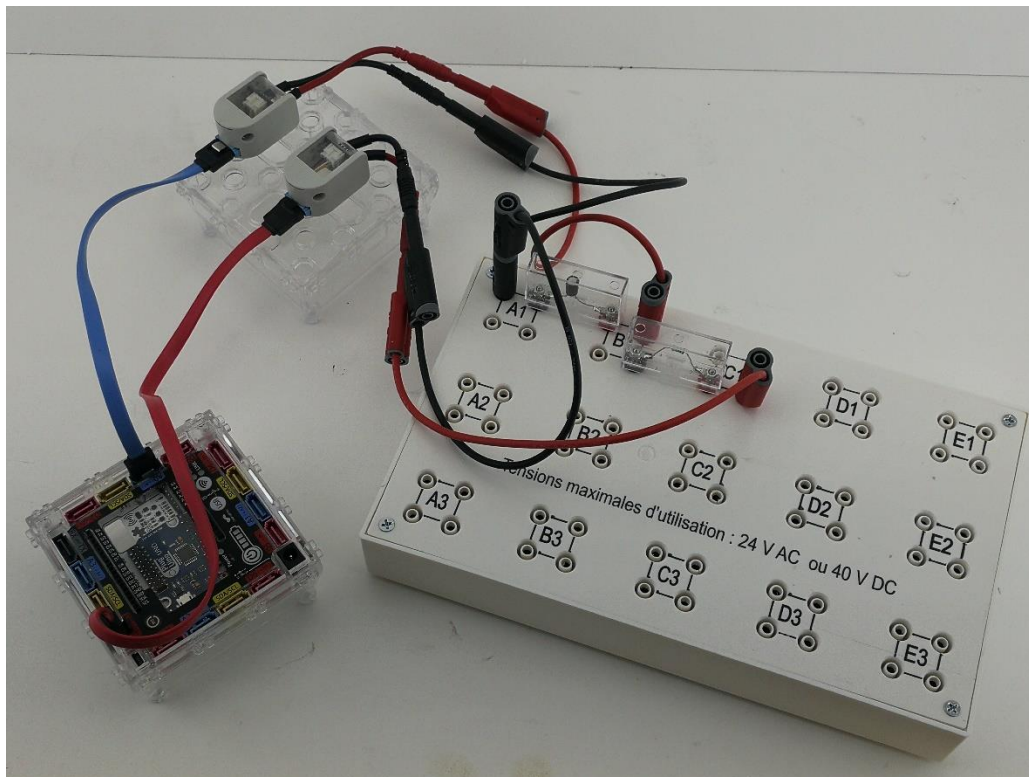


Niveau : Tale Spé PC	Ondes et signaux : Etudier la dynamique d'un système électrique
Modèle du circuit RC série : charge d'un condensateur par une source idéale de tension, décharge d'un condensateur, temps caractéristique.	Étudier la réponse d'un dispositif modélisé par un dipôle RC. Déterminer le temps caractéristique d'un dipôle RC à l'aide d'un microcontrôleur, d'une carte d'acquisition ou d'un oscilloscope. Réaliser un montage électrique pour étudier la charge et la décharge d'un condensateur dans un circuit RC.

Dossier TP Plug'Uino® : Etude de la charge et décharge d'un condensateur Programmation en langage Arduino



1. Résumé de l'activité

Cette activité expérimentale a pour but de :

- Mesurer et enregistrer à l'aide d'un microcontrôleur, l'évolution en fonction du temps de la tension aux bornes d'un condensateur lors de sa charge et de sa décharge.
- Déterminer le temps caractéristique d'un dipôle RC à l'aide d'un microcontrôleur.

2. Thème du programme abordé

Niveau : Tale	Ondes et signaux : Etudier la dynamique d'un système électrique
Modèle du circuit RC série : charge d'un condensateur par une source idéale de tension, décharge d'un condensateur, temps caractéristique.	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, d'une carte d'acquisition ou d'un oscilloscope. Réaliser un montage électrique pour étudier la charge et la décharge d'un condensateur dans un circuit RC.

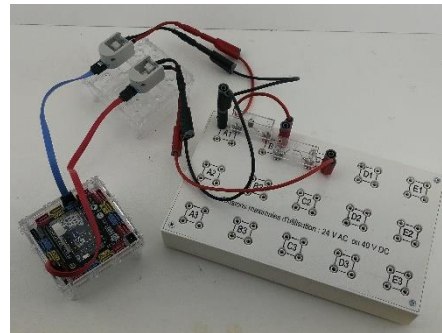
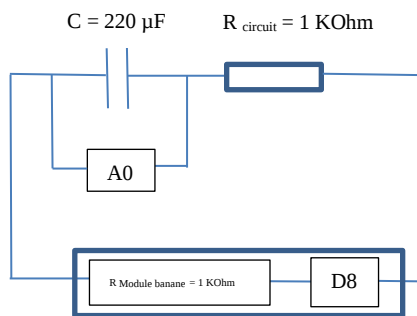
3. Matériel mis en œuvre

- 1 interface Plug'Uino® Réf. 650 003
- 2 modules de connexion Plug'Uino®, SATA -Connecteurs 2 fiches bananes femelle Réf. 651 056
(Le module intègre une résistance de 1 kOhms qui s'ajoute en série dans le circuit RC)
- 2 cordons SATA 25 cm Réf. 655 013
- 1 boîtier de montage Plug'Uino® Réf. 656 016
- 1 Platine de montage électronique Réf. 000 017
- 1 Boîtier dipôle Condensateur 220 μ F
- 1 Boîtier dipôle Résistance 1 kOhms

4. Présentation de l'expérience réalisée

Le montage comporte en série :

Un condensateur $C = 220 \mu\text{F}$, une résistance $R_{\text{circuit}} = 1 \text{ k}\Omega$, et la résistance interne de $1 \text{ k}\Omega$ du module Plug'Uino® de connexion SATA-Bananes qui alimente le circuit (connecté à la sortie D8). À l'aide de la sortie numérique D8 du microcontrôleur, on applique une tension de 5 V aux bornes du circuit RC. On peut schématiser le montage comme ci-dessous :



Simultanément on mesure sur l'entrée analogique A0 du microcontrôleur, l'évolution de la tension aux bornes du condensateur (capacité), à l'aide du deuxième module de connexion banane Plug'Uino®.

Le programme calcule le temps caractéristique du circuit RC.

Le microcontrôleur est particulièrement apte à faire de nombreuses mesures en un temps limité, il est donc parfaitement adapté à un enregistrement de l'évolution de la tension au cours du temps.

5. Programme Arduino

Le programme Arduino fonctionnel, permettant d'enregistrer et d'afficher le graphique de la tension est donné ci-après.

Ce programme est téléchargeable sur notre site au format «.ino »

(mot de passe pour ouvrir le fichier téléchargé : SCIENCETHIC).

charge_condensateur | Arduino 1.8.10

Fichier Édition Croquis Outils Aide

```

charge_condensateur

/***** Charge et décharge d'un condensateur *****/
*
* On mesure la tension aux bornes d'un condensateur pendant qu'il
* se charge, puis pendant qu'il se décharge. La mesure de la tension
* est prise toutes les 10 ms.
*
*
*****/

int etat = 0; // 0 au début, 1 pendant la charge
              // 2 pendant la décharge, 3 quand c'est fini
long previousMillis = 0;
long interval = 10; // Nombre de millisecondes entre chaque mesure

void setup() {

  pinMode(8, OUTPUT); // Définition de D8 en sortie pour alimenter le circuit RC
  Serial.begin(9600); // Définition de la vitesse de communication en bauds (Bit/seconde)

  // Dans un premier temps, on s'assure que le condensateur est
  // complètement déchargé

  Serial.println("Charge Décharge t=");
  digitalWrite(8,LOW); // Mise à l'état bas (0 V) de la sortie D8
  delay(2000); //Délai d'attente de 2000 ms

  // nouvel état:charge du condensateur
  etat = 1;

  digitalWrite(8,HIGH); // On alimente le circuit avec une tension de 5 V (délivrée par le microcontrôleur sur D8 )

}

```

```
void loop() {

    int currentMillis = millis();
    float tension; //Définition de "tension" comme un nombre à virgule flottante
    char t[50]; // Chaîne de caractères pour stocker l'information de la valeur du temps caractéristique
    float TensionCaract; //Définition de "TensionCaract" comme la valeur de la tension à t = Temps caractéristique

    if(currentMillis - previousMillis >= interval) {
        // Prendre une mesure à chaque intervalle de temps définit par la variable "interval" (interval = 10 ms)

        previousMillis = currentMillis;

        if (etat == 1) { // Charge du condensateur avec une tension de 5 V
            tension = analogRead(A0)*5.0/1024.0; //Conversion de la valeur numérisée de A0 en une valeur de tension en Volts.
            TensionCaract = 0;
            if (tension < 4.7){ // Si le condensateur n'est pas complètement chargé on rentre dans la condition ci-dessous
                if ( tension > 3.0 && tension < 3.1){ //Si la tension aux bornes du condensateur atteint 63% de la valeur max (5 V) alors...

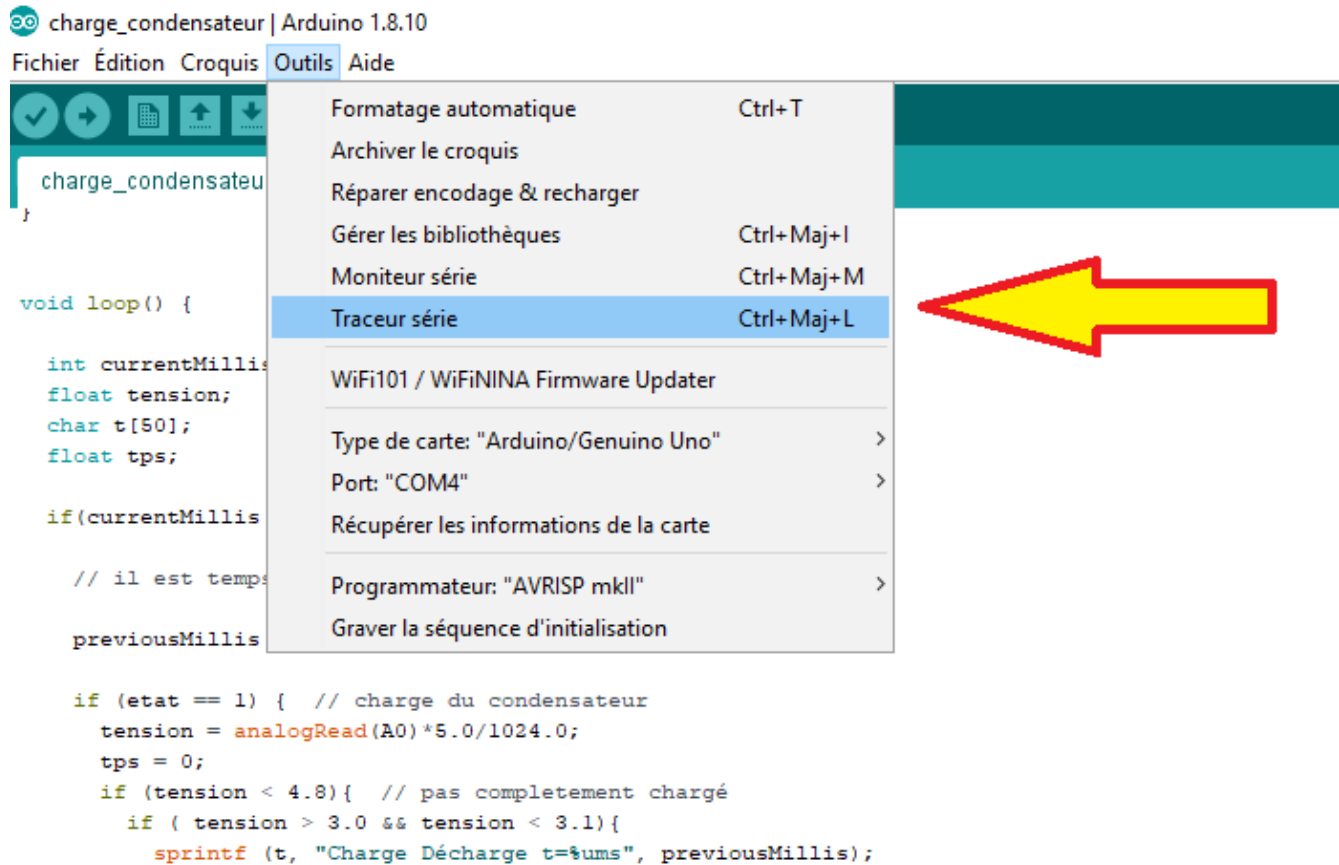
                    sprintf(t, "Charge Décharge t=%ums", previousMillis);
                    //Enregistre le temps caractéristique t (=RC) dans la légende en haut du graphique
                    TensionCaract = tension;
                    // Enregistre le moment où la tension atteint une valeur entre 3 et 3,1 V,
                    // c'est-à-dire le temps caractéristique à 63% de la valeur finale de la tension aux bornes du condensateur chargé.

                } //Trace les valeurs de la tension aux bornes du condensateur pendant la charge
                Serial.print(tension); //Trace en bleu la courbe de charge
                Serial.print(","); //Séparation des courbes
                Serial.print(0); //Trace en rouge la courbe de décharge
                Serial.print(","); //Séparation des courbes
                Serial.print(TensionCaract); //Trace en vert le moment du temps caractéristique
                Serial.println(); //Arrêt du tracage des courbes
            }

            else{ // Sinon, si le condensateur est complètement chargé, on change d'état: décharge du condensateur
                etat = 2;
                Serial.println(t); //Affiche le temps caractéristique t (=RC) dans la légende en haut du graphique
                digitalWrite(8, LOW); // On coupe l'alimentation du circuit, en mettant la sortie D8 à l'état bas (0 V)
            }
        }

        if (etat == 2) { // décharge
            tension = analogRead(A0)*5.0/1024.0; //Conversion de la valeur numérisée de A0 en une valeur de tension en Volts.
            if (tension > 0.01){ // pas complètement déchargé
                //Trace les valeurs de la tension aux bornes du condensateur pendant la charge
                Serial.print("Charge,"); // Maintien de la légende "Charge" et séparation de la couleur des courbes
                Serial.print(tension); //Trace en rouge la courbe de décharge
                Serial.println(); //Arrêt du tracage des courbes
            }
            else { // complètement déchargé
                etat = 3; // on change d'état: tout est terminé
            }
        }
    }
}
```

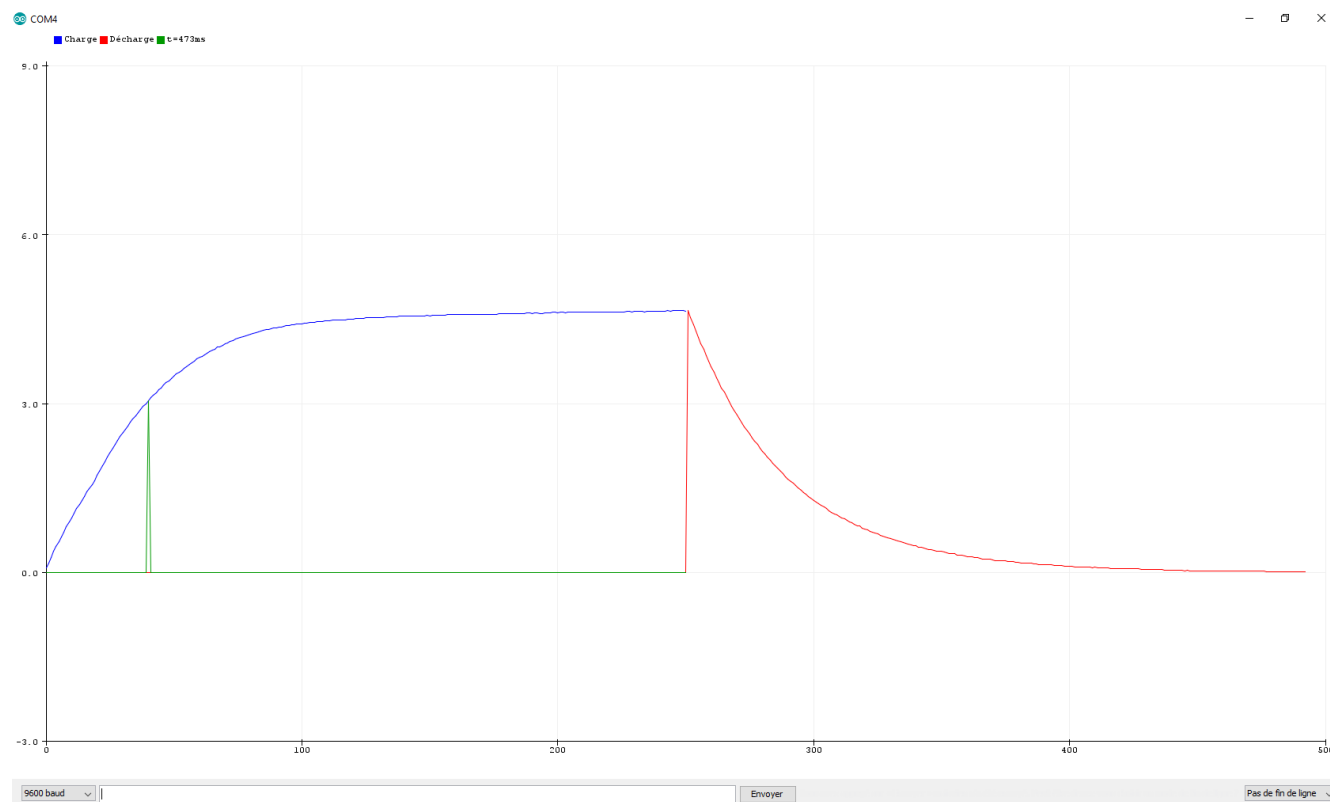
Aller dans le menu Outil/Traceur série



6. Résultats obtenus

Pour ce circuit RC, $R = (1+1) \text{ KOhm}$, $C = 220 \mu\text{F}$, la valeur théorique du temps caractéristique est $t = 440 \text{ ms}$.

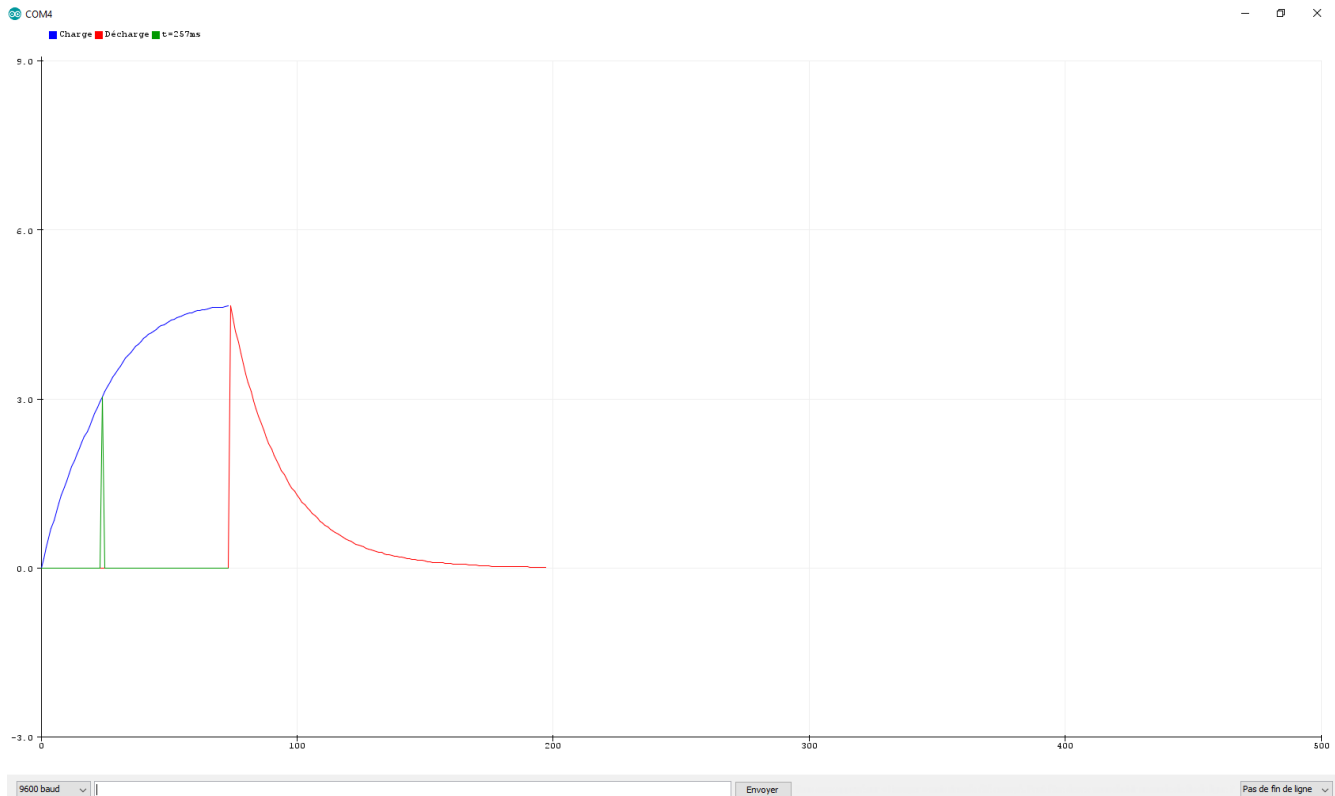
Nous trouvons expérimentalement $t = 473 \text{ ms}$, soit un écart de 7.5 %



En supprimant la résistance de 1 KOhm du circuit, il reste donc uniquement la résistance de 1 kOhms du module connecteurs bananes Plug'Uino.

La valeur théorique du temps caractéristique est $t = 0,00022 \times 1\,000 = 220\text{ ms}$

Nous trouvons expérimentalement $t = 257\text{ ms}$, soit un écart de 16.8 %



7. Exploitation

En fonction des capacités de programmation des différents élèves on peut leur demander diverses choses :

- à un niveau **d'initiation**, on peut enlever certains commentaires du programme et demander aux élèves à quoi correspond cette ligne ;
- à un niveau **plus avancé**, on peut effacer certaines lignes et demander aux élèves de programmer eux-mêmes ces lignes ;