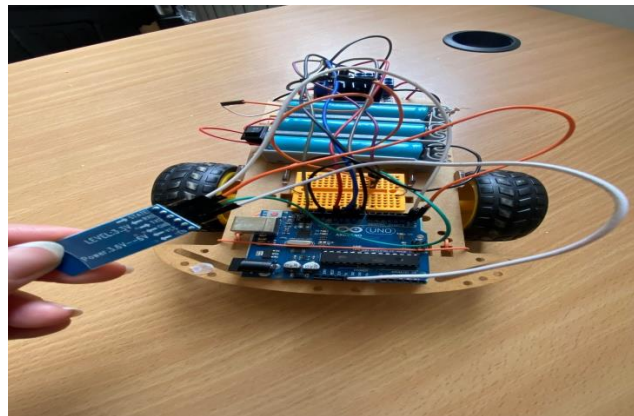


République Algérienne Démocratique et populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université des sciences et de la technologie houari Boumediene



Faculté de Génie Electrique
Master Electronique des systèmes embarqués

Projet Module Internet Of Things
Thème :
La conception et la réalisation d'un Smart car



Présenté par :

Cheraitia islam

Matricule :

191931042293

Les titres abordés dans ce rapport :

- Définition de l'Internet Des Objets
- Principe de fonctionnement
- Matériel utilisés
- Montage
- Programme
- Explication détaillée du programme
- Les figure : les figures des composants ,le fonctionnement du pont H et l'application mobile

Définition de l'internet des objets :

L'Internet des objets, ou IdO, est un système d'appareils informatiques interconnectés qui peuvent collecter et transférer des données sur un réseau sans fil sans intervention humaine. Il ne s'agit pas seulement des ordinateurs portables et des smartphones : presque tout appareil qui dispose d'un interrupteur peut potentiellement se connecter à Internet, ce qui en fait un composant de l'IdO. Par exemple, un « objet » dans l'Internet des objets pourrait être une personne portant un implant de surveillance cardiaque, une caméra diffusant des images en direct d'animaux sauvages dans les eaux côtières ou une voiture avec des capteurs intégrés pour avertir le conducteur des risques. Essentiellement, tout objet auquel on peut attribuer une adresse IP (Internet Protocol) et qui peut transférer des données sur un réseau peut faire partie de l'Internet des objets.

Principe de fonctionnement :

En utilisant Internet of Things (IoT) qui décrit le réseau de terminaux physiques, les « objets », qui intègrent des capteurs, des logiciels et d'autres technologies en vue de se connecter à d'autres terminaux et systèmes sur Internet et d'échanger des données avec eux.

Donc notre principe de fonctionnement consiste à contrôler notre smart car à distance avec module bluetooth HC-05 à l'aide d'une application mobile équivalente d'une télécommande

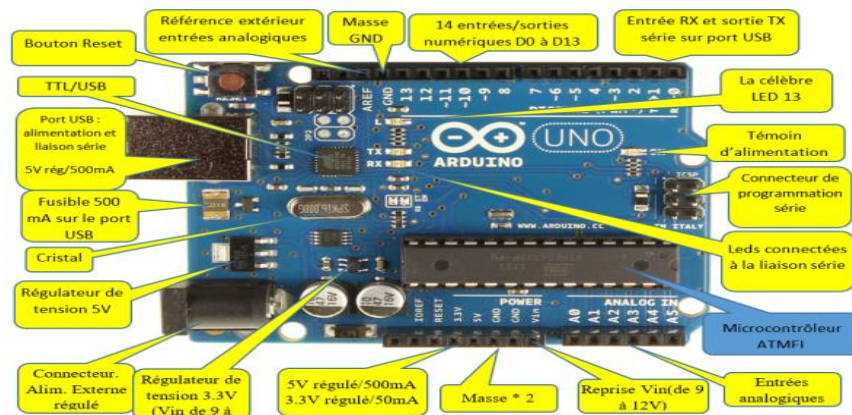
I) Matériel Utilisés :

- Carte Arduino.
- Module contrôleur des moteurs L298N.
- Module bluetooth hc-05.
- Deux moteurs DC.
- Une batterie.
- Des câbles jumper.
- Un châssis et des roues à bille.

II) Définition des composants :

1) Carte Arduino :

Arduino est une plateforme électronique open source qui repose sur un logiciel et un matériel facile à utiliser. Les cartes Arduino sont capables de lire des entrées (capteur) et de les transformer en sorties (comme l'activation de moteurs). Nous pouvons dire à notre carte ce qu'elle doit faire en envoyant un groupe d'instructions au microcontrôleur de la carte.



Carte arduino uno

2) Module bluetooth HC-05 :

le module Bluetooth joue le rôle de passerelle de données transparente. Toutes les données reçues via la connexion Bluetooth est acheminé à la broche TX du module, et les données envoyées au RX de broches du module sont émises par la connexion Bluetooth. Ce module est programmé en tant que esclave avec ce module on a pu voir le score du jeu au terminal.

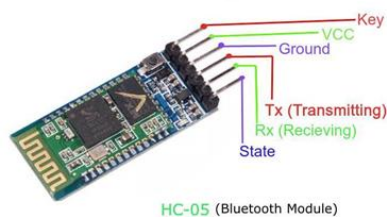


Figure 2:module bluetooth HC-05

3) Module contrôleur des moteurs L298N :

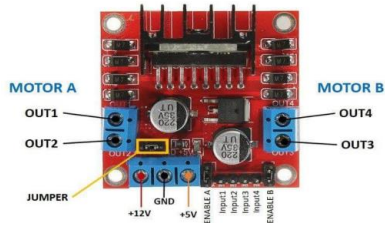


Figure 3: module controleur des moteurs

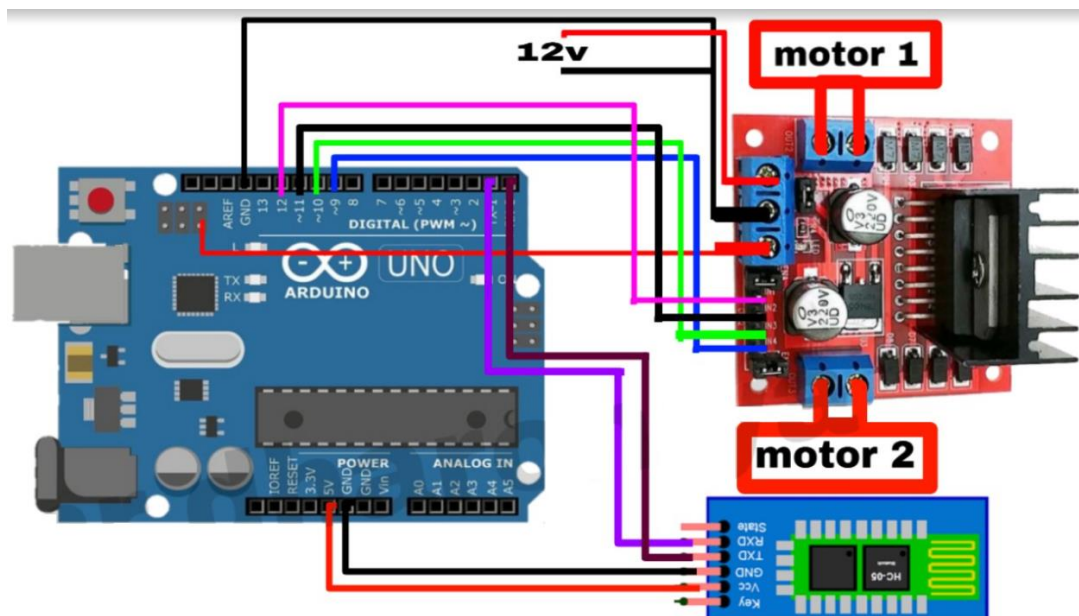
4) Moteur DC (bo_motor) :

Le moteur DC, appelé aussi moteur à courant continu, fait partie de la classe des moteurs électriques et sert essentiellement à transformer de l'énergie électrique en énergie mécanique.



Figure 4:moteur DC

5) Montage :



6) Programme :

For the programme contact the owner

Explication détaillée du programme :

pour le programme on a déclaré les variables de type int pour le module **contrôleur des moteurs L298N** pin (9, 10) pour le premier motor et pin (11, 12) pour le 2eme et la variable val de type caratere pour le message transmis par le bluetooth.

Ensuite dans le bloc void setup on a configuré les pins déjà déclaré comme sortie (OUTPUT) Ces pins sont reliees avec les 2 moteurs DC c'est l'équivalent des roues gauche et droite avec l'instruction PinMode qui fait la configuration des entrees et sorites .
et on a fixé la vitesse de transmission serie entre moniteur serie (sur ordinateur) et arduino avec 9600bauds par l'instruction Serial.begin(9600).

Dans le bloc void loop, on a commencé par la boucle While (Serial.available()> 0)
Si une donnée est arrivée, Serial.available() sera supérieure à 0. La file d'attente du buffer peut recevoir jusqu'à 128 octets.
Qui veut dire tant que les messages reçus par le bluetooth seront supérieur a 0 après on a fait comme séquence val = Serial.read();
Instruction Serial.read est une fonction de la bibliothèque série Arduino et ce qu'il fait est de lire le premier octet disponible

Et `Serial.println(val);` qui veut dire lire la valeur de la donnée reçue et l'affiche dans le moniteur série.

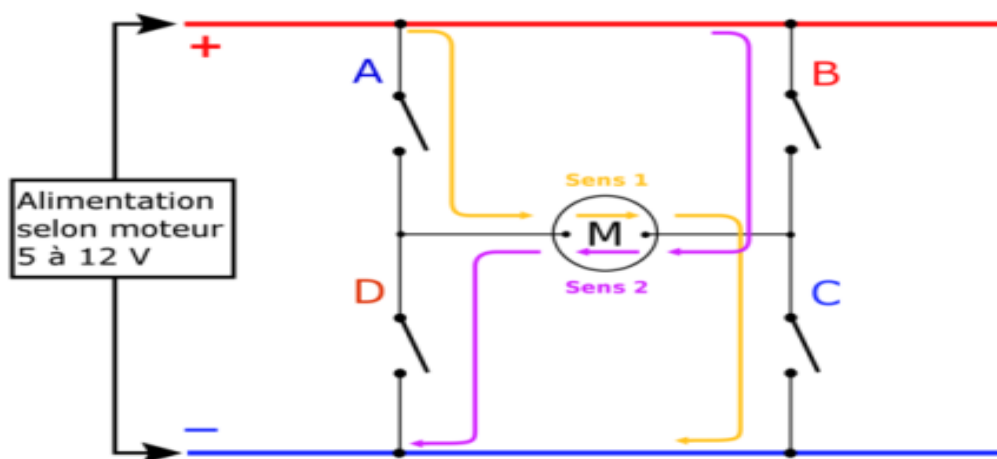
les boutons d'application bluetooth RC controller téléchargé par play store ces boutons sont considérés comme des messages (F,B,L,R,G,I,H,J,S,W,U ;la signification de chaque lettre est sur la figure ci-dessous).



Les options des boutons

a l'aide de la boucle { if } on fait la configuration des rotations des moteurs DC (m1a ,m1b ;m2a,m2b) avant , arrière, gauche ,droite .

la configuration est faite avec l'instruction `digitalWrite` deux états HIGH ou LOW high pour état haut du moteur c'est l'état d'une rotation et low état bas du moteur aucune rotation et voici la figure des options du bouton pour la manipulation du smart car marche avant ,arrière droite, gauche avec le principe de fonctionnement du pont H on configure les rotations souhaitées



Principe de fonctionnement du pont H

Convention : passant=HIGH bloquée=LOW

S'il n'y a pas un passage de courant on dit qu'il est bloqué on obtient donc niveau bas la configuration avec low (0v)

S'il y a un passage de courant on dit qu'il est passant on obtient donc niveau haut la configuration avec HIGH (5v)

1/La première boucle condition if de **marche avant** :

M1a et m2a sont à l'état **passant** donc niveau haut **HIGH** comme on a dit précédemment avec instruction digitalwrite,

m1b et m2b sont à l'état **bloqué** donc niveau haut **low**

2/ condition **marche arrière** avec else if :

M1a et m2a sont à l'état **bloqué** donc niveau haut **low** comme on a dit précédemment avec instruction digitalwrite,

m1b et m2b sont à l'état **passant** donc niveau haut **HIGH**

3/ condition **gauche** avec else if :

M1a et m1b et m2b sont à l'état **bloqué** donc niveau haut **low** comme on a dit précédemment avec instruction digitalwrite,

m2a sont à l'état **passant** donc niveau haut **HIGH**

4/ condition **droite** avec else if :

M2a et m1b et m2b sont à l'état **bloqué** donc niveau haut **low** comme on a dit précédemment avec instruction digitalwrite,

m1a est à l'état **passant** donc niveau haut **HIGH**

5/ condition **stop** avec else if :

m1a et M2a et m1b et m2b sont à l'état **bloqué** donc niveau haut **low** comme on a dit précédemment avec instruction digitalwrite,

6/ condition **avant droite** avec else if :

M2a et m1b et m2b sont à l'état **bloqué** donc niveau haut **low** comme on a dit précédemment avec instruction digitalwrite,

m1a est à l'état **passant** donc niveau haut **HIGH**.

7/ condition **arrière droite** avec else if :

M2a et m1a et m2b sont à l'état **bloqué** donc niveau haut **low** comme on a dit précédemment avec instruction digitalwrite,

m1b est a l'état **passant** donc niveau haut **HIGH**.

8/ condition **avant droite** avec else if :

M1b et m1a et m2b sont a l'état **bloqué** donc niveau haut **low** comme on a dit précédemment avec instruction digitalWrite,

m2a est a l'état **passant** donc niveau haut **HIGH**.

8/ condition **arrière gauche** avec else if :

M1a et m2a et m1b sont a l'état **bloqué** donc niveau haut **low** comme on a dit précédemment avec instruction digitalWrite,

m2b est a l'état **passant** donc niveau haut **HIGH**.

on a programmée 8 rotations grace au fonctionnement de ce pont .

L'application mobile : (BLUETOOTH RC CONTROL)

