

TP2 : Mouvement Va_Et_Vient d'un chariot

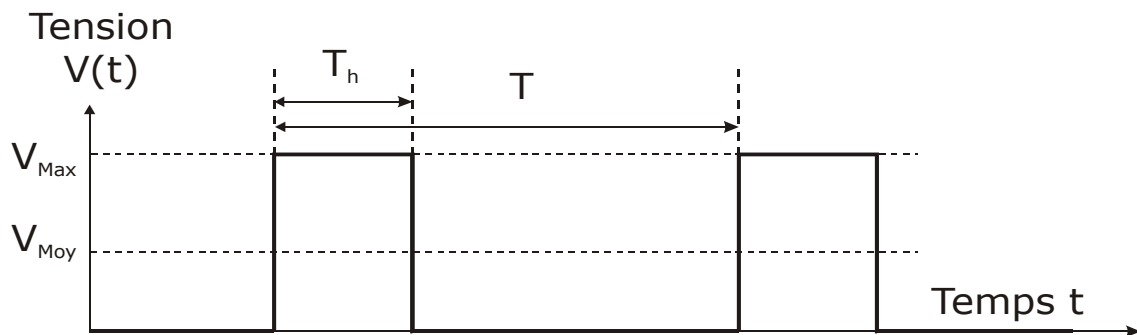
Objectif du TP :

- ✿ Schéma de câblage des différents composants avec la carte ARDUINO MEGA.
- ✿ Exploitation du capteur de distance à ultrason comme détecteur de proximité.
- ✿ Utilisation de l'IDE ARDUINO pour la programmation (transcription) d'un GRAFCET.
- ✿ Utilisation de la commande PWM comme régulateur de vitesse à boucle ouverte.
- ✿ Utilisation du pont H pour la commande et le changement du sens de rotation d'un MCC.

1. Introduction : Commande par PWM ou MLI

1.1 Définition

La commande par PWM (Pulse Width Modulation) ou MLI (Modulation en Largeur d'Impulsions) est un signal numérique carré de période T où la tension ne peut prendre que les logiques "0" et "1". Le niveau bas "0" correspond généralement à 0 Volt et le niveau haut "1" correspond à V_{Max} . La durée de l'impulsion pour laquelle la tension est à l'état haut est notée T_h .

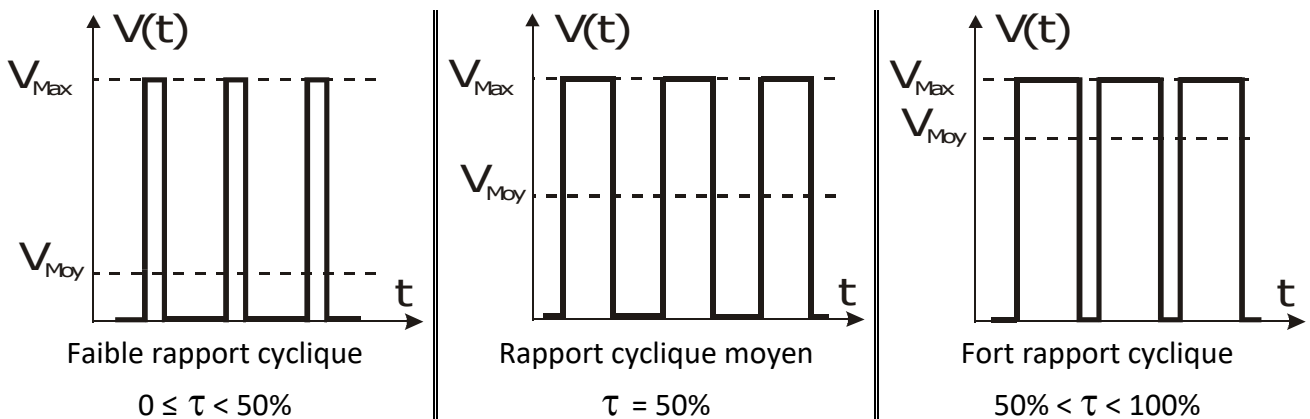


$$\begin{cases} V(t) = V_{Max} & \text{si } 0 \leq t \leq T_h \\ V(t) = 0 & \text{si } T_h < t \leq T \end{cases} \quad \left\{ \begin{array}{l} T : \text{ Période du signal} \\ T_h : \text{ Durée du signal à l'état haut} \\ V_{Max} : \text{ Tension maximale} \\ V_{Moy} : \text{ Tension moyenne} \end{array} \right.$$

$$V_{Moy} = \frac{1}{T} \int_0^T V(t) dt = \frac{1}{T} [V_{Max} \cdot t]_0^{T_h} + 0 = \frac{T_h}{T} V_{Max}$$

On définit le rapport cyclique $\tau = \frac{T_h}{T}$ [%] ; $0 \leq \tau \leq 1$; Alors, $V_{Moy} = \tau V_{Max}$.

$$\tau = \frac{T_h}{T} = \frac{V_{Moy}}{V_{Max}}$$



1.2 Principe de fonctionnement

La variation de la vitesse d'un MCC se fait par la variation de la tension d'alimentation aux bornes de l'induit. Cependant, pour les faibles valeurs de la tension, le moteur a du mal à démarrer et à conserver sa vitesse. En effet, les MCC exigent une tension minimale afin de vaincre surtout le couple résistant de démarrage (couple des pertes T_p) : frottement mécaniques et résistances ferromagnétiques. Il devient donc difficile d'ajuster les faibles vitesses de façon précise. La solution à ce problème consiste à utiliser une commande par PWM. Il suffit de fournir au moteur la tension maximale qui ne sera appliquée que par de très courts instants. En ajustant le rapport cyclique τ , on arrive à faire varier la vitesse du MCC, même pour les faibles vitesses. Selon le type des moteurs (temps de réponse), la fréquence du signal PWM varie de 100 Hz à 30 KHz. Si la fréquence du signal PWM devient importante, le MCC risque de ne pas fonctionner correctement vu son faible temps de réponse.

1.3 Application pour un moteur à aimant permanent

$$U_i = E + R_i I_i \rightarrow E = U_i - R_i I_i$$

$$E = K_i \Phi \Omega \rightarrow \Omega = E / K_i \Phi \rightarrow \boxed{\Omega = (U_i - R_i I_i) / K_i \Phi}$$

$$\text{Puisque } U_i = U_{\text{Moy}} = \tau \cdot U_{\text{max}} \quad \text{donc } \boxed{\Omega = (\tau U_{\text{max}} - R_i I_i) / K_i \Phi}$$

si τ augmente (diminue) alors Ω augmente (diminue)

1.4 Programmation de τ sur les microcontrôleurs

Pour programmer le rapport cyclique τ , on écrit la valeur de T_h correspondante dans un registre du microcontrôleur.

$$\boxed{\tau = \frac{T_h}{T} = \frac{V_{\text{Moy}}}{V_{\text{Max}}}}$$

La période du signal dépend du nombre de bits du CAN n . $T = 2^n$, d'où

$$\boxed{T_h = T \frac{V_{\text{Moy}}}{V_{\text{Max}}} = 2^n \frac{V_{\text{Moy}}}{V_{\text{Max}}}}$$

Si T_h n'est pas entier, $T_h = \text{ARRONDI}(T_h)$.

Exemple : pour les cartes ARDUINO, $n = 8$; $V_{\text{Max}} = 5 \text{ V}$

Si on veut avoir en sortie une tension $V_{\text{Moy}} = 3.8 \text{ V}$

$$T_h = 2^8 * (3.8/5) = 194.56 \text{ d'où } T_h = \text{ARRONDI}(194.56) = 195.$$

La valeur à écrire dans le registre est $T_h = 195$, on aura en sortie exactement $V_{\text{Moy}} = 3.809 \text{ V}$

2. Problématique : Mouvement Va_Et_Vient d'un chariot

Un chariot, équipé de deux capteurs de fin de course placés à ses bords droit et gauche, effectue un mouvement de Va_Et_Vient le long d'un axe rectiligne. Le premier capteur de fin de course (à droite) est un capteur de distance à ultrason jouant le rôle d'un détecteur de proximité. Le deuxième capteur de fin de course (à gauche) est un bouton poussoir.

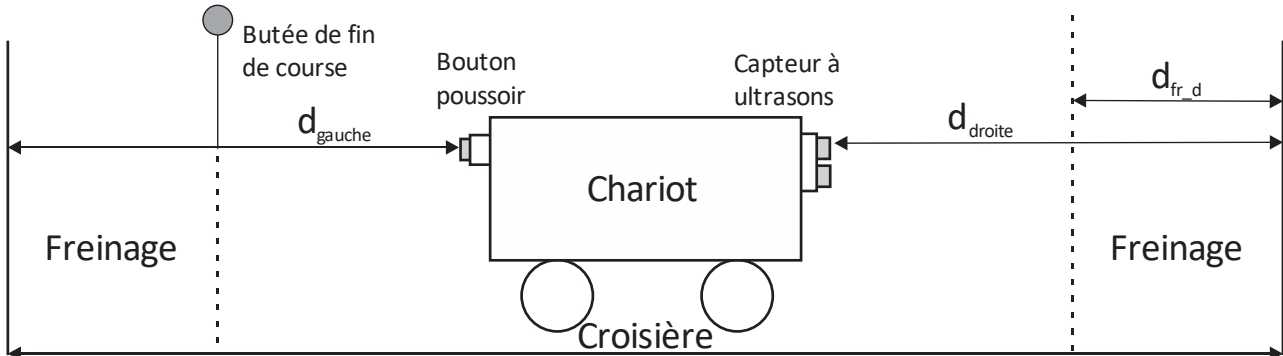


Fig. 1 : Mouvement Va_Et_Vient du chariot

Légende:

d_{fr_d} : Distance de freinage droite

d_{droite} : Distance séparant le chariot du bord droit.

3. Fonctionnement du chariot

3.1 Mode 1 : Arrêt immédiat, démarrage immédiat

Phase 1 : Le chariot se déplace à $V_{croisière}$ (vitesse de croisière) vers la droite.

Phase 2 : Quand $d_{droite} \leq d_{fr_d}$, le chariot s'arrête immédiatement et reste immobile pendant t_{imm} .

Phase 3 : Le chariot reste immobile pendant t_{imm} .

Phase 4 : Le chariot redémarre immédiatement en sens inverse vers la gauche.

Phase 5 : Le chariot se déplace à $V_{croisière}$ (vitesse de croisière) vers la gauche

Phase 6 : A l'appui sur le bouton poussoir, le chariot s'arrête immédiatement.

Phase 7 : Le chariot reste immobile pendant t_{imm} .

Phase 8 : Le chariot redémarre immédiatement en sens inverse vers l'autre borne.

3.2 Mode 2 : Démarrage avec accélération, arrêt avec décélération, allumage des LED

Phase 1 : Le chariot se déplace à $V_{croisière}$ (vitesse de croisière) vers la droite.

Phase 2 : Quand $d_{droite} \leq d_{fr_d}$, le chariot s'arrête avec une phase de décélération. La vitesse doit diminuer progressivement pendant $t_{déc}$ jusqu'à ce que chariot s'arrête.

Phase 3 : Le chariot reste immobile pendant t_{imm} .

Phase 4 : Le chariot redémarre en sens inverse avec une phase d'accélération. La vitesse augmente progressivement pendant t_{acc} jusqu'à atteindre $V_{croisière}$.

Phase 5 : Le chariot se déplace à $V_{croisière}$ (vitesse de croisière) vers la gauche.

Phase 6 : A l'appui sur le bouton poussoir, le chariot s'arrête avec une phase de décélération. La vitesse doit diminuer progressivement pendant $t_{déc}$ jusqu'à ce que chariot s'arrête.

Phase 7 : Le chariot reste immobile pendant t_{imm} .

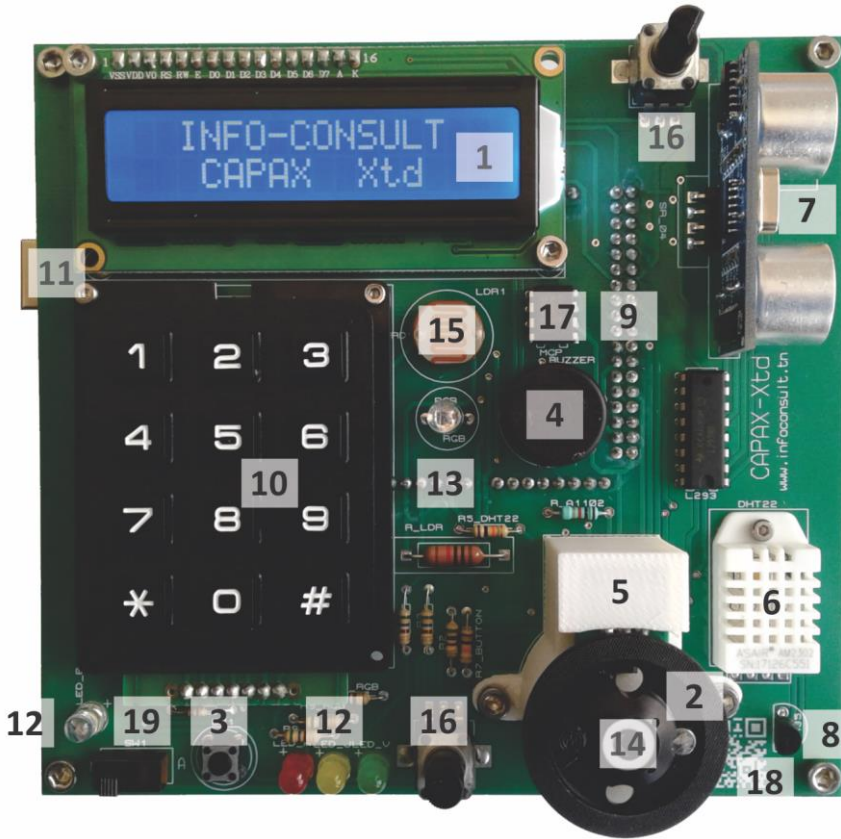
Phase 8 : Le chariot redémarre en sens inverse avec une phase d'accélération. La vitesse augmente progressivement pendant t_{acc} jusqu'à atteindre $V_{croisière}$.

Remarques :

- ✿ Pour les 2 modes, le mouvement de Va_Et_Vient se répète indéfiniment.
- ✿ Le chariot démarre à droite en premier lieu.

t_{imm}	t_{acc}	$t_{déc}$	d_{fr}	$V_{croisière}$	V_{son}	Allumage LED	Accélération	Croisière	Décélération	Arrêt
3 s	4 s	5 s	12 cm	160	344.7 m/s	Vert	1	1	0	0
						Orange	1	0	1	0
						Rouge	0	0	1	1

4. Utilisation des composants du TP sur la carte CAPAX-Xtd®



1. Afficheur LCD 16*2
2. Aimant
3. Bouton poussoir
4. Buzzer
5. Capteur à effet HALL
6. Capteur DHT22
7. Capteur distance US
8. Capteur LM35
9. Carte ARDUINO MEGA
10. Clavier 12 Touches
11. Connecteur USB
12. Diodes LED
13. Diode RGB
14. Moteur MCC
15. Photorésistance LDR
16. Pot. analogique
17. Pot. numérique (SPI)
18. QR code
19. Sélecteur

Légende		E/S correspondante sur la carte ARDUINO
7	HC-SR04 TRIG ECHO	D23 (entrée numérique) D25 (sortie numérique)
12	LED Verte	D42 (sortie numérique)
12	LED Orange	D44 (sortie numérique)
12	LED Rouge	D43 (sortie numérique)
16	Bouton poussoir	D47 (entrée numérique)
14	MCC et C.I L293 (pont H) Enable In1 In2	D2 (sortie analogique) : vitesse de rotation D3 (sortie numérique) : sens de rotation* D5 (sortie numérique) : sens de rotation*

* Le sens de rotation du moteur est défini selon le tableau suivant :

In1	In2	Sens de rotation	Sens de déplacement du chariot
HIGH	LOW	Horaire	vers la droite
LOW	HIGH	Trigonométrique	vers la gauche

Attention : Le moteur doit tourner à vide. Ne pas mettre le CD-ROM charge moteur.

4. Déroulement du TP

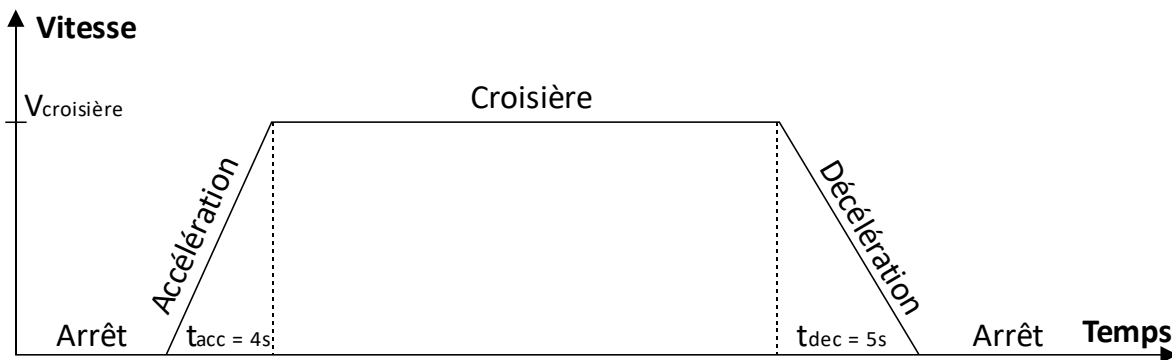
- 4.1 Faire une recherche sur Internet sur le principe de fonctionnement du pont H.
 4.2 En respectant la notice du constructeur (datasheet du CI L293) et le brochage des entrées-sorties comme spécifié dans le tableau ci-dessous, terminer le câblage de tous les composants avec la carte Arduino MEGA (Voir schéma détaillé dans l'annexe).

4.3 Etude du mode 1

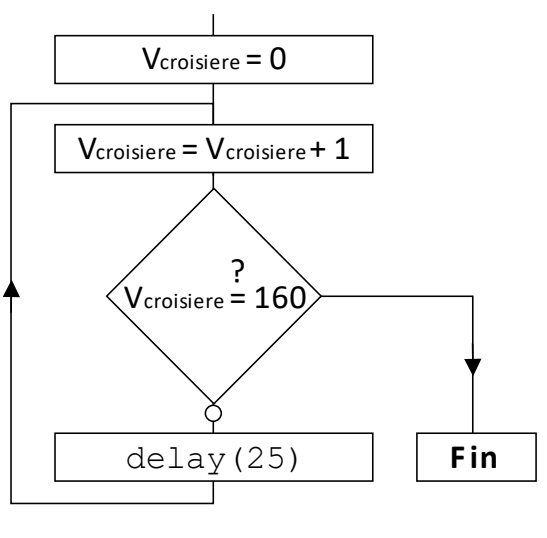
- 4.3.1 Compléter le GRAFCET mode 1 (voir annexe).
 4.3.2 En déduire le logigramme ainsi que le code ARDUINO pour le mode 1

4.4 Etude du mode 2.

- 4.4.1 Etude des phases accélération et décélération



En phase de d'accélération, le chariot doit atteindre la vitesse $V_{\text{croisière}} = 160$ en $t_{\text{acc}} = 4$ s. Donc, on doit augmenter graduellement la vitesse de "1" jusqu'à atteindre $V_{\text{croisière}} = 160$ et patienter à chaque augmentation une $\text{Tempo} = 4\text{s}/160 = 0.025$ s = 25 ms. L'instruction `delay (Tempo)` permet de patienter pendant un temps Tempo [ms].



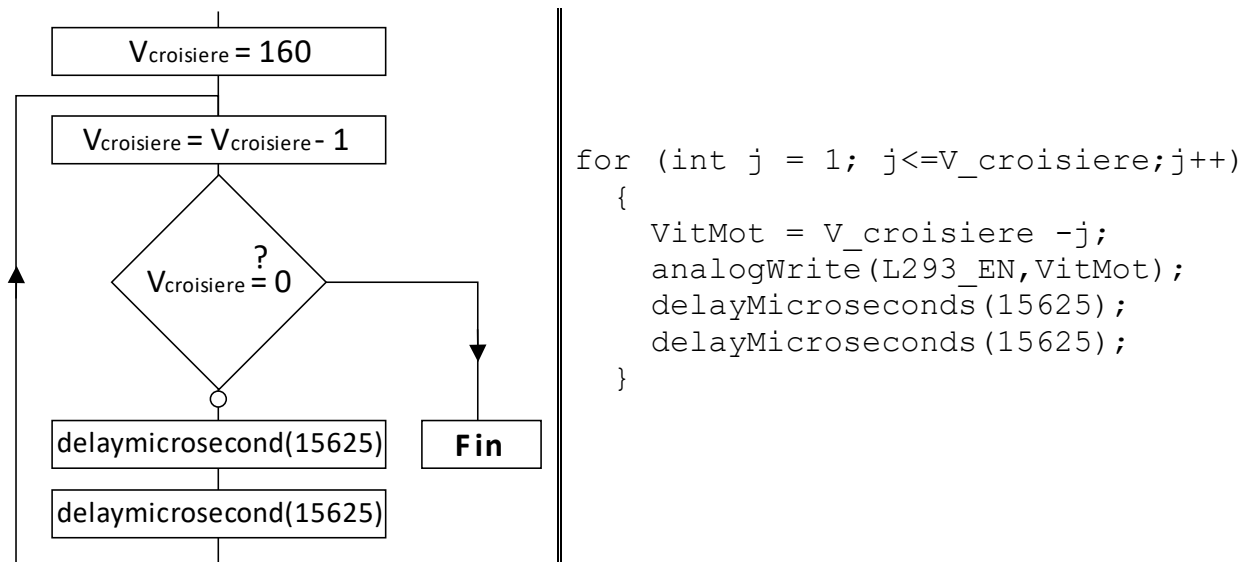
```

for (int j = 1; j<=V_croisiere;j++)
{
  VitMot = j;
  analogWrite(L293_EN,VitMot);
  delay(25);
}
  
```

En phase de décélération, le chariot doit passer de $V_{\text{croisière}} = 160$ à $V_{\text{croisière}} = 0$ en $t_{\text{dec}} = 5\text{s}$.

Donc, on doit diminuer graduellement la vitesse de "1" jusqu'à atteindre $V_{\text{croisière}} = 0$ et patienter à chaque diminution une $\text{Tempo} = 5\text{s}/160 = 0.03125\text{ s} = 31.25\text{ ms}$.

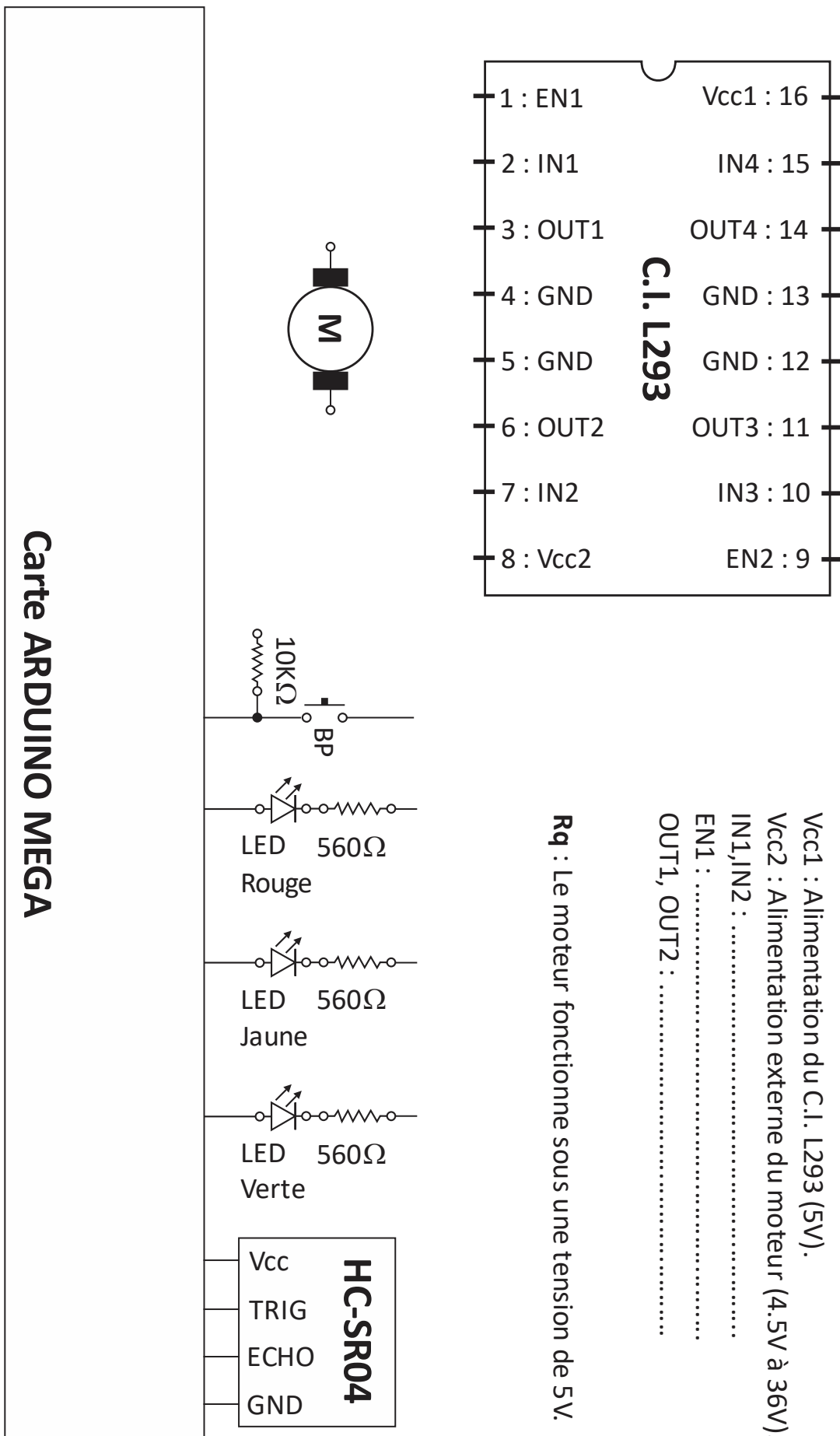
Comme l'instruction `delay` ne peut pas patienter les sous-multiples de la milliseconde, on utilisera l'instruction `delaymicrosecond` qui permet de patienter le temps en μs , donc $\text{Tempo} = 31250\ \mu\text{s}$. Or, la valeur maximale du temps dans l'instruction `delaymicrosecond` ne peut pas dépasser 16383. Donc on divise Tempo par 2 ($31250/2=15625$) et on utilisera 2 fois de suite l'instruction `delaymicrosecond(15625)`.



4.4.2 Compléter le GRAFCET mode 2 (voir annexe).

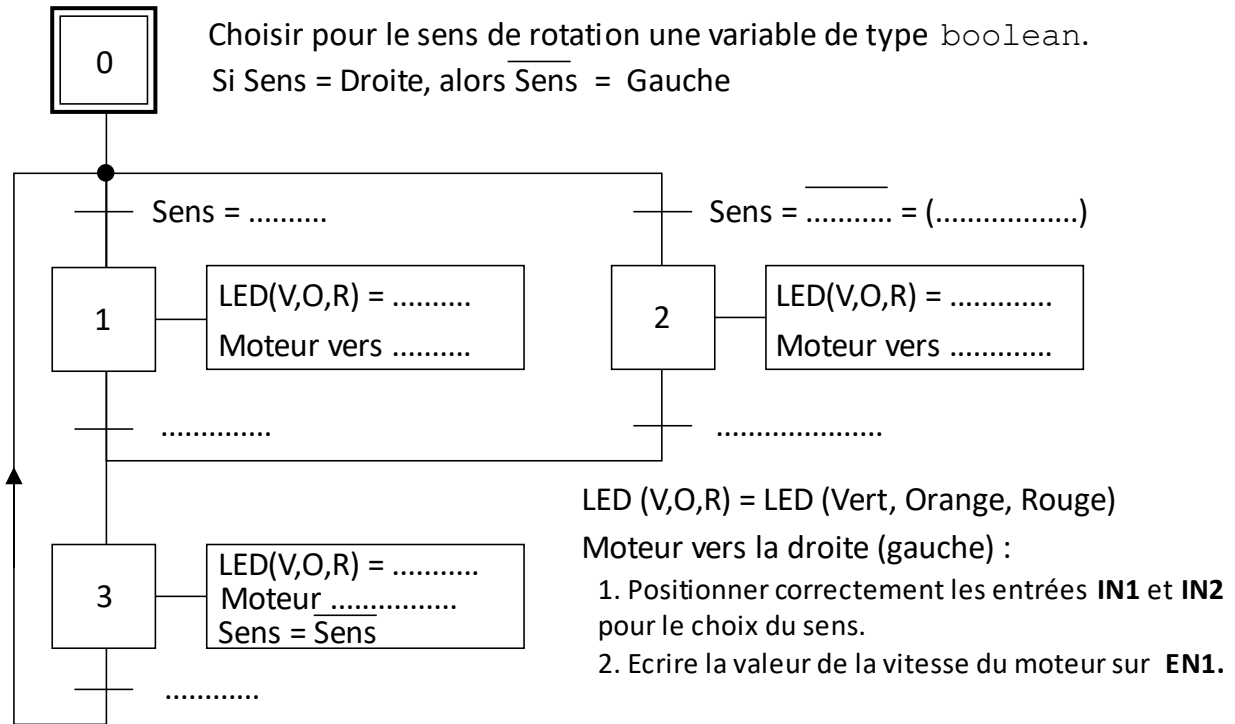
4.4.3 En déduire le logigramme ainsi que le code ARDUINO pour le mode 2

Annexe 1 : A remettre avec le compte rendu du TP.



Annexe 2 : A remettre avec le compte rendu du TP.

Grafcet mode 1



Annexe 3 : A remettre avec le compte rendu du TP.

Grafcet mode 2

