

Matériel et logiciels :

- Banc de traction avec alimentation et jauge de contrainte.
- Fil, poulie, ensemble de masses calibrées.
- Ressort spiral, scotch, film alimentaire, élastique.
- Logiciel Arduino, pyzo, regressi.

Objectifs du TP :

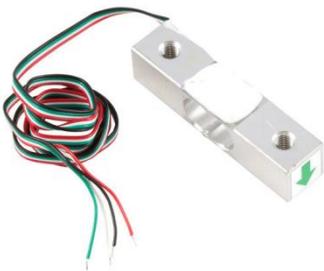
- Calibrer un dispositif expérimental de traction.
- Analyser le comportement en traction de différents matériaux.
- Mesurer l'élasticité des matériaux

I Présentation du dispositif expérimental

1.) Les différents éléments :

On dispose d'un banc de traction constitué :

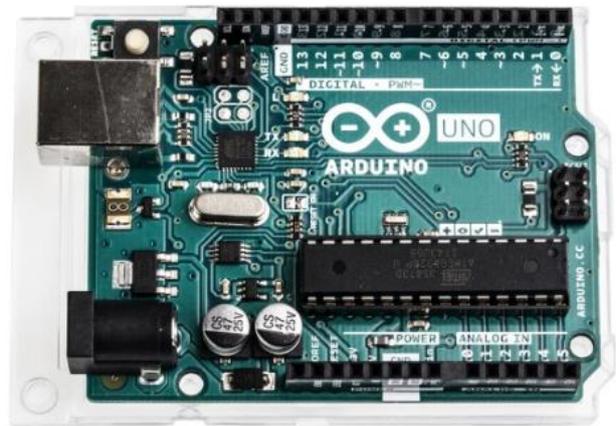
- d'un moteur pas à pas relié à une courroie permettant de mettre en mouvement un support horizontal. Le moteur pas-à-pas permet de contrôler précisément un positionnement angulaire : on positionne le rotor en modifiant la direction d'un champ magnétique créé par les bobinages du stator. Il est commandé par un dispositif qui comporte une partie logique et une commande de puissance. La partie logique détermine pour chaque pas quelles sont les bobines alimentées et le sens de rotation.
- d'un capteur de force qui délivre une tension électrique à ses bornes évoluant lorsque le capteur se déforme sous l'effet d'une action extérieure dans la direction indiquée par la flèche (dans les deux sens).
- d'n microcontrôleur Arduino associé à un dispositif de puissance permettant de commander le moteur et de récupérer les données du capteur.



Capteur de force



Moteur pas à pas



Carte Arduino

2.) Pilotage de l'expérience par Arduino :

Tout d'abord, copier le répertoire **Traction_arduino** présent dans le dossier travail de la classe dans votre dossier personnel. Ce répertoire contient trois sous-dossiers : **Calibration**, **Traction** et **Python**.

Un microcontrôleur est un système autonome programmé pour réaliser une tâche donnée.

Pour démarrer une expérience, la procédure est la suivante :

- Brancher (c'est déjà fait !) le microcontrôleur sur le port USB de l'ordinateur.
- Brancher (c'est déjà fait !) les quatre fils d'alimentation du moteur sur le boîtier, en respectant le code couleur.

NE PAS TOUCHER A CES BRANCHEMENTS !

- Fermer l'interrupteur d'alimentation présent sur le boîtier.
- Lancer le programme Arduino. Pour démarrer une expérience, ouvrir le fichier d'extension **.ino** situé dans le dossier adhoc, puis cliquer sur **téléverser** situé dans le menu **Croquis**.
- Les codes sont écrits pour générer des affichages que l'on peut visualiser via le **moniteur série** à lancer à partir de l'onglet **Outils**.

On "téléverse" dans le microcontrôleur le code informatique qui est ensuite exécuté. Le code est écrit en langage C. Vous pouvez regarder sa structure à titre informatif mais ne devez pas le modifier. Le microcontrôleur est dès lors autonome, et peut être débranché de l'ordinateur. Il faut alors prévoir un dispositif interne pour récupérer les données (comme par exemple une carte SD).

II Présentation du dispositif

Si erreur lors du téléversement, aller dans Outils et Changer le port (com 13 Arduino) ou la version d'Arduino (comparer avec un poste sur lequel ça fonctionne).

1.) Calibration du capteur de force :

Le but de cette partie est de vérifier la linéarité du capteur de force.

- Fixer une corde au capteur de force et accrocher à l'autre extrémité une masse calibrée, tout en faisant passer la corde par la poulie présente à l'extrémité du banc.
- Téléverser le script **calibration.ino** et lancer le **moniteur série**. Vous devez voir s'afficher régulièrement des valeurs U_J , proportionnelles à la tension électrique aux bornes de la jauge de contrainte.

1) Pour une masse m donnée, quelle est la valeur attendue de la force de tension T exercée par le fil sur la jauge de contrainte ?

2) Pour une masse donnée $m = 100g$, mesurer la valeur U_J . Pour cela copier une dizaine de valeurs affichées dans le moniteur série (CTRL-C) dans un fichier libreoffice **calc** (CTRL-V). Faire la moyenne de ces valeurs et la noter.

Attention : Il faut remplacer les points par des virgules dans l'expression des nombres, en utilisant dans l'onglet *Edition* : *Rechercher et remplacer*.

3) Répéter l'opération pour les quatre autres masses.

4) A l'aide du programme python **calib.py** dans le répertoire **python**, représenter la tension T en fonction de U_J . Vérifier que T est une fonction affine de U_J : $T = a U_J + b$. Relever les résultats de la modélisation.

5) Compléter alors les lignes de la fin du programme **analyse.py** présent dans le dossier **python** en définissant a et b .

Dans le script "analyse2" les résultats sont déjà rentrés

Fermer **calibration.ino**.

2.) Calibration du moteur :

La commande du moteur pas-à-pas permet de déplacer le support mobile en contrôlant son déplacement : un pas du moteur correspond à une rotation d'un angle de $1,8^\circ$. Entre deux mouvements successifs, le moteur effectue 10 pas, ce qui entraîne une rotation de la roue dentée, convertie par une courroie de transmission en une translation du support.

6) En mesurant l'écart entre deux dents de la courroie et sachant qu'il y a 18 dents au total au niveau du moteur, déterminer la distance parcourue lorsque le support effectue un mouvement. Compléter la ligne à la fin du programme **analyse.py** définissant la variable dL . *prendre la valeur dans "analyse2"*

Eteindre l'alimentation du moteur et positionner le support proche du capteur de force. Allumer l'alimentation puis téléverser le script **traction.ino** et vérifier le bon fonctionnement du dispositif : le support doit effectuer un aller / retour puis s'immobiliser (en restant bloqué car le moteur est alimenté).

Pour déplacer à la main le support, il faut d'abord éteindre l'alimentation du moteur.

III Etude en traction : ressort spiral

Couper l'alimentation du moteur quand on ne fait pas de mesures pour ne pas qu'il chauffe.

Fixer le ressort spiral au capteur de force et au support, de sorte que le ressort soit horizontal et parallèle à la courroie. Initialement, le ressort doit être légèrement tendu. Si ce n'est pas le cas, ouvrir l'interrupteur d'alimentation du boîtier, puis déplacer le support.

Attention :

- Arduino garde en mémoire le dernier programme téléversé. Ne rebrancher l'alimentation qu'une fois le dispositif en place !
- Le moteur ayant tendance à chauffer, penser à couper l'alimentation une fois les mesures effectuées.

Relancer le script **traction.ino**. Dans le moniteur série, les valeurs de tension s'affichent pour le mouvement aller puis pour le mouvement retour. Copier les données numériques correspondant au mouvement aller et stocker les dans un fichier texte (par exemple avec le logiciel **Notepad++**) que vous enregistrerez dans le dossier **Traction_arduino/python** sous le nom **ressort_aller.txt**. Faire de même pour le mouvement retour sous le nom **ressort_retour.txt**.

7) Exécuter alors le script python **analyse.py**. Que constate-t-on ? Vous pouvez sauvegarder et imprimer la courbe obtenue.

8) Déterminer alors la constante de raideur du ressort.

9) Déterminer expérimentalement la constante de raideur en effectuant plusieurs mesures d'allongement vertical sur le support fourni et en entrant les valeurs sous un tableur comme Regressi. Conclure.

On accroche au ressort les masses fournies, on trace sous regressi $P = mg$ en fonction de l'allongement.

Une regression linéaire donne la valeur de k .

Insister pour qu'ils fassent cette étude correctement plutôt qu'ils passent à la suite.

IV Etude d'autres matériaux facultatif

10) Remplacer le ressort par un élastique (non tendu au départ) et relancer le script **traction.ino**. Tracer la courbe obtenue. Que remarque-t-on ? Peut-on établir une constante de raideur de l'élastique ? L'élastique se comporte-t-il de la même façon en traction et en compression ?

11) Réitérer la procédure avec du scotch puis avec du film alimentaire. Pour chaque matériau, établir la force à ne pas dépasser pour rester dans la zone de comportement linéaire.