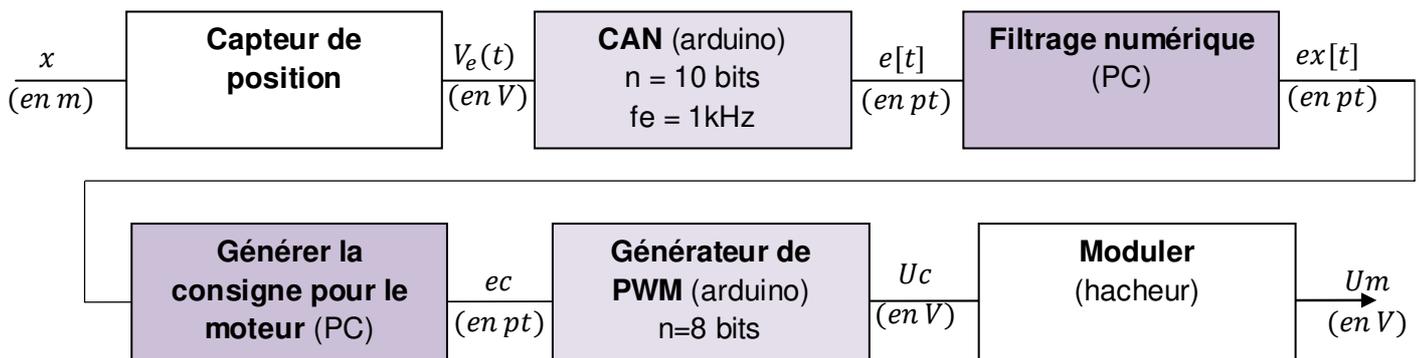


td	td ACQ 4.1	TS11 (Période 4)
	Filtrage numérique	0h30
	Cycle 8 : Acquérir conditionner et traiter l'information	4 semaines

ANALYSER Caractériser un constituant de la chaîne d'information.
ANALYSER Analyser la structure d'un programme informatique. ⇔ I
ANALYSER Analyser un algorithme. ⇔ I
EXPERIMENTER Générer un programme et l'implanter dans le système cible.

Problème technique :

Piloter la tension d'alimentation par modulation d'amplitude (pwm) d'un moteur en fonction de la valeur échantillonnée d'un capteur (avec filtrage numérique du signal d'acquisition).



1 Paramétrage des entrées-sorties de la carte Arduino

Le driver **fyrmata** a été téléversé sur l'arduino connecté au port '**COM5**'.

Le module **pyfyrmata** (driver python côté PC a été importé en début de programme) :

```
import pyfirmata
conn = pyfirmata.Arduino('COM5') # établissement de la connexion entre le PC et la carte Arduino.
```

L'instruction **conn.get_pin('x :y :z')** permet d'initialiser la broche concernée pour la connexion **conn** :

- **x** vaut **a** pour une grandeur analogique ; **d** pour une grandeur numérique ; **p** pour PWM
- **y** : entier définissant le numéro de la broche
- **z** vaut **i** : input (s'il s'agit d'une entrée à lire) ; **o** : output (s'il s'agit d'une sortie à affecter)

1) Ecrire l'instruction qui permet d'associer à la variable :

- **a1** la broche analogique **1** pour un accès en lecture de la position atteinte,
- **p3** la broche digitale **3** pour la génération d'une commande pwm à destination du moteur.

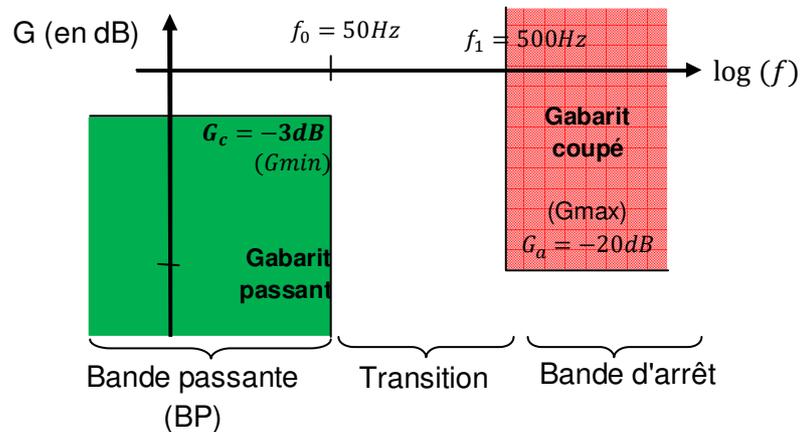
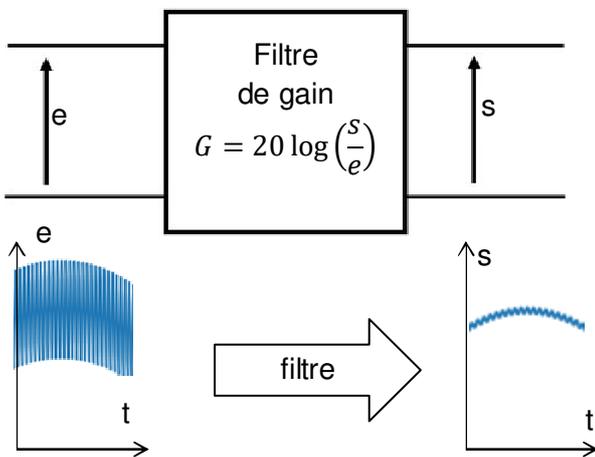
Fonctionnalité	en python	Résultat
Lire la valeur de la broche a , paramétrée comme une entrée analogique	a.read()	un flottant compris entre 0. et 1.
Ecrire une valeur 0.5 sur la broche p, paramétrée comme une sortie pwm L'argument doit être compris entre 0. et 1. (rapport cyclique de 0 à 100%)	p.write(0.5)	la tension en sortie aura un rapport cyclique de 50%

- 2) Ecrire l'instruction qui permet d'ajouter à une liste **e** la valeur lue sur la broche **a1**.
- 3) Ecrire l'instruction qui permet d'écrire sur la broche **p** la commande pwm qui correspond à la valeur d'indice 101 d'une liste **ex**.

2 Equation de récurrence du filtre

Le filtre étudié admet :

- en entrée $e[n]$ comportant des évolutions numériques à différentes fréquences,
- en sortie $s[n]$ dont une partie des évolutions numériques de l'entrée sont atténués à l'aide d'un filtre passe-bas d'ordre 1.



- 4) Quels éléments du gabarit nous permet de savoir que le filtre est un filtre passe bas d'ordre 1 ?

Le filtre passe-bas d'ordre 1 est caractérisé par l'équation différentielle suivante :

$$\tau \frac{ds(t)}{dt} + s(t) = e(t)$$

avec $\tau = \frac{1}{2\pi f_0}$: la constante de temps du filtre (en s)

En sortie de CAN, les grandeurs analogiques deviennent des grandeurs échantillonnées (numériques) :

- **e[n]** : signal d'entrée à l'instant t_n ;
- **s[n]** : signal de sortie à l'instant t_n .

La durée entre 2 échantillons vaut **Te=0.001s**.

- 5) Déterminer l'équation de récurrence permettant de calculer **s[n]**. Pour cela, il convient d'intégrer l'équation différentielle du filtre passe bas par la méthode des rectangles à gauche entre les instants t_{n-1} et t_n .

3 Implémentation du filtre et génération de la commande pwm

- 6) Ecrire une fonction **filtrer()** qui admet en argument une liste de flottants **e** et qui renvoie en sortie une liste **s** obtenue par application du filtre à **e** (la période d'échantillonnage et la constante de temps du filtre sont affectées respectivement aux variables globales **Te** et **tau** dans le corps du programme).
- 7) Ecrire l'instruction qui permet d'affecter à **Ls** la valeur renvoyée par la fonction **filtrer()** avec la liste des valeurs d'entrée **Le**.

Les valeurs de commande du moteur sont stockées dans une liste **Lc** et sont construites par soustraction avec la valeur de la consigne de position **xc**.

- 8) Ecrire l'instruction qui permet d'affecter à la sortie **p3** la dernière valeur calculée dans la liste **Lc**.