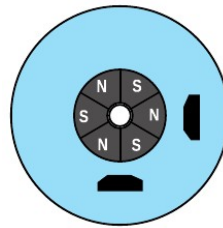
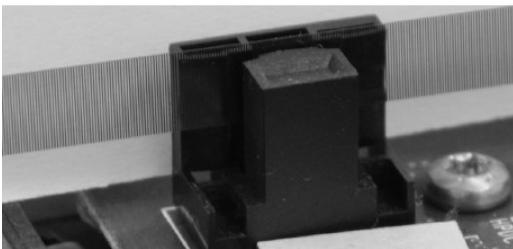


Commande numérique d'un moteur à courant continu

Devoir Maison : Décodage des signaux en quadrature d'un codeur incrémental

Un codeur incrémental est un capteur renvoyant deux signaux carrés entre 0 et 5V, en quadrature de phase, permettant de compter ou décompter des incréments de déplacement, ou de mesurer une vitesse de déplacement.

Ils sont généralement optiques (exemple de la tête d'impression d'imprimante) ou magnétiques (cas du moteur CC qui sera utilisé en TP)



Les deux signaux proviennent de deux capteurs (photo-transistors ou capteurs à effet Hall) en décalage d'un quart de période.

Les figures ci-contre représentent les deux voies d'acquisition pour un sens de rotation horaire et trigonométrique. Lors d'une inversion de sens, les signaux inversent leur déphasage relatif.

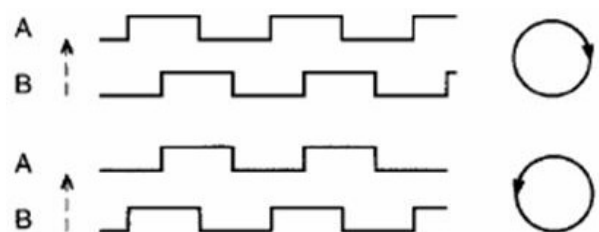
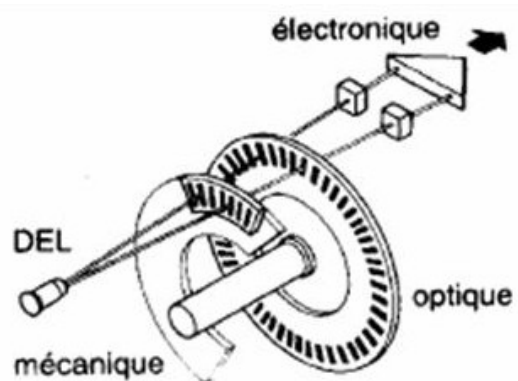
Chaque front du signal correspond à un événement que le micro-contrôleur est capable de détecter (un top). Il y a donc 4 front (4 tops) par période des signaux provenant du capteur.

Les deux voies du capteur sont branchées sur des entrées "interruption" du micro-contrôleur, c'est-à-dire des entrées susceptibles de détecter le changement d'état du signal (le front).

Lorsqu'un changement d'état est détecté, le programme principal est temporairement arrêté pour traiter une fonction "interrupt", puis reprend à la suite. Cette fonction d'interruption peut traiter le comptage ou décomptage en fonction de l'état des signaux.

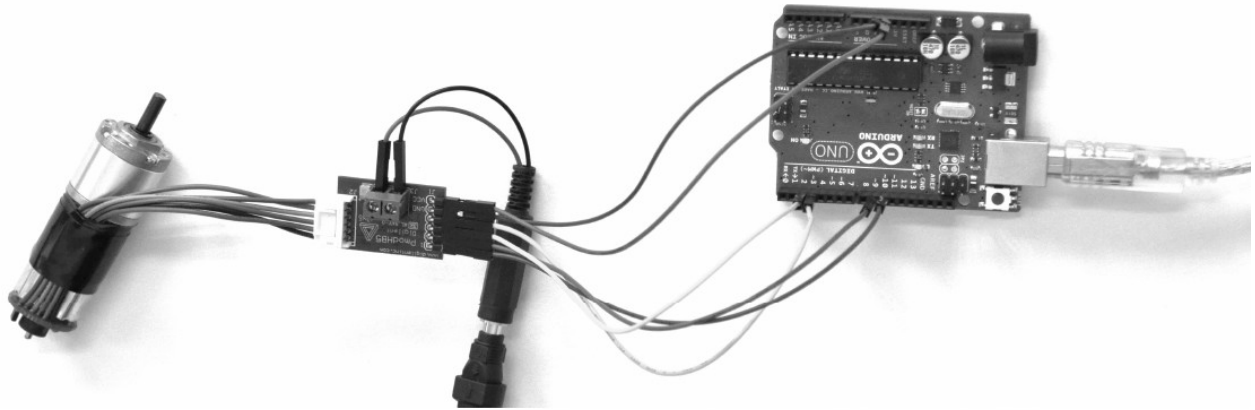
En notant A et B les états des signaux juste après le front détecté, et MA et MB les états des signaux mémorisé avant le front détecté (variable logiques 0 ou 1), exprimer la condition booléenne conduisant à incrémenter le compteur de top (sens horaire) ou à décrémenter le compteur de tops (sens trigonométrique), noté "cpt".

Ecrire le code en C à l'aide d'une structure "if" (les opérations booléennes s'écrivent || pour le OU logique et && pour le ET logique)



Commande du moteur en boucle ouverte

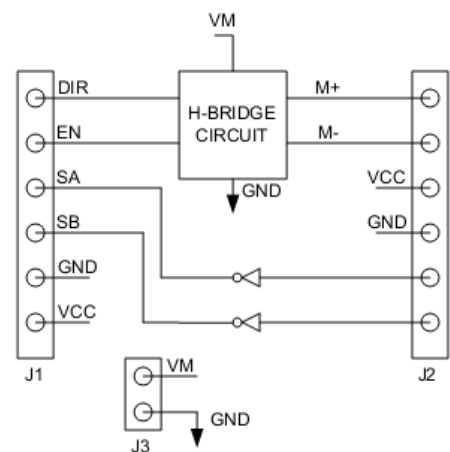
Le moteur est alimenté par un hacheur PMOD5. Brancher le moteur sur le hacheur. Viser la fiche d'alimentation sur le hacheur si ce n'est pas déjà fait en veillant à brancher le câble noir sur la borne GND et le câble rouge sur la borne VM. **Ne pas brancher tout de suite l'alimentation sur la fiche du hacheur.**



Sur le bornier "signal" du hacheur PMOD5, brancher VCC et GRD respectivement sur le 5V de l'arduino et sur la masse de l'arduino.

La photographie ci-dessus montre un cablage direct pour visualiser les branchements mais vous pourrez utilement utiliser un report sur la platine labdec.

Brancher l'entrée DIR (direction) sur la broche 8 et l'entrée EN (enable) sur la broche 9 (permettant de générer un signal modulé PWM).



Brancher le potentiomètre de façon à faire une acquisition analogique sur l'entrée AN0.

Proposer un programme permettant de commander le moteur en boucle ouverte, en envoyant 0V sur la broche 8 et un signal modulé PWM sur la broche 9, proportionnel à la tension du potentiomètre.

Au moment de tester votre programme, **vérifier le bon sens de branchement** de la prise male de l'alimentation : du coté où les deux flèches se font face, le logo doit indiquer que le moins est à l'extérieur et le plus à l'intérieur, tel que sur la photo ci-contre. Brancher alors la fiche de l'alimentation sur la prise du hacheur.



Vérifier que le moteur tourne (après avoir branché l'alimentation sur la fiche du hacheur) à vitesse variable, commandé par le potentiomètre. Changer l'état logique de la broche 8 et vérifier l'inversion du sens de rotation.

Proposer un programme permettant de tourner dans un sens quand le potentiomètre renvoie une valeur entre 0 et 512, et dans l'autre sens quand le potentiomètre renvoie entre 512 et 1023.

Prise en compte d'une entrée en interruption

L'objectif de cette partie est de détecter le changement d'état d'un signal logique. Pour cela, nous allons utiliser une entrée interruption (entrée 2) qui sera maintenu à 0V par une résistance (résistance de pull-down) sauf lorsqu'un contact électrique impose une tension à 5V.

Brancher la résistance entre la masse et la broche 2 de l'arduino. Implanter le programme suivant dans l'arduino et téléverser le programme.



```
sketch_nov20b | Arduino 1.0.5
Fichier  Édition  Croquis  Outils  Aide

sketch_nov20b $

int cpt=0;

void setup(){
  Serial.begin(115200);
  attachInterrupt(0,interrupt2,CHANGE);
  pinMode(13,OUTPUT);
}

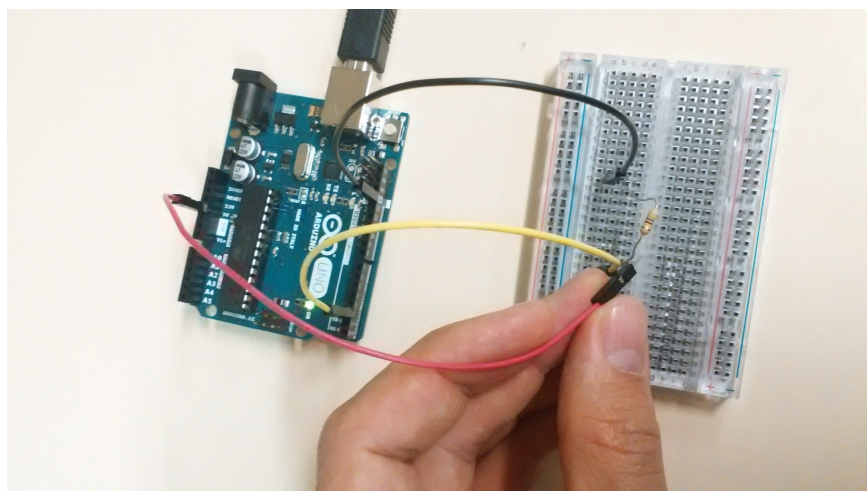
void interrupt2(){
  digitalWrite(13,!digitalRead(13));
  cpt++;
}

void loop(){
  Serial.println(cpt);
}

Formatage automatique terminé.
Taille binaire du croquis : 3 198 octets (d'un max de 32 256 octets)

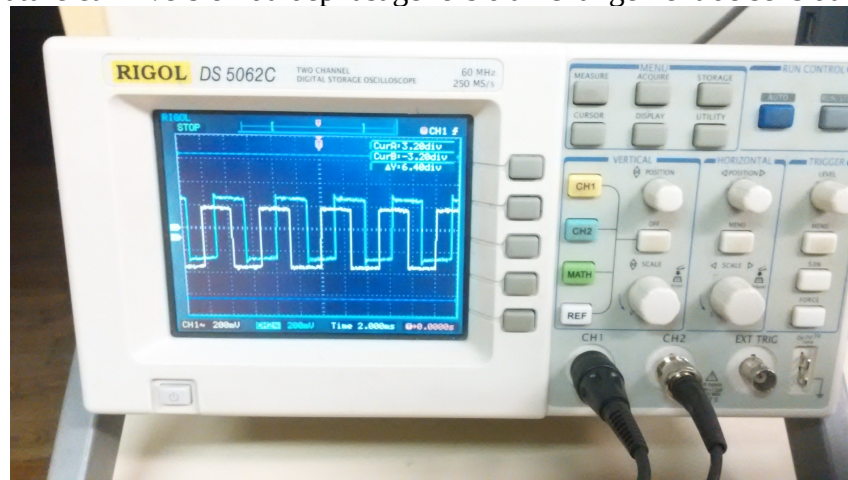
14 Arduino Uno on COM8
```

A l'aide d'un fil relié au 5V, toucher le câble de la résistance du coté relié à la broche 2 pour provoquer un changement d'état logique de la broche. Vérifier que la LED 13 change d'état. Afficher sur le port série la valeur du compteur de front.



Décodage du codeur à quadrature

Brancher les deux voies des deux codeurs sur la platine labdec et tracer les signaux à l'oscilloscope. Vérifier la quadrature et l'inversion du déphasage lors d'un changement de sens du moteur.



Implanter le programme de décodage de la quadrature suivant :

```
sketch_nov20b | Arduino 1.0.5
Fichier  Édition  Croquis  Outils  Aide

sketch_nov20b $

int cpt=0;

void setup(){
  Serial.begin(115200);
  attachInterrupt(0,interrupt2,CHANGE);
  pinMode(13,OUTPUT);
}

void interrupt2(){
  digitalWrite(13,!digitalRead(13));
  cpt++;
}

void loop(){
  Serial.println(cpt);
}

Formatage automatique terminé.
Taille binaire du croquis : 3 198 octets (d'un max de 32 256 octets)

14                               Arduino Uno on COM8
```

Vérifier en pilotant le moteur en boucle ouverte que le comptage et décomptage de la position est correct.

Asservissement du moteur CC en position – Correction proportionnelle

A partir d'une consigne de position issue du potentiomètre (exprimée en top codeurs), proposer un programme d'asservissement reproduisant le schéma bloc suivant :

COMMANDE NUMÉRIQUE D'UN MOTEUR CC

Notion d'interruption sur un flip flop de LED

Décodage des signaux codeurs en position

Mesure de vitesse

Asservissement de vitesse type P

Asservissement de vitesse type PI

Asservissement de position