Ecole d'Eté sur

L'INTELLIGENCE ARTIFICIELLE

EDITION 2021

Voiture automatique avec NodeMCU Esp32 et Raspberry Pi

Généralités sur la robotique

Qu'est ce que la robotique ?

La **robotique** est l'ensemble des techniques permettant la conception et la réalisation de machines automatiques ou de robots.

Un robot est: «une machine effectuant, grâce à un système de commande automatique à base de micro-processeur, une tâche précise pour laquelle il a été conçu dans le domaine industriel, scientifique, militaire ou domestique ».

De cette définition découlent deux interprétations : la première serait de voir le robot comme une machine, qui possède des capteurs, un système logique et des actionneurs. Il est matériel. La deuxième laisse penser qu'un robot peut aussi être virtuel (voir Bot informatique).

Qu'est ce que la robotique ?

La robotique actuelle trouve des applications dans différents domaines (liste non exhaustive) :

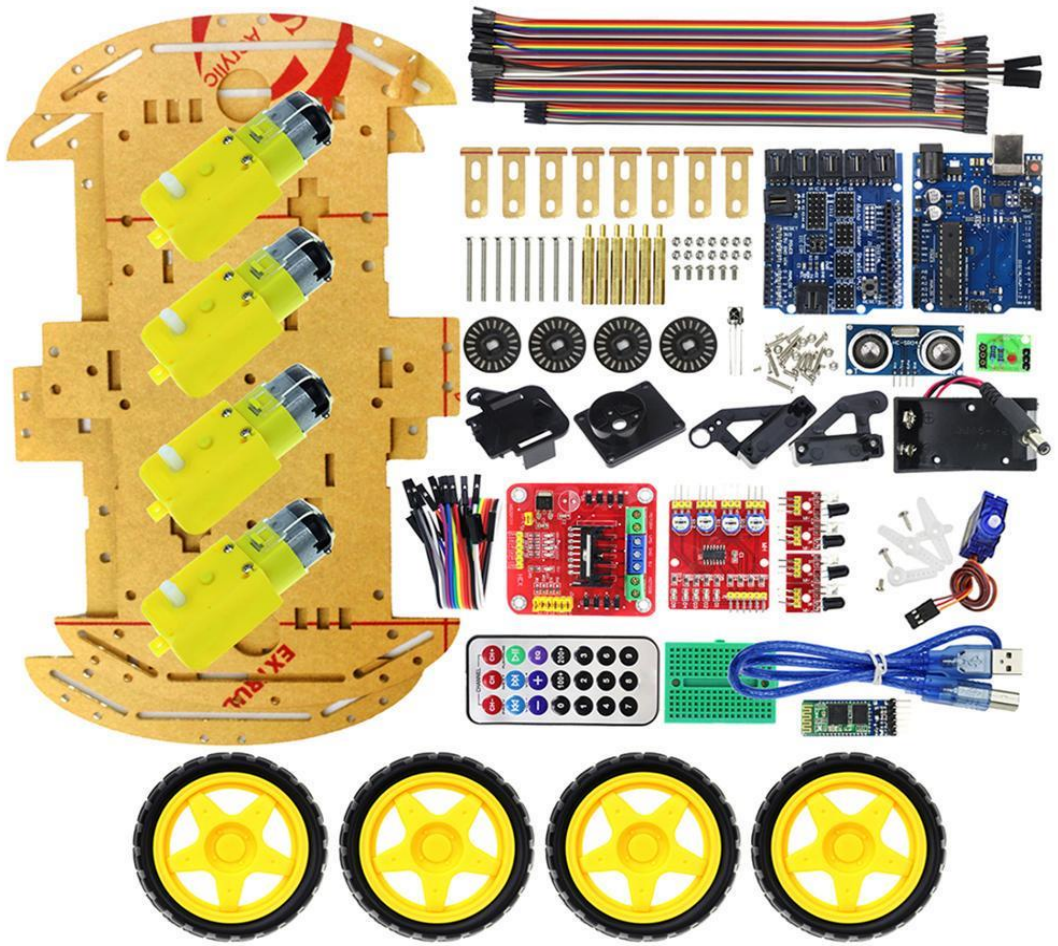
- la robotique industrielle,
- la robotique domestique,
- la robotique médicale,
- la robotique militaire,
- la robotique scientifique, par exemple pour l'exploration de l'espace (aérobot), des fonds marins (robots sous-marins autonomes), dans les laboratoires d'analyse (robotique de laboratoire), etc., ou encore
- la robotique de transport (de personnes et de marchandises), avec par exemple ROPITS (Robot for Personal Intellingent Transport System), Robosoft, RoboCourier, etc.

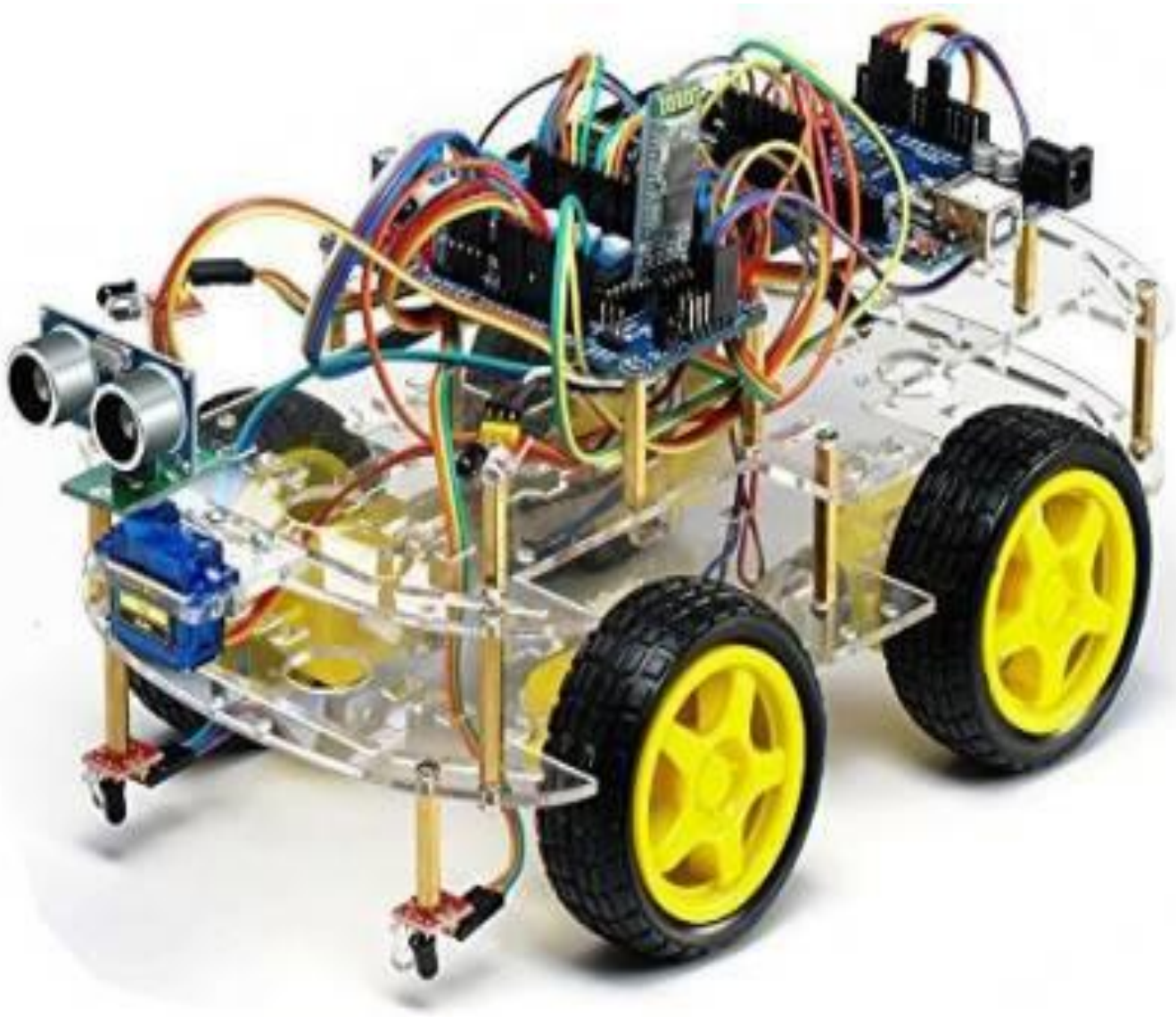
Généralités sur la robotique

Différents types de robots matériels

Voiture 4 roues

Contenu du kit de voiture

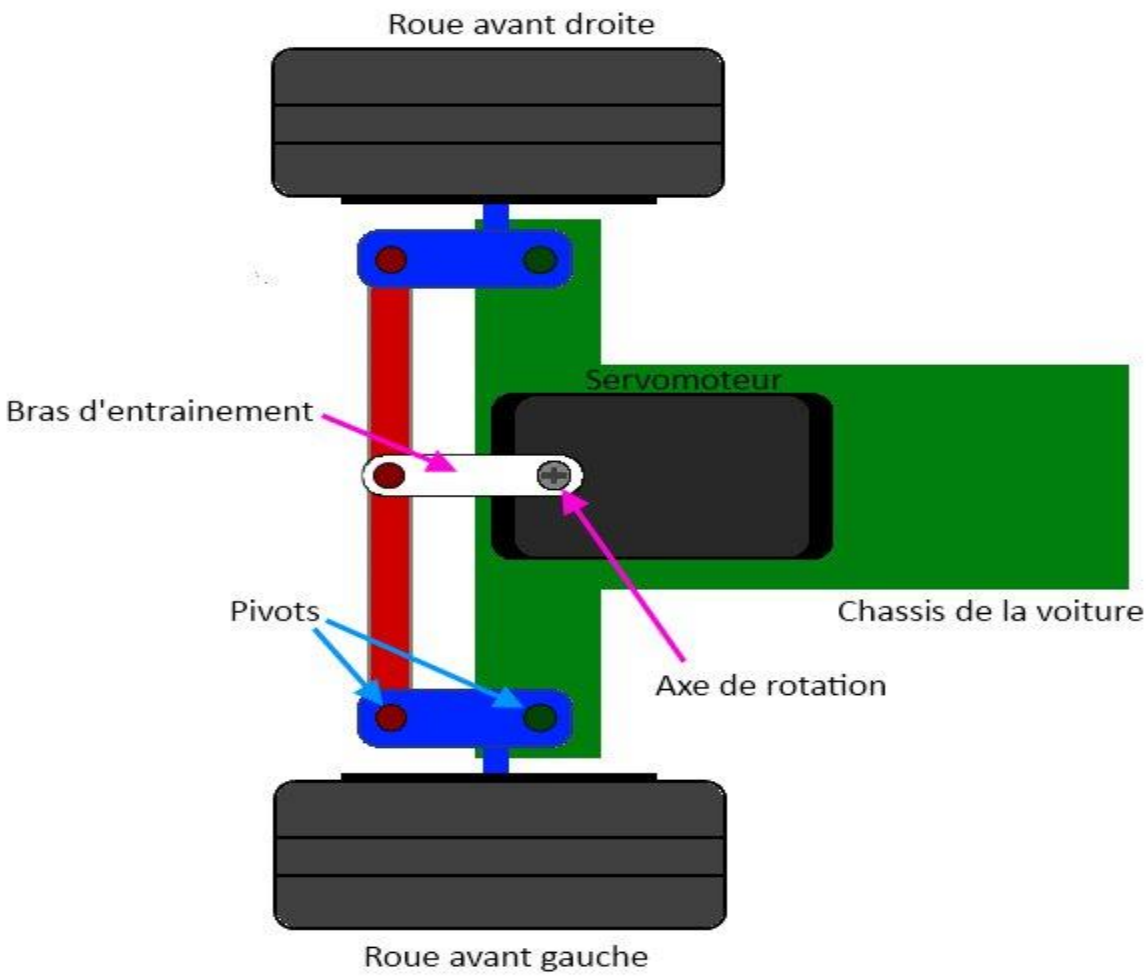


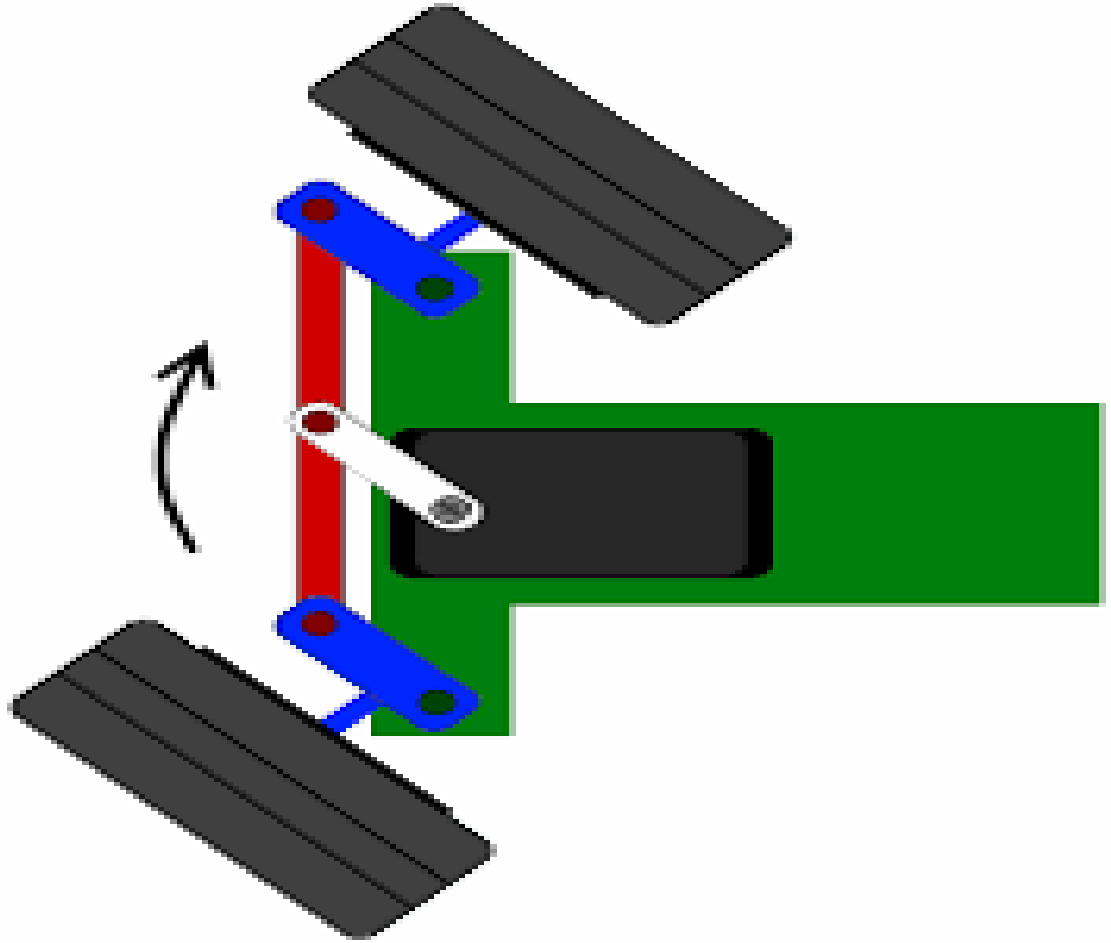


Commande la voiture

Comprendre la mécanique

Voiture avec 2 moteurs CC et un servomoteur

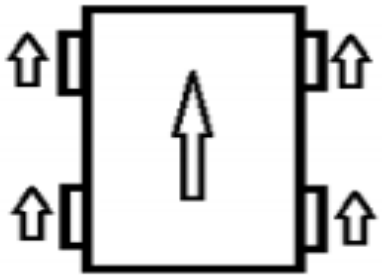




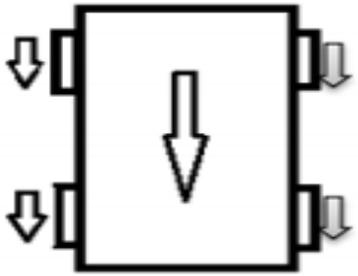
Commande la voiture

Comprendre la mécanique

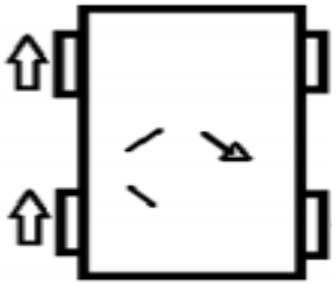
Voiture avec 4 moteurs CC



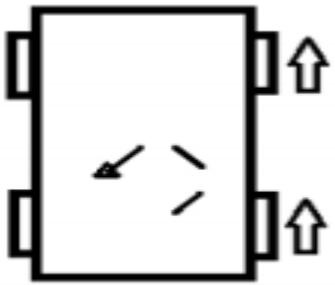
Forward



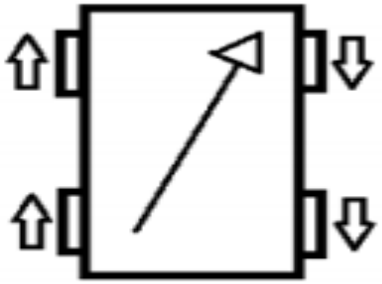
Reverse



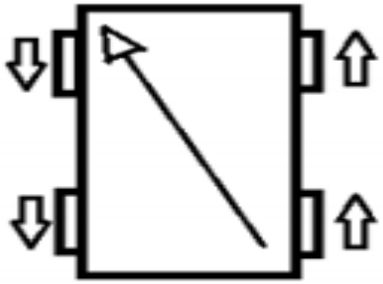
Clockwise



Anti-Clockwise



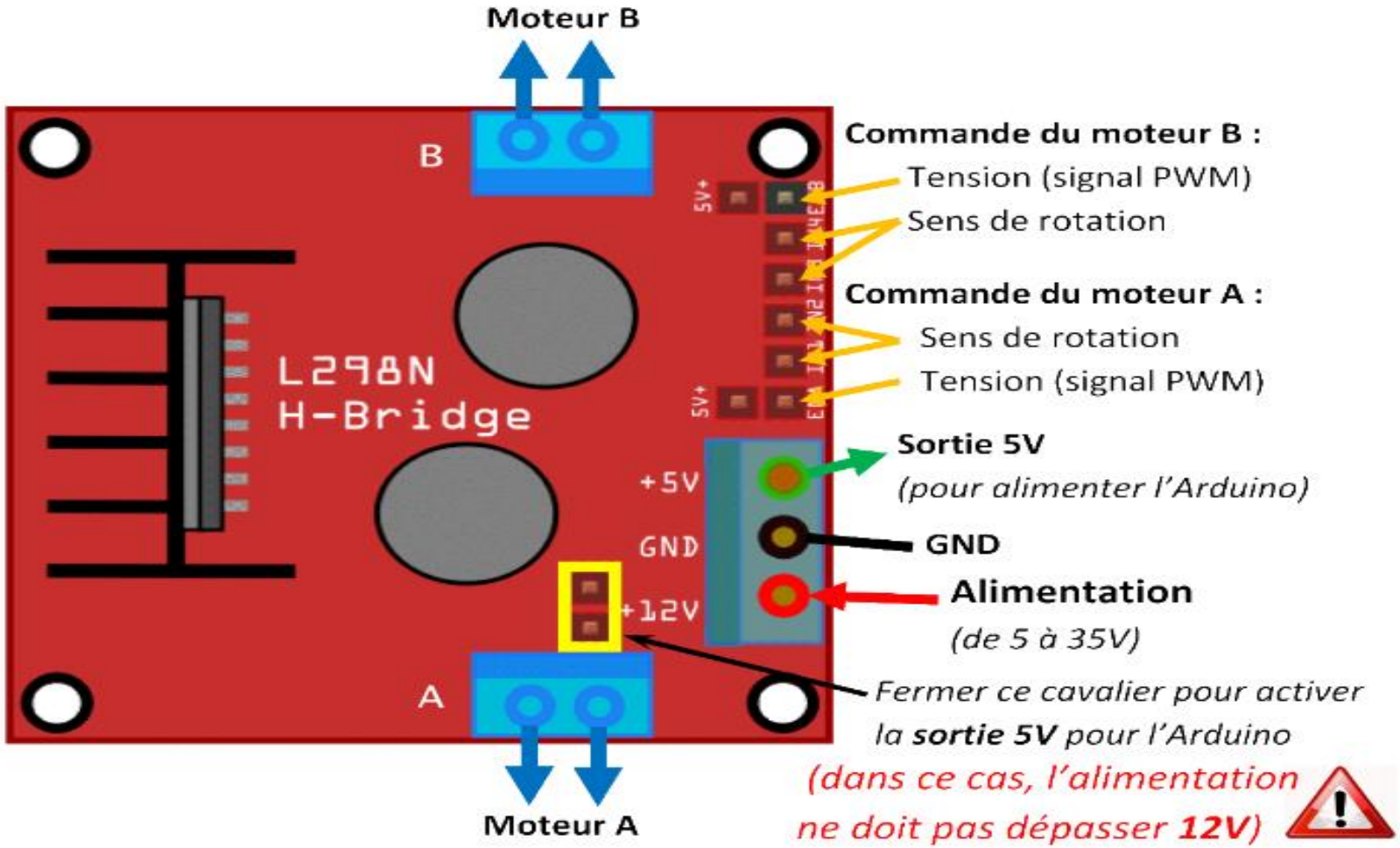
Turn Right



Turn Left

Le module L298N

Le L298N est un module de commande des moteurs à courant continu.

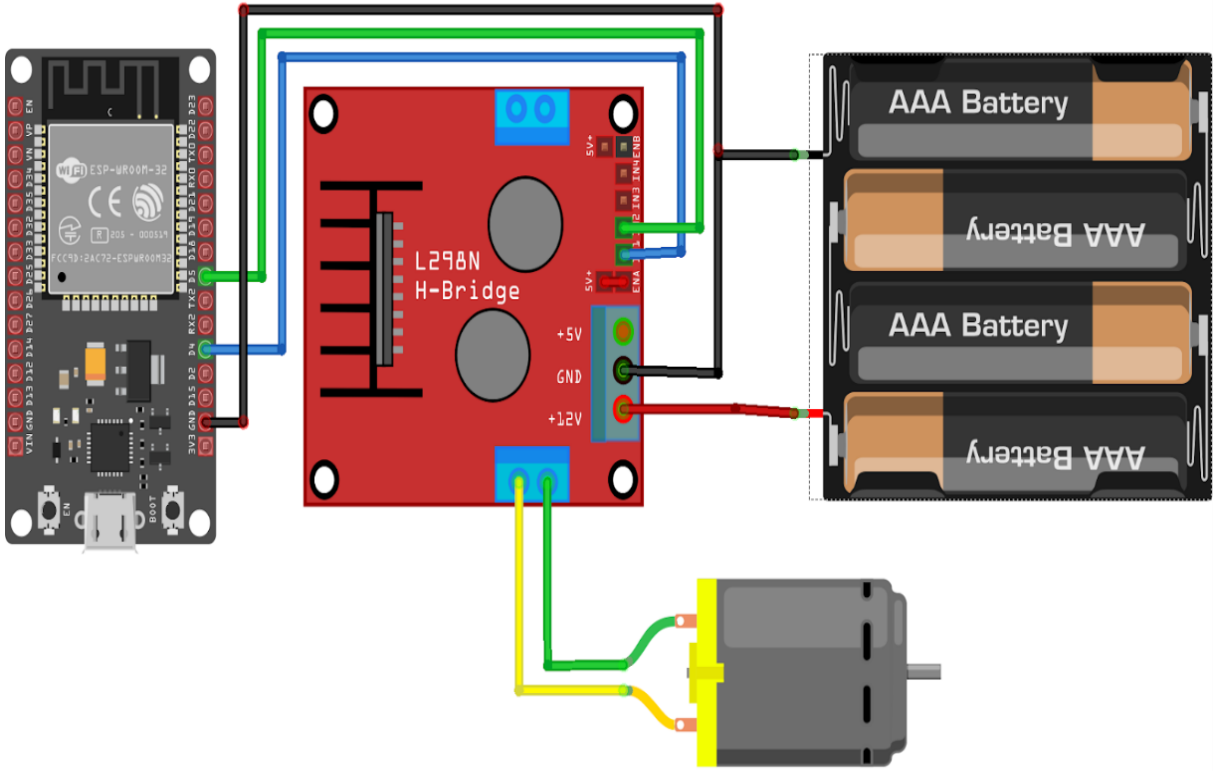


Moteur CC

Commande du moteur avec le L298N

On commande le moteur en jouant sur les pins In et En du module L298N.

Entrées			Fonctionnement du moteur
EnA (Ven)	In1 (A)	In2 (B)	
0	X	X	moteur non alimenté (à **l'arrêt**, arbre libre)
1	0	0	moteur bloqué (alimenté en **frein**)
1	1	1	moteur bloqué (alimenté en **frein**)
1	1	0	moteur alimenté, tourne dans le sens **direct**
1	0	1	moteur alimenté, tourne dans le sens **inverse**



Commande la voiture

MicroPython sur Esp32

Préliminaires

```
from machine import Pin, PWM
import time
# Pin Moteur Gauche
enA_Mot_Gauche = PWM(Pin(13), freq=500, duty=900)
in1_Mot_Gauche = Pin(12,Pin.OUT)
in2_Mot_Gauche = Pin(14,Pin.OUT)
# Pin Moteur Droit
in3_Mot_Droit = Pin(27,Pin.OUT)
in4_Mot_Droit = Pin(26,Pin.OUT)
enB_Mot_Droit = PWM(Pin(25), freq=500, duty=900)

pins = [in1_Mot_Gauche,in2_Mot_Gauche,in3_Mot_Droit,in4_Mot_Droit]
pwms = [enA_Mot_Gauche,enB_Mot_Droit]
```

Commande la voiture

La fonction change_pins_state()

```
def change_pins_state(pins,cmd="0000"):
    for i in range(4):
        if cmd[i] == "0":
            pins[i].off()
        else:
            pins[i].on()
```

Commande la voiture

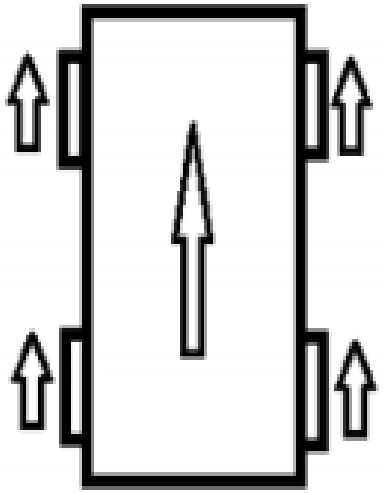
La fonction stop()

```
def stop(pins):
    change_pins_state(pins, cmd="0000")
```

Commande la voiture

La fonction forward()

```
def forward(pins,pwms,motor_speed = 1023):
    stop(pins)
    time.sleep_ms(10)
    change_pins_state(pins,cmd="0101")
    pwms[0].duty(motor_speed)
    pwms[1].duty(motor_speed)
```

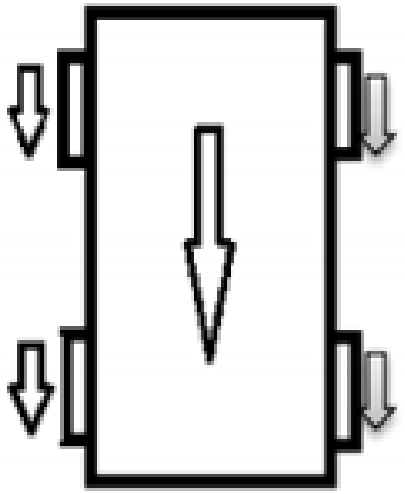


Forward

Commande la voiture

La fonction reverse()

```
def reverse(pins,pwms,motor_speed = 1023):
    stop(pins)
    time.sleep_ms(10)
    change_pins_state(pins,cmd="1010")
    pwms[0].duty(motor_speed)
    pwms[1].duty(motor_speed)
```

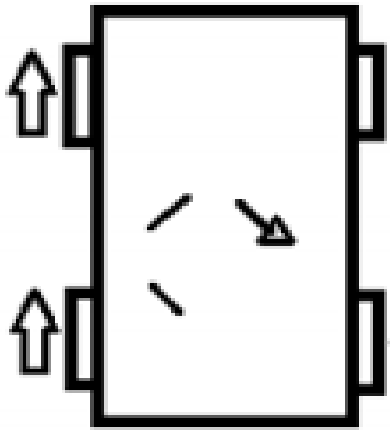


Reverse

Commande la voiture

La fonction clockwise()

```
def clockwise(pins,pwms,motor_speed = 1023):
    stop(pins)
    time.sleep_ms(10)
    change_pins_state(pins,cmd="0100")
    pwms[0].duty(motor_speed)
```

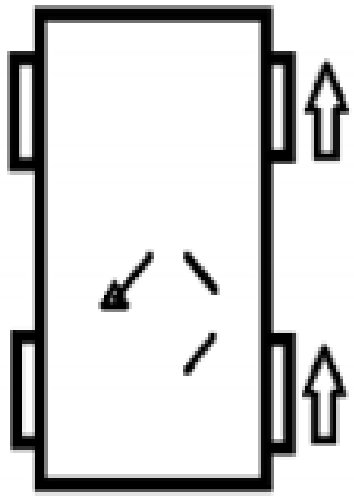


Clockwise

Moteur

La fonction anti_clockwise()

```
def anti_clockwise(pins,pwms,motor_speed = 1023):
    stop(pins)
    time.sleep_ms(10)
    change_pins_state(pins,cmd="0001")
    pwms[1].duty(motor_speed)
```

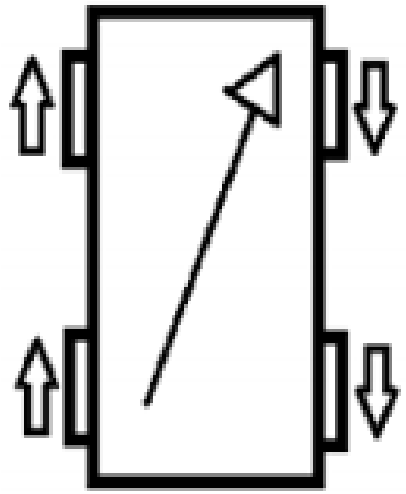


Anti-Clockwise

Moteur

La fonction turn_right()

```
def turn_right(pins,pwms,motor_speed = 1023):
    stop(pins)
    time.sleep_ms(10)
    change_pins_state(pins,cmd=" 0110 ")
    pwms[0].duty(motor_speed)
    pwms[1].duty(motor_speed)
```

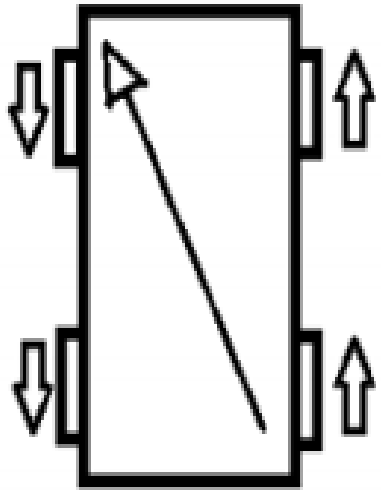


Turn Right

Commande la voiture

La fonction turn_left()

```
def turn_left(pins,pwms,motor_speed = 1023):
    stop(pins)
    time.sleep_ms(10)
    change_pins_state(pins,cmd=" 1001 ")
    pwms[0].duty(motor_speed)
    pwms[1].duty(motor_speed)
```



Turn Left

Applications

Test des différents mouvements

Ecrire un programme qui teste les différents mouvements de la voiture.

Applications

Eviter les obstacles

Ecrire un programme qui, en utilisant le capteur HCSR04, détecte les obstacles et les évite en s’arrettant.

Applications

Eviter les obstacles et rechercher un nouveau chemin

Ecrire un programme qui, en utilisant le capteur HCSR04, détecte les obstacles et les évite en s'arrettant. Ensuite, avec le servomoteur, le programme explore le coté droit et le coté gauche, choisit le meilleur chemin et continue d'avancer.