



FICHE-TP

Etude du dipôle RC série soumis à un échelon de tension

NIVEAU	Terminale spécialité phys-chim.
Partie du programme officiel	Ondes et signaux
Capacité expérimentale ou numérique	Étudier la réponse d'un dispositif modélisé par un dipôle RC. Déterminer le temps caractéristique d'un dipôle RC à l'aide d'un microcontrôleur

Présentation du TP :

Objectif : Utiliser un microcontrôleur pour étudier au cours du temps la tension aux bornes du condensateur d'un dipôle RC série soumis à un échelon de tension.

Démarche de l'étude et outils :

Il s'agit de programmer la carte type UNO avec l'EDI Arduino pour qu'elle soumette le dipôle RC à une tension de 5V (charge) ou 0V (décharge), enregistre la tension du condensateur au cours du temps, affiche les mesures sur un graphique et enfin mesure et affiche le temps caractéristique :

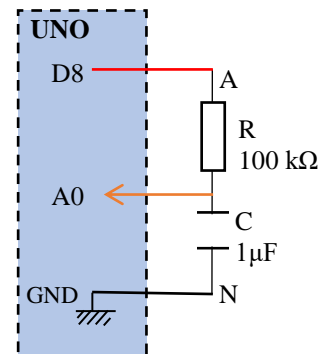
- la carte de type UNO joue à la fois le rôle de générateur de tension et d'interface d'acquisition ;
- l'IDE Arduino sert à la fois à la programmation et comme grapheur à l'exécution.

Pour des instructions et explications détaillées sur l'utilisation de l'EDI Arduino, et notamment du traceur avec RC, consulter la ressource académique en ligne [Document-3-Programmation-avec-Arduino-IDE.pdf](#).

Matériel : carte type UNO R3 + EDI Arduino, platine d'essais, résistor et condensateur tels que $RC \sim 10^{-1} s$ + fils Dupont M/M.

Principe du dispositif :

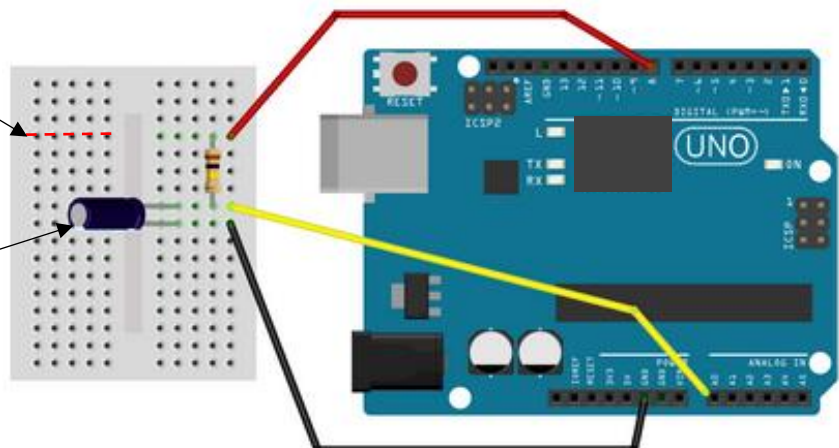
on enregistre sur l'entrée analogique A0 la tension $u_C(t)$ aux bornes du condensateur quand le dipôle série RC est soumis à un échelon de tension entre la masse et la sortie D8 (cf. schéma ci-contre).



1) Schéma du montage

Les points de la platine sont reliés selon la largeur

! Condensateur polarisé :
La borne du condensateur marquée par un signe - doit être connectée à GND



2) Travail demandé aux élèves : *exemple de déroulé possible*

Place dans la progression : on suppose que le condensateur a déjà été présenté et étudié ; il s'agit d'une vérification expérimentale (démarche déductive).

1) **Introduction** (accroche et objectif ou problème posé).

2) **Mise en œuvre du dispositif d'étude de la charge du condensateur :**

Réaliser le montage à l'aide du schéma fourni.

Avec l'IDE Arduino, ouvrir le programme [RC charge traceur .ino](#) (cf. ci-après) qui permet de commander la charge du condensateur, de tracer le graphique de $u_c(t)$ et de mesurer le temps caractéristique.

Le téléverser puis ouvrir le traceur série pour visualiser son exécution.

La définition du temps caractéristique est donnée dans l'énoncé du TP.

3) **Adaptation du code pour étudier la décharge :**

Modifier le code pour que la carte UNO commande la décharge du condensateur, trace le graphique de $u_c(t)$ et mesure le temps caractéristique. Sauvegarder sous un autre nom.

4) **Validation :**

Comparer la valeur mesurée de τ à la valeur calculée à partir des données R et C des fabricants.

3) Fichiers ino

Fichier de départ proposé aux élèves :

Nom du fichier :

Capture d'écran :

[RC charge traceur .ino](#)

```
7 // Déclaration des constantes
8 byte Te = 10; // durée fixe entre deux acquisitions (petite valeur entière en ms)
9
10 // Déclaration des variables
11 unsigned long t; // temps qui prendra des grandes valeurs entières positives
12 unsigned long t0; // origine des temps (début charge)
13 unsigned long t1 = 0; // servira à mémoriser l'instant de l'acquisition
14 unsigned int tau = 0; // temps caractéristique en ms et en valeur entière positive
15 float uc; // tension uc qui prendra des valeurs décimales
16
17 void setup() {
18 // Initialisation des ports
19 pinMode(8, OUTPUT); // définit la broche numérique 8 comme une sortie de la carte UNO
20 Serial.begin(9600); //ouvre une communication avec le port série de l'ordinateur
21
22 // Dans un premier temps, on s'assure que le condensateur est complètement déchargé
23 digitalWrite(8,LOW); // sortie numérique 8 mise à 0,0V
24 Serial.println("Patientez_3s_SVP décharge..."); // affiche un texte via le port série (en légende si traceur OU suivi d'un
25 delay(3000); // délai de décharge (valeur à adapter au temps caractéristique théorique RC)
26
27 // Dans un deuxième temps, charge du condensateur
28 Serial.println("uc=f(t) charge...");
29 uc = float(analogRead(A0))*5/1023; // lit la valeur numérique sur l'entrée A0 et la convertit en volts (0,0-5,0V = 0-1023)
30 Serial.println(uc); // affiche via le port série la valeur de uc (0,0V attendu), suivi d'un passage à la ligne si moniteur
31 digitalWrite(8,HIGH); // sortie numérique 8 mise à 5,0V (= 1023)
32 t0 = millis(); // définition de l'origine des temps à l'aide de la fonction millis() qui renvoie la
33 // date en ms de l'horloge interne d'Arduino prise à partir de sa mise sous tension
34
35 while( analogRead(A0) < 1022 ) { // boucle qui s'exécute tant que uc < 5,0V ;
36 // éviter 1023 comme valeur de comparaison car parfois cette valeur n'est pas atteinte
37 t = (millis() - t0) ; // mesure du temps écoulé depuis l'origine des temps t0
38 if ((t - t1) >= Te) { // bloc qui ne sera exécuté que si 10ms se sont écoulées depuis t1
39 uc = float(analogRead(A0))*5/1023;
40 Serial.println(uc); // affiche via le port série la valeur de uc, suivi d'un passage à la ligne si moniteur OU s
41 t1 = t; // mémorise l'instant de l'acquisition
42 if ((analogRead(A0) >= 0.63*1023) and (tau == 0)) { // condition de mesure de tau : à 63% de la tension de charge
43 tau = (t-5); // mesure de tau à mi-intervalle avec l'acquisition précédente de uc
44 }
45 }
46 }
47 digitalWrite(8,LOW); // remise à 0,0V de la sortie 8
48 Serial.print("uc(V)=f(t(cs)) "); // affiche du texte via le port série
49 Serial.print("tau="); //
50 Serial.print(tau); // affiche la valeur de tau via le port série
51 Serial.println("ms+/-5ms"); // affiche du texte via le port série, et passe à la ligne si moniteur
52 }
```