

AE.18B - Constante de temps τ d'un circuit RC

Le contexte

Vérifier la capacité affichée sur le condensateur

Documents mis à disposition

Document 1 : Le circuit RC série

Un circuit RC série est un circuit électrique, composé d'une résistance et d'un condensateur montés en série.

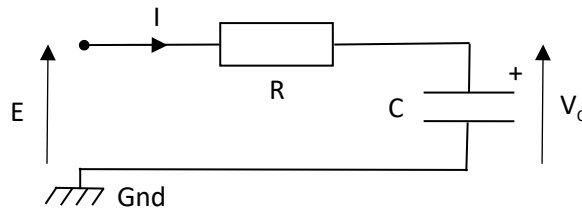


Figure 2 : Schéma électrique d'un circuit RC série

Document 3 : Réponse d'un circuit RC à un échelon de tension

On appelle **échelon de tension** le passage brutal de la tension appliquée à l'ensemble $\{R+C\}$ d'une valeur nulle à une valeur non nulle : on suppose qu'à $t < 0$, la **tension d'alimentation** du circuit est nulle et qu'à partir de $t = 0$ elle est égale à une **constante** E .

Le condensateur initialement déchargé se charge. **Au cours de sa charge** la tension V_C à ses bornes s'écrit :

$$V_C = E \times \left(1 - e^{-\frac{t}{RC}}\right)$$

Le produit RC est homogène à une durée appelée **constante de temps**, notée τ et exprimée en s.

Constante de temps

Au cours de la charge d'un condensateur à travers une résistance R et sous une tension E du générateur :

- ✚ à $t = \tau$, $V_C \approx 0,63 \times E$ (soit 63% de E) ;
- ✚ à $t = 3\tau$, $V_C \approx 0,95 \times E$ (soit 95% de E) ;
- ✚ à $t = 5\tau$, $V_C \approx 0,99 \times E$ (soit 99% de E) ;

de E) (Le condensateur est pratiquement chargé).

La constante donne donc un ordre de grandeur de la charge d'un condensateur.

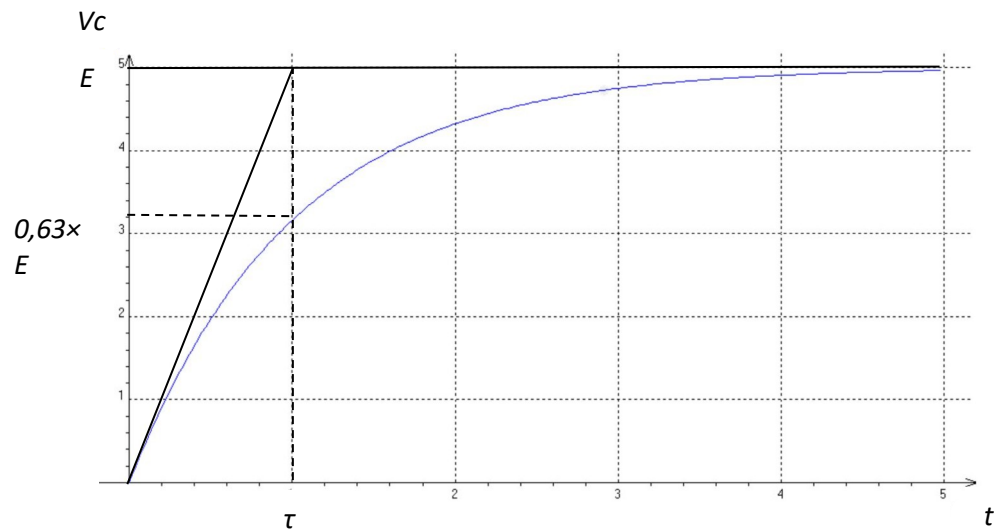


Figure 3 : Tension V_c aux bornes d'un condensateur au cours de sa charge

Document 4 : Le montage

Montage proposé par Christophe BELLESSORT (académie de Normandie)

Principe

Le microcontrôleur est utilisé, d'une part pour alimenter le circuit RC série (broche numérique 2) et d'autre part pour relever mesurer la tension aux bornes du condensateur (broche analogique A0).

On place un bouton poussoir en parallèle du condensateur pour permettre une décharge instantanée avant de mesurer la durée de la charge.

Une fois le condensateur déchargé, la sortie numérique 2 de la carte passe de l'état bas à l'état haut et alimente le circuit série sous une tension de 5,0 V.

Schématisation

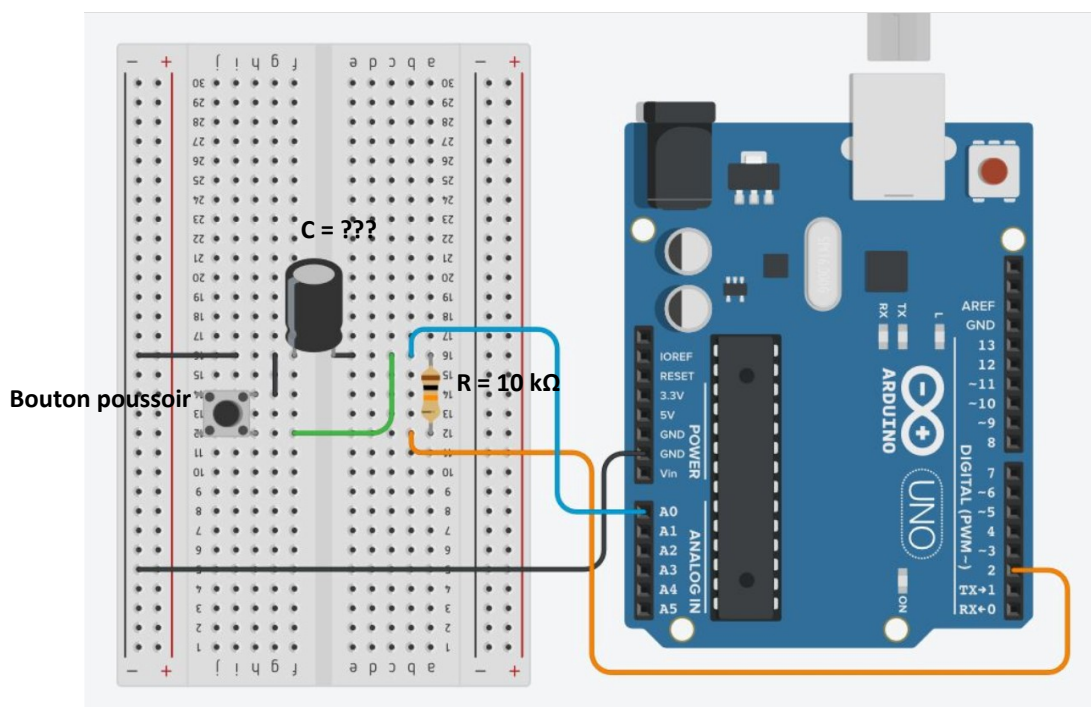


Figure 4 : Schématisation du montage réalisée avec le simulateur Tinkercad

A. Problématique et démarche expérimentale

1. A l'aide des documents proposer une démarche expérimentale permettant de répondre à la problématique ? **ANA**

APPEL n°1		
	Appeler le professeur pour lui présenter votre démarche expérimentale ou en cas de difficultés	

B. Etude préliminaire de la charge du condensateur

1. Réaliser le câblage du montage présenté dans le **Document 4**. **REA**
2. Téléverser le programme « charge_capa.ino » dans la carte. Suivre la charge du condensateur à partir du moniteur série. Lorsque le message « *Fin de la charge du condensateur* » s'affiche, décocher le **Défilement automatique** et copier les valeurs de « *Vc* » et de « *temps* » associées à la charge. Copier ces valeurs dans *Regressi* (**Edition** → **Copier document**) et représenter $V_c = f(t)$. **REA**
3. Déterminer graphiquement la valeur de τ . **REA**

APPEL n°2		
	Appeler le professeur pour lui présenter votre travail ou en cas de difficultés	

C. Réponse à la problématique

Document 5 : Script Arduino pour la détermination de τ

```

/* Détermination de la constante de temps d'un circuit RC - Script rédigé par Cristophe BELLESSORT (académie de Normandie)
 * Modifications mineures apportées par Marie-Anne DEJOAN (académie de la Guyane) */

// Définition de l'entrée numérique utilisée
const int alimentation = 2; // La broche numérique 2 alimente le circuit RC
// Définition de l'entrée analogique utilisée et des variables décimales
int Lecture = A0; // La broche analogique A0 permet la mesure de la tension aux bornes du condensateur
float tau; // "tau" est une variable décimale
// Définition et mise à zéro des durées mesurées et intervenant dans le calcul de "tau"
unsigned long tempsDebut = 0; // unsigned long : variable de type long non signé
unsigned long tempsFin = 0;

// BOUCLE D'INITIALISATION : exécutée une seule fois
void setup() {
  Serial.begin(9600); // Ouvre le port série et fixe le débit de communication à 9600 bauds
  pinMode(alimentation, OUTPUT); // "alimentation" est une broche de sortie
  digitalWrite(alimentation, LOW); // A l'état initial "alimentation" est à l'état BAS
  pinMode(Lecture, INPUT); // "Lecture" est une broche d'entrée
}

// BOUCLE PRINCIPALE : exécutée à l'infini
void loop() {
  digitalWrite(alimentation, LOW);
  Serial.println("Décharge du condensateur");
  Serial.println("Pressez le bouton poussoir de court-circuit si la décharge trop longue");
  while (analogRead(Lecture) > 0) { // On s'assure que le condensateur se décharge
    float Vmesuree = analogRead(Lecture);
    float Ucapa = (Vmesuree/1023)*5; // On convertit "Vmesuree" en une tension comprise entre 0 et 5V
    Serial.print("La tension aux bornes du condensateur est égale à : ");
    Serial.print(Ucapa); // On lit la tension aux bornes du condensateur (valeur entre 0 et 1023)
    Serial.println(" V");
    delay(500); // Temporisation de 500 ms
  }
  Serial.println("Nouvelle mesure dans 5 secondes");
  Serial.println("Lachez l'interrupteur de court-circuit");
  delay(5000);
  digitalWrite(alimentation, HIGH); // On alimente le circuit RC avec une tension de 5V
  tempsDebut = micros(); // Mémorisation du temps au début de l'alimentation du circuit RC
  while (analogRead(Lecture) < 647) { // Tant que le niveau lu est inférieur à 647 (i.e. 63.2% de la valeur numérique 1023)
    // Ne rien faire
  }
  tempsFin = micros(); // Durée écoulée depuis le début du lancement du programme
  tau = (tempsFin - tempsDebut) * 1e-6; // Calcul de tau exprimé en s
  Serial.print("La constante de temps du circuit RC est égale à : ");
  Serial.print(tau, 3); // "tau" avec 3 décimales
  Serial.println(" s");
  delay(5000);
}

```

4. Ouvrir et téléverser dans la carte, le programme « circuit_rc.ino » dont le script est présenté dans le **Document 5**. Comparer la valeur de τ qui s'affiche à celle déterminée précédemment. **REA**
5. Modifier le script afin de répondre plus explicitement à la problématique. **REA**



Appeler le professeur pour lui présenter les modifications apportées au programme

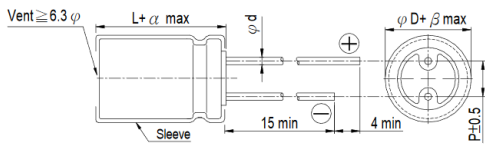


6. Une fois vérifié par votre professeur, téléverser le programme modifié. **REA**
7. Répondre à la problématique. **VAL**

D. Prolongement : Etude statistique

On mène à présent une étude statistique en renouvelant la mesure un grand nombre de fois pour plus de précision. On calculera ainsi la valeur moyenne de la capacité, elle constitue le meilleur estimateur ainsi que l'incertitude-type qui est une estimation de l'étendue des valeurs que l'on peut raisonnablement attribuer à une grandeur.

Document 6 : Extrait de la fiche technique du condensateur

Items	Performance																																																													
Category Temperature Range	-40℃ ~ +85℃																																																													
Capacitance	100 μF																																																													
Capacitance Tolerance	±5 %																																																													
Diagram of Dimensions	 <table><tr><th colspan="9">Lead Spacing and Diameter</th><th>Unit: mm</th></tr><tr><td>φ D</td><td>5</td><td>6.3</td><td>8</td><td>10</td><td>12.5</td><td>16</td><td>18</td><td>22</td><td>25</td></tr><tr><td>P</td><td>2.0</td><td>2.5</td><td>3.5</td><td>5.0</td><td>5.0</td><td>7.5</td><td>7.5</td><td>10</td><td>12.5</td></tr><tr><td>φ d</td><td colspan="2">0.5</td><td colspan="3">0.6</td><td colspan="2">0.8</td><td colspan="2">1.0</td></tr><tr><td>α</td><td colspan="3">1.0</td><td colspan="6">L<20: 1.5, L≥20: 2.0</td><td>2.0</td></tr><tr><td>β</td><td colspan="9">0.5</td></tr></table>	Lead Spacing and Diameter									Unit: mm	φ D	5	6.3	8	10	12.5	16	18	22	25	P	2.0	2.5	3.5	5.0	5.0	7.5	7.5	10	12.5	φ d	0.5		0.6			0.8		1.0		α	1.0			L<20: 1.5, L≥20: 2.0						2.0	β	0.5								
Lead Spacing and Diameter									Unit: mm																																																					
φ D	5	6.3	8	10	12.5	16	18	22	25																																																					
P	2.0	2.5	3.5	5.0	5.0	7.5	7.5	10	12.5																																																					
φ d	0.5		0.6			0.8		1.0																																																						
α	1.0			L<20: 1.5, L≥20: 2.0						2.0																																																				
β	0.5																																																													

Document 7 : Protocole pour l'étude statistique de la mesure de C

- 🔧 Ouvrir le programme « circuit_rc_stat.ino ».
- 🔧 Modifier le programme afin de réaliser 30 mesures consécutives de la valeur de la capacité. Exécuter le programme.
- 🔧 Sélectionner dans le moniteur série toutes les valeurs numériques (n° de la mesure et valeur de C en μ F).
Attention : ne pas sélectionner les chaînes de caractères (en-tête des colonnes et remarque finale).
- 🔧 Copier ces valeurs (raccourci clavier : **ctrl+c**).
- 🔧 Ouvrir le tableur **Excel** et coller (raccourci clavier : **ctrl+v**) les valeurs préalablement sélectionnées.

 Sauvegarder le fichier (dans le dossier spécifié par votre professeur) en le nommant « **capa.csv** » :

CSV (séparateur : point-virgule) (*.csv)

 Enregistrer

 A partir du logiciel **Edupython** ouvrir le programme : « Etude_statistique.py ». Interpréter le programme.

8. Mettre en œuvre le protocole du **Document 7**. Dans la console d'Edupython, relever la **valeur moyenne** de la capacité et l'**incertitude-type**.

Attention : conserver un nombre de décimales cohérent pour ces deux valeurs.

Observer l'étendue de mesures grâce à l'histogramme. **REA**

9. Renouveler les étapes du protocole pour 100 puis 200 mesures consécutives. **REA**

10. Quelles observations portant sur l'incertitude-type et l'étendue de mesures peut-on réaliser lorsqu'on augmente le nombre de mesures ? **VAL**

11. A l'aide de la fiche technique donner un intervalle dans lequel se situe la valeur réelle de la capacité. **VAL**

12. Le **résultat de la mesure (valeur moyenne et incertitude-type)** est-il cohérent avec les indications de la fiche technique ? **VAL**

13. Quelle expression de la valeur de la capacité est la plus précise ? **VAL**

Annexe 1 : Programme Arduino pour une étude statistique

```
/* Détermination de la valeur de la capacité d'un condensateur à partir de la constante de temps d'un circuit RC
 * Ce script a été écrit par Cristophe BELLESSORT (académie de Normandie)
 * Mme Marie-Anne DEJOAN (académie de Guyane) a apporté quelques modifications
 * Le programme effectue 100 mesures successives de la capacité d'un condensateur.*/

// Définition de l'entrée numérique utilisée
const int alimentation = 2; // La broche numérique 2 alimente le circuit RC

// Définition de l'entrée analogique utilisée et des variables décimales
int Lecture = A0; // La broche analogique A0 permet la mesure de la tension aux bornes du condensateur
float tau; // "tau" est une variable décimale
float capa; // "capa" est une variable décimale

// Définition de la constante R
const int R = 10000; // Valeur de la résistance R exprimée en Ohm

// Définition et mise à zéro des durées mesurées et intervenant dans le calcul de "tau"
unsigned long tempsDebut = 0; // unsigned long : variable de type long non signé
unsigned long tempsFin = 0;

int compteur = 0; // Initialisation du Compteur de mesures

void setup() {
  Serial.begin(9600) ; // Ouvre le port série et fixe le debit de communication à 9600 bauds
  pinMode(alimentation, OUTPUT); // "alimentation" est une broche de sortie
  pinMode(Lecture, INPUT); // "Lecture" est une broche d'entrée
  Serial.print("\n");
  Serial.print("\t"); // Ajout d'un espace
  Serial.println("C en uF");
}
```

```

// BOUCLE PRINCIPALE : exécutée à l'infini
void loop() {
  while (compteur < 100){
    digitalWrite(alimentation, LOW);
    compteur++; // Incrémente "compteur" de un
    while (analogRead(Lecture) > 0) { // Décharge du condensateur
      // Ne rien faire
    }
    digitalWrite(alimentation, HIGH); // On alimente le circuit RC avec une tension de 5V
    tempsDebut = micros(); // Mémorisation du temps au début de l'alimentation du circuit RC
    while (analogRead(Lecture) < 647) { // Tant que le niveau lu est inférieur à 647
      // i.e. 63.2% de la valeur numérique 1023
      // Ne rien faire
    }
    // Une fois sorti de la boucle on mesure tempsFin
    tempsFin = micros(); // Durée écoulée depuis le début du lancement du programme
    tau = (tempsFin - tempsDebut) * 1e-6; // Calcul de tau exprimé en s
    capa = (tau/R)*1e6; // Calcul de C exprimé en micro Farad
    Serial.print(compteur);
  }
}

## CALCUL DE LA MOYENNE + ECART TYPE + INCERTITUDE TYPE

n=len(capa) # Compte Le nombre d'éléments présents dans Le tableau à 1 dimension et L'affecte à La variable n
print('Nombre de mesures :', n)
print('Moyenne de C :',mean(capa),' s') # Affiche La moyenne sur L'ensemble des mesures effectuées
s=std(capa, ddof = 1) # Calcul L'écart-type à n-1 (ou déviation standard au sens statistique) et L'affecte à La variable s
print('Ecart-type non biaisé pour la série de mesures de C :', s, 'uF') # Affiche L'écart-type non biaisé
U=s/sqrt(n) # Calcul de L'incertitude-type
print('Incrtitude-type sur la mesure de C :', U, 'uF') # Affiche L'incrtitude-type

## TRACE DES HISTOGRAMMES

figure("Capacité du condensateur") # Définit Le nom de La figure
hist([capa],range=(min(capa),max(capa)),bins=30,edgecolor = 'blue') # La fonction bins est associée
# au nombre d'intervalles pour L'axe des abscisses

ylabel('Nb de mesures') # Définit Le nom des ordonnées
legend(["Capacité en uF"]) # Définit La légende
show() # Affiche L'histogramme

```

re.

