



Ecole supérieur de l'informatique

Ex. INI (institut nationale de formation en  
informatique)

---

## Rapport de projet SCI

---

Réalisé par

- Chelihi Soufiane
- Boumendjel Mohamed Islam

Groupe : SIL1

Encadré par : Mr Sehad

Année : 2022-2023

## Table des matières

Table des figures .....	2
Introduction .....	3
Description de projet .....	3
Principe de fonctionnement.....	3
Mode automatique.....	3
Mode manuel.....	3
Principe de fonctionnement de contrôleur de moteur.....	3
Principe de fonctionnement de capteur ultrasonique.....	5
Les composants utilisés.....	6
Schéma détaillé du montage.....	7
Interface de contrôle.....	8
Code source .....	8
Photo du résultat final.....	12
Conclusion.....	13
Références .....	13

## Table des figures

Figure 1 contrôleur de moteurs DC modèle L298N.....	4
Figure 2 contrôle de direction des moteurs par le contrôleur L298N.....	4
Figure 3 capteur ultrasonique modèle HC-SR04.....	5
Figure 4 fonctionnement du capteur ultrasonique .....	5
Figure 5 schéma de montage du projet.....	7
Figure 6 interface de contrôle.....	8

## Introduction

L'IoT a le potentiel de transformer de nombreux aspects de notre vie quotidienne, en offrant des solutions intelligentes pour résoudre des problèmes dans des domaines tels que la santé, la sécurité, l'agriculture, l'environnement, l'industrie et bien plus encore. Les objets connectés peuvent être des appareils ménagers, des capteurs, des véhicules, des équipements industriels, des robots, des drones, des vêtements ...etc.

Dans le cadre de notre projet du module SCI, on a pensé à automatiser une des tâches quotidiennes qui est le nettoyage du sol en concevant un aspirateur intelligent contrôlable par WIFI.

## Description de projet

Il s'agit d'un aspirateur intelligent réalisé à base d'arduino. Cet aspirateur automatise le nettoyage de sol. Il fonctionne en deux modes :

- Mode manuel : l'utilisateur peut le diriger manuellement à travers une plateforme accessible par WIFI.
- Mode automatique : l'aspirateur change de direction automatiquement tout en évitant les obstacles.

Le projet est réalisé en utilisant un arduino modèle MKR WIFI 1010 qui est le composant principal du projet qui se charge d'orchestrer tous les autres composants notamment : les moteurs de mouvement et le capteur ultrasonique.

## Principe de fonctionnement

### Mode automatique

En mode automatique, l'arduino calcule la distance qui le sépare de plus proche obstacles. Si la distance est inférieure à un certain seuil (30cm), l'aspirateur s'arrête dans sa place puis change de direction.

Si ce n'est pas le cas, l'aspirateur va avancer l'avant tout en faisant fonctionner le moteur d'aspiration qui fait absorber la poussière dans un petit boîtier.

### Mode manuel

En mode manuel, l'utilisateur doit accéder à l'interface de contrôle de l'aspirateur qui est accessible en WIFI par téléphone ou par ordinateur en connectant au réseau WIFI généré par l'aspirateur nommé « Arduino » puis ouvrir son navigateur web et taper l'URL <http://192.168.4.1>.

L'interface de contrôle permet de faire avancer l'aspirateur, tourner à gauche, tourner à droite, reculer ou s'arrêter en place. Elle permet aussi de basculer entre le mode automatique ou manuel.

### Principe de fonctionnement de contrôleur de moteur

Pour contrôler les moteurs DC, on a utilisé le contrôleur de moteur modèle L298N. Il permet de contrôler les directions de deux moteurs DC ainsi que modifier leurs vitesses de rotation. L'image suivante représente ce contrôleur :

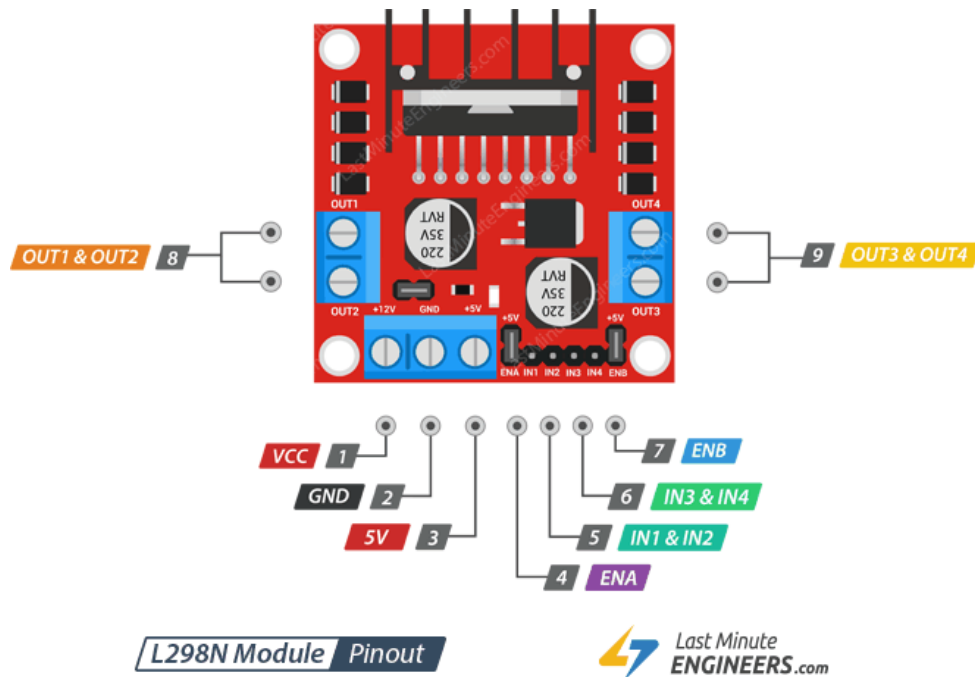


Figure 1 contrôleur de moteurs DC modèle L298N

1. Vcc : cette pin doit être connectée à la batterie. Elle accepte jusqu'à 35V et 2A.
2. GND : la masse
3. 5V : c'est une pin de sortie qui délivre un voltage de 5V qu'on utilise pour alimenter l'arduino
4. ENA : pin de contrôle de vitesse de rotation du moteur gauche
5. IN1 et IN2 : les pins de contrôle de direction du moteur gauche
6. IN3 et IN4 : les pins de contrôle de direction du moteur droite
7. ENB : pin de contrôle de vitesse de rotation du moteur droite
8. OUT1 et OUT2 : les pins d'alimentation du moteur gauche
9. OUT3 et OUT4 : les pins d'alimentation du moteur droite

Les pins ENA et ENB doivent être connectées à des pins PWM dans l'arduino afin de contrôler la vitesse de rotation des moteurs.

Le contrôle de direction est expliqué dans le tableau suivant :

IN1	IN2	Etat de moteur
LOW (0)	LOW (0)	Moteur OFF
LOW (0)	HIGH (1)	Moteur tourne aux sens inverse des aiguilles de montre
HIGH (1)	LOW (0)	Moteur tourne aux sens des aiguilles de montre
HIGH (1)	HIGH (1)	Moteur OFF

Figure 2 contrôle de direction des moteurs par le contrôleur L298N

Le contrôle de direction du deuxième moteur se fait de la même façon avec les pins IN3 et IN4

## Principe de fonctionnement de capteur ultrasonique

Dans le mode automatique, on utilise un capteur ultrasonique modèle HC-SR04 pour calculer la distance devant l'aspirateur et l'obstacle le plus proche afin de le faire éviter les obstacles automatiquement.

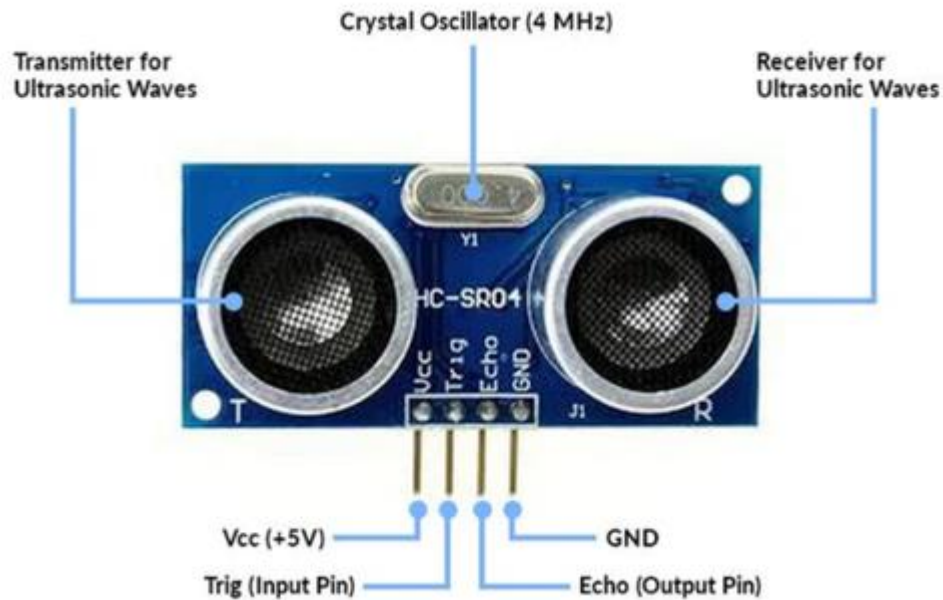


Figure 3 capteur ultrasonique modèle HC-SR04

- Vcc : pin d'alimentation
- GND : pin de la masse
- Trig : pin entrée. Quand elle est dans l'état HIGH, le transmetteur envoie des ondes ultrasoniques
- Echo : pin sortie. Elle délivre un état à HIGH quand le récepteur reçoit des ondes ultrasoniques

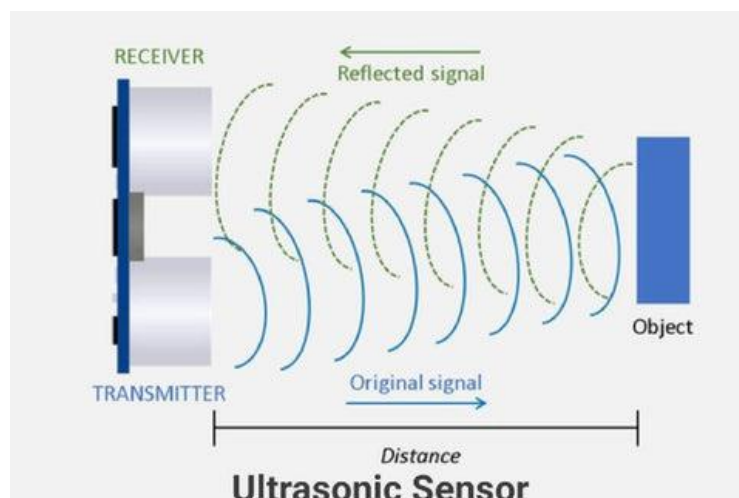


Figure 4 fonctionnement du capteur ultrasonique

Quand la pin Trig est à l'état HIGH, le transmetteur envoie des ondes ultrasoniques qui vont traverser l'espace jusqu'à ce que ces ondes soient reflétées par un objet. Quand le récepteur reçoit ces ondes reflétées, il met la pin Echo à l'état HIGH.

On sait que la formule de calcul de distance est la suivante :

$$d = V \times t$$

Tel que :

- d : distance
- V : vitesse du son
- t : temps nécessaire pour traverser la distance d

On considère la vitesse de son à 20°C :  $V = 343 \text{ m/s} = 0.034 \text{ cm /}\mu\text{s}$

Le temps « t » peut être calculé en mesurant le temps pendant lequel la pin Echo est à l'état HIGH

La distance « d » est égale à la distance traversée par les ondes d'aller et de retour. Donc, pour calculer la distance entre le capteur et un objet, on doit diviser « d » par 2.

## Les composants utilisés

composant	quantité
Arduino MKR WIFI 1010	1
Châssis de voiture	1
Moteurs DC	3
Roues	2
Contrôleur de moteur modèle L298N	1
Capteur ultrasonique modèle HC-SR04	1
Fils de connexion	Selon besoins
Breadboard	1
Batteries 9V	3
Batteries 1.5V	4
Power bank	1

## Schéma détaillé du montage

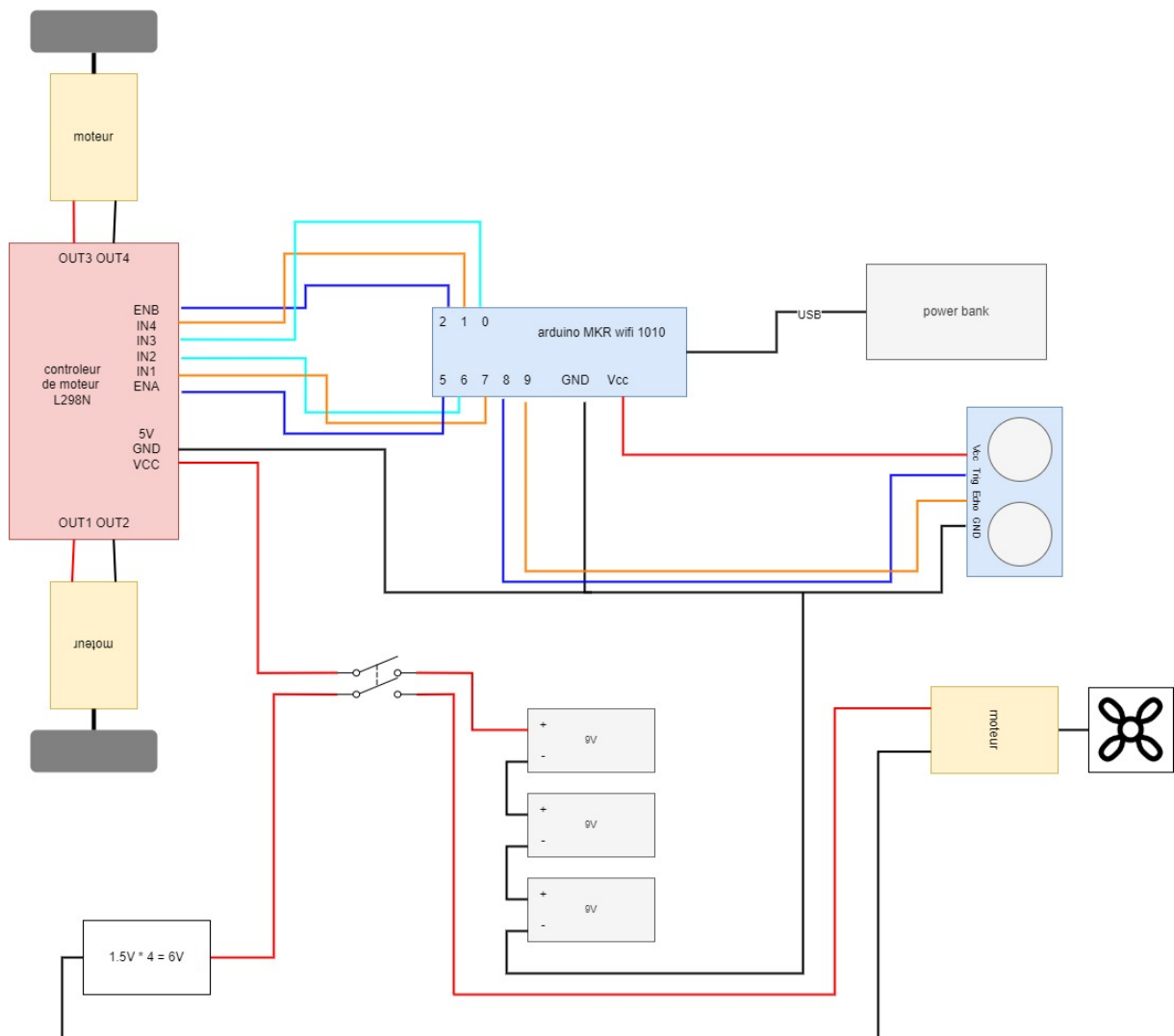


Figure 5 schéma de montage du projet

## Interface de contrôle

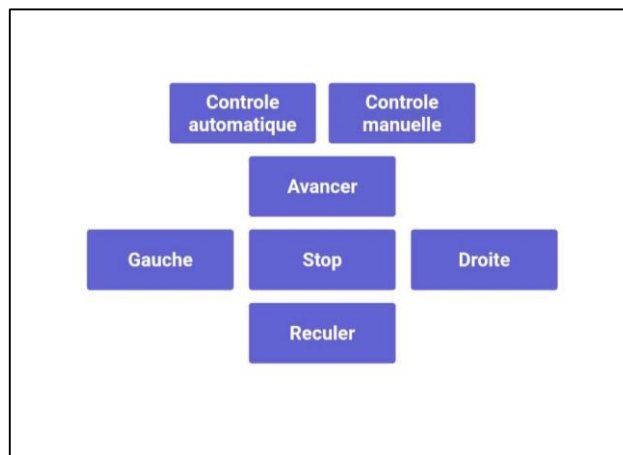


Figure 6 interface de contrôle

## Code source

```
#include <WiFiNINA.h>

int motor11 = 0; // moteur droite
int motor12 = 1;
int motor1pmw = 2;

int motor21 = 6; //moteur gauche
int motor22 = 7;
int motor2pmw = 5;

int trig = 9; //trig pour capteur ultrasonic
int echo = 8; //echo pour capteur ultrasonic
int vitess = 60; // vitesse de rotation des moteurs

int autom = 0; // si il fontionne en mode automatique

char ssid[] = "Arduino"; // nom du réseau wifi
char pass[] = "12345678"; // mot de passe du réseau
int keyIndex = 0;

WiFiServer server(80); // serveur

//faire avancer l'aspirateur
void avancer() {
  digitalWrite(motor11, LOW);
  digitalWrite(motor12, HIGH);
  analogWrite(motor1pmw, vitess);

  digitalWrite(motor21, HIGH);
  digitalWrite(motor22, LOW);
  analogWrite(motor2pmw, vitess);
}

//faire reculer l'aspirateur
void reculer() {
  digitalWrite(motor11, HIGH);
  digitalWrite(motor12, LOW);
  analogWrite(motor1pmw, vitess);

  digitalWrite(motor21, LOW);
```



```

digitalWrite(motor22, HIGH);
analogWrite(motor2pmw, vitess);
}

//tourner l'aspirateur à gauche
void tourner_gauche() {
digitalWrite(motor11, LOW);
digitalWrite(motor12, HIGH);
analogWrite(motor1pmw, vitess+30);

digitalWrite(motor21, LOW);
digitalWrite(motor22, HIGH);
analogWrite(motor2pmw, vitess+30);
}

//arreter l'aspirateur dans sa place
void stop(){
digitalWrite(motor11, LOW);
digitalWrite(motor12, LOW);
digitalWrite(motor21, LOW);
digitalWrite(motor22, LOW);
}

//tourner l'aspirateur à gauche

void tourner_droite() {
digitalWrite(motor11, HIGH);
digitalWrite(motor12, LOW);
analogWrite(motor1pmw, vitess+30);

digitalWrite(motor21, HIGH);
digitalWrite(motor22, LOW);
analogWrite(motor2pmw, vitess+30);
}

//calculer la distance via le capteur ultrasonique
//formule de calcul expliquée dans le rapport
int calculer_distance() {
digitalWrite(trig, LOW);
delay(2);
digitalWrite(trig, HIGH);
delay(10);
digitalWrite(trig, LOW);
long temps = pulseIn(echo, HIGH);
int cm = temps / 29 / 2;
Serial.println(cm);
return cm;
}

void setup() {
Serial.begin(9600);
pinMode(motor11, OUTPUT);
pinMode(motor12, OUTPUT);
pinMode(motor1pmw, OUTPUT);

pinMode(motor21, OUTPUT);
pinMode(motor22, OUTPUT);
pinMode(motor2pmw, OUTPUT);

pinMode(trig, OUTPUT);
pinMode(echo, INPUT);

Serial.println("Access Point Web Server");

```

```

//verifier l'etat du wifi
if (WiFi.status() == WL_NO_MODULE) {
    Serial.println("Communication with WiFi module failed!");
    // don't continue
    while (true);
}

//verifier la version du firmware
String fv = WiFi.firmwareVersion();
if (fv < WIFI_FIRMWARE_LATEST_VERSION) {
    Serial.println("Please upgrade the firmware");
}

Serial.print("Creating access point named: ");
Serial.println(ssid);

// Créer le point d'access
status = WiFi.beginAP(ssid, pass);
if (status != WL_AP_LISTENING) {
    Serial.println("Creating access point failed");
    while (true);
}

delay(10000);

// demarrer le serveur
server.begin();

// afficher les informations de wifi
printWiFiStatus();
}

void loop() {
    //verifier l'etat de wifi
    if (status != WiFi.status()) {
        status = WiFi.status();

        if (status == WL_AP_CONNECTED) {
            // si l'utilisateur se connecte au wifi
            Serial.println("Device connected to AP");
        } else {
            // s'il se deconnecte
            Serial.println("Device disconnected from AP");
        }
    }

    WiFiClient client = server.available();

    if (client) {
        Serial.println("new client");
        String currentLine = "";
        while (client.connected()) { //tant que l'utilisateur est connecté
            if (client.available()) {
                char c = client.read(); //lire un caractère
                Serial.write(c);
                if (c == '\n') {

                    if (currentLine.length() == 0) {

                        client.println("HTTP/1.1 200 OK");
                        client.println("Content-type:text/html");
                        client.println();

```

```

        //afficher le code html de l'interface
        client.print("<!DOCTYPE html><html><head><title>Controle aspirateur automatique
</title></head><style>body{display: flex; justify-content: center; align-items: center;
flex-direction: column; height: 100vh;}button{width: 230px; height: 95px; border-radius:
5px; color: white; background-color: rgb(97, 97, 209); padding: 6px; font-weight: 700;
font-size: 30px; border: none; margin: 10px; cursor: pointer;}</style><body> <div><button
onmousedown='auto()'>Controle automatique</button><button onmousedown='man()'>Controle
manuelle</button></div><button onmousedown='avancer()'>Avancer</button> <div><button
onmousedown='gauche()'>Gauche</button> <button onmousedown='stop()'>Stop</button> <button
onmousedown='droite()'>Droite</button></div><button
onmousedown='reculer()'>Reculer</button> <script>function avancer(){fetch('/A')}function
gauche(){fetch('/G')}function droite(){fetch('/D')}function reculer(){fetch('/R')}function
stop(){fetch('/S')}function auto(){fetch('/T')}function
man(){fetch('/M')}</script></body></html>");
        client.println();

        break;
    }
    else {
        currentLine = "";
    }
}
else if (c != '\r') {
    currentLine += c;
}

if (currentLine.endsWith("GET /A")) {
    avancer();
}
if (currentLine.endsWith("GET /G")) {
    tourner_gauche() ;
    delay(490);
    stop();
}
if (currentLine.endsWith("GET /D")) {
    tourner_droite();
    delay(490);
    stop();
}
if (currentLine.endsWith("GET /R")) {
    reculer() ;
}
if (currentLine.endsWith("GET /S")) {
    stop();
}

if (currentLine.endsWith("GET /T")) {
    autom = 1;
}
if (currentLine.endsWith("GET /M")) {
    autom=0;
    stop();
}

}
}
// quand l'utilisateur se deconnecte
client.stop();
Serial.println("client disconnected");
}

//en mode automatique
if(autom==1){

    int distance = calculer_distance();

```

```

        if(distance <= 30) { //si il y a obstacle
            stop();
            delay(500);
            tourner_gauche();
            delay(400);
        } else { //si le chemin est libre
            avancer();
        }
    }
}

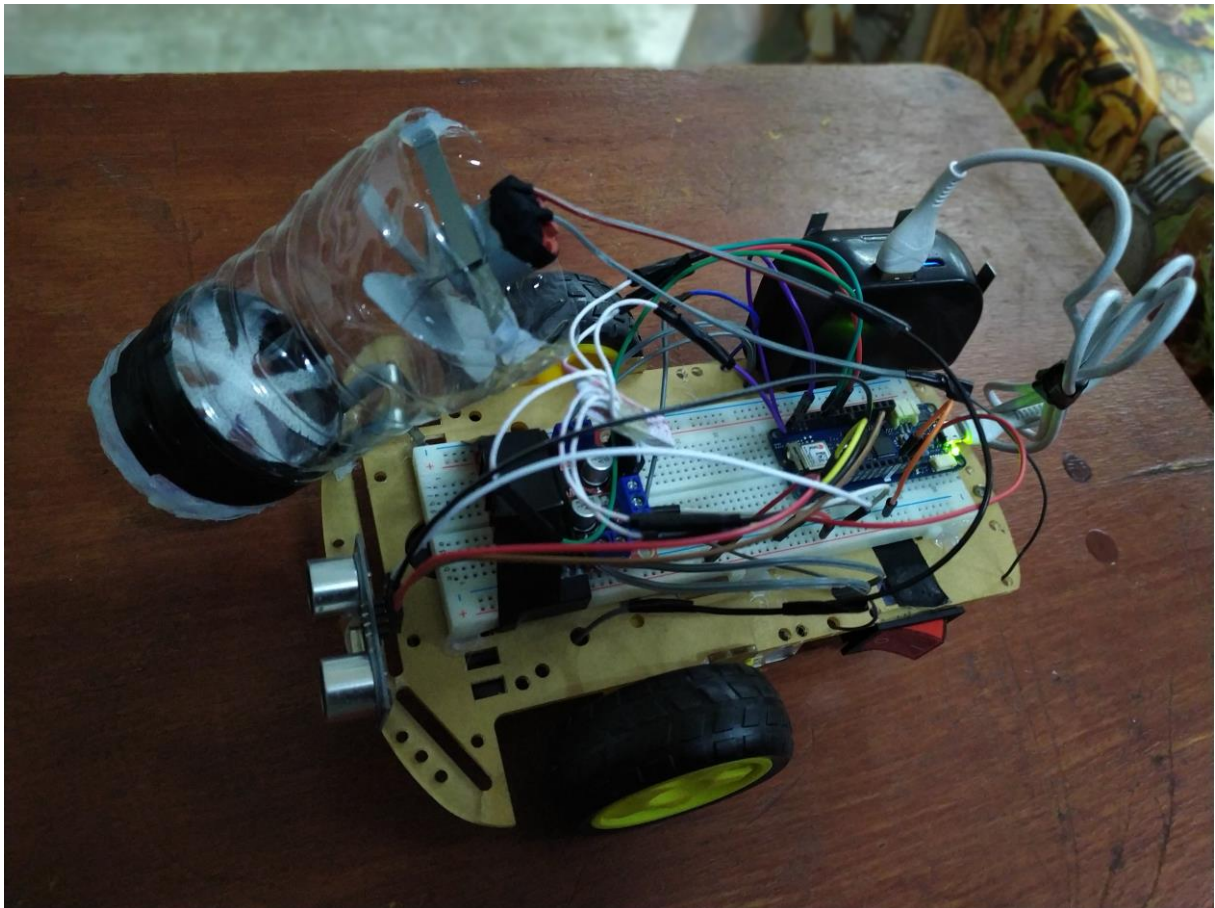
//afficher les informations du wifi
void printWiFiStatus() {
    Serial.print("SSID: ");
    Serial.println(WiFi.SSID());

    IPAddress ip = WiFi.localIP();
    Serial.print("IP Address: ");
    Serial.println(ip);

    Serial.print("To see this page in action, open a browser to http://");
    Serial.println(ip);
}

```

Photo du résultat final



## Conclusion

En conclusion, le développement d'un aspirateur intelligent en IoT est un projet passionnant qui peut apporter de nombreux avantages. Grâce à cette technologie, l'aspirateur peut être contrôlé à distance, automatisé et doté de nombreuses fonctionnalités intelligentes telles que la détection des obstacles et l'adaptation à l'environnement. Cela permettrait aux utilisateurs de gagner du temps, d'améliorer l'efficacité et de réduire les efforts nécessaires pour nettoyer leur maison. En résumé, le développement d'un aspirateur intelligent en IoT peut être bénéfique pour les utilisateurs et l'environnement, et peut ouvrir la voie à de nouvelles innovations dans le domaine de la maison intelligente.

Pour ce projet, on a des perspectives qui peuvent l'améliorer tel que la déclenchement automatique et le chargement automatique de la batterie ainsi que la détection des déchets par une caméra spéciale et un algorithme Computer vision et l'intégration avec les plateformes existant tel que Alexa et Google Home.

## Références

- [In-Depth: Interface L298N DC Motor Driver Module with Arduino \(lastminut  
https://www.volta.ma/comment-utiliser-un-capteur-de-distance-a-ultrasons-hc-sr04-avec-arduino/arduino/#:~:text=Comment%20fonctionne%20le%20HC%20DSR04&text=Le%20capteur%20cr%C3%A9e%20alors%20une,sonores%20ont%20parcouru%20en%20microsecondes.eengineers.com\)](https://www.volta.ma/comment-utiliser-un-capteur-de-distance-a-ultrasons-hc-sr04-avec-arduino/arduino/#:~:text=Comment%20fonctionne%20le%20HC%20DSR04&text=Le%20capteur%20cr%C3%A9e%20alors%20une,sonores%20ont%20parcouru%20en%20microsecondes.eengineers.com)
- <https://www.volta.ma/comment-utiliser-un-capteur-de-distance-a-ultrasons-hc-sr04-avec-arduino/arduino/#:~:text=Comment%20fonctionne%20le%20HC%20DSR04&text=Le%20capteur%20cr%C3%A9e%20alors%20une,sonores%20ont%20parcouru%20en%20microsecondes>