

Activité - Etude d'un système de détection d'un passager (d'après Bac 2021 Métropole)

Pour renforcer la sécurité routière, les voitures sont équipées d'un système de détection de la présence d'un passager pour lui signaler si sa ceinture de sécurité est bien attachée.

Dans le cadre d'un projet scientifique, un groupe d'élèves réalise un système de détection semblable à celui d'une voiture. Il est composé d'un capteur de pression capacitif « artisanal » associé à un microcontrôleur.

Le condensateur « artisanal » est constitué de deux feuilles d'aluminium séparées par une feuille de papier isolante. Lorsqu'un objet de masse m est posé dessus, il exerce une pression sur les deux feuilles d'aluminium et les déforme, ce qui modifie la capacité électrique du condensateur « artisanal ».

Après un traitement numérique des signaux électriques, le microcontrôleur peut détecter la présence de l'objet.



Figure 1. Schéma de l'installation d'un capteur capacitif dans l'assise d'un siège de voiture



Figure 2. Photographie d'une face du capteur de pression capacitif « artisanal ».

L'objectif de cette activité est d'illustrer le principe de fonctionnement d'un tel capteur.

Partie 1 : Étude de la charge d'un condensateur

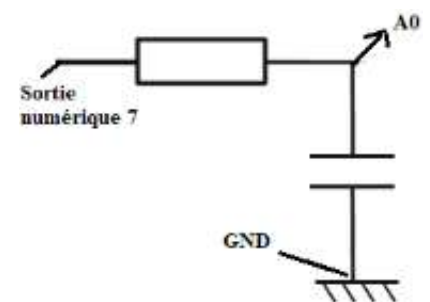
Le montage ci-contre permet d'étudier la charge d'un condensateur :

Le dipôle constitué par le conducteur ohmique et le condensateur, nommé « dipôle RC », est alimenté par une tension de 5,0 V fournie par la sortie numérique 7 d'une carte Arduino.

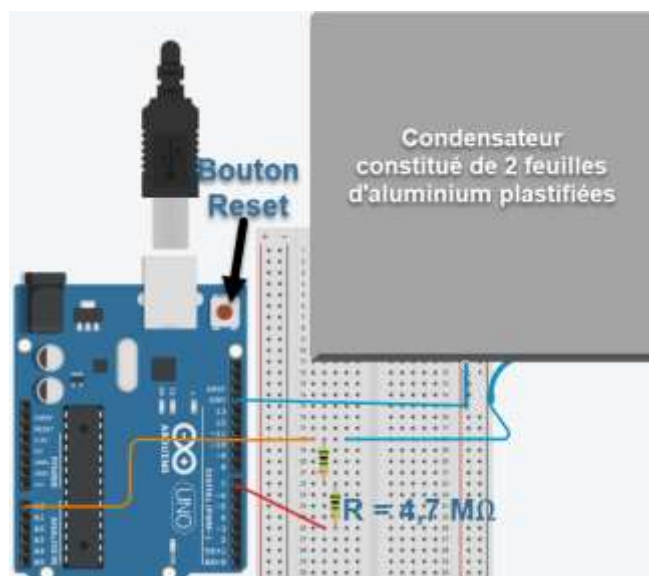
À la date $t = 0$ (origine des temps), le dipôle RC est soumis à un échelon de tension de 0 à 5,0 V et la charge du condensateur démarre.

À la date $t = \tau = R \times C$, le condensateur est chargé à 63 %.

On considère que la charge du condensateur est totale au bout de $5 \times \tau$.



Réaliser le montage suivant



```

1 // Définition des différentes grandeurs
2 unsigned long origine_temps;
3 float C;
4 float R;
5 float Tau;
6
7 void setup() {
8   pinMode(7, OUTPUT); // alimentation branchée sur l'entrée 7
9   Serial.begin(9600);
10  // dans un premier temps, on s'assure que le condensateur est complètement déchargé
11  Serial.println("Préparation du condensateur");
12  digitalWrite(7, LOW); // alimentation à 0V
13  delay(1000); // délai au cours duquel l'alimentation est maintenue à 0V, on prend ici 1000 ms
14
15  // dans un deuxième temps, charge du condensateur
16  Serial.println("Charge de condensateur");
17  digitalWrite(7, HIGH); // alimentation à 5V (1023 bits)
18  origine_temps = millis(); // définition de l'origine des temps à l'aide de la fonction
19  // millis() qui renvoie la date en ms de l'horloge interne d'Arduino
20  // puis à partir de sa mise sous tension
21  while(analogRead(A0) < 0.63*1023) {
22    // affichage de la constante de temps
23    Tau = millis() - origine_temps;
24    Serial.print("Tau = ");
25    Serial.print(Tau);
26    Serial.println(" ms");
27    R = 4400000;
28  }
29
30  //
31
32
33
34
35
36 void loop() {
37

```

NB :

- La mesure de la tension aux bornes du condensateur se fait à l'aide de l'entrée analogique A0 (fil orange).
- La sortie numérique 7 (fil rouge) permet de créer une tension de 5 V aux bornes du dipôle RC.
- Le CAN (Convertisseur Analogique Numérique) de la carte Arduino est sur 10 bits soit $2^{10} = 1024$ valeurs possibles. Ainsi 0 V correspond à 0 en numérique et 5 V correspond à 1023.


Le logiciel de programmation : Arduino

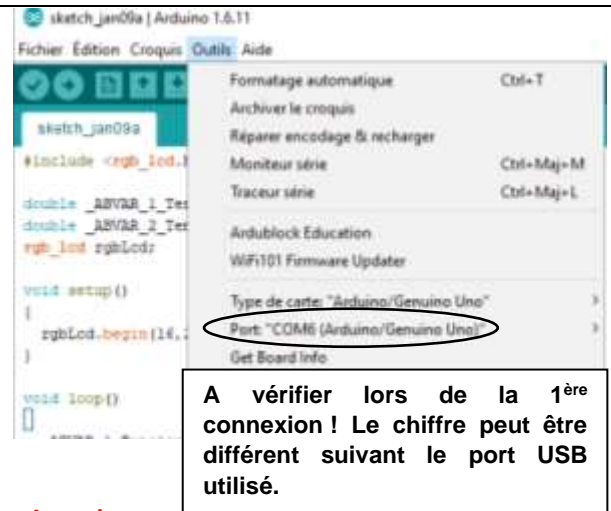
- La programmation peut être réalisée par le codage en langage Arduino (proche du langage C).
- Les deux icônes ci-dessous vont suffire à communiquer avec le microcontrôleur :

1- Vérification du programme



2 - Téléversement (= envoi) du programme sur la carte

- Brancher le câble USB pour relier le microcontrôleur à l'ordinateur puis choisir le port de communication (outils).
- Copier sur votre espace personnel le **Dossier : TP Condensateur artisanal**
- Ouvrir le programme *TP_Condensateur_artisanal* à partir du logiciel Arduino (Fichier / Ouvrir) et l'enregistrer dans travail. Téléverser le programme puis cliquer sur le **moniteur série**  (qui se trouve en haut à droite).

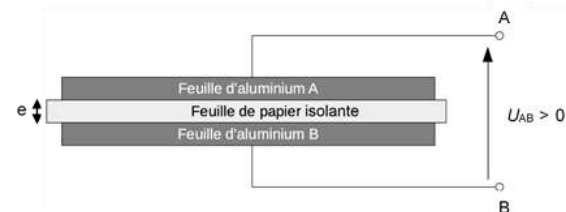


Questions :

- 1- Expliquer la signification de la ligne 21 du programme ci-dessus.
- 2- Faire un reset (bouton blanc). Quelle est la valeur affichée sur l'écran. Pourquoi ?
- 3- Modifier le programme, ligne 29, pour que la capacité C du condensateur soit calculée en nF. Sur le même principe que les lignes 25 à 27, compléter le programme (lignes 30 à 32) pour afficher la valeur de la capacité C du condensateur avec son unité. Recopier les lignes 30 et 32.

Partie 2 : Étude du capteur de pression capacitif « artisanal »

Le capteur de pression capacitif « artisanal » est représenté en coupe sur la figure ci-contre :



Questions :

- 1- Justifier l'utilisation de l'adjectif « capacitif » dans l'expression « capteur de pression capacitif » couramment utilisée pour désigner ce genre de capteurs.
- 2- Si le capteur est soumis à une tension positive constante U_{AB} entre ses bornes A et B, des charges électriques apparaissent sur chacune des feuilles, notées Q_A sur la feuille d'aluminium A et Q_B sur la feuille d'aluminium B. On note C la capacité électrique de ce capteur. Donner l'expression littérale de la charge Q_A puis celle de la charge Q_B en fonction de C et U_{AB} .
- 3- La capacité électrique C d'un tel capteur dépend de ses caractéristiques géométriques (S : la surface en regard des feuilles d'aluminium, e : l'épaisseur de la feuille de papier isolante et ϵ : une constante caractéristique de la feuille de papier isolante).

En utilisant le montage précédent, déterminer la formule qui permet de relier la capacité du condensateur à ses caractéristiques géométriques : $C = \frac{e}{\epsilon \times S}$ $C = \frac{e \times S}{\epsilon}$ $C = \frac{\epsilon \times S}{e}$ $C = \frac{\epsilon \times e}{S}$

- 4- Indiquer, en justifiant la réponse, le sens de variation de la capacité électrique C du capteur quand un objet est posé sur le condensateur « artisanal ».
- 5- Expliquer comment le capteur peut détecter la présence de l'objet.