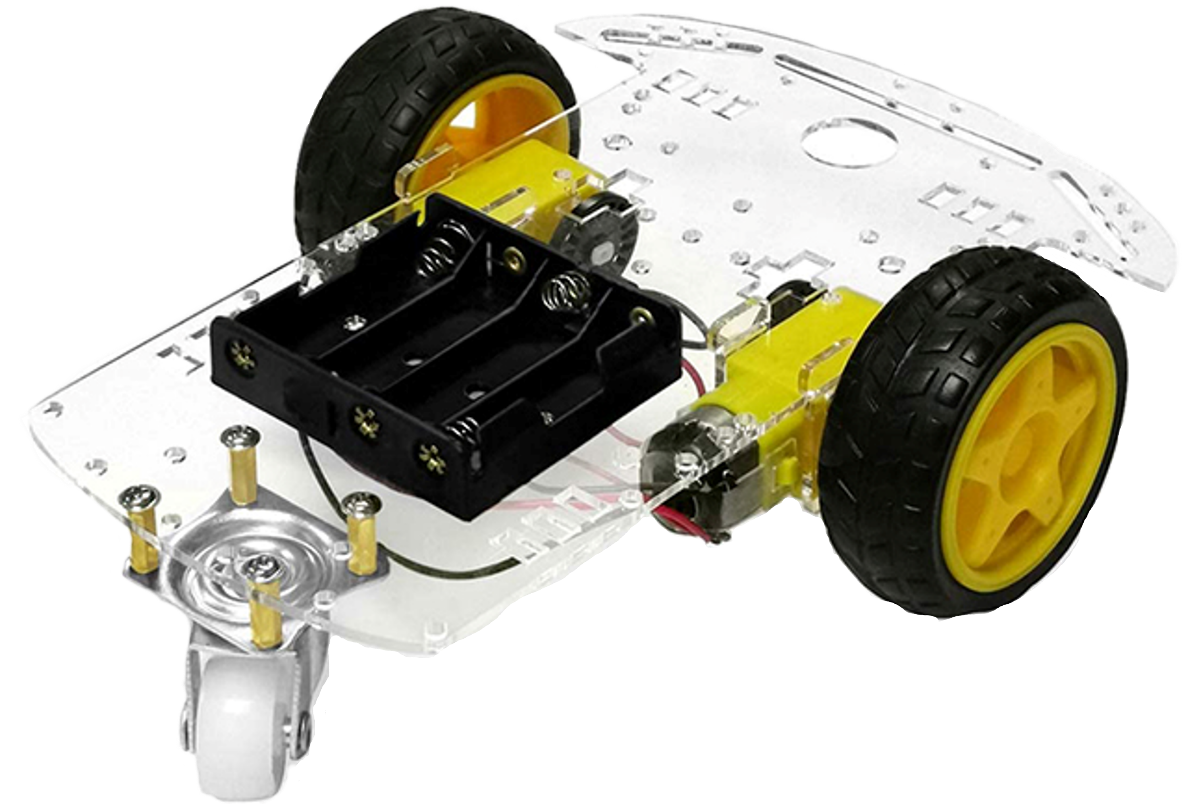
13/05/2022 -- 16/06/2022

UNIVERSITE PARIS XIII

I.U.T. DE VILLETANEUSE

DÉPARTEMENT G.E.I.I.



PROJET ROBOT

Noumane

BUT GEII TP INFO C

Introduction

Notre réalisation pratique consistera à réaliser un robot commandé par Wi-Fi grâce à une application nommée Blynk. Pour cela, nous utiliserons un microcontrôleur qui intègrent la gestion du Wi-Fi. Nous utiliserons également la plateforme open-source qui permet de coder le microcontrôleur, Arduino. Dans la première partie, nous allons étudier tous les composants à notre disposition puis dans un second temps, nous allons monter le robot et téléverser des programmes.

Organisation du projet

* Nous réalisons le diagramme de Gantt prévisionnelle du projet.

Une image contenant table

Description générée automatiquement

* Répartition des rôles pour les différentes revues

|  |  |  |  |
| --- | --- | --- | --- |
| REVUES NOMS | REVUE 1 | REVUE 2 | REVUE 3 |
| ASSAHOU S | Chef de projet | Animateur | Secrétaire |
| Jafais Noumane | Animateur | Secrétaire | Chef de projet |
|  | Secrétaire | Chef de projet | Animateur |

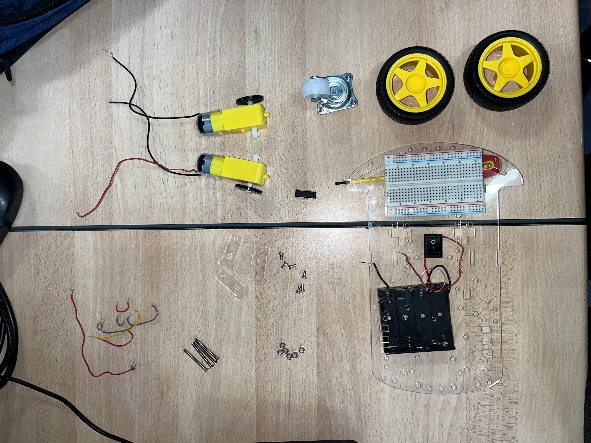
Une image contenant texte

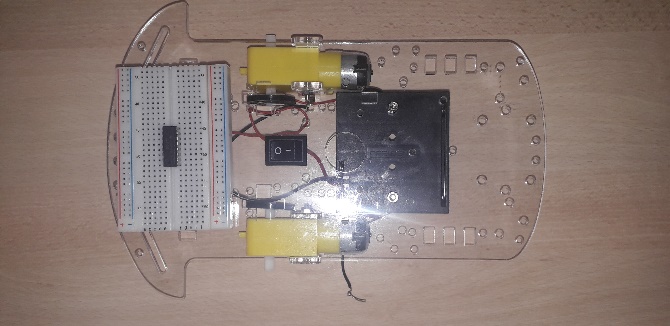
Description générée automatiquementPrise en main du matériel

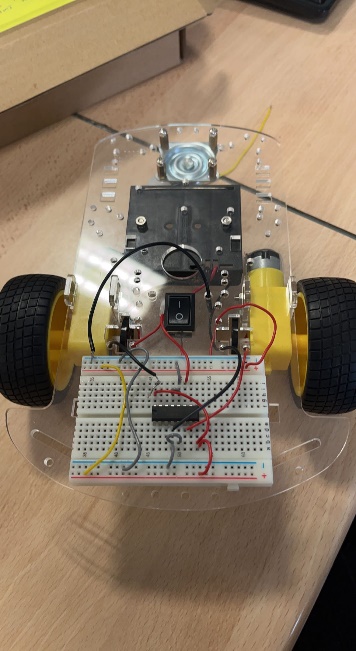
Nous avons à notre disposition :

* Un kit châssis robot (socle, roues, moteurs …)
* Un circuit intégré L293D
* Un microcontrôleur ESP32 programmable depuis Arduino

Nous réalisons le montage du robot :



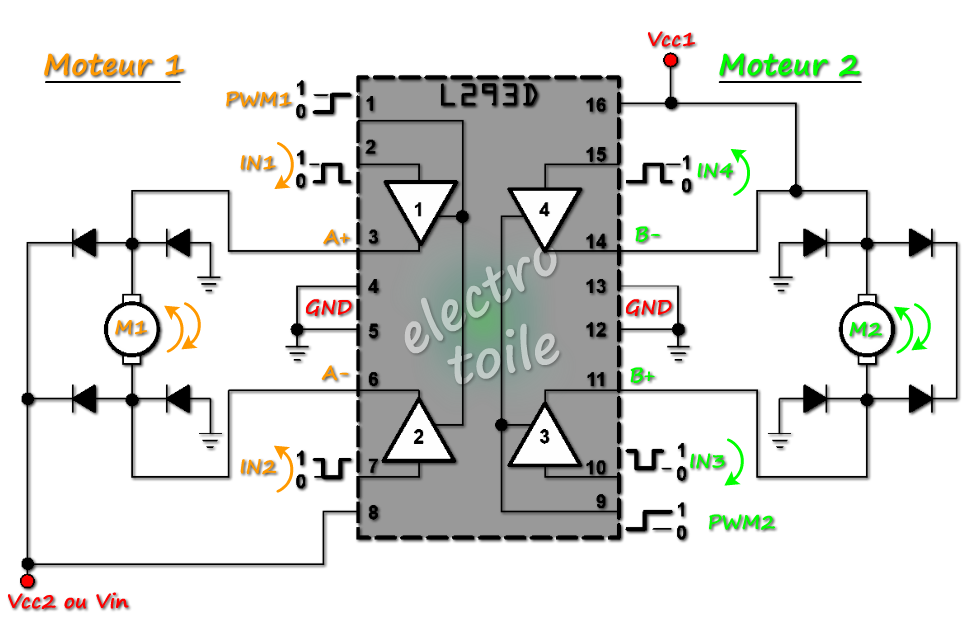




Circuit intégré L293D

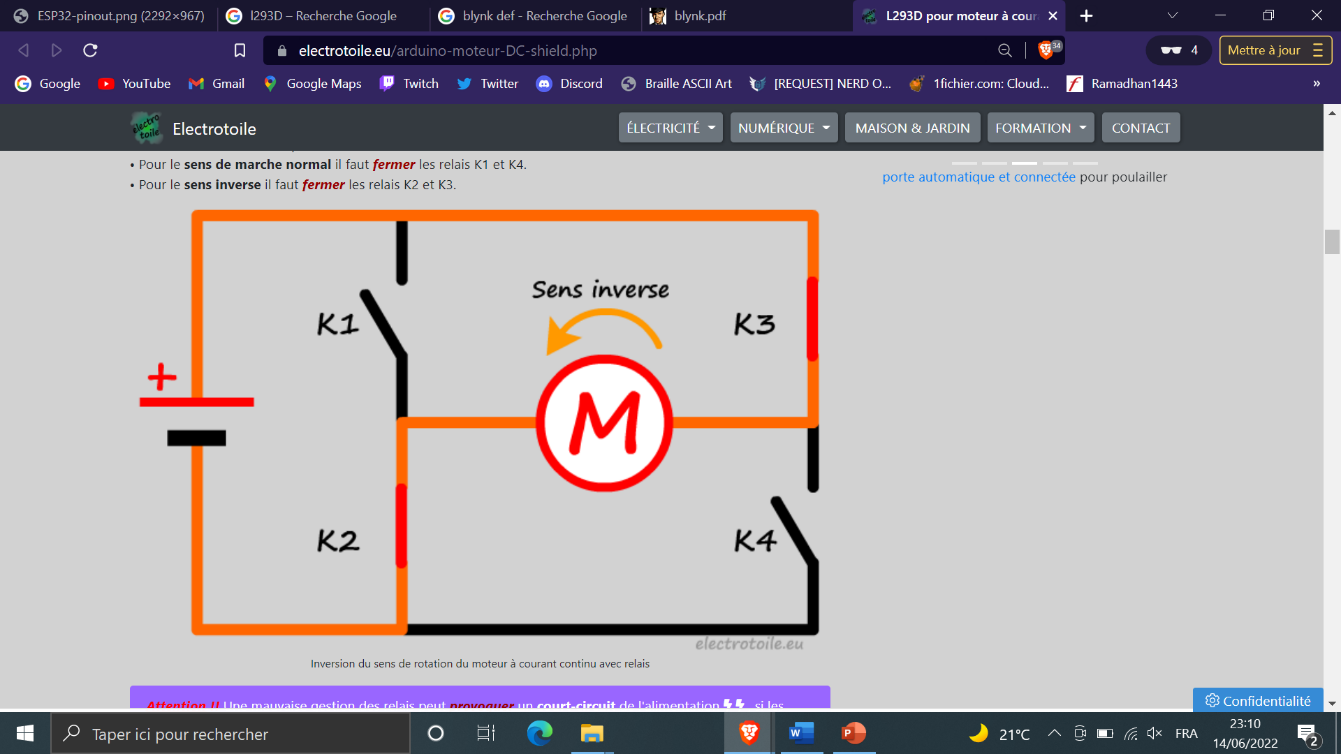
Nous étudions maintenant le circuit intégré L293D. Ce circuit permet de piloter 2 moteurs à courant continu dans les deux sens de rotation et de faire varier la vitesse. Ce circuit a 8 bornes et est composé de 4 transistors.

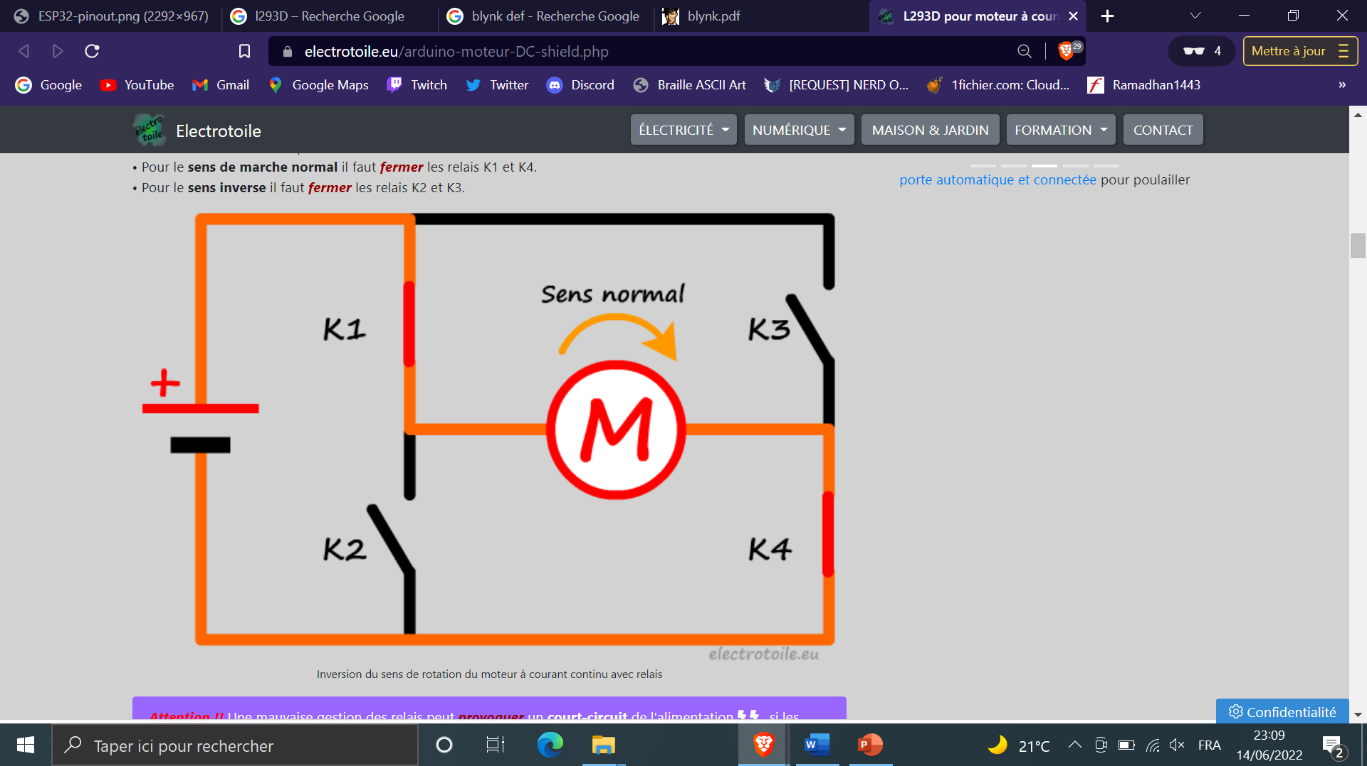
Voici le schéma électrique du L293D avec les 2 moteurs :



Le circuit est équipé d’un pont en H qui permet d’inverser de sens de rotation du moteur 1 avec les bornes IN1 et IN2, les bornes IN3 et IN4 pour le moteur 2.

Les bornes ENABLE1 (PWM1) et ENABLE2 (PWM2) servent à recevoir des informations de vitesse pour faire varier les moteurs.

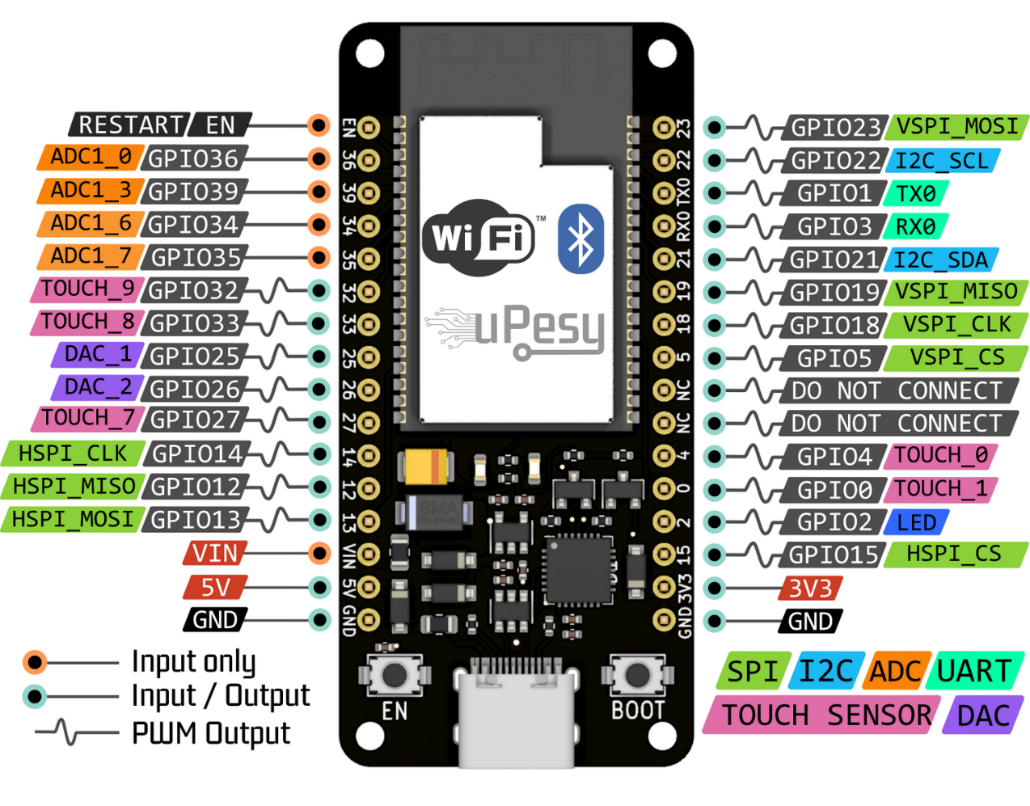
Les bornes A+, A-, B+, B- sont reliées aux moteurs et permettent de faire tourner le moteur selon le sens des bornes IN.

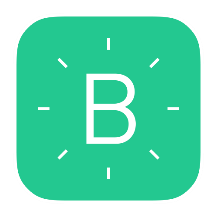


D’apres ces figures, nous pouvons voir que si on active PWM alors K1 et K4 se ferme, le courant circule et fait tourner le moteur. A l’inverse, si on désactive PWM, alors K1 et K4 s’ouvre mais K2 et K3 se ferme, donc le courant circuler toujours et le moteur tourne dans le sens inverse.

Module ESP32

Nous étudions maintenant la carte de développement l’ESP32. Cette carte est équipée de la gestion du WIFI, du Bluetooth, du « touch ». Cette carte est programmable sur Arduino. Pour cela, il faudra installer les librairies de la carte ESP32. Voici la carte ESP32 avec les bornes associées :



Application Blynk

Pour pouvoir contrôler la carte ESP32 à distance via WIFI, nous allons utiliser une application mobile nommée « Blynk ». Blynk est une plateforme pour IoT (Internet of Things = l’Internet des Objets). Elle permet de contrôler et de visualiser les données d’un système embarqué via un serveur cloud. La conception de l’application mobile à base de widgets est réalisée par des simple réglage et sans ligne de code.

Pour utiliser sur Arduino, il faut inclure la bibliothèque Blynk dans le code Arduino.

Nous avons ici un programme qui permet de connecter la carte ESP32 à l’application mobile Blynk via wifi avec une clé d’authentification fourni par Blynk, le nom ainsi que le mot de passe du réseau.

Une image contenant texte

Description générée automatiquement

Maintenant nous allons réaliser le montage pour commander les moteurs du robot en utilisant le circuit intégré L293D et la carte ESP32.

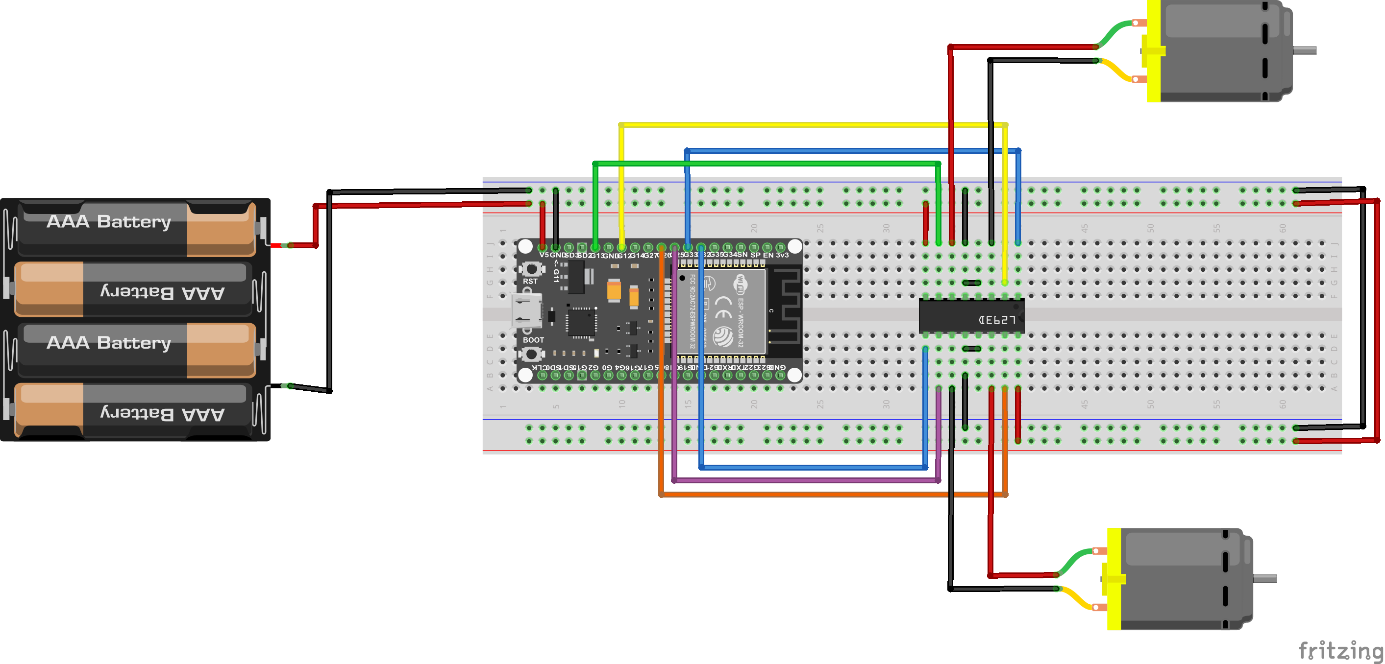
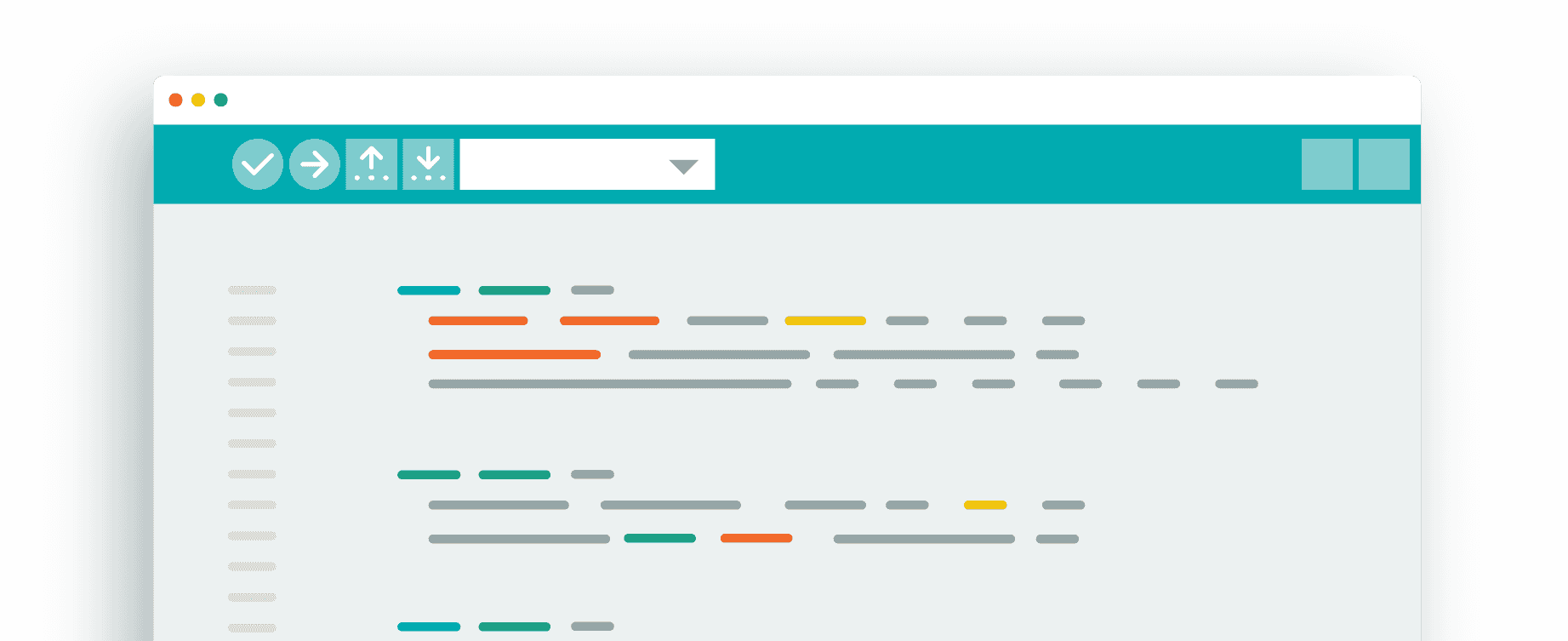
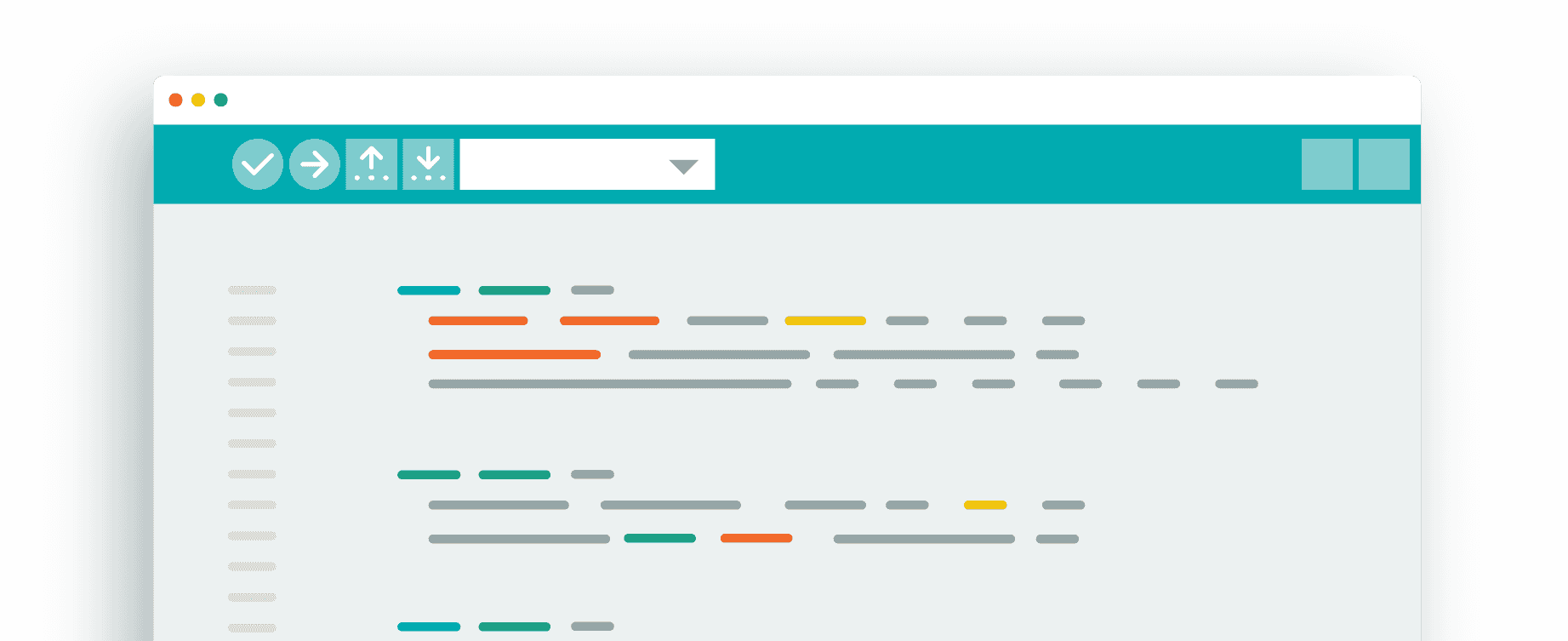


Image câblage

Code Arduino

Ce code permet de piloter le robot via Wifi.



#define BLYNK\_PRINT **Serial**

#include <**WiFi**.h>

#include <**WiFiClient**.h>

#include <BlynkSimpleEsp32.h>

#define BLYNK\_TEMPLATE\_ID "TMPLfN0q466A"

#define BLYNK\_DEVICE\_NAME "ESP32 Robot"

#define BLYNK\_AUTH\_TOKEN "QlPxhbpOdLE2wCVjThEYipcBMTJUtWdt"

#define MoteurIN1 13 //IN1

#define MoteurIN2 12 //IN2

#define MoteurIN3 26 //IN3

#define MoteurIN4 25 //IN4

#define ENA        33 //PWM1

#define ENB        32 //PWM2

bool avant = 0;

bool arriere = 0;

bool gauche = 0;

bool droite = 0;

int Speed;

char auth[] = "QlPxhbpOdLE2wCVjThEYipcBMTJUtWdt";

char ssid[] = "SMJ6";

char pass[] = "12345678";

void setup()

{

**Serial**.begin(115200);

pinMode(ENA, OUTPUT);

pinMode(ENB, OUTPUT);

pinMode(MoteurIN1, OUTPUT);

pinMode(MoteurIN2, OUTPUT);

pinMode(MoteurIN3, OUTPUT);

pinMode(MoteurIN4, OUTPUT);

**Blynk**.begin(auth, ssid, pass);

}

void loop()

{

**Blynk**.run();

if (avant == 0 && arriere == 0 && gauche == 0 && droite == 0)

{

digitalWrite(MoteurIN1, LOW);

digitalWrite(MoteurIN2, LOW);

digitalWrite(MoteurIN3, LOW);

digitalWrite(MoteurIN4, LOW);

}

}

BLYNK\_WRITE(V0)

{

avant = param.asInt();

if (avant == 1)

{

analogWrite(ENA, Speed);

analogWrite(ENB, Speed);

digitalWrite(MoteurIN1, LOW);

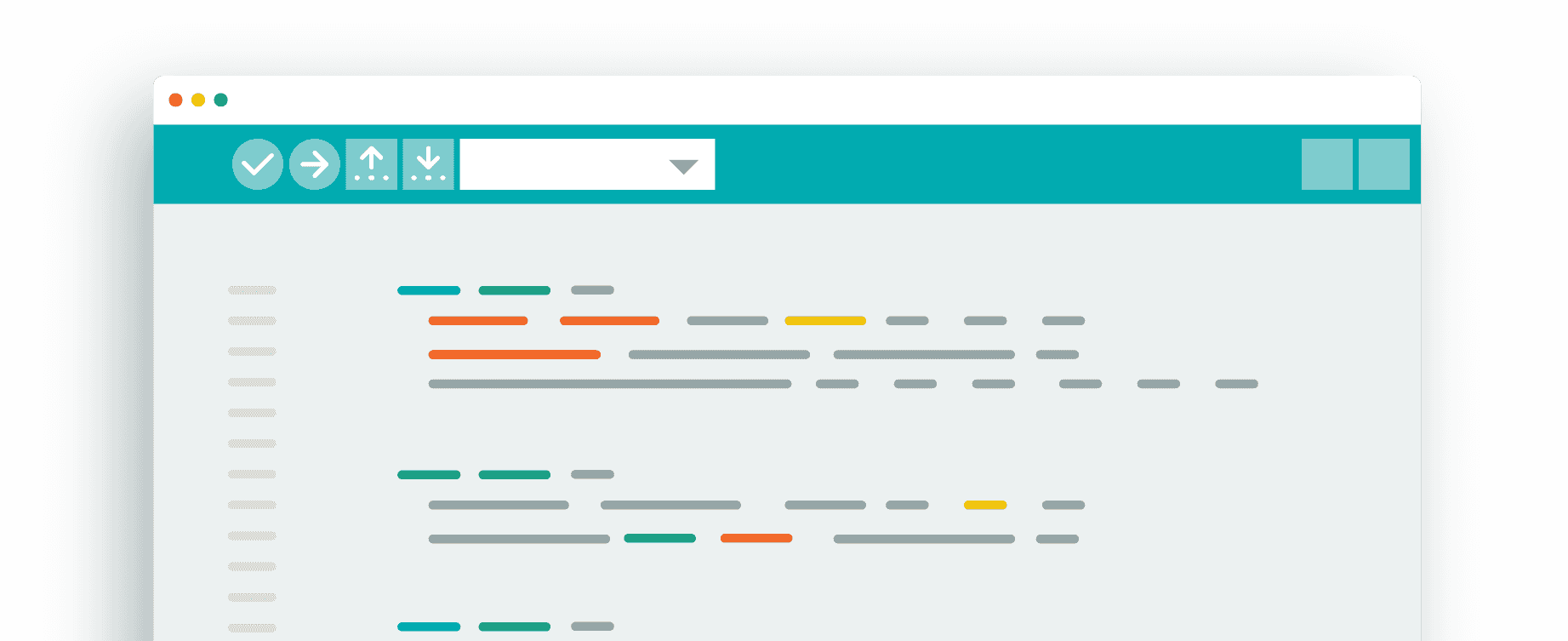
digitalWrite(MoteurIN2, HIGH);

digitalWrite(MoteurIN3, HIGH);

digitalWrite(MoteurIN4, LOW);

}

}



BLYNK\_WRITE(V1)

{

arriere = param.asInt();

if (arriere == 1)

{

analogWrite(ENA, Speed);

analogWrite(ENB, Speed);

digitalWrite(MoteurIN1, HIGH);

digitalWrite(MoteurIN2, LOW);

digitalWrite(MoteurIN3, LOW);

digitalWrite(MoteurIN4, HIGH);

}

}

BLYNK\_WRITE(V2)

{

gauche = param.asInt();

if (gauche == 1)

{

analogWrite(ENA, Speed);

analogWrite(ENB, Speed);

digitalWrite(MoteurIN1, HIGH);

digitalWrite(MoteurIN2, LOW);

digitalWrite(MoteurIN3, HIGH);

digitalWrite(MoteurIN4, LOW);

}

}

BLYNK\_WRITE(V3)

{

droite = param.asInt();

if (droite == 1)

{

analogWrite(ENA, Speed);

analogWrite(ENB, Speed);

digitalWrite(MoteurIN1, LOW);

digitalWrite(MoteurIN2, HIGH);

digitalWrite(MoteurIN3, LOW);

digitalWrite(MoteurIN4, HIGH);

}

}

BLYNK\_WRITE(V4)

{

Speed = param.asInt();

}