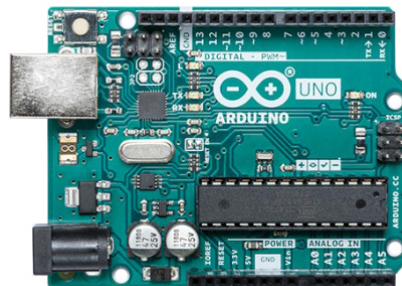
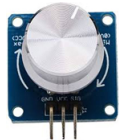
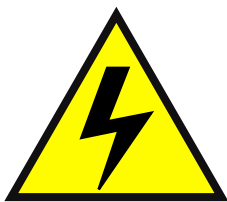


	Sciences et Technologies de l'Industrie et du Développement Durable		
	INGÉNIERIE ET DÉVELOPPEMENT DURABLE		
	Approche comportementale des produits	TP4	I2D

## ACQUISITION ET PILOTAGE SERVO



### INSTRUCTIONS PERMANENTES DE SÉCURITÉ

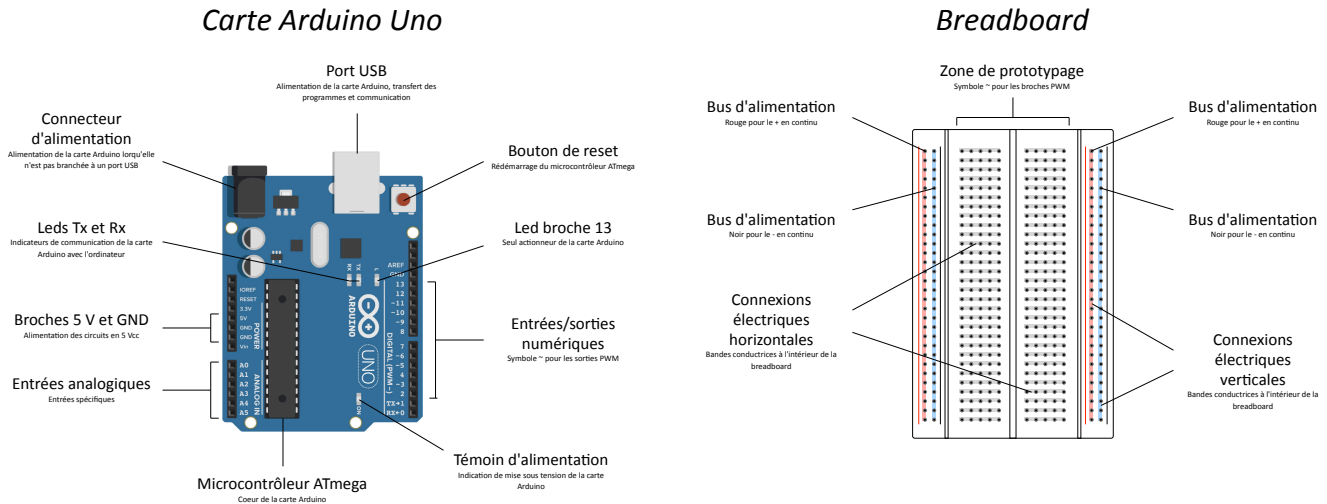


1. Avant toute mise sous tension, le professeur vérifie le montage et contrôle le calibrage des appareils de mesure.
2. La mise sous tension et hors tension du poste (consignation, déconsignation) est effectuée en présence du professeur.
3. Toute intervention nécessitant l'ouverture d'un circuit électrique (installation d'un appareil) est effectuée hors tension.
4. Pendant la phase où le poste est sous tension, l'élève travaille sans modifier le câblage du circuit (relevés de mesures ...).
5. En cas de problèmes sur un poste de travail voisin, vous devez impérativement couper l'alimentation du poste en activant le bouton d'arrêt d'urgence le plus proche.

**C'EST LE PROFESSEUR QUI DONNE, APRÈS AVOIR  
PROCÉDÉ À LA CONSIGNATION DU POSTE,  
L'AUTORISATION DE DÉMONTAGE**

## 1. Rappels

Arduino est une plateforme matérielle et logicielle de développement d'applications embarquées. Elle se compose d'une carte électronique basée autour d'un microcontrôleur comportant un certain nombre d'entrées et de sorties permettant la connexion de capteurs, ou d'actionneurs.

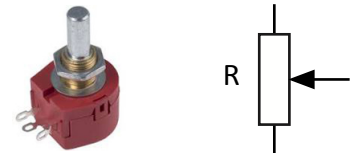


Le logiciel de programmation des modules Arduino est une application Java libre servant d'éditeur de code et de compilateur, et qui peut transférer le firmware et le programme au travers de la liaison USB.

Le langage de programmation utilisé est un mélange de C et de C++, restreint et adapté aux possibilités de la carte.

## 2. Acquisition de la valeur d'un potentiomètre

Un potentiomètre est une résistance variable à trois bornes. En général, les deux bornes extérieures donnent la résistance totale du potentiomètre et la borne du milieu donne la résistance entre le curseur et une des deux bornes extérieures.



Le potentiomètre mis à votre disposition doit avoir une résistance totale comprise entre 1 k $\Omega$  et 10 k $\Omega$ .

### Question n° 1 :

- Proposer un schéma de montage permettant de mesurer la résistance totale du potentiomètre mis à votre disposition.
- Préciser si le montage doit être sous tension ou non.
- **Faire vérifier votre schéma par le professeur.**
- Réaliser votre montage.

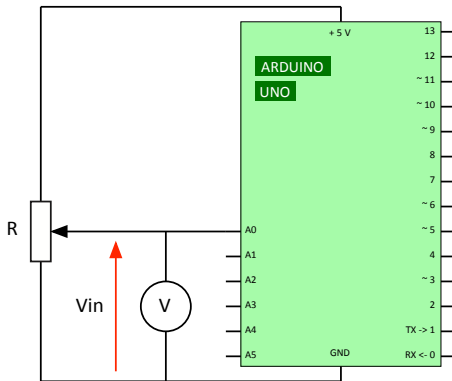
**STOP**

**FAIRE VÉRIFIER PAR LE PROFESSEUR**

- Mesurer la résistance totale du potentiomètre mis à votre disposition.
- Valider le choix du potentiomètre mis à votre disposition.

**Question n° 2 :**

- Après avoir vérifié, **en présence du professeur**, la mise **hors tension** de la carte Arduino, réaliser le montage ci-dessous.

*Montage potentiomètre**Programme*

```
int potpin = A0;

void setup() {
  Serial.begin(9600);
}

void loop() {
  int potval = analogRead(potpin);
  Serial.print("valeur potentiometre :");
  Serial.println(potval);
  delay (100);
}
```

**STOP****FAIRE VÉRIFIER PAR LE PROFESSEUR**

- Saisir le programme fourni sur le logiciel Arduino.
- Téléverser le programme dans la carte Arduino.
- Ouvrir le **moniteur série** et **faire valider le fonctionnement de votre programme par le professeur**.

**Question n° 3 :**

- Compléter le tableau de mesure fourni sur le document réponse pour différentes valeurs de Vin.
- Tracer sur le document réponse fourni la caractéristique  $N = f(V_{in})$  (code obtenu en fonction de la tension Vin mesurée).
- Indiquer le nombre de bits utilisés pour coder l'information Vin.
- Calculer le quantum du Convertisseur Analogique Numérique de l'entrée A0 de l'Arduino.

**Question n° 4 :**

- Modifier le programme précédent comme indiqué ci-dessous.

```
void loop() {
  int potval = analogRead(potpin);
  Serial.print("valeur potentiometre :");
  Serial.print(potval);

  float voltage = (potval/1024.0)*5.0;
  Serial.print(" , tension :");
  Serial.println(voltage);
  delay(100);
}
```

- Téléverser le programme modifié dans la carte Arduino.
- Ouvrir le **moniteur série** et **faire valider le fonctionnement du programme modifié par le professeur**.

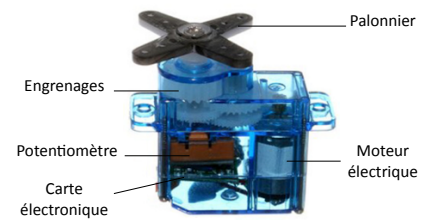
**Question n° 5 :**

- Comparer les valeurs de tensions affichées avec les valeurs Vin mesurées.
- Modifier éventuellement la formule pour ajuster les valeurs affichées.
- Mettre la carte Arduino **hors tension**.

### 3. Pilotage servomoteur

Un servomoteur est un système permettant de mettre en rotation un objet. Comparé à un simple moteur, un servomoteur est capable de se positionner à un angle précis et de maintenir cette position même si un effort est exercé sur son axe.

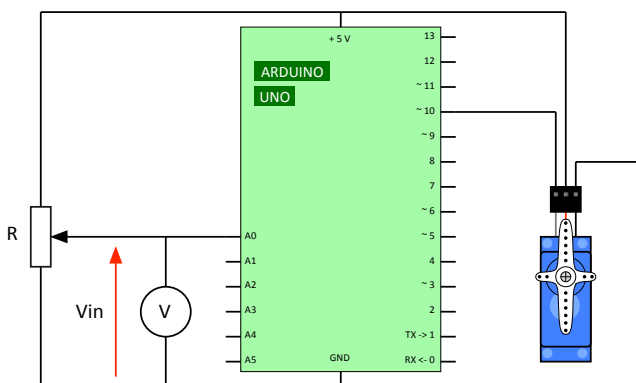
Le servomoteur utilisé pour ce TP est commandé directement par un angle compris entre 0 à 180 degrés.



#### Question n° 6 :

- Après avoir vérifié, **en présence du professeur**, la mise **hors tension** de la carte Arduino, réaliser le montage ci-dessous.

Montage servomoteur



#### Remarque :

La ligne `#include <Servo.h>` est obtenue à partir de la bibliothèque Servo intégrée dans le logiciel Arduino (Croquis/Inclure une bibliothèque/Servo).

#### Connecteur servomoteur :

- Noir : 0 V ou GND
- Rouge : Alimentation + 5 V
- Autre couleur : Signal de commande

Programme

```
#include <Servo.h>
Servo myservo;
int potpin = A0;
int potval;
int angle;

void setup() {
  myservo.attach(10);
  Serial.begin(9600);
}

void loop() {
  potval = analogRead(potpin);
  Serial.print("valeur potentiometre :");
  Serial.print(potval);

  angle = map(potval, 0, 1023, 0, 180);
  Serial.print(" , angle :");
  Serial.println(angle);

  myservo.write(angle);

  delay (100);
}
```

**STOP**

**FAIRE VÉRIFIER PAR LE PROFESSEUR**

- Saisir le programme fourni sur le logiciel Arduino.
- Compléter l'organigramme fourni sur le document réponse avec les lignes de programme correspondantes.
- Téléverser le programme dans la carte Arduino.
- Ouvrir le **moniteur série** et **faire valider le fonctionnement de votre programme par le professeur**.

**Question n° 7 :**

- Compléter le tableau de mesure fourni sur le document réponse pour différentes valeurs de  $V_{in}$ .
- Vérifier les valeurs de l'angle données par le moniteur série à l'aide d'un rapporteur.
- Tracer sur le document réponse fourni la caractéristique  $\alpha = f(V_{in})$  (angle obtenu en fonction de la tension  $V_{in}$  mesurée).
- Commenter la courbe obtenue.

**Question n° 8 :**

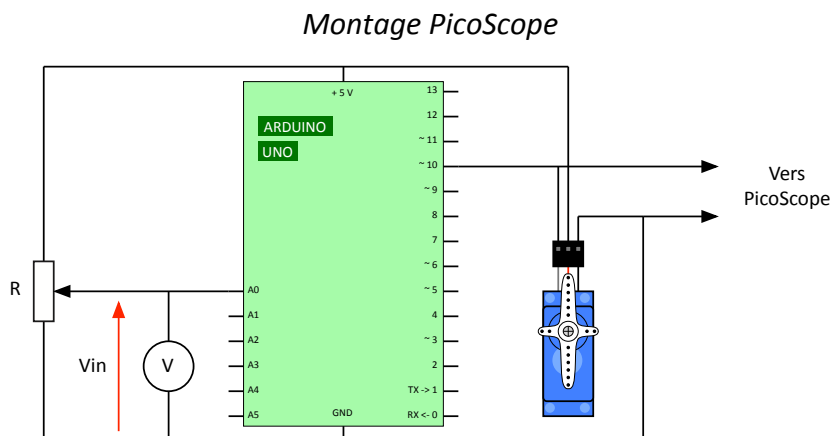
- Déterminer par calcul la valeur théorique de  $V_{in}$  pour un angle de  $90^\circ$ .
- Vérifier votre résultat avec le montage précédent.
- Mettre la carte Arduino **hors tension**.

## 4. Visualisation signal de commande

On souhaite visualiser le signal de commande fourni au servomoteur par la carte Arduino à l'aide d'un PicoScope et du logiciel associé.

**Question n° 9 :**

- Après avoir vérifié, **en présence du professeur**, la mise **hors tension** du poste, réaliser le montage ci-dessous.

**STOP****FAIRE VÉRIFIER PAR LE PROFESSEUR**

- Ouvrir le logiciel du PicoScope.
- Visualiser, à l'aide du PicoScope, le signal de commande du servomoteur.
- Mesurer précisément la période du signal.
- Vérifier si la période du signal varie ou non en fonction de l'angle demandé.
- Recopier le signal sur le document réponse fourni pour un angle quelconque.

**Question n° 10 :**

- Relever, à l'aide du PicoScope, l'allure du signal de commande du servomoteur pour un angle de 0 °.
- Mesurer précisément la durée pendant laquelle le signal est à 1.
- Compléter le graphique fourni sur le document réponse pour cet angle.
- Répéter l'opération pour un angle de 90 ° et pour un angle de 180 ° (angle maxi).
- Comparer les résultats obtenus et conclure sur la commande du servomoteur pour faire varier la position angulaire de son axe.

**5. Remarques**

---

- Toutes les réponses doivent être justifiées.
- Tous documents autorisés.
- Ressource internet : <https://www.arduino.cc/reference/en/>