

TD N°1 SYSTMES DE COMMUNICATION

Exercice 1

On désire commander deux moteurs dans le même sens, en utilisant la figure 1 compléter le schéma de câblage (figure2) et le programme 1.

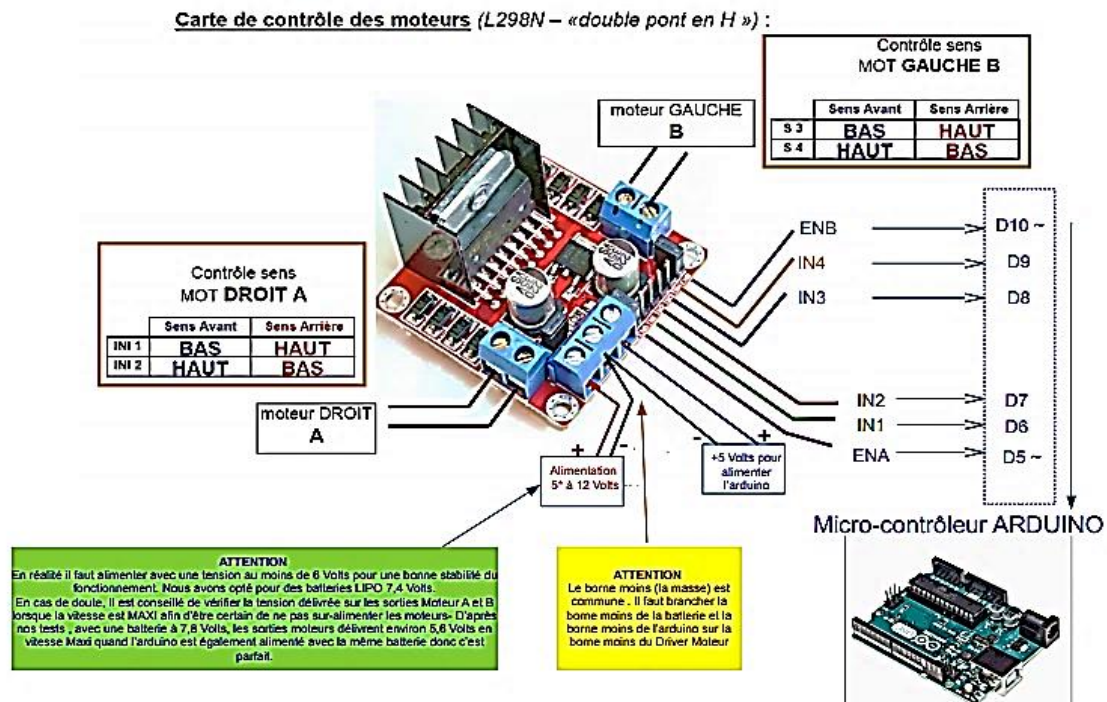


Figure 1

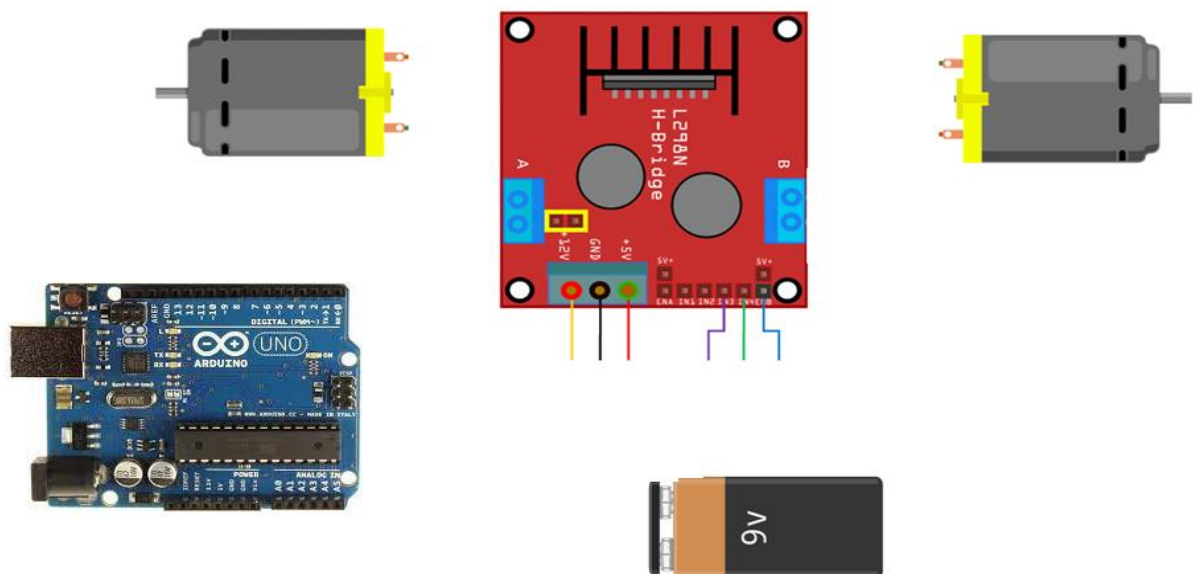


Figure 2 : Schéma de câblage

Programme 1 :

```
// Moteur 1
int IN1 = 5;
int IN2 = 6;
```

```
// Moteur 2
int IN3 = 10;
int IN4 = 11;
```

```
void setup() {
  pinMode(---, ---);
  -----;
  -----;
  -----;
}
```

```
void loop() {
```

```
  // Marche Avant
  digitalWrite(---, 1);
  digitalWrite(---, 0);
  digitalWrite(---, 1);
  digitalWrite(---, 0);
```

```
  // Attente 2s
  -----;
```

```
  // Arrêt des moteurs
  digitalWrite(---, 0);
  -----
  -----
  -----
  -----
```

```
  // Attente 1s
  -----;
```

```
  // Marche arrière
  -----
  -----
  -----
  -----
  -----
  -----
```

```
  // Attente 2s
  -----
```

```
  // Arrêt des moteurs
  -----
  -----
  -----
  -----
```

```
  // Attente 1s
  -----
```

```
}
```

Exercice 2

Le montage à réaliser devra comporter :

- Un servomoteur qui jouera le rôle de barrière
- Un bouton pour demander l'ouverture de la barrière
- Un feu bicolore qui passera au vert lorsque la barrière sera complètement ouverte

Le scénario sera le suivant :

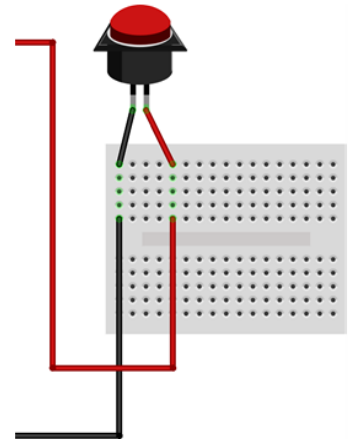
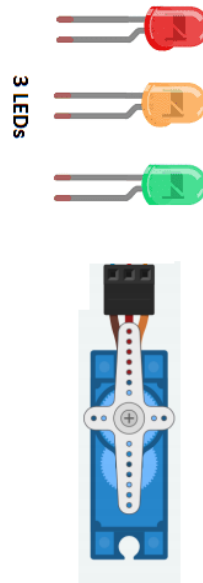
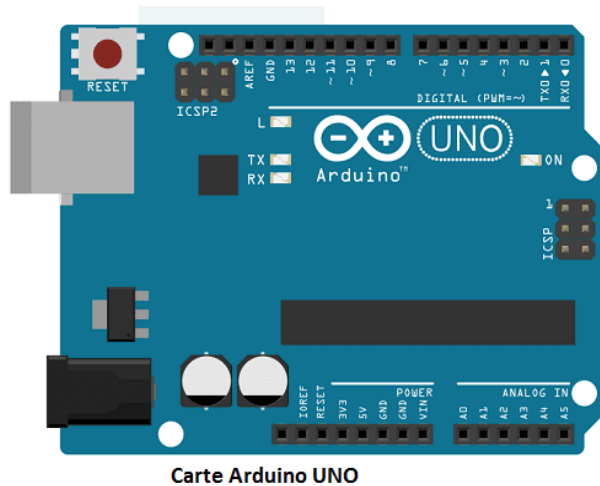


Le fonctionnement normal est un feu allumé au rouge et une barrière fermée (0°).

Le fonctionnement normal est interrompu par l'appui sur un bouton poussoir.

Si l'appui du bouton est détecté, alors la barrière (actionnée par le servomoteur) se relève doucement ; Lorsque la barrière est à la verticale (90°), le feu vert s'allume pendant 5 secondes pendant lesquelles la barrière reste ouverte (90°). Après les 5 secondes, le feu repasse au rouge, la barrière redescend doucement et le fonctionnement normal reprend.

Question : Compléter le schéma de câblage (figure3) et le programme2.

**Figure 3****Programme 2 :**

```
#include <Servo.h> //Appel de la librairie "SERVO"

----- // Crée l'objet SERVO pour contrôler le SERVOMOTEUR
----- // variable qui enregistre la position du SERVO
// définition des bornes d'entrée et sortie
----- //La led rouge est connectée à la PIN 3
----- //La led verte est connectée à la PIN 4
----- //Le bouton est connecté à la PIN 2
----- //Le servo est connecté à la PIN 9
----- // Vérification de l'état du bouton

void setup() {
  -----
  ----- // Définition des entrées/sorties
  -----
}

void loop() {
  // On initialise dès le début l'état des led

  -----
  -----
  -----
  ----- // Lit l'état du bouton poussoir
  -----// si bouton appuyé, alors on lève la barrière
  et on fait changer l'état des led, sinon, on reboucle

  ----- // on commence à actionner dans
  // un sens la barrière, jusqu'à 90°, degré par degré
  ----- // va à la position "POS"
```

```

----- // attends 15ms

}
// la barrière est levée, on peut allumer la led verte et éteindre la rouge
-----
----- // Attends 5 secondes
// La led rouge s'allume et on éteint la verte
-----
-----
//On referme ensuite la barrière
-----// cette fois si, on passe de 90° à 0°
// (décroit à chaque fois de -1)
myservo.write(pos); // va à la position "POS"
delay(15); // attends 15ms
}
}
}

```

Exercice 3

Q1 : Parmi ces affirmations, cochez celles qui vous paraissent justes :

- | | |
|--|--|
| <input type="checkbox"/> Le BUS I2C est un BUS parallèle | <input type="checkbox"/> Le BUS I2C est unidirectionnel |
| <input type="checkbox"/> Le BUS I2C est un BUS série | <input type="checkbox"/> Le BUS I2C permet le transfert à 100kb ^{s-1} |
| <input type="checkbox"/> Le BUS I2C est un BUS synchrone | <input type="checkbox"/> Le BUS I2C est un bus adressable |
| <input type="checkbox"/> Le BUS I2C est un BUS asynchrone | <input type="checkbox"/> Plusieurs octets peuvent être transmis en une seule trame |
| <input type="checkbox"/> N'importe quel circuit peut prendre la main sur le BUS I2C pour communiquer | <input type="checkbox"/> On ne peut relier que deux circuits sur un BUS I2C |

Q2 : Combien de signaux sont utilisés sur un BUS I2C ? Donnez leur nom et leur rôle :

Q3 : Comment sont reliés les différents circuits I2C sur le BUS ?

Q4 : Comment un circuit peut-il prendre la main sur le BUS ? Comment appelle-t-on cette opération dans la trame ?

Q5 : Comment un circuit libère-t-il le BUS ? Comment appelle-t-on cette opération dans la trame ?

Q6 : Quelle est la condition pour placer un bit (0 ou 1) sur le signal SDA ? A quel moment ce bit sera-t-il lu par l'esclave ?

Q7 : Quel est l'élément de transmission permettant de vérifier que la communication des requêtes entre deux circuits se passe bien ?

Q8 : Sur combien de bits est codée une adresse I2C ? Quel est le rôle du bit R/W ?

Q9 : Pourquoi une communication entre Maître / Esclave commence-t-elle toujours par une écriture ?