmd=turnOnLights');\">L</button>Turn ON
lights</td></tr>");
    client.println("<tr>");
    client.println("<td></td>");
    client.println("<td><button style=\"width: 50px;\"
onclick=\"window.location.replace('?cmd=forward');\">Z</button>Go forward</td>");
    client.println("<td></td>");
    client.println("</tr>");
    client.println("<tr>");
    client.println("<td><button style=\"width: 50px;\"
onclick=\"window.location.replace('?cmd=turnLeft');\">Q</button>Turn left</td>");
    client.println("<td><button style=\"width: 50px;\"
onclick=\"window.location.replace('?cmd=backward');\">S</button>Go backward</td>");
    client.println("<td><button style=\"width: 50px;\"
onclick=\"window.location.replace('?cmd=turnRight');\">D</button>Turn right</td>");
    client.println("</tr>");
    client.println("<tr>");
    client.println("<td></td>");
    client.println("<td><button style=\"width: 50px;\"
onclick=\"window.location.replace('?cmd=stopCar');\">X</button>Stop</td>");
    client.println("<td></td>");
    client.println("</tr>");
    client.println("</table>");
    client.println("</body>");
    client.println("</html>");
}

// fait la voiture tourner à gauche
void turnLeft(){
    // on tourne les roues à gauche (moteur de devant)
    digitalWrite(left,HIGH);
    digitalWrite(right,LOW);
}

// fait la voiture tourner à droite
void turnRight(){
    // on tourne les roues à droite (moteur de devant)
    digitalWrite(left,LOW);
    digitalWrite(right,HIGH);
}

// fait la voiture aller en avant
void goForward(){
    // on va en avant (moteur derrière)
    digitalWrite(forward,HIGH);
    digitalWrite(backward,LOW);
    // on attend 1 seconde
    delay(1000);
    // et on arrête la voiture
    stopCar();
}

```

```

// fait la voiture aller en arrière
void goBackward(){
    // on va en arrière (moteur derrière)
    digitalWrite(forward,LOW);
    digitalWrite(backward,HIGH);
    // on attend 1 seconde
    delay(1000);
    // et on arrête la voiture
    stopCar();
}

// arrête le moteur de devant
void stopFrontMotor(){
    digitalWrite(left,HIGH);
    digitalWrite(right,HIGH);
}

// arrête le moteur de derrière
void stopBackMotor(){
    digitalWrite(forward,HIGH);
    digitalWrite(backward,HIGH);
}

// arrête les 2 moteurs
void stopCar(){
    stopBackMotor();
    stopFrontMotor();
}

```