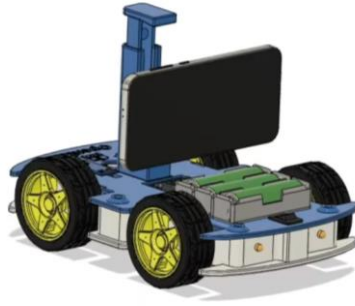
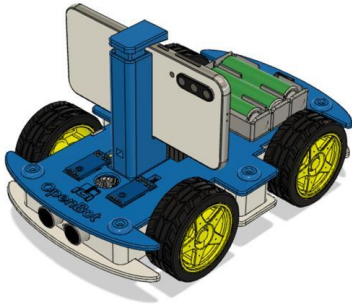


OpenBot



UNIVERSITÉ DE TECHNOLOGIE DE BELFORT-MONTBÉLIARD

Conception / réalisation d'un robot suiveur connecté smartphone (open source)

Rapport TX 51

A2020

Wang Jie
Li Xingchen

Département Génie mécanique et conception

Tuteur en UV

Frederic Demoly

Introduction

General

Dans ce projet TX51, nous avons choisi le projet de réaliser un robot suiveur connecté smartphone mené par M. Demoly. Le créateur de ce projet est un chercheur d'Intel Corporation, ce projet est un projet open source, c'est-à-dire que le chercheur partage les ressources du projet avec tous les participants. L'intention initiale du projet est de fabriquer une petite voiture pouvant être utilisée avec un smartphone à faible coût. Les chercheurs espèrent que les participants pourront améliorer leurs robots afin de continuer à développer de nouvelles applications pour eux à l'avenir.

Après avoir étudié attentivement ces dossiers de projet avec notre professeur, nous avons déterminé notre direction de recherche et développement. La direction de recherche de notre projet est de réaliser le contrôle à distance de cette voiture. Notre inspiration est venue principalement d'une série télévisée américaine, et nous avons examiné de manière globale la situation mondiale actuelle. 2020 est une année difficile en raison de l'épidémie qui fait rage. Nous avons vu de nombreuses personnes incapables de rentrer chez elles pendant cette période difficile pour diverses raisons. Nous voulons donc développer une voiture qui peut être contrôlée à distance, et installer un téléphone portable sur cette voiture. Avec cette condition, nous pouvons discuter en vidéo avec notre famille ou amis en contrôlant la position de la perspective du chat vidéo. Notre projet est principalement divisé en deux parties, la première est la modification et l'optimisation de modules 3D, et la seconde est principalement notre programme.

Source d'inspiration

L'idée vient d'une série, le protagoniste n'a pas pu aller à une fête famille à cause d'un voyage d'affaires, il est très créatif, et il a créé une machine qui lui permet de participer à distance à la fête, ce qu'il a besoins de faire, c'est de contrôler à distance, et faire l'appel vidéo. Comparé avec la situation actuelle, sous l'épidémie Coronavirus, cela sera beaucoup demandé pour les personnes qui ne peuvent pas visiter leurs familles. Donc, nous avons proposé cette idée.

Sommaire

Introduction1

 General 1

 Source d'inspiration..... 1

La première partie3

 Corps de la voiture3

La deuxième partie7

 Partie programme et assemblage..... 7

 Montage 9

 Les idées de design 11

 L’interprétation du programme HTML..... 13

 Code de Arduino..... 17

 Code de commande du moteur L298 :17

 Partie 1 : Variables18

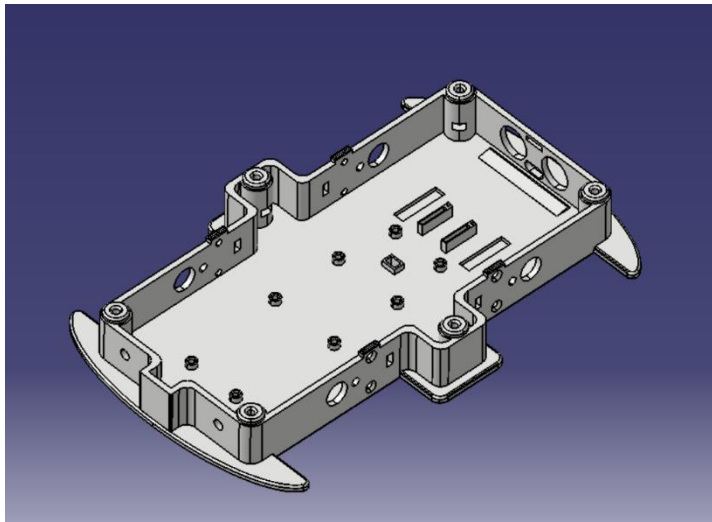
 Partie 2 : code principal18

Annexe: La fonction d’option20

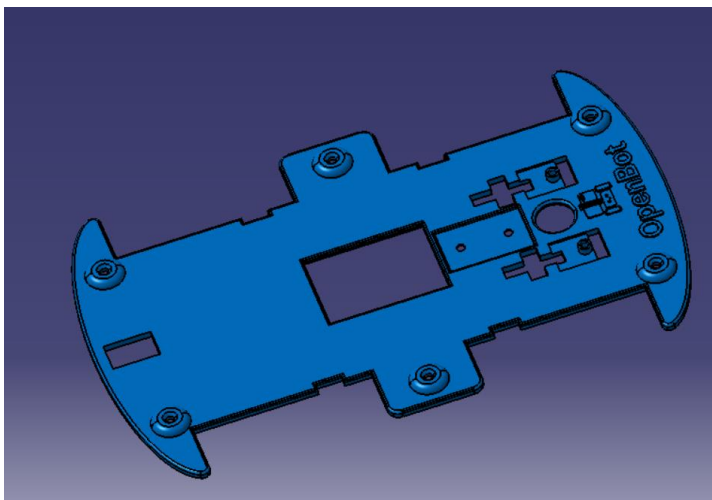
La première partie

Corps de la voiture

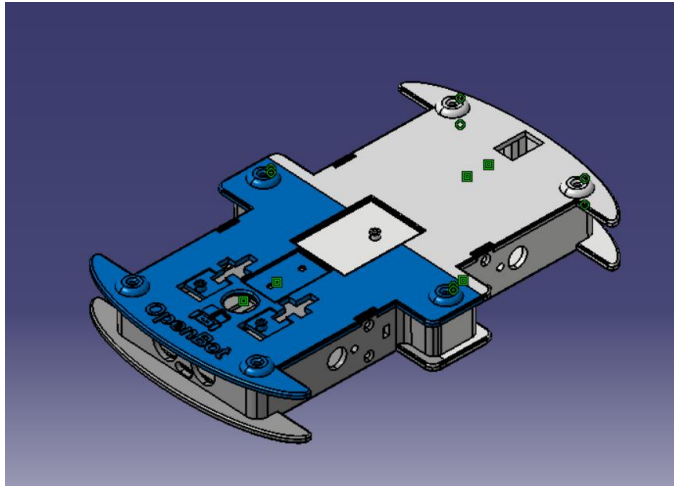
Le chercheur a bien conçu la taille et la forme de la voiture afin de réserver des positions appropriées pour l'Arduino, le moteur, les roues, etc. Dans notre projet, nous n'avons pas changé les composants électroniques et les pièces à utiliser, nous n'avons donc pas besoin de modifier la partie de la voiture.



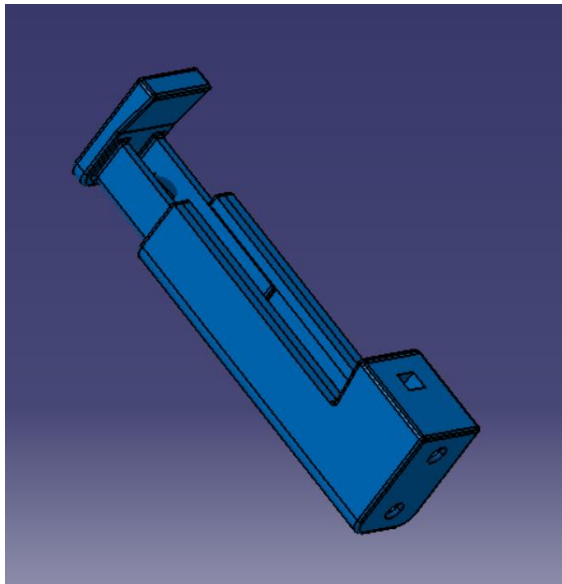
Corps de support



Couverture

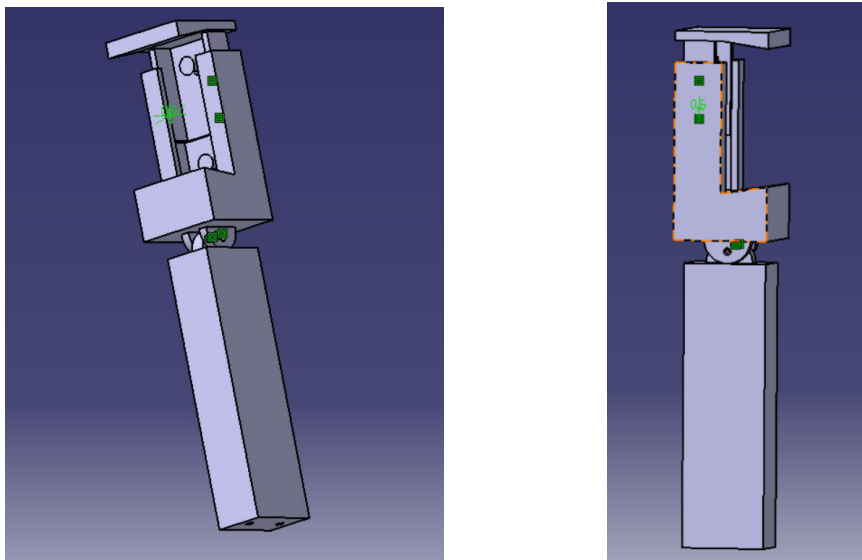


Vue d'assemblage de corps de voiture

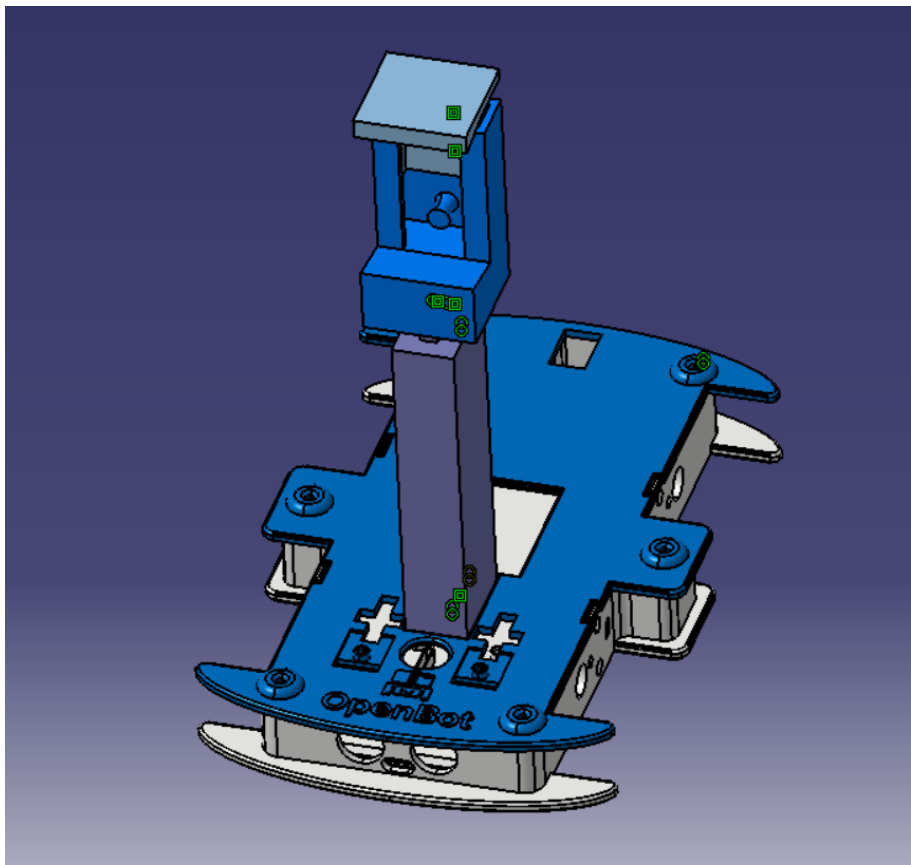


Pièce de support de téléphone portable

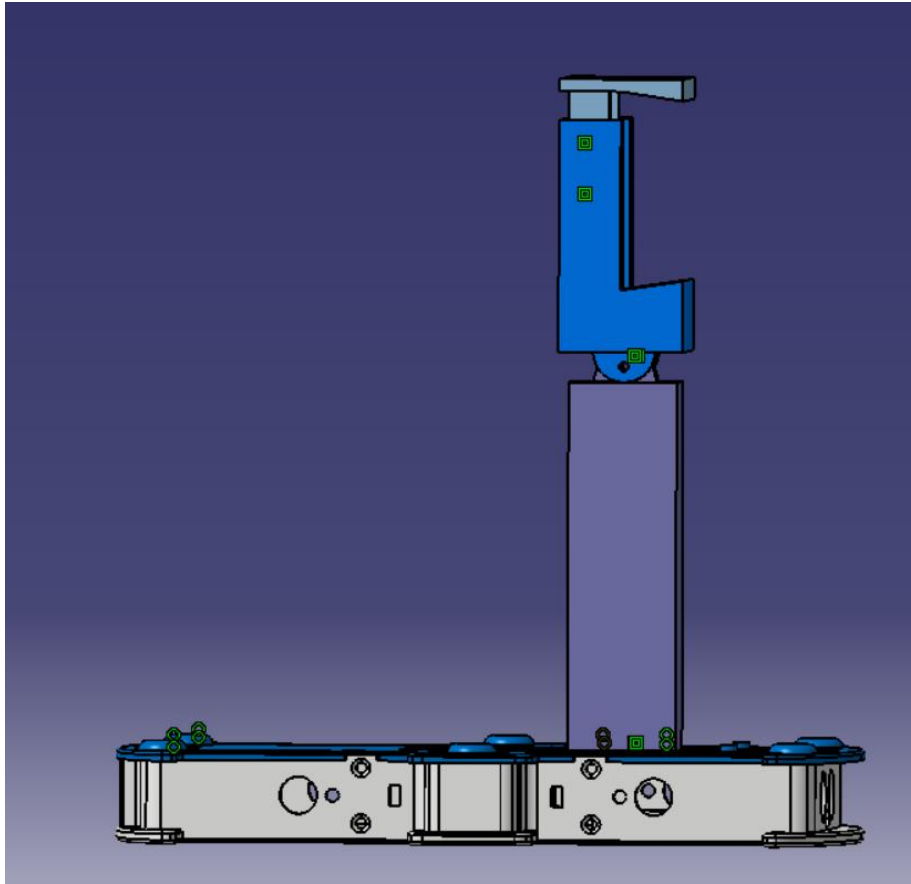
Afin d'être plus en phase avec l'intention initiale de notre projet, nous avons repensé un nouveau support de téléphone portable. Si nous ne comptons que sur le support de téléphone portable d'origine, lorsque nous sommes dans un chat vidéo, notre caméra ne peut prendre que des photos du sol. Notre modification a donc augmenté la hauteur du téléphone et ajusté certains angles d'objectif. Mais se fondant sur les matériaux existants nous ne pouvons pas faire le support trop élevé pour l'instant, parce que nous devons considérer le centre de gravité et de l'équilibre.



Entre les deux parties, nous pouvons d'abord
ajuster l'angle, puis le fixer avec une vis.



Vue d'assemblage 1



Vue d'assemblage 2

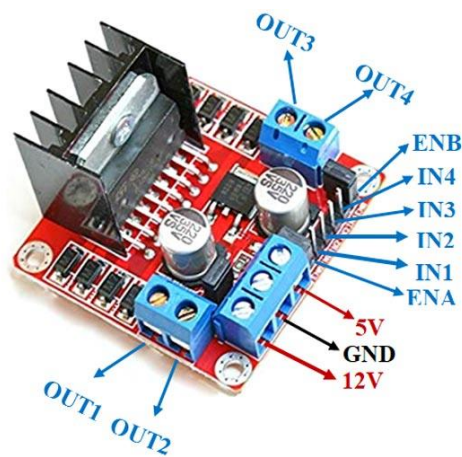
La deuxième partie

Partie programme et assemblage

Tout d'abord, nous devons vous présenter les matériaux et composants électroniques que nous utilisons actuellement pour réaliser cette fonction.

La liste:

L298N Motor Driver



TT motors with tires



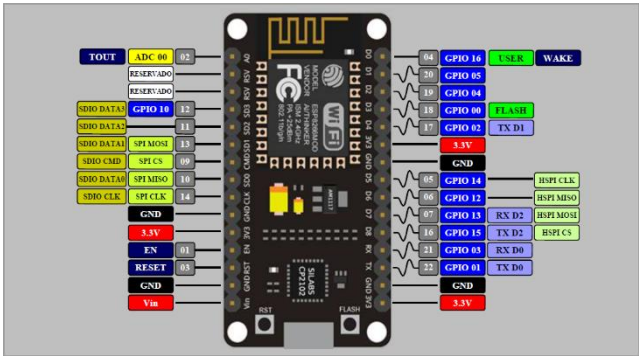
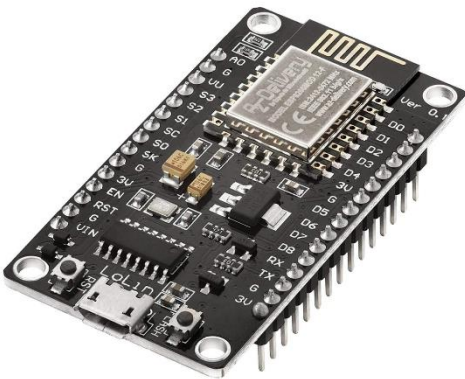
18650 battery



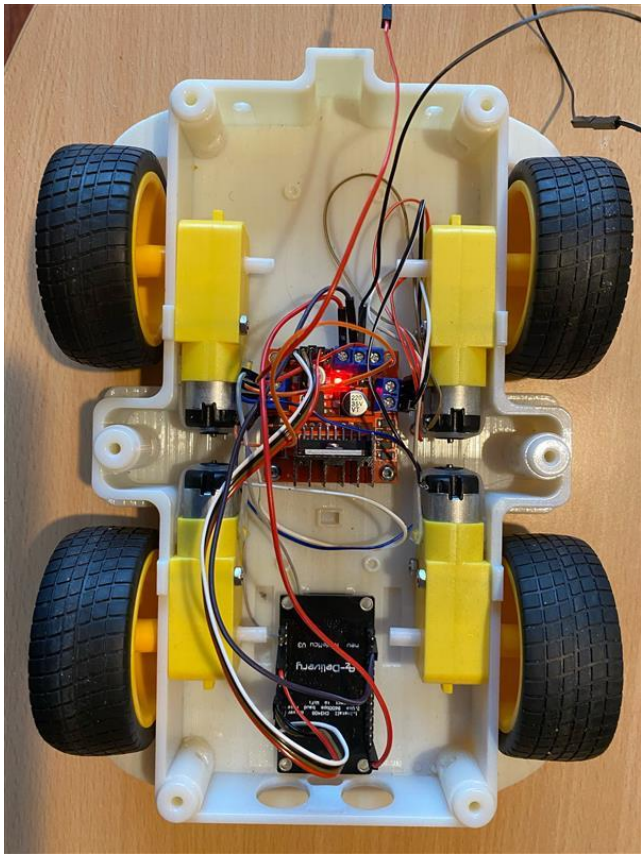
18650 battery holder



esp8266 nodemcu



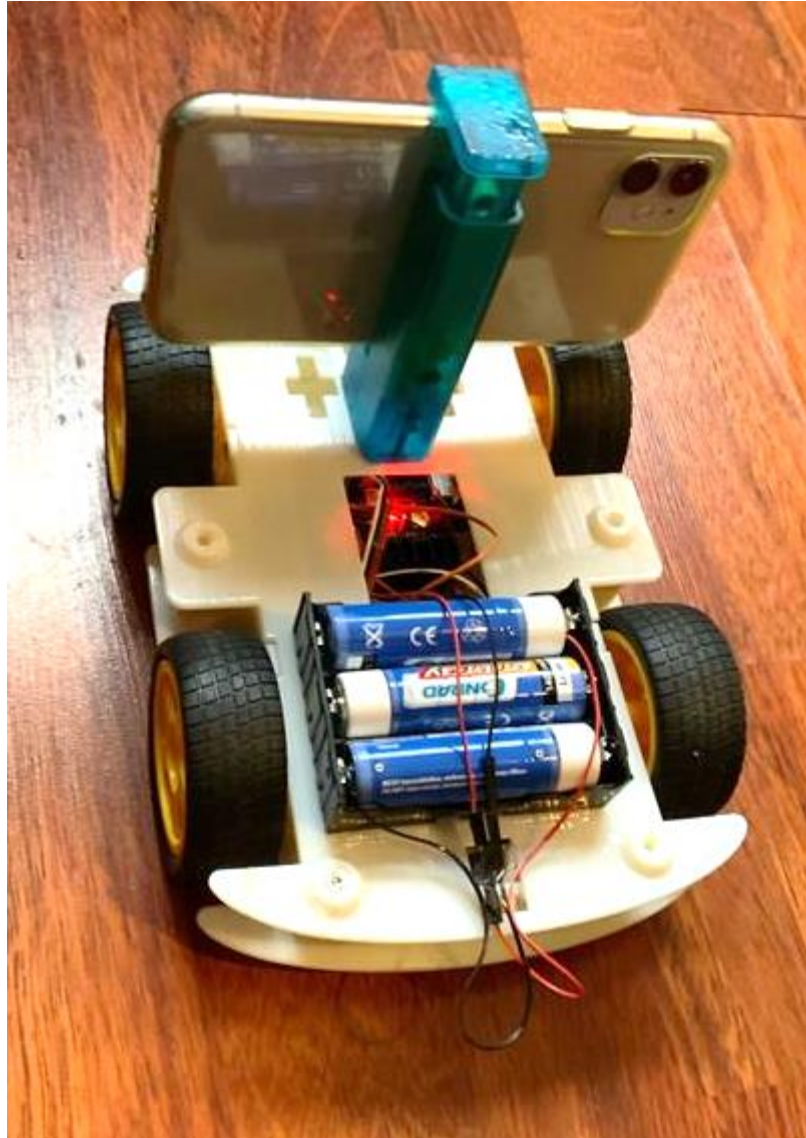
Montage



Montage 1



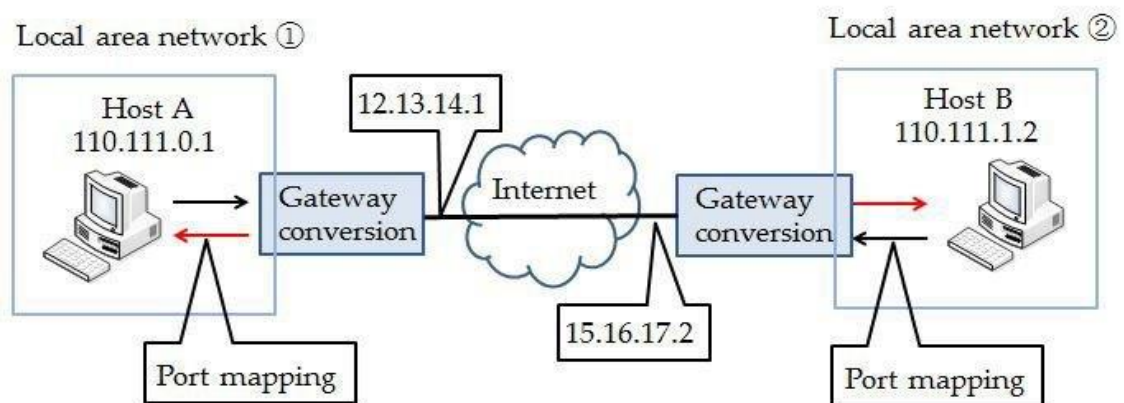
Montage 2



Montage 3

Les idées de design

1. Nous utilisons un module wifi (esp8266 nodemcu), donc nous pouvons directement connecter cette voiture à un wifi, mais cette connexion se présente sous la forme d'un accès au réseau local LAN (Local Area Network).
Pour le moment, il suffit de compléter le code de contrôle de la partie moteur L298 pour contrôler la rotation ou l'arrêt de ces moteurs sur une courte distance, c'est-à-dire sous le réseau local (sous le même wifi).
2. Mais si l'on veut réaliser un contrôle à distance, c'est évidemment impossible que sous le réseau local. Pour parvenir à cette fonction, nous devons utiliser la méthode de pénétration intranet.



La pénétration intranet est un terme utilisé lorsque nous nous connectons au réseau, également appelé pénétration NAT. Lorsque l'ordinateur est dans un réseau local, la connexion et la communication requises lorsque les nœuds du réseau externe et du réseau interne sont connectés, parfois le réseau interne ne peut pas être pénétré.

La fonction de la pénétration intranet est que lorsque nous la définissons lors du mappage de port, la pénétration intranet a la fonction de traduction d'adresse. C'est-à-dire traduire l'adresse du réseau public en une adresse privée. Ensuite, il utilise le routeur haut débit ADSL (Asymmetric Digital Subscriber Line) pour générer une adresse IP publique dynamique ou fixe. Enfin, l'ADSL est directement mappé sur le commutateur, afin que tous les ordinateurs puissent partager Internet.

La pénétration de l'intranet peut être obtenue via des ports tiers ouverts. Ici, nous utilisons un port de service tiers d'une société chinoise. Nous connectons l'adresse IP et le port du module wifi au port du logiciel de pénétration intranet, puis achetons un nom de domaine réseau que nous utilisons, afin de pouvoir connecter notre module wifi au réseau public.

3. Une fois que les problèmes de connexion ci-dessus sont résolus, nous devons bien sûr également examiner les moyens de contrôler. Autrement dit, l'interface de la console. Au début du projet, nous voulions à l'origine développer une application Android pour réaliser ces contrôles. Mais après une réflexion répétée, nous avons constaté que les applications

Android ont de nombreuses limitations et ne sont pas très pratiques à utiliser. Afin d'obtenir un effet utilisable multi-plateforme, nous avons décidé de développer un HTML comme l'interface de contrôle. Enfin, lorsque la partie programme HTML est terminée, nous intégrons directement cette partie du code dans le programme de contrôle, afin que ces signaux soient mappés ensemble sur le réseau public.

L'interprétation du programme HTML

Tout d'abord, afin de concevoir une interface de contrôle, nous devons créer un HTML, HTML s'appelle Hypertext Markup Language, qui est un langage de balisage. Il comprend une série de balises. Grâce à ces balises, le format du document sur le réseau peut être unifié et les ressources Internet dispersées peuvent être connectées comme un tout logique. Le texte HTML est un texte descriptif composé de commandes HTML. Les commandes HTML peuvent décrire du texte, des graphiques, des animations, des sons, des tableaux, des liens, etc. L'interprète de base de ces codes est basé sur le langage C.

```
<html>
<head>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="https://unpkg.com/element-ui/lib/theme-chalk/index.css">
</head>
<body>
  <div id="app" ref="body" style="display: none; height: 100%;">

    <script type="text/javascript">
      function buttonclickup() {
        window.location = " / up ";
      }

      function buttonclickleft() {
        window.location = " / left ";
      }

      function buttonclickstop() {
        window.location = " / stop ";
      }

      function buttonclickdown() {
        window.location = " / down ";
      }

      function buttonclickright() {
        window.location = " / right ";
      }
    </script>
```

Nous devons nous assurer que notre HTML est basé sur le format UTF-8 (Universal Character Set/Unicode Transformation Format), ensuite, nous avons introduit un CSS (Cascading Style Sheets), qui sert de bibliothèque de performances de style HTML. Puis nous définissons notre page principale.

Dans un premier temps, nous avons mis en place cinq fonctions. Le programme ici signifie que tant que nous cliquons sur ces boutons, ces fonctions connecteront la page Web à différentes commandes, et réaliseront le saut ou l'actualisation de la page Web.

```
<div class="left">
  <div class="leftLight">
    <div class="bubble" ref="left"></div>
  </div>
  <el-button class="leftBtn" @click="leftClick()" onclick="buttonclickleft()" />LEFT</el-button>
</div>

<div class="middle">
  <div class="up">
    <div class="bubble tc" ref="up"><el-button class="middleUpBtn tc" @click="middleUpClick()" onclick="buttonclickup()" />UP</el-button></div>
  </div>

  <div class="stop">
    <div class="bubble tc" ref="stop"><el-button class="middleStopBtn tc" @click="middleStopClick()" onclick="buttonclickstop()" />STOP</el-button></div>
  </div>

  <div class="down">
    <el-button class="middleDownBtn tc" @click="middleDownClick()" onclick="buttonclickdown()" />DOWN</el-button>
    <div class="bubble tc" ref="down"></div>
  </div>
</div>

<div class="right">
  <div class="rightLight">
    <div class="bubble ac" ref="right"></div>
  </div>
  <el-button class="rightBtn" @click="rightClick()" onclick="buttonclickright()" />RIGHT</el-button>
</div>

<el-dialog :visible.sync="visible" title="Hello world">
</el-dialog>
</div>
```

Dans ce code, nous définissons les paramètres, les catégories, les fonctions et les styles des cinq boutons que nous devons utiliser.

```
<script src="https://unpkg.com/vue/dist/vue.js"></script>
<script src="https://unpkg.com/element-ui/lib/index.js"></script>
```

Ici, nous avons également cité deux fichiers js comme environnement de compilation et bibliothèque de langage de script pour notre page HTML.

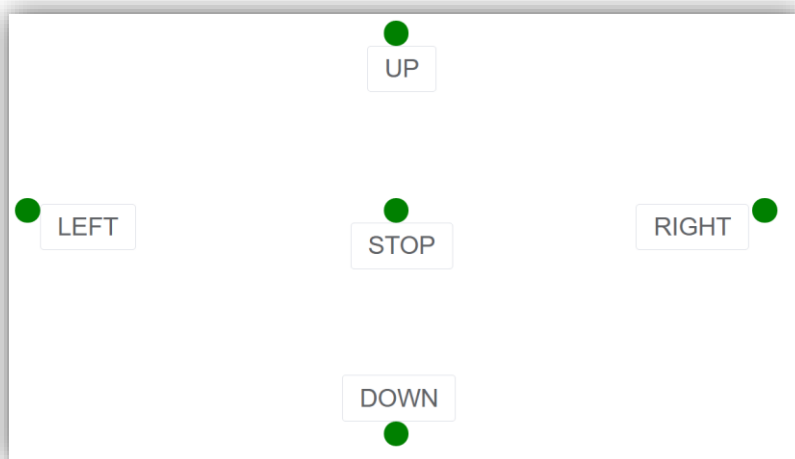
```
new Vue({
  el: '#app',
  data: function () {
    return {
      visible: false,
      position: ''
    }
  },
  methods: {
    leftClick() {
      this.$refs.left.style.cssText = 'background-color:red;'
      this.$refs.up.style.cssText = 'background-color:green;'
      this.$refs.stop.style.cssText = 'background-color:green;'
      this.$refs.down.style.cssText = 'background-color:green;'
      this.$refs.right.style.cssText = 'background-color:green;'
    },
    middleUpClick() {
      this.$refs.left.style.cssText = 'background-color:green;'
      this.$refs.up.style.cssText = 'background-color:red;'
      this.$refs.stop.style.cssText = 'background-color:green;'
      this.$refs.down.style.cssText = 'background-color:green;'
      this.$refs.right.style.cssText = 'background-color:green;'
    },
    middleStopClick() {
      this.$refs.left.style.cssText = 'background-color:green;'
      this.$refs.up.style.cssText = 'background-color:green;'
      this.$refs.stop.style.cssText = 'background-color:red;'
      this.$refs.down.style.cssText = 'background-color:green;'
      this.$refs.right.style.cssText = 'background-color:green;'
    },
  },
})
```

Le code de cette section est que nous avons conçu cinq bulles qui peuvent afficher des couleurs en même temps.

Comme indiqué dans le code de la figure, par exemple, le premier bouton gauche, lorsque nous cliquons sur le bouton gauche, la bulle du bouton gauche est affichée en rouge et les autres bulles sont affichées en vert.

Cette fonction peut être un bon rappel du bouton sur lequel nous appuyons actuellement, et elle peut également être utilisée comme signal de retour, ce qui signifie que notre opération fonctionne efficacement et que la page Web et la connexion ne sont pas mortes.

L'effet de page :




```

<style>
    .left {
        width: 35%;
        height: 100%;
        float: left;
        left: 0;
        top: 0;
    }

    .leftLight {
        position: absolute;
        top: 30%;
        font-size: 50px;
    }

    .leftBtn {
        position: absolute;
        top: 31%;
        font-size: 50px;
        margin-left: 100px;
    }

    .up {
        position: absolute;
        height: 66%;
    }

```

Tous les codes après <style> sont utilisés pour ajuster la mise en page de la page Web, Il contient la position et la taille en pixels des bulles, du texte et des boutons.

```

html+="

Enfin, nous intégrons l'intégralité du programme HTML et du code dans le système de code contrôlé par Arduino, pour réaliser la connexion entre notre html et le système de contrôle back-end.



16


```

Code de Arduino

Code de commande du moteur L298 :

```
void go_up() {  
    digitalWrite(D5,0);  
    analogWrite(D2,Speed_D);  
    digitalWrite(D6,0);  
    analogWrite(D8,Speed_D);  
    server.sendHeader("Location","/"); // return to the first page  
    server.send(303); // send 303 to go back to the first page  
}  
|  
void go_down() {  
    analogWrite(D5,Speed_D);  
    digitalWrite(D2,0);  
    analogWrite(D6,Speed_D);  
    digitalWrite(D8,0);  
    server.sendHeader("Location","/"); // return to the first page  
    server.send(303); // send 303 to go back to the first page  
}  
  
void go_left() {  
    analogWrite(D2,Speed_T);  
    digitalWrite(D5,0);  
    analogWrite(D6,Speed_T);  
    digitalWrite(D8,0);  
    server.sendHeader("Location","/"); // return to the first page  
    server.send(303); // send 303 to go back to the first page  
}  
  
void go_right() {  
    digitalWrite(D2,0);  
    analogWrite(D5,Speed_T);  
    analogWrite(D8,Speed_T);  
    digitalWrite(D6,0);  
    server.sendHeader("Location","/"); // return to the first page  
    server.send(303); // send 303 to go back to the first page  
}
```

<DigitalWrite> est une fonction qui règle la tension de sortie de la broche sur haute ou basse. Cette fonction a deux paramètres pin et valeur : le paramètre pin représente le pin à régler, le paramètre valeur représente la tension de sortie.

<Analogwrite> peut sortir la valeur analogique (onde PWM) sur la broche, Il peut être utilisé pour régler la luminosité des diodes électroluminescentes à différents niveaux de lumière ou pour entraîner des moteurs à différentes vitesses. Après avoir appelé analogWrite (), la broche générera une onde carrée stable avec un rapport cyclique spécifié jusqu'au prochain appel à analogWrite ().

Ici, nous réalisons que le principe de la rotation de la voiture est : ces moteurs n'ont pas le matériel pour régler l'angle de braquage, si nous voulons que la voiture vire à gauche, nous gardons les deux roues à gauche immobiles et les deux roues à droite continuent à tourner, pour que nous puissions tourner à gauche. Tourner vers la droite est le réglage opposé.

Partie 1 : Variables

```
int Speed_D=350;
int Speed_T=300;

ESP8266WebServer server(80);

const int led = 13;
```

Speed_D est la vitesse d'avancer et reculer.

Speed_T est la vitesse de tourner.

LED est un entier en défini le numéro de piste.

Partie 2 : code principal

```
void setup(void) {

    pinMode(led, OUTPUT);
    digitalWrite(led, 0);
    pinMode(D5, OUTPUT);
    digitalWrite(D5, 0);
    pinMode(D6, OUTPUT);
    digitalWrite(D6, 0);
    pinMode(D2, OUTPUT);
    digitalWrite(D2, 0);
    pinMode(D8, OUTPUT);
    digitalWrite(D8, 0);

    Serial.begin(115200);
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password);
    Serial.println("");

    // Wait for connection
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println("");
    Serial.print("Connected to ");
```

```
server.on("/", handleRoot);
server.on("/up", go_up);
server.on("/down", go_down);
server.on("/left", go_left);
server.on("/right", go_right);
server.on("/stop", go_stop);
server.onNotFound(handleNotFound);
server.begin();
Serial.println("HTTP server started");
}

void loop(void) {
    server.handleClient();
    MDNS.update();
}
```

Dans cette partie, nous avons utilisé 5 pistes (D2, D5, D6, D8, LED) de Esp8266 Nodemcu (c'est un module de WIFI, ceci va aider le contrôle en distance) totalement. Pour contrôler le module L298N pouvant contrôler les 4 roues.

Le principe, c'est d'utiliser le module Esp8266 comme un web serveur, et établir le web en code html, dans la côte de l'opérateur, il peut appuyer les 5 boutons, et puis selon le code d'Arduino, le module va mettre la réaction de web et la sortie des 5 piste en relation, et puis nous pouvons contrôler les 4 roues en appuyer l'un des 5 boutons sur web.

Mais la limite est que le web n'est pas établi que dans l'internet local, cela veut dire que nous pouvons seulement la contrôler dans 10 mètres, pour résoudre ce problème, nous avons pensé à « la perçage dans l'internet local », cela veut dire nous allons utiliser un logiciel pour mettre l'internet local en l'internet public (expliqué dans la partie « Les Idées de design »), ce logiciel s'appelle « ngrok ».

Annexe: La fonction d'option

En plus d'implémenter cette fonction de télécommande, nous avons initialement prévu de développer une autre fonction pour assister le contrôle. Cette fonction auxiliaire consiste à installer un télémètre à ultrasons sur la voiture. Lorsque nous contrôlons la voiture pour conduire, l'avance, tourne à gauche et à droite de la voiture, nous pouvons voir les obstacles devant et les obstacles sur le côté grâce au chat vidéo en temps réel. Mais lorsque nous reculons, nous ne pouvons pas voir les obstacles derrière, à ce stade, nous pouvons utiliser ce télémètre à ultrasons pour réaliser la fonction d'évitement d'obstacles.

Nos idées de conception devront finalement être réalisées via notre html.

Notre idée de conception est de :

1. L'installation un télémètre à ultrasons derrière de la voiture, et connecter à notre puce de contrôle. Nous utilisons Arduino comme serveur pour lire la distance sur le télémètre en temps réel. Une fois que l'Arduino a lu ces données, elles seront envoyées à l'ordinateur en temps réel via le module wifi. Puis utiliser un programme similaire au logiciel de port série pour convertir les données en une format numérique de côté ordinateur, enregistrer-les et mettre-les à jour dans un fichier similaire à TXT en temps réel.
2. Du côté HTML, nous avons écrit un morceau de code de programme Ajax (Asynchronous Javascript And XML), nous pouvons utiliser ce programme pour afficher ces chiffres en temps réel enregistrés dans le fichier txt sur notre page Web. Nous mettons en place certaines conditions de jugement, par exemple, lorsque la distance de l'obstacle est inférieure ou égale à 10 cm, ces chiffres affichés deviennent rouges. Cela peut nous rappeler que la voiture est sur le point de heurter un obstacle et ne peut plus reculer.

Cependant, en raison de l'épidémie et du temps, nous n'avons pas reçu le soutien des enseignants concernés. Nous n'avons pas trouvé de moyen de réaliser la fonction d'Arduino en envoyant des données à l'ordinateur en temps réel et en générant des fichiers texte.

Mais nous avons réalisé la fonction de capture en temps réel des chiffres et mise à jour sur le frontal d'Ajax côté HTML. Les code Ajax comme ci-dessous :

```
<html>
<head>
  <meta charset="UTF-8">
  <title></title>
  <script src="https://apps.bdimg.com/libs/jquery/2.1.4/jquery.min.js"></script>
</head>
```

Ce morceau de code définit notre format HTML et notre bibliothèque de fonctions référencées.

```
<script>
var index = 0;
function loadData() {
  index++;
  console.log(index);
  $.ajax({
    url: "data.txt",
    data: {index: index},
    success: function (data) {
      var lines = data.split('\n');
      var numbers = [];
      var str = '';
      for (var i = 0; i < lines.length; i++) { // for (var i = lines.length-1; i < lines.length; i++) {
        if (lines[i] != "") {
          numbers.push(Number(lines[i]));
          if (Number(lines[i]) <= 10) {
            str += '<p class="red">' + lines[i] + '</p>';
          } else {
            str += '<p class="green">' + lines[i] + '</p>';
          }
        }
      }
      document.getElementById("result").innerHTML = str;
    }
  })
}

setInterval(loadData, 1000);
```

Ici, url est l'emplacement du fichier auquel nous voulons référencer les données.

Ce code indique que lors de la lecture de ce fichier texte, Ajax extraira les nombres du texte et les affichera sur HTML. < if (Number(lines[i]) <= 10) {> : Prendre 10 comme critère. Lorsque la distance est inférieure ou égale à 10, le chiffre vire au rouge, indiquant un signal de danger. Lorsque le nombre est supérieur à 10, c'est la distance de sécurité.