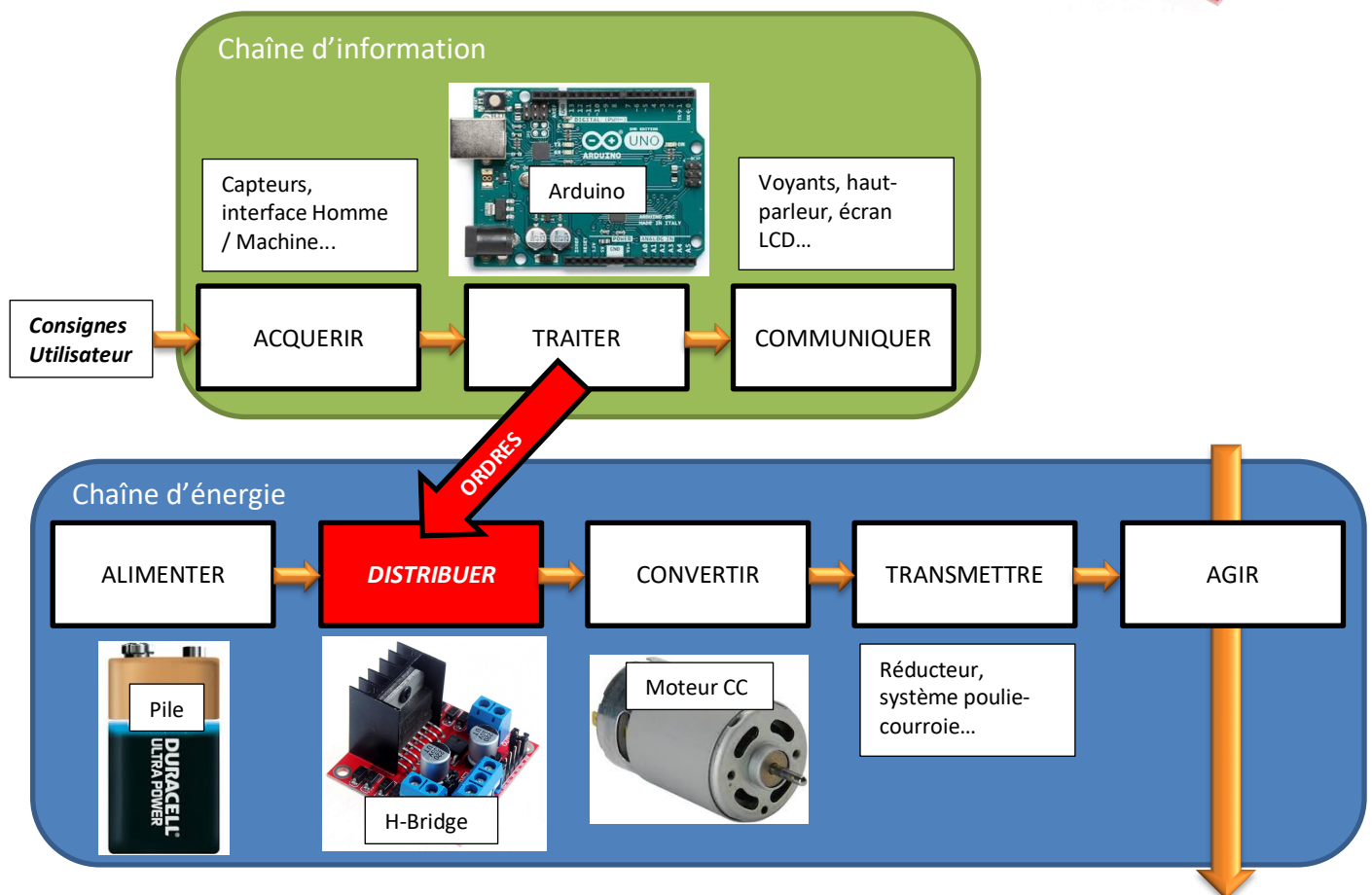
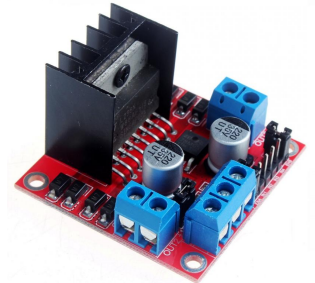


1. Mise en situation

1.1. Introduction

Les microcontrôleurs traitent des informations, mais ne sont généralement pas conçus pour commander directement des organes de puissance tels que les moteurs.

L'interface entre une carte Arduino et un moteur CC peut s'effectuer via un module H-Bridge (pont en H) L298N, qui réalise dans ce cas la fonction distribuer :



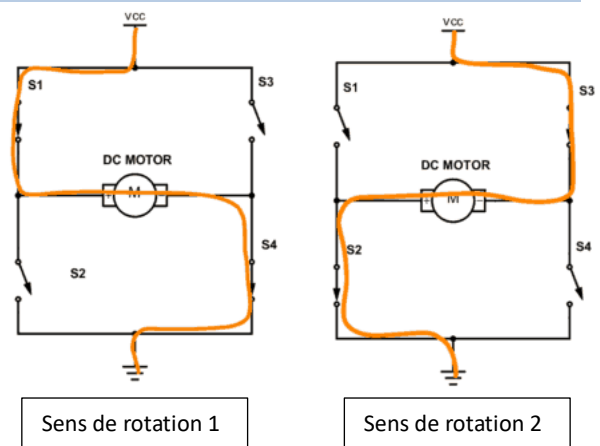
1.2. Fonctionnement du module L298N

1.2.1. Commande en pont en H (d'après Wikipedia)

https://fr.wikipedia.org/wiki/Pont_en_H

Le pont en H est une structure électronique servant à contrôler la polarité aux bornes d'un dipôle. Il est composé de 4 éléments de commutation généralement disposés schématiquement en une forme de H d'où le nom. Les commutateurs peuvent être des **relais**, des **transistors**, ou autres éléments de commutation en fonction de l'application visée.

En fonction de l'état des éléments de commutation, la **polarité aux bornes du dipôle peut s'inverser**. On peut ainsi commander le **sens de rotation** d'un moteur CC.



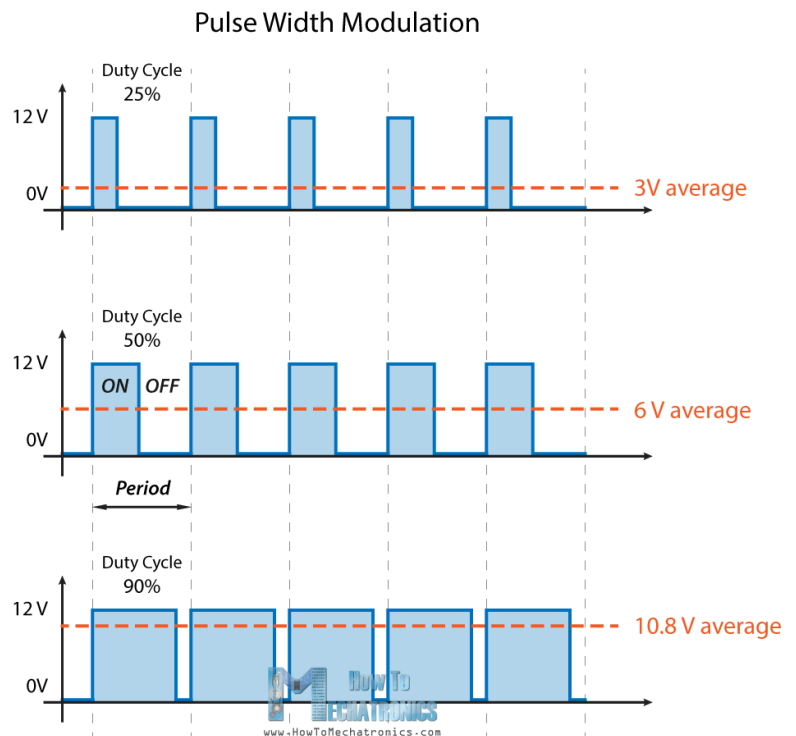
1.2.2. Commande en modulation de largeur d'impulsion (MLI - d'après Wikipedia)

https://fr.wikipedia.org/wiki/Modulation_de_largeur_d%27impulsion

En anglais : Pulse Width Modulation (PWM)

Le principe est de créer un signal logique (valant 0 ou 1), à fréquence fixe mais dont le **rapport cyclique (duty cycle)** est contrôlé numériquement, la valeur moyenne de ce signal étant une grandeur analogique, égale au produit du rapport cyclique par l'amplitude maximale du signal.

Lors de la commande d'un moteur CC, la commande PWM permet alors de faire varier la tension moyenne et donc la vitesse de rotation.



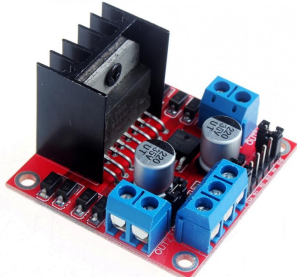



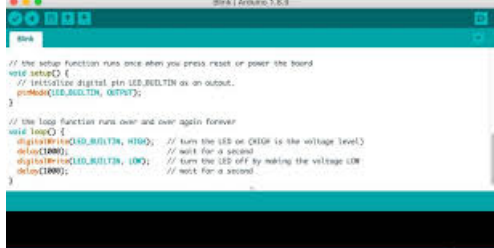
1.3. Objectif du tutoriel

On souhaite réaliser la commande d'un moteur CC avec une carte Arduino par l'intermédiaire du module H-Bridge L298N :

- **Objectif 1 :** Modifier le sens de rotation
- **Objectif 2 :** Faire varier la vitesse de rotation

1.4. Matériel et logiciels utilisés

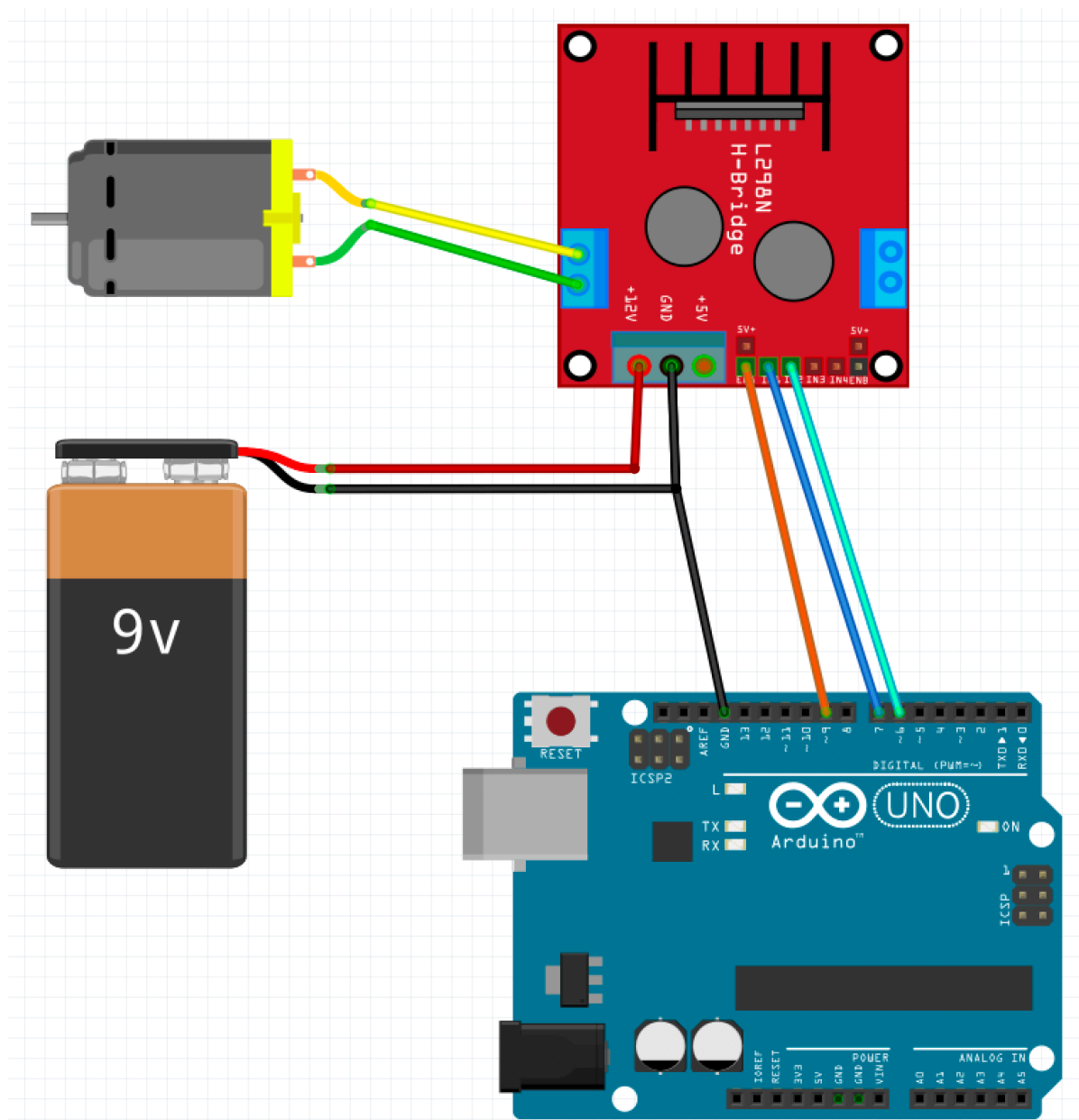
| Désignation | Illustration | Fonction |
|-------------------|---|---|
| Carte Arduino UNO |  | Exécuter le programme afin réaliser les interactions entre les différents équipements |
| Pile 9V |  | Alimenter en énergie électrique |
| Module L298N |  | Réaliser la fonction distribuer |

| | | |
|-------------|---|--------------------------------|
| Moteur CC |  | Réaliser la fonction convertir |
| Arduino IDE |  | Programmation de l'Arduino |

2. Câblage

Note : Le câblage doit se faire HORS TENSION (Arduino débranché et pile retirée).

FAIRE VERIFIER VOTRE MONTAGE AVANT LA MISE SOUS TENSION !!



3. Programmation Arduino

Ouvrir un nouveau programme sur l'IDE Arduino.

| | |
|---|--|
| Début du programme : Déclaration des pins de commande du module L298N | <pre>// Arduino DC Motor Control - PWM H-Bridge L298N - Example 01 #define enA 9 // Pin de commande PWM #define in1 6 // Pin de controle de sens #define in2 7 // Pin de controle de sens</pre> |
| Fonction setup() : Déclaration du mode des pins (OUTPUT = sortie) | <pre>void setup() { pinMode(enA, OUTPUT); pinMode(in1, OUTPUT); pinMode(in2, OUTPUT); }</pre> |
| Fonction principale loop() : <ul style="list-style-type: none"> <u>Partie 1 : Vitesse maximale dans le sens 1</u> <ul style="list-style-type: none"> Appel de la fonction sens1() Écriture de la vitesse maximale (255) sur la pin de commande PWM Temporisation de 5 secondes <u>Partie 2 : Arrêt du moteur</u> <ul style="list-style-type: none"> Écriture de la vitesse nulle (0) sur la pin de commande PWM Temporisation de 0.5 secondes <u>Partie 2 : Accélération progressive dans le sens 2</u> <ul style="list-style-type: none"> Appel de la fonction sens2() Appel de la fonction accel() <u>Partie 4 : Arrêt du moteur</u> | <pre>void loop() { sens1(); analogWrite(enA, 255); delay(5000); analogWrite(enA, 0); delay(500); sens2(); accel(); analogWrite(enA, 0); delay(500); }</pre> <p>Partie 1</p> <p>Partie 2</p> <p>Partie 3</p> <p>Partie 4</p> |
| Fonction accel() : Boucle « while » qui incrémente toutes les 50 ms une variable « i » entre 0 et 255. A chaque boucle, la variable « i » est écrite sur la pin de commande PWM. | <pre>void accel() { int i = 0; while (i < 255) { i++; analogWrite(enA, i); delay(50); } }</pre> |
| Fonction sens1() : Écriture des valeurs des pin de contrôle de sens | <pre>void sens1() { digitalWrite(in1, HIGH); digitalWrite(in2, LOW); }</pre> |
| Fonction sens2() : Écriture des valeurs des pin de contrôle de sens | <pre>void sens2() { digitalWrite(in1, LOW); digitalWrite(in2, HIGH); }</pre> |

Téléverser le programme sur la carte Arduino et vérifier le fonctionnement.