

Mesurer de faibles capacités avec Arduino

Méthode d'utilisation du programme Arduino

Le programme Arduino fourni (hatier-clic.fr/pct558) permet de mesurer des capacités de l'ordre de quelques dizaines de picofarads, parfois inaccessibles aux capacimètres disponibles dans les lycées. Cette méthode utilise la capacité résiduelle sur les entrées analogiques du microcontrôleur Arduino. La valeur de cette capacité étant variable d'une console à l'autre, il est nécessaire d'effectuer un étalonnage.

Cette mesure n'est pas précise dans toutes les gammes de capacités. On se contentera de valeurs à quelques picofarads près, ce qui est suffisamment précis.

Protocole 1 Étalonnage du microcontrôleur Arduino

Il faut disposer d'un condensateur de capacité $C_{\text{réf}}$ connue, de quelques dizaines de picofarads (idéalement vers 60 pF). Pour étalonner l'appareil, le plus simple est de procéder par tâtonnements, en suivant le protocole ci-dessous :

- Brancher ce condensateur aux bornes A0 et A2 de la console Arduino.
- Téléverser le programme et visualiser le résultat sur la console. La valeur affichée n'est pas forcément la valeur de référence $C_{\text{réf}}$ indiquée par le constructeur.
- Modifier dans le programme la constante `CAPA_ENTREE` (l'augmenter si la valeur affichée est inférieure à la valeur de référence $C_{\text{réf}}$, la diminuer si elle est supérieure) puis téléverser à nouveau le programme modifié.
- Procéder ainsi jusqu'à avoir une valeur correcte à quelques picofarads près.

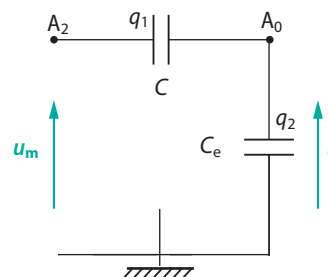
Protocole 2 Mesure d'une capacité grâce à un microcontrôleur Arduino

- Une fois le microcontrôleur étalonné, brancher le condensateur dont on veut mesurer la capacité aux entrées analogiques A0 et A2.
- Téléverser le programme obtenu après étalonnage et lire la valeur de capacité affichée.
- La mesure est faite toutes les secondes, ce qui permet de visualiser les variations éventuelles de la capacité mesurée sous l'effet des variations des paramètres (espace entre les plaques, etc.).

Explications

Le microcontrôleur Arduino ayant un comportement capacitif interne, on utilise la capacité d'entrée des entrées analogiques pour faire la mesure.

Le schéma équivalent du circuit est le suivant. C est la capacité du condensateur que l'on cherche à mesurer, C_e la capacité d'entrée résultante des deux entrées.



Le programme agit ainsi pour chaque mesure :

- ① L'entrée A2, initialement déchargée, est chargée (A2 passée à HIGH), ce qui correspond à une valeur numérique de 1023 (environ 5 V).
- ② Les condensateurs se chargent instantanément (car la résistance du circuit est très faible). Étant donné que ces condensateurs étaient initialement déchargés et qu'il y a conservation de la charge, on en déduit que les charges q_1 et q_2 portées par les deux condensateurs sont identiques. De plus, la tension aux bornes de C étant $u_m - u$, on peut écrire : $C(u_m - u) = C_e u$.
- ③ Le programme mesure la valeur de u (nommée `val`) en numérique, puis fait le calcul de C selon la relation précédente : $C = \frac{C_e u}{u_m - u}$.
- ④ Le programme affiche la valeur de la capacité C puis décharge les condensateurs (A2 passée à LOW) puis attend une seconde (`delay(1000)`) puis recommence.

```

1 //Brancher le condensateur entre A0 et A2
2 //Calibrer la valeur suivante avec un condensateur connu
3 const float CAPA_ENTREE = 24;
4
5 void setup()
6 {
7   pinMode(A2, OUTPUT);
8   pinMode(A0, OUTPUT);
9   Serial.begin(9600);
10 }
11
12 void loop()
13 {
14   pinMode(A0, INPUT);
15   digitalWrite(A2, HIGH);
16   int val = analogRead(A0);
17   //Calcul et affichage
18   float CAPA=(float)val*CAPA_ENTREE/(float)(1023-val);
19   Serial.print(F("C = "));
20   Serial.print(CAPA, 0);
21   Serial.println(F("pF"));
22   digitalWrite(A2, LOW);
23   pinMode(A0, OUTPUT);
24   delay(1000);
25 }

```

Étalonnage
Cette valeur est à modifier pour étalonner le microcontrôleur (**protocole 1**).

Configuration
Les entrées analogiques auxquelles est branché le condensateur sont indiquées ici.

① Charge instantanée des condensateurs.

③ Mesure la valeur de u (notée `val`).

③ Calcul de la capacité du condensateur.

④ Affichage des résultats grâce à `Serial.print`.

④ Décharge des condensateurs.

Remarques

- On voit (étape ④) que la valeur de C mesurée est proportionnelle à C_e , qui n'est pas connue. Ceci explique pourquoi l'étalonnage préalable (**protocole 1**) est indispensable pour régler la valeur de C_e (`CAPA_ENTREE`).
- La relation utilisée ($C(u_m - u) = C_e u$) montre aussi les limites de la méthode. Si C est très grande devant C_e , la valeur de $u_m - u$ est très faible, quasiment nulle. Or, cette valeur n'est connue qu'au bit près, avec 1024 valeurs entre 0 et 1023. Lorsque cette valeur en bits est à quelques unités, l'incertitude est très grande. On limite donc l'utilisation de cette méthode de mesure aux capacités assez faibles. Par exemple, pour un écart relatif inférieur à 10 %, il faut que $u_m - u$ soit supérieur ou égal à 10 en numérique (bits). Si $u_m - u$ est égal à 10, alors u vaut 1013. Pour une capacité d'entrée C_e typiquement égale à 25 pF, cela donne la capacité maximale pouvant être mesurée : $C_{\max} = 1013 \times \frac{25 \text{ pF}}{10} = 2,5 \text{ nF}$.