



Nom

Note :

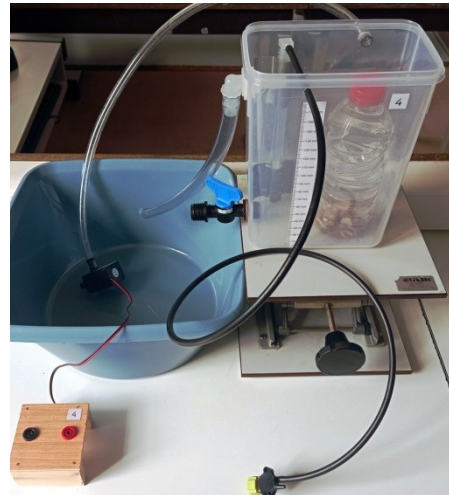
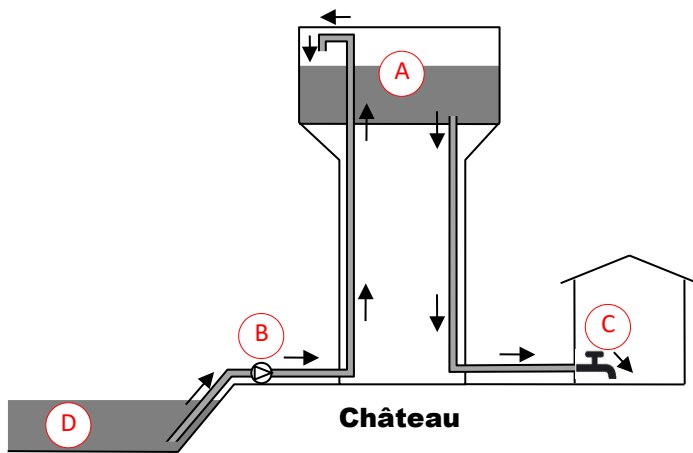
Prénom

AE.3B - Régulation TOR

Partie 1 : Introduction

La maquette de l'activité 3A simule un château d'eau dont on veut maintenir la hauteur d'eau constante pour avoir une pression constante chez l'utilisateur. On souhaite que, quelle que soit l'ouverture du robinet de puisage, le niveau d'eau puisse être maintenu constant :

- à 90 mm dans un premier temps ;
- à 140 mm dans un second temps, après un appui sur un bouton de commande.



1. Indiquer la lettre du schéma du château d'eau qui correspond à : **(1 pt)**
 bac d'alimentation du château d'eau ; robinet de puisage ;
 réservoir de contrôle de niveau ; pompe.
2. Placer aussi ces quatre mêmes lettres sur la photo de la maquette. **(1 pt)**

Partie 2 : Étalonnage du capteur de niveau (2 pts)

3. Avant de réaliser l'acquisition des mesures, il est indispensable d'étalonner le capteur de pression. La pression variant avec le niveau d'eau dans la cuve, on souhaite exploiter le capteur de pression pour afficher le niveau d'eau. Autrement dit, on souhaite déterminer expérimentalement une relation entre le niveau d'eau et la valeur numérique du capteur de pression. Pour cela :
 - Ouvrir le programme "*etalonnage_niveau.ino*." avec le logiciel **Arduino** et le téléverser sur la carte; ouvrir le moniteur série ;
 - Ouvrir le robinet puis suivre les indications situées dans le moniteur Série.
Grâce à l'étalonnage, on connaît les valeurs de **a** et de **b** qui permettent de calculer le niveau d'eau (en mm) avec la relation **niveau = a × valeur numérique du capteur de pression + b** ;
Noter ces valeurs !



Partie 3 : Algorithme

4. En s'inspirant de l'activité **3A**, rédiger ici un algorithme permettant de maintenir un niveau égal au seuil de 90 mm. **(1 pt)**

Partie 4 : Réponse à des changements d'ouverture du robinet de puisage

Après que le capteur ait été étalonné, on laisse le robinet ouvert et on réalise une régulation TOR du niveau d'eau :

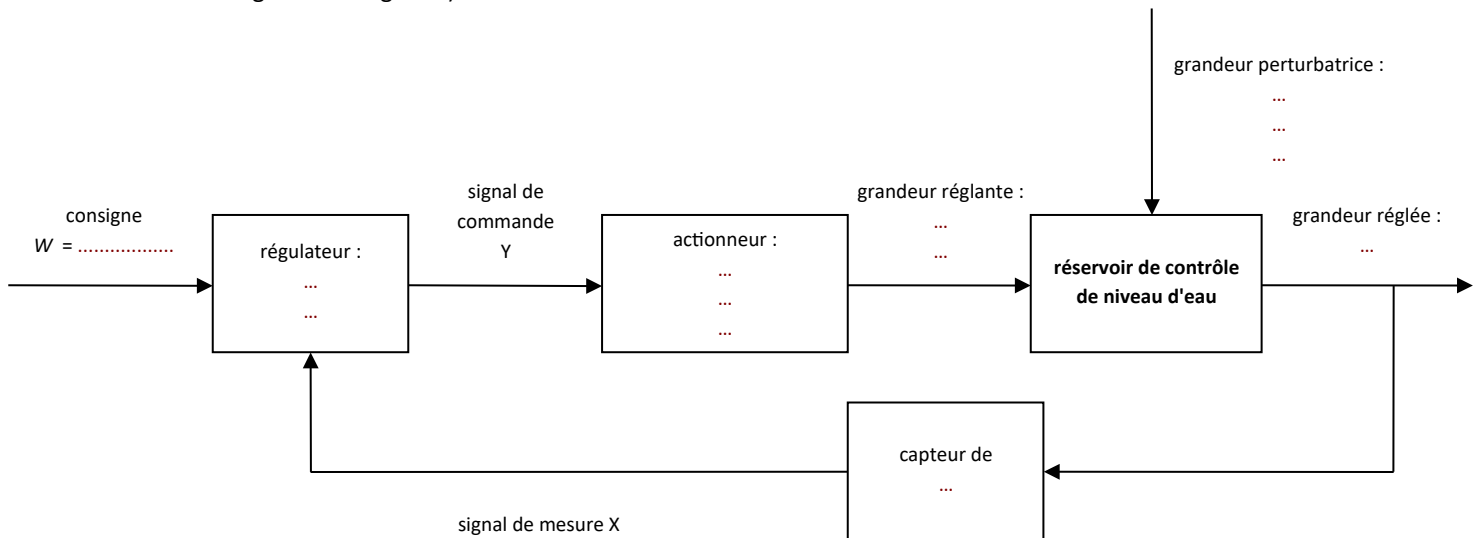
5. Ouvrir le robinet de puisage.
6. Marche à suivre strictement
 - a. Avec le logiciel Arduino, ouvrir le programme "*regul_tor.ino*".
Modifier les lignes 39, 40 et 43 du programme avec les valeurs de consigne1 (le niveau d'eau voulu dans un 1^{er} temps), et les valeurs **a** et **b** déterminées au cours de l'étalonnage.
 - b. Téléverser le programme sur le microcontrôleur.
 - c. Avec le logiciel EduPython, ouvrir le programme *graphique_regulation_tor.py*. et l'enregistrer dans vos documents. Attendre que la fenêtre graphique s'affiche.
 - d. Appuyer sur le bouton de la carte.
 - e. La régulation démarre et l'Arduino commence à envoyer les données.
7. Quand le niveau d'eau semble à peu près stabilisé, attendre une vingtaine de secondes puis changer modérément l'angle d'ouverture du robinet de puisage et observer l'évolution du système.
8. Pour finir, comme indiqué sur la fenêtre graphique, un appui long sur le bouton pour arrêter la maquette et l'acquisition.
9. Le fichier (au format .csv) contenant les résultats de l'acquisition se trouve dans le même dossier que le programme "*graphique_regulation_tor.py*".
À partir du tableur-grapheur, ouvrir le fichier (au format .csv) contenant les résultats de l'acquisition et imprimer le graphique et le joindre à votre compte-rendu **(2 pts)**
Observer l'évolution temporelle des différentes grandeurs et commenter **(1,5 pt)**.

Partie 5 : Schéma fonctionnel de la boucle de régulation

10. Compléter le schéma fonctionnel de la boucle de régulation ci-dessous en ajoutant : **(3,5 pts)**
 - le nom de la grandeur réglée (grandeur dont on veut maintenir la valeur constante) ;
 - le nom du type de capteur utilisé (pour mesurer la grandeur réglée) ;
 - la valeur de la consigne (valeur que l'on veut imposer à la grandeur réglée) ;



- le nom de la grandeur réglante (grandeur sur laquelle le système automatisé agit afin de tenter d'imposer une valeur à la grandeur réglée) ;
- le nom de la grandeur perturbatrice (grandeur non contrôlée par le système automatisé mais qui a une influence sur la grandeur réglée)
- le nom de ce qui sert de régulateur (l'électronique qui compare la mesure à la consigne et indique comment il faut agir suivant la situation) ;
- le nom de ce qui sert d'actionneur (l'ensemble d'objets qui, piloté par le signal provenant du régulateur, va modifier la grandeur réglante).



Partie 6 : Réponse à un changement brusque de niveau souhaité

1. Lancer `regul_tor.ino`.
2. Lancer `graphique_regulation_tor.py`.
3. Appui bref : démarrage de l'expérience.
4. Appui bref pendant l'expérience : échelon de consigne.
5. Appui long : arrêt, sauvegarde du CSV.

11. Le fichier (au format texte) contenant les résultats de l'acquisition se trouve dans le même dossier que le programme "`graphique_regulation_tor.py`".

À partir du tableur-grapheur, ouvrir le fichier (au format .csv) contenant les résultats de l'acquisition. Imprimer le graphique et le joindre au compte-rendu **(1 pt)**

Observer l'évolution temporelle des différentes grandeurs et commenter. **(1 pt)**

Partie 7 : Cas d'une régulation tout ou rien à deux seuils

12. Avec la régulation TOR à 1 seuil, la pompe s'allume et s'éteint très brièvement et très souvent. Ceci entraîne une usure prématurée de l'actionneur. Pour éviter ce problème, on peut utiliser une régulation TOR à 2 seuils : pour avoir un niveau d'environ 110 mm, on fixe par exemple une consigne basse de 106 mm et une consigne haute de 114 mm.

Quel doit être l'état de la commande de la pompe si le niveau d'eau est supérieur à la consigne haute ? **(1 pt)**

Qu'en est-il si le niveau d'eau est inférieur à la consigne basse ? **(1 pt)**



13. En s'aidant de la question précédente, compléter les 6 pointillés de l'extrait de programme simplifié ci-dessous pour que la hauteur d'eau soit régulée entre 106 mm et 114 mm.

```
# Branchement du capteur de pression sur la broche A0 du microcontrôleur
# Branchement du signal du relais sur la broche 9 du microcontrôleur

consigne_haute = 114 ; # valeur de la consigne haute (en mm)
consigne_basse = 106 ; # valeur de la consigne basse (en mm)
a = 1.35119 ; # coefficient a pour le calcul de la hauteur d'eau
b = -56.3035 ; # coefficient b pour le calcul de la hauteur d'eau

commande_pourcent = 0 ; # signal de commande (en %)
digitalWrite(9,0) ; # met le signal de commande à 0 (éteint la pompe)

while True:
    hauteur = a * analogRead(A0) + b ; # hauteur d'eau (en mm)
    if ( ..... ){..
        commande_pourcent = ...
        digitalWrite(9,..)
    }
    if ( ..... ){
        commande_pourcent = ..... ;
        ...
        sleep(200) ; # pause de 200 ms
```

Si le capteur a déjà été étalonné lors de cette séance, l'acquisition peut être réalisée. Sinon, refaire la partie 3.

14. Ouvrir le robinet de puisage.

15. Protocole :

1. Téléverser / lancer `regul_tor_2_seuils.ino` dans l'Arduino.
2. Lancer `graphique_regulation_tor.py` sur l'ordinateur.
3. Appui bref sur le bouton : démarrage de l'expérience.
4. Appui long sur le bouton : arrêt de l'expérience et sauvegarde du CSV

16. Le fichier (au format texte) contenant les résultats de l'acquisition se trouve dans le même dossier que le programme "`graphe_python`" et se nomme `regulniv00`. Avant de l'utiliser et/ou de démarrer une nouvelle acquisition, le renommer en remplaçant `00` par `TOR2seuils`.

À partir du tableur-grapheur, ouvrir le fichier (au format `.csv`) contenant les résultats de l'acquisition et imprimer le graphique à joindre avec votre compte-rendu (2 pts).

Observer l'évolution temporelle des différentes grandeurs et commenter. (1 pt)