

## TP N° 2 : La variation de vitesse d'un moteur à courant continu

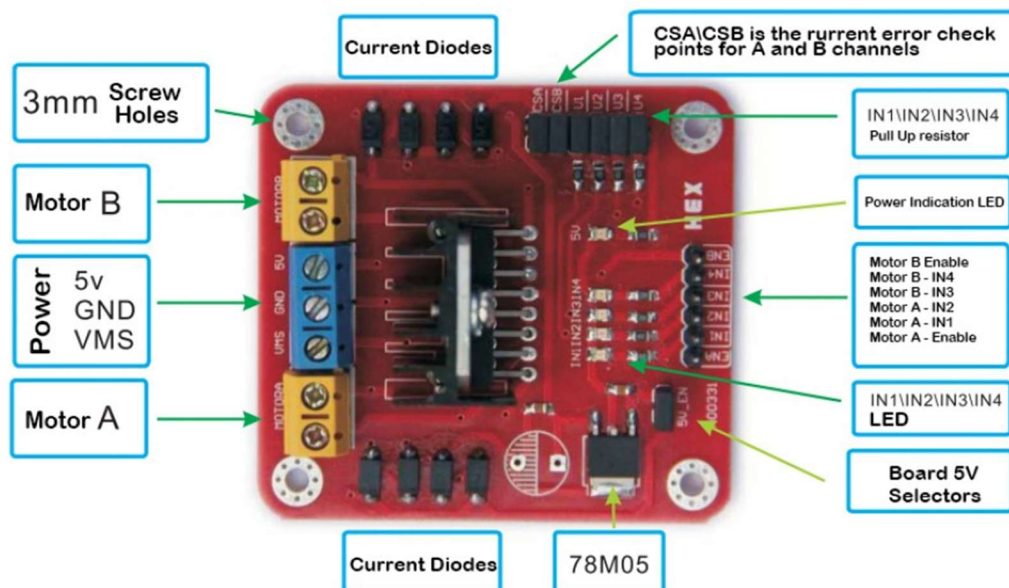
### I. Objectifs

A travers ce TP l'étudiant sera capable de :

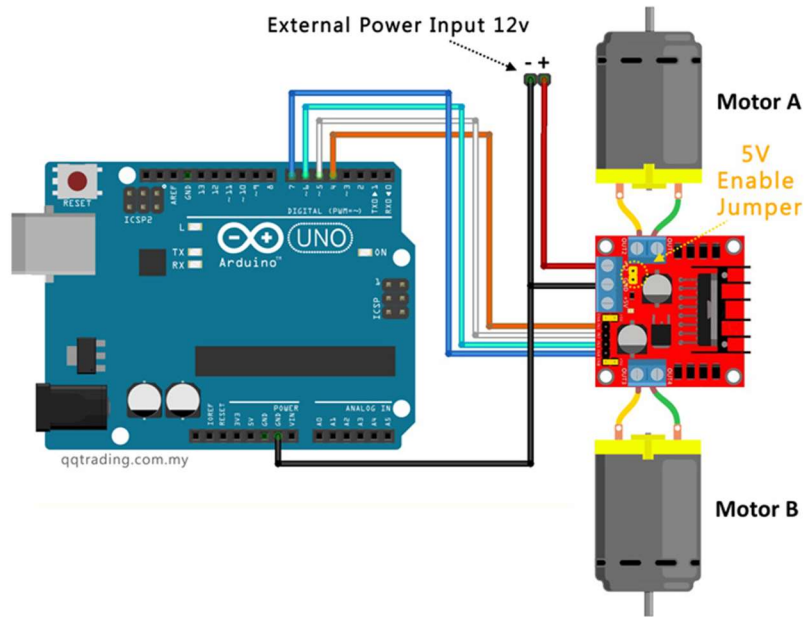
- Commander un moteur à courant continu dans les deux sens de rotation en utilisant une carte de commande et une interface de puissance.
- Varier automatiquement la vitesse du moteur à courant continu en utilisant un signal PWM.

### II. Partie 1 : La commande du moteur à courant continu dans les deux sens de rotation avec Arduino et le L298N :

- Le driver utilisé est un Shield Arduino pour moteur à courant continu basé sur le L298 :



- Construire le montage suivant avec un seul moteur à courant continu :



- En utilisant l'IDE Arduino taper le programme suivant, téléverser le sketch dans la carte Arduino.

```

Variation_de_vitesse_§
// TP 2: Variation de vitesse d'un mcc
// Partie 1

// the setup function runs once when you press reset or power the board

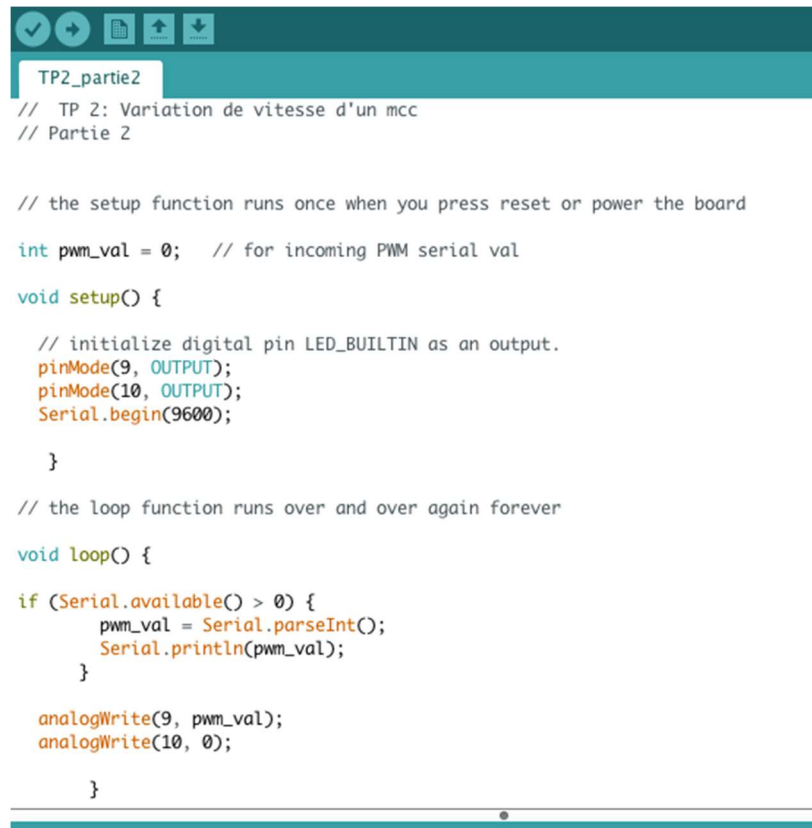
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {

  // Pour le sens positif de rotation
  analogWrite(9, 60);
  analogWrite(10, 0);
  delay(4000);
  // Pour stoper le moteur
  analogWrite(9, 0);
  analogWrite(10, 0);
  delay(4000);
  // Pour le sens négatif de rotation
  analogWrite(9, 0);
  analogWrite(10, 60);
  delay(4000);
}
    
```

### III. Partie 2 : Variation de vitesse du moteur à courant continu

- En utilisant le montage de la partie 1 taper le programme suivant dans un sketch Arduino IDE :



```
TP2_partie2
// TP 2: Variation de vitesse d'un mcc
// Partie 2

// the setup function runs once when you press reset or power the board

int pwm_val = 0; // for incoming PWM serial val

void setup() {

  // initialize digital pin LED_BUILTIN as an output.
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  Serial.begin(9600);

}

// the loop function runs over and over again forever

void loop() {

  if (Serial.available() > 0) {
    pwm_val = Serial.parseInt();
    Serial.println(pwm_val);
  }

  analogWrite(9, pwm_val);
  analogWrite(10, 0);

}
```

- Téléverser le programme puis lancer le moniteur série et envoyer des commandes PWM vers la carte Arduino.

