

Connaissances	Diagramme d'état
Capacités	Régler les paramètres de fonctionnement d'un système Générer un programme et l'implanter dans le système cible
Activités (1 H)	Compléter un diagramme d'état Implanter le programme dans un microcontrôleur Arduino Mega

Ressources documentaires	Dossier technique scooter électrique
Ressources matérielles	Maquette mesure de vitesse Micro-ordinateur avec logiciel MATLAB

Ce TP a pour objectif d'implanter un diagramme d'état dans un microcontrôleur Arduino Mega afin de :

- réaliser la commande du moteur du scooter ;
- mesurer et afficher la vitesse du scooter.

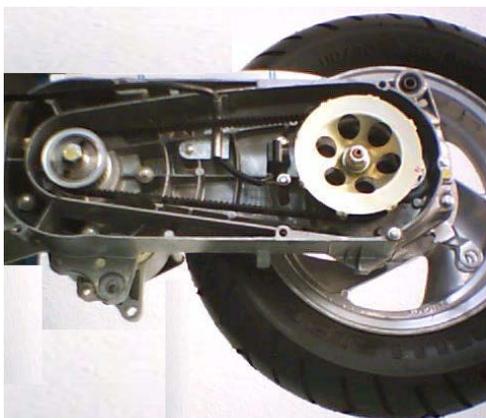


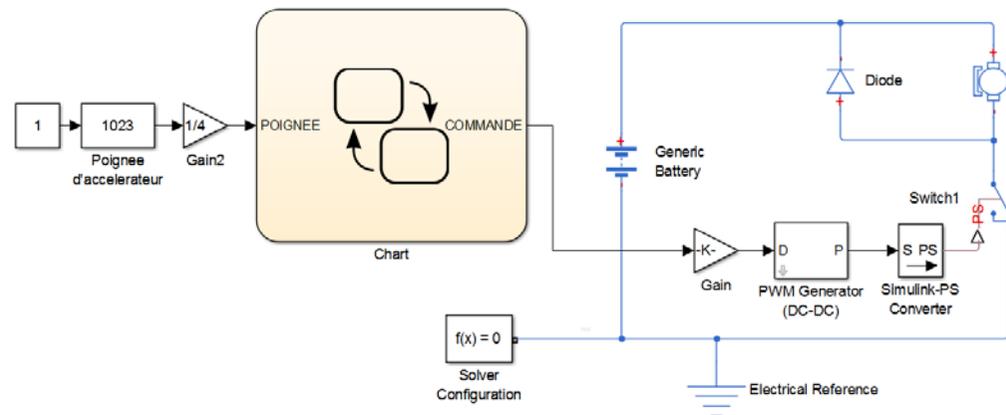
Figure 1

1. PRÉSENTATION DE LA MAQUETTE DIDACTIQUE

La maquette didactique représentée à la figure 1 reprend le fonctionnement partiel du scooter électrique. Elle intègre :

- un moteur électrique à courant continu qui entraîne le capteur de vitesse ;
- un microcontrôleur Arduino qui produit les signaux pour la commande du moteur et l'affichage de la vitesse du scooter ;
- un hacheur 1 quadrant qui module l'énergie électrique à destination du moteur ;
- un compteur de vitesse constitué de 10 leds ;
- le capteur magnétoélectrique utilisé pour la mesure de la vitesse du scooter ;
- un potentiomètre qui simule la poignée d'accélérateur ;
- un afficheur qui indique la fréquence de rotation du disque cranté.

2. MODÈLE DE SIMULATION DE LA COMMANDE DU MOTEUR



2.1. DESCRIPTION DES ENTRÉES ET SORTIES DU DIAGRAMME D'ÉTAT DE COMMANDE DU MOTEUR

SORTIES

- COMMANDE est un nombre binaire codé sur 8 bits. Il commande un hacheur dont le rapport cyclique dépend de la valeur du signal COMMANDE. Si COMMANDE = 0, le moteur est à l'arrêt. Si COMMANDE = 255, le moteur tourne à sa vitesse maximale.

ENTRÉES

- POIGNEE est un nombre binaire codé sur 8 bits. Ce nombre est l'image de la tension de sortie du potentiomètre. POIGNEE = 0 lorsque la poignée d'accélérateur est relâchée. POIGNEE = 255 lorsqu'elle est tournée au maximum.

3. DIAGRAMME D'ÉTAT DE LA COMMANDE DU MOTEUR

On donne, à la figure 2, le diagramme d'état (Chart) de la commande du moteur :

3.1. ÉTATS

Le diagramme comporte un seul état (State) désigné par le label Commande. L'action associée à cet état est écrite à l'intérieur de l'état. Elle est conclue par un point-virgule :



Figure 2

3.2. SIMULATION DU FONCTIONNEMENT

☞ Copier, sur votre lecteur personnel, le dossier *TP12* présent dans le répertoire *Documents en consultation* du lecteur classe. Double cliquer sur le fichier *commande_scooter1.slx* pour ouvrir le modèle de simulation.

☞ Dans la fenêtre du modèle, lancer la simulation pour tester le fonctionnement du modèle de la commande du scooter.

☞ Relever la valeur de la poignée. Indiquer la vitesse atteinte par le scooter

☞ Diminuer la valeur de la poignée en double cliquant dessus.

☞ Indiquer comment évolue la vitesse du scooter. Le fonctionnement simulé correspond-il au fonctionnement attendu :

3.3. CORRECTION DU DIAGRAMME D'ÉTAT

Pour corriger le problème rencontré, il faut préciser le moment où l'action doit avoir lieu en écrivant un mot-clé (*entry*: ou *exit*: ou *during*:) devant l'action :

- *entry*: l'action est effectuée à l'entrée dans l'état ;
- *exit*: l'action est effectuée à la sortie de l'état ;
- *during*: l'action est effectuée pendant toute la durée de l'état.

☞ Écrire (au-dessus de l'action figure 2) le mot-clé qui permet d'obtenir le fonctionnement voulu.

☞ Compléter le diagramme d'état conformément à la réponse trouvée.

☞ Dans la fenêtre du modèle, lancer la simulation pour tester le fonctionnement simulé du modèle de la commande du scooter.

FAIRE VÉRIFIER PAR LE PROFESSEUR.

4. PROGRAMMATION DU MICROCONTRÔLEUR

Le fonctionnement du diagramme d'état étant validé par la simulation, il est maintenant possible de l'implanter dans la carte microcontrôleur Arduino Mega. Pour cela, il faut indiquer les numéros des broches utilisées du microcontrôleur et préciser la nature des informations qu'elles traitent.

4.1. AFFECTATION DES ENTRÉES ET SORTIES DU MICROCONTRÔLEUR

☞ Sélectionner le diagramme d'état créé et copier le dans un nouveau modèle (Ctrl+N) nommé *Commande_moteur_arduino*.

☞ Cliquer sur le *Simulink library browser* afin d'accéder aux bibliothèques de composants. Aller dans *Simulink Support package for Arduino Hardware* et placer sur le modèle les blocs associés aux entrées et sorties conformément à la figure 3.

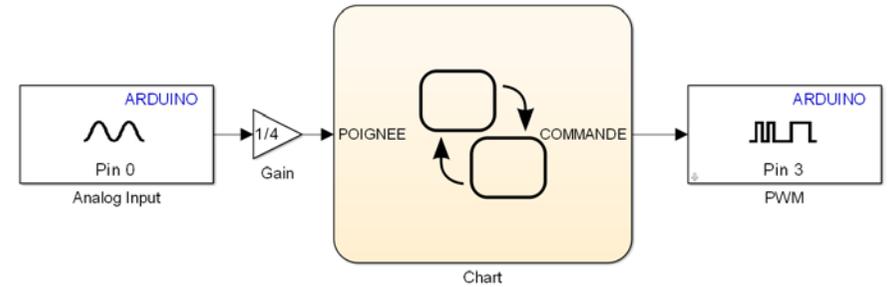
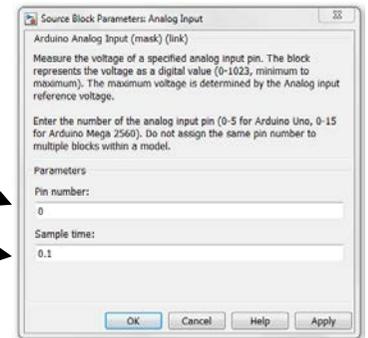


Figure 3

☞ Double cliquer sur les blocs pour affecter les numéros de broche (*Pin number*).



Pour l'entrée, régler la période de scrutation (*Sample time*) à 0,1 sec.

4.2. IMPLANTATION DU PROGRAMME DANS LE MICROCONTRÔLEUR

☞ Relier la maquette au PC par un port USB. Régler l'alimentation stabilisée à 9 V (limitée à 3 A).

FAIRE VÉRIFIER PAR LE PROFESSEUR.

☞ Dans le menu du modèle, cliquer sur *Tools > Run on Target Hardware > Prepare to run...*

☞ Choisir la cible (*Hardware Board*) : Arduino Mega 2560

Dans la même fenêtre, sélectionner *Solver* et écrire la valeur *inf* dans *Stop time*.

☞ Pour programmer le microcontrôleur, cliquer sur le bouton *Deploy to Hardware* :

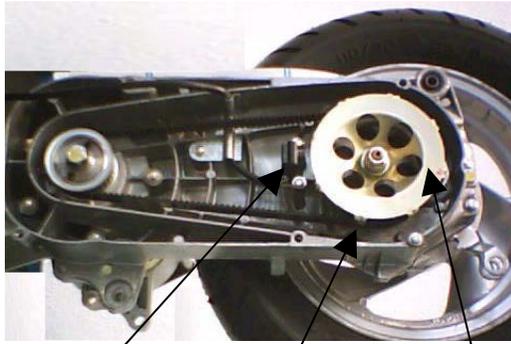


☞ Alimenter la maquette et tester le fonctionnement.

☞ La maquette a-t-elle le fonctionnement attendu ?

5. MESURE DE LA VITESSE DU SCOOTER

5.1. PRINCIPE DE LA MESURE DE VITESSE



Captur magnétoélectrique Cran Disque cranté

La mesure de la vitesse du scooter est réalisée par un capteur magnétoélectrique et un disque cranté doté de 8 crans. Le disque cranté est monté sur un pignon du réducteur de vitesse. Il provoque l'apparition d'impulsions électrique en sortie du capteur. Il y a une impulsion à chaque passage d'un cran du disque devant le capteur.

Le nombre d'impulsions produit et la fréquence du signal issu du capteur sont donc proportionnels à la vitesse du scooter.

5.2. DÉTERMINATION DE LA FRÉQUENCE DE ROTATION DU DISQUE CRANTÉ PAR SIMULATION

Reprendre le fichier *commande_scooter.slx*. Lancer la simulation.

- Pour les valeurs de la poignée données dans le tableau 1, mesurer et relever les valeurs de :
 - la vitesse du scooter ;
 - la fréquence de rotation du disque cranté.

Position de la poignée	Vitesse du scooter (en km/h)	Fréquence de rotation du disque (en tr/min)	Nombre d'impulsions du capteur en 1 sec.
0			
1023			

Tableau 1

5.3. MESURE DU NOMBRE D'IMPULSIONS DU CAPTEUR (SIMULATION EN MODE EXTERNE)

La mesure de la vitesse du scooter s'effectue en comptant les impulsions produites par le capteur en une seconde. Le modèle *mesure_vitesse.slx* inclut un diagramme d'état qui réalise ce comptage.

La mesure du nombre d'impulsions comptées chaque seconde par une simulation en mode externe. Dans ce mode de simulation, les opérations de calcul programmées sont exécutées sur l'ordinateur et non sur la maquette qui fonctionne uniquement comme interface de puissance.

- Ouvrir le fichier *mesure_vitesse.slx*. Sélectionner le mode *External* (à la place de *Normal*). Lancer la simulation.

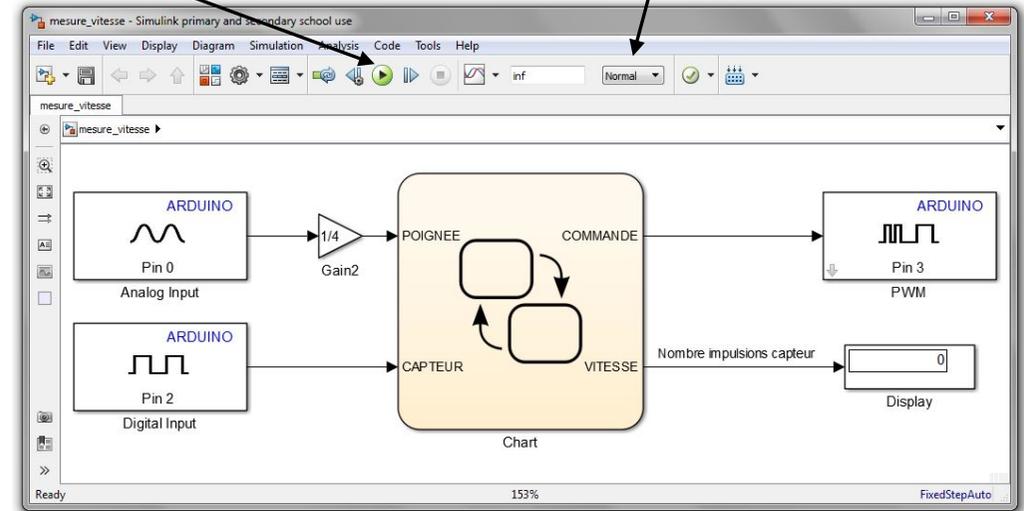


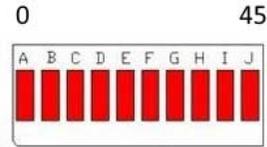
Figure 4

- Sur la maquette, tourner le potentiomètre pour régler la fréquence de rotation du disque correspondant à la vitesse de 45 km/h.

- Relever et reporter dans le tableau 1, le nombre d'impulsions comptées en une seconde (*Display* sortie *VITESSE*) :

6. AFFICHAGE DE LA VITESSE DU SCOOTER

L'affichage de la vitesse utilise un bargraphe constitué de 10 leds. La première (A) indique 0 km/h, la deuxième (B) indique une vitesse d'au moins 5 km/h, la troisième (C) une vitesse d'au moins 10 km/h... jusqu'à la dixième (J) qui indique 45 km/h.



Ces leds, de A à J, sont connectées respectivement aux sorties 4 à 13 du microcontrôleur.

Les leds sont allumées lorsqu'elles reçoivent un niveau logique '1'. Elles sont pilotées par un comparateur qui compare la valeur VITESSE à une constante.

✍ À partir des résultats du tableau 1, calculer le nombre d'impulsions correspondant à une vitesse de 5 km/h :

✍ Compléter la valeur de chaque comparateur du modèle de la figure 5 pour obtenir le fonctionnement décrit ci-dessus.

🔗 Ouvrir le fichier `affichage_vitesse.slx`.

🔗 Renseigner la valeur de la constante pour chaque comparateur conformément à la figure 6.

🔗 Programmer le microcontrôleur en utilisant le même protocole qu'au paragraphe 4.2.

FAIRE VÉRIFIER LE FONCTIONNEMENT PAR LE PROFESSEUR.

✍ La maquette a-t-elle le fonctionnement attendu ?

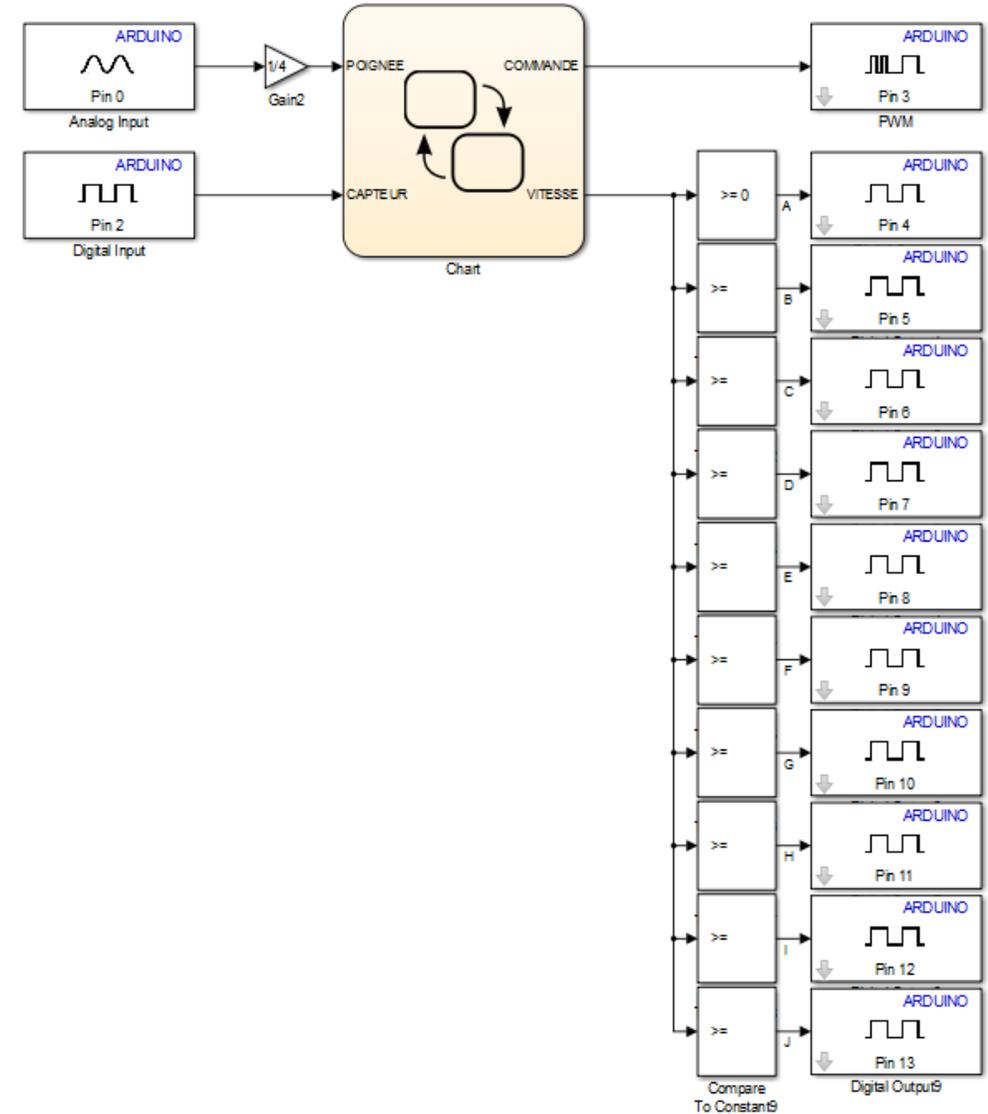


Figure 5