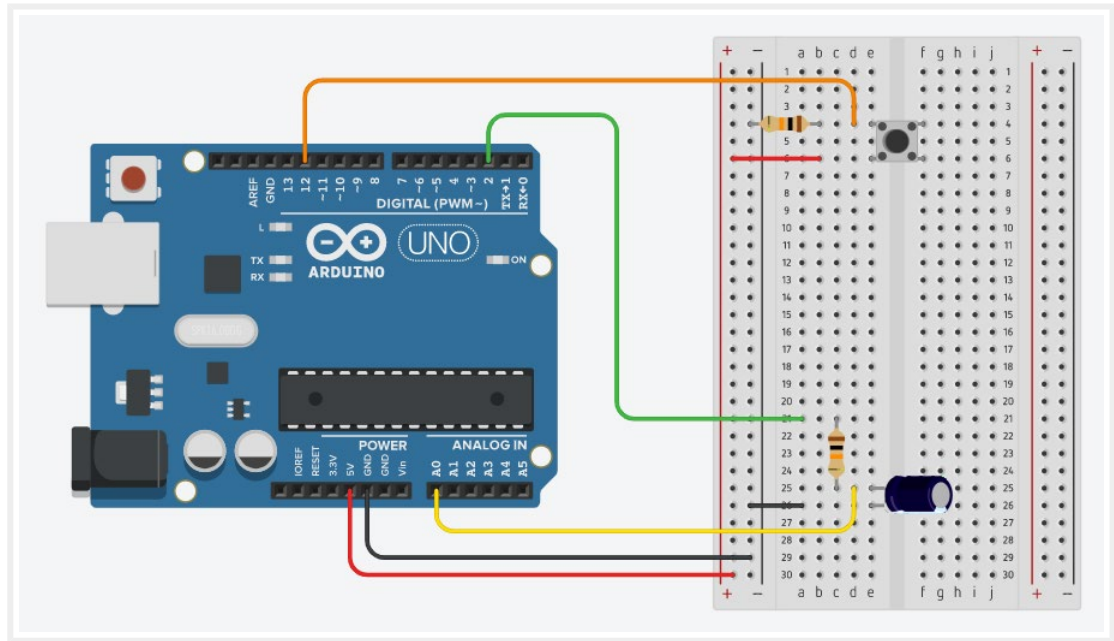


# Dipôles RC – Charge

## (Étude de la charge d'un condensateur d'un dipôle RC)



### Liste des composants :

- . 1 condensateur de 100  $\mu\text{F}$  (C chimique : **attention à la polarité**)
- . 2 résistances de 10  $\text{k}\Omega$  (résistance du circuit du bouton poussoir et du dipôle RC)
- . 1 bouton poussoir
- . 1 plaque d'essais
- . Fils de connexion

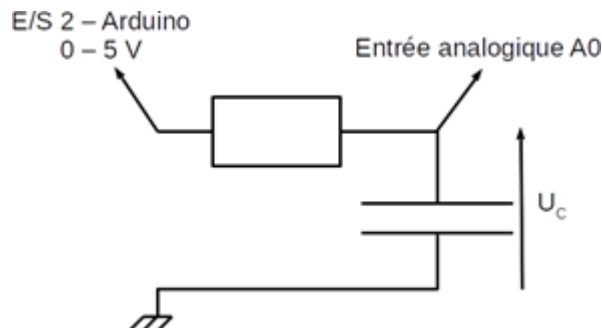
### Objectif

L'objectif de l'activité est suivre l'évolution temporelle de la tension aux bornes du condensateur lors de sa charge afin de vérifier la relation :

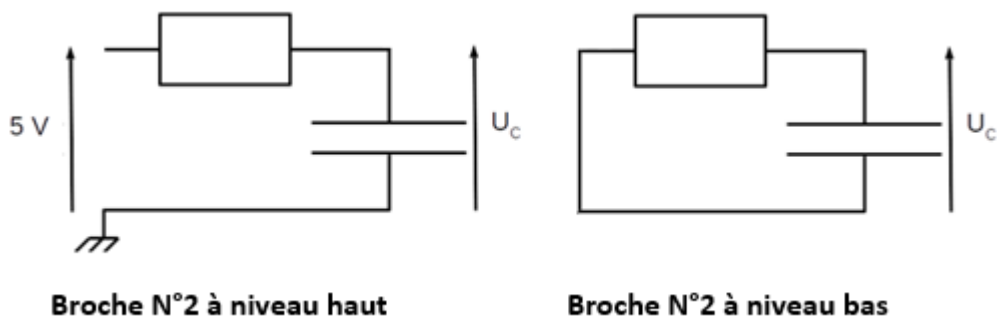
$$u_C(t) = E.(1 - e^{-\frac{t}{RC}})$$

Dans ce montage, La carte Arduino est utilisée :

- . pour appliquer une tension de 5 volts aux bornes du dipôle résistance – condensateur grâce à une sortie numérique (broche 2),
- . pour mesurer la tension aux bornes du condensateur à l'aide d'une entrée analogique (broche A0).



Suivant l'état logique de broche N°2 déclarée en sortie numérique, le schéma électrique sera équivalent à :



Avec un Arduino, pour obtenir des mesures suffisamment précises, il faut utiliser un dipôle RC conduisant à des constantes de temps longues (quelques centaines de millisecondes au minimum).

## . Descriptif de l'activité

Après avoir déchargé le condensateur, la mesure de la tension aux bornes du condensateur  $U_c$ , lors de la charge, à l'aide de l'entrée analogique A0 est lancée, à  $t = 0$  s, par un appui sur le bouton poussoir.

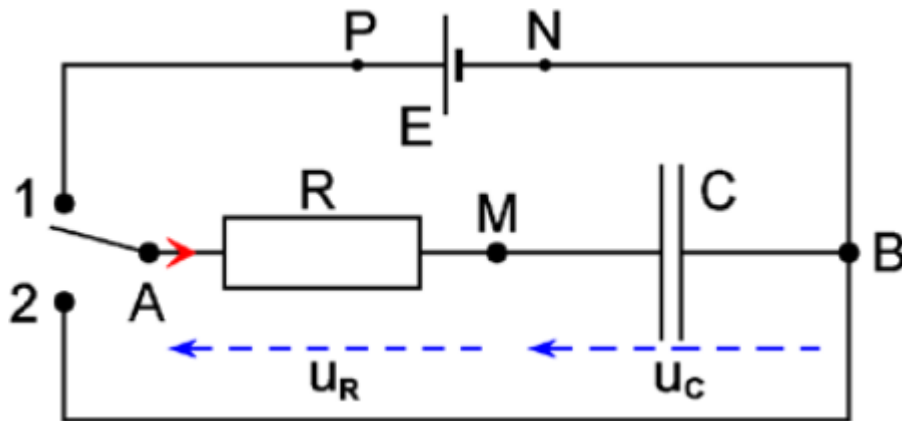
La valeur de la tension en V est affichée dans le moniteur série toutes les 100 ms.

Les mesures sont arrêtées en appuyant sur le bouton poussoir. Le condensateur est alors déchargé afin de pouvoir effectuer de nouvelles mesures en appuyant de nouveau sur le bouton poussoir.

Il est donc possible d'acquérir des couples de données  $(t, U_c)$  afin de vérifier la relation  $U_c = f(t)$  théorique.

## . Rappel : Le dipôle RC

Un dipôle RC est l'association en série d'un conducteur ohmique de résistance  $R$  et d'un condensateur de capacité  $C$ , comme dans le schéma ci-dessous :



### - Charge du condensateur

Le condensateur étant déchargé, on bascule, à l'instant  $t=0$ , l'interrupteur en position 1.

A chaque instant  $t > 0$ , on a :

$$V_A - V_B = u_R(t) + u_C(t) = R.i(t) + u_C(t) = E$$

$$\text{On a } i(t) = \frac{dq(t)}{dt} \text{ et } q(t) = C.u_C(t)$$

$$\text{donc } i(t) = C.\frac{du_C(t)}{dt} \text{ soit } R.C.\frac{du_C(t)}{dt} + u_C(t) = E$$

D'où :

$$\boxed{\frac{du_C(t)}{dt} + \frac{1}{R.C}.u_C(t) = \frac{E}{R.C}}$$

On dit que  $u_C(t)$  satisfait à une équation différentielle non homogène du premier ordre.

Avec  $q(t) = C.u_C(t)$ , on a

$$\boxed{\frac{dq(t)}{dt} + \frac{1}{R.C}.q(t) = \frac{E}{R}}$$

À  $t = 0$ , le condensateur est déchargé et  $q(0) = C.u_C(0) = 0$ , on en déduit :  $\left. \frac{du_C}{dt} \right|_{t=0} = \frac{E}{R.C}$

On a donc un point de la courbe représentative de  $u_C(t)$  :  $(0 ; 0)$  ainsi que la valeur de la pente de la tangente à cette courbe à l'origine des dates.

Au bout d'un temps assez long ( $t = \infty$ ) on peut considérer que le condensateur est chargé et qu'il ne passe plus de charge dans le circuit :  $\left. \frac{dq}{dt} \right|_{t=\infty} = C. \left. \frac{du_C}{dt} \right|_{t=\infty} = 0$  donc  $\left. \frac{du_C}{dt} \right|_{t=\infty} = 0$

De l'équation différentielle, on tire :  $u_C(\infty) = E$  tension aux bornes du générateur.

La courbe représentative de  $u_C(t)$  tend vers la valeur  $E$  qui représente une asymptote avec une pente nulle. La courbe tend exponentiellement vers cette valeur.

Les solutions de l'équation différentielle sont de la forme :

$u_C(t) = A.e^{-m.t} + B$  où  $m > 0$ ,  $m$  et  $B$  constantes d'intégration,  $A$  constante non définie.

Introduisons cette expression dans l'équation :  $\frac{d(A.e^{-m.t} + B)}{dt} + \frac{1}{R.C} \cdot (A.e^{-m.t} + B) = \frac{E}{R.C}$

D'où :  $-m.A.e^{-m.t} + \frac{A}{R.C} \cdot e^{-m.t} + \frac{B}{R.C} = \frac{E}{R.C}$

Cette équation doit être vérifiée à chaque instant, on en déduit :  $B = E$

D'où :  $-m.A.e^{-m.t} + \frac{A}{R.C} \cdot e^{-m.t} = 0$  et  $m = \frac{1}{R.C}$

La solution générale de l'équation différentielle s'écrit :  $u_C(t) = A.e^{-\frac{t}{R.C}} + E$

$A$  est une constante qui dépend des conditions initiales.

Ici, à  $t = 0$ , le condensateur est déchargé, donc :  $u_C(0) = 0 = A.e^{-0} + E$  d'où  $A = -E$

Compte tenu des conditions initiales imposées par l'expérience, la solution est :

$$u_C(t) = E.(1 - e^{-\frac{t}{R.C}})$$

En appliquant les relations :  $q(t) = C.u_C(t)$  et  $i(t) = C. \frac{du_C(t)}{dt}$

On a :  $q(t) = C.E.(1 - e^{-\frac{t}{R.C}})$  et  $i(t) = \frac{E}{R} \cdot e^{-\frac{t}{R.C}}$

Au bout d'un temps long :  $q(\infty) = C.E = Q_0$  le condensateur est chargé

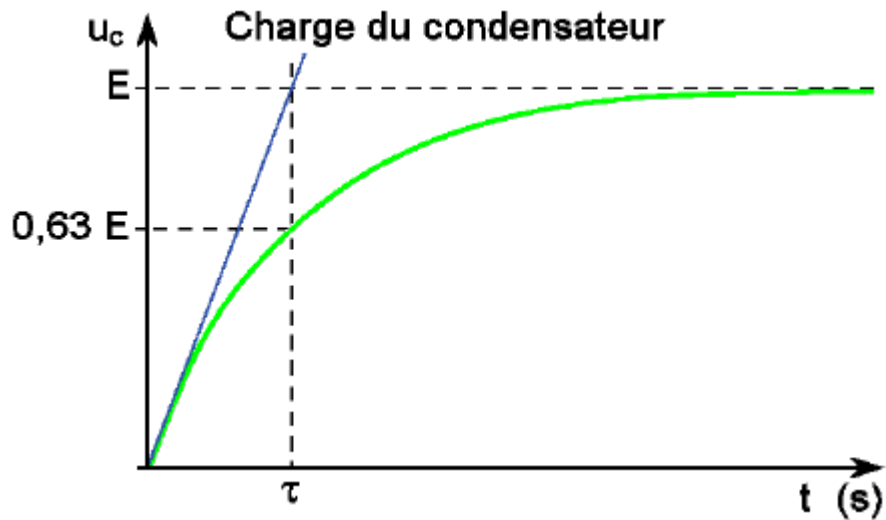
Et :  $i(\infty) = 0$  il n'y a plus de courant

## - Constante de temps du dipôle RC

Le produit  $R.C$  est homogène à un temps et est appelé constante de temps  $\tau$  du dipôle RC.

. Lors de la charge :

$$u_C(t) = E.(1 - e^{-\frac{t}{R.C}})$$



La tangente à la courbe à l'origine coupe l'asymptote  $y = E$  au point d'abscisse  $t = \tau$ .

En effet, la tangente à la courbe représentative de  $u_C(t)$ , à l'origine des dates, a pour équation :

$$y = \left. \frac{d[u_C(t)]}{dt} \right|_{t=0} . t = - E. \left( - \frac{1}{R.C} \right) . t = \frac{E}{R.C} . t$$

Elle coupe l'asymptote  $y = E$  en un point d'abscisse  $t : E/(R.C).t = E$  soit  $t = R.C = \tau$

Et :

$$u_C(\tau) = E.(1 - e^{-1}) \approx 0,63.E.$$

Par lecture graphique de l'abscisse du point de la courbe dont l'ordonnée est égale à  $0,63.E$ , on obtient la valeur de  $\tau$ .

## . Le programme

Voici le code de l'activité :

### Dipoles\_RC\_Charge

```
// Déclaration des constantes et variables

const int PinUC = 0;
const int PinButton = 12;
const int PinAlimC = 2;

int ValPinUC = 0;
float UC = 0.0;
unsigned long t0;
float dt;

int ValButton = 0;
int OldValButton = 0;
int State = 0;
int OldState = 0;

// Déclaration de fonctions

void decharge(){
    digitalWrite(PinAlimC, LOW);
    Serial.println("Decharge du condensateur");
    while (analogRead(PinUC) > 0) {
        delay(200);
    }
    Serial.println("Condensateur decharge");
}

// Initialisation des entrées et sorties

void setup() {
    Serial.begin(9600);
    pinMode(PinButton, INPUT);
    pinMode(PinAlimC, OUTPUT);
    decharge();
    Serial.println("Appuyez sur le bouton poussoir pour commencer les mesures.");
}
```

```

// Fonction principale en boucle

void loop() {
    ValButton = digitalRead(PinButton);
    delay(10);

    if ((ValButton == HIGH) && (OldValButton == LOW))
    {
        State=1-State;
    }
    OldValButton = ValButton;

    if (State==1)
    {
        if (OldState == 0)
        {
            Serial.println("Charge du condensateur en cours.");
            Serial.println("");
            Serial.println ("Temps (S);Uc (V):");
            t0 = micros();
            digitalWrite(PinAlimC, 1);
            OldState=1;
        }
        ValPinUC = analogRead(PinUC);
        UC = (ValPinUC/1023.0)*5.0;
        dt = (micros() - t0)* 1e-6;
        Serial.print(dt,2);
        Serial.print(";");
        Serial.println(UC,2);

        delay(100);
    }
    else
    {
        if (OldState == 1){
            Serial.println("Fin des mesures.");
            digitalWrite(PinAlimC, 0);
            OldState = 0;}
    }
}

```

### Déroulement du programme :

– 1. Déclaration des constantes et variables :

- . **const int PinUc = 0** (broche du condensateur : A0)
- . **const int PinButton = 12** (broche du bouton poussoir)
- . **const int PinAlimC = 2** (broche d'alimentation du dipôle RC)
- . **int ValPinUc = 0** (variable nombre entier pour stocker la valeur de la broche du condensateur)
- . **float Uc = 0.0** (variable nombre décimal pour stocker le résultat du calcul de la tension Uc)
- . **unsigned long t0** (variable nombre entier long pour stocker la date du début de charge )

- . **float dt** (variable nombre décimal pour stocker la différence de temps entre les mesures de Uc)
- . **int ValButton = 0** (variable nombre entier pour stocker la valeur de la broche du bouton poussoir)
- . **int OldValButton = 0** (variable nombre entier pour stocker la valeur précédente de la broche du bouton poussoir)
- . **int State = 0** (variable nombre entier correspondant à l'action à effectuer)
- . **int OldState = 0** (variable nombre entier correspondant à l'action effectuée précédemment)

– 2. Déclaration de fonctions :

–> Fonction permettant de décharger le condensateur :

– Mise à niveau bas de la broche d'alimentation du dipôle RC :

**digitalWrite(PinAlimC, LOW)**

– Attente de la fin de la décharge du condensateur :

**while (analogRead(PinUC) > 0) )**

– 3. Initialisation des entrées et sorties :

. **Initialisation de la liaison série à un débit de 9600 bauds,**

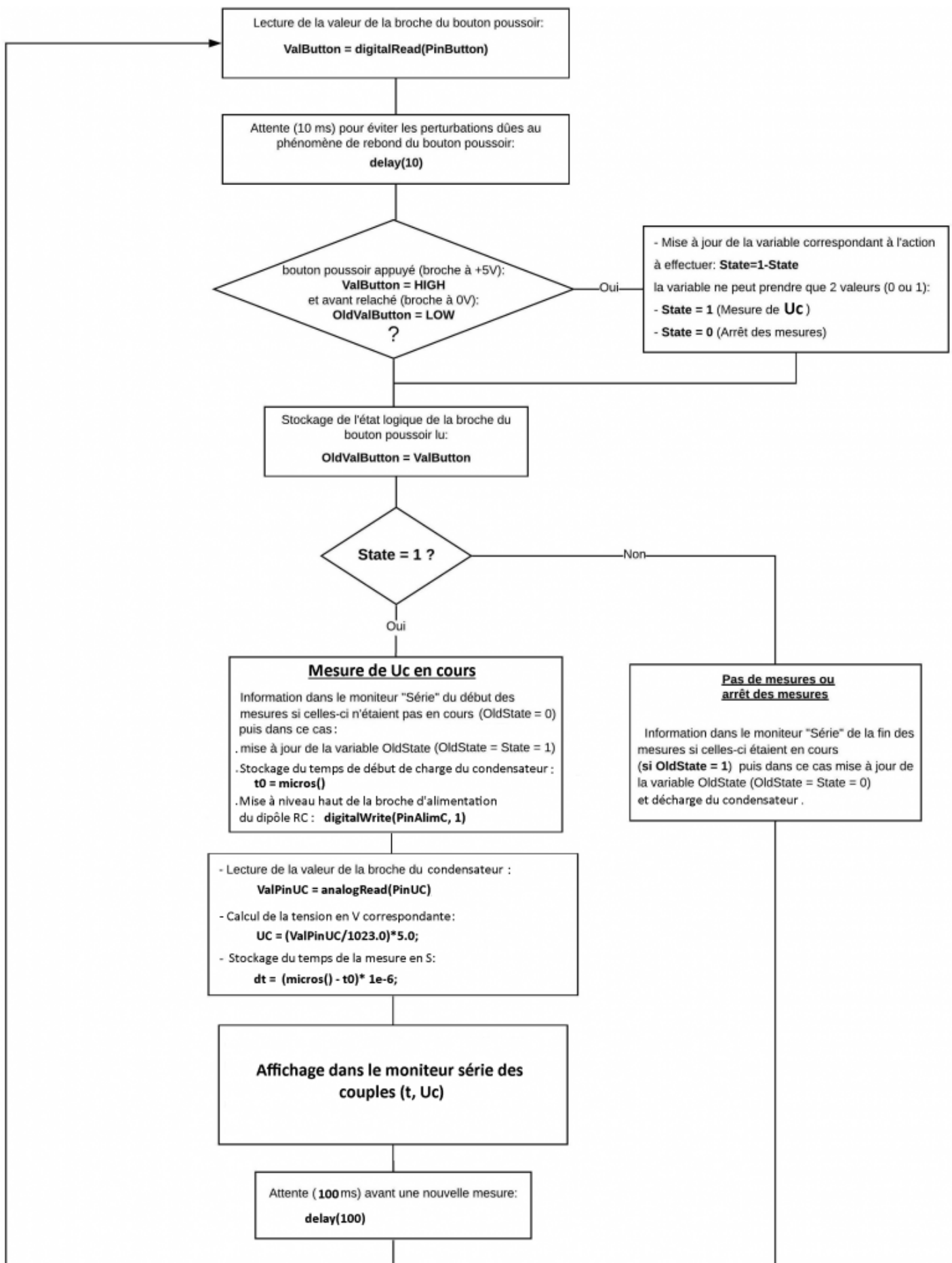
. **Initialisation de la broche du bouton poussoir en entrée,**

. **Initialisation de la broche d'alimentation du dipôle RC en sortie,**

. **Décharge du condensateur.**

– 4. Fonction principale en boucle :





Remarque :

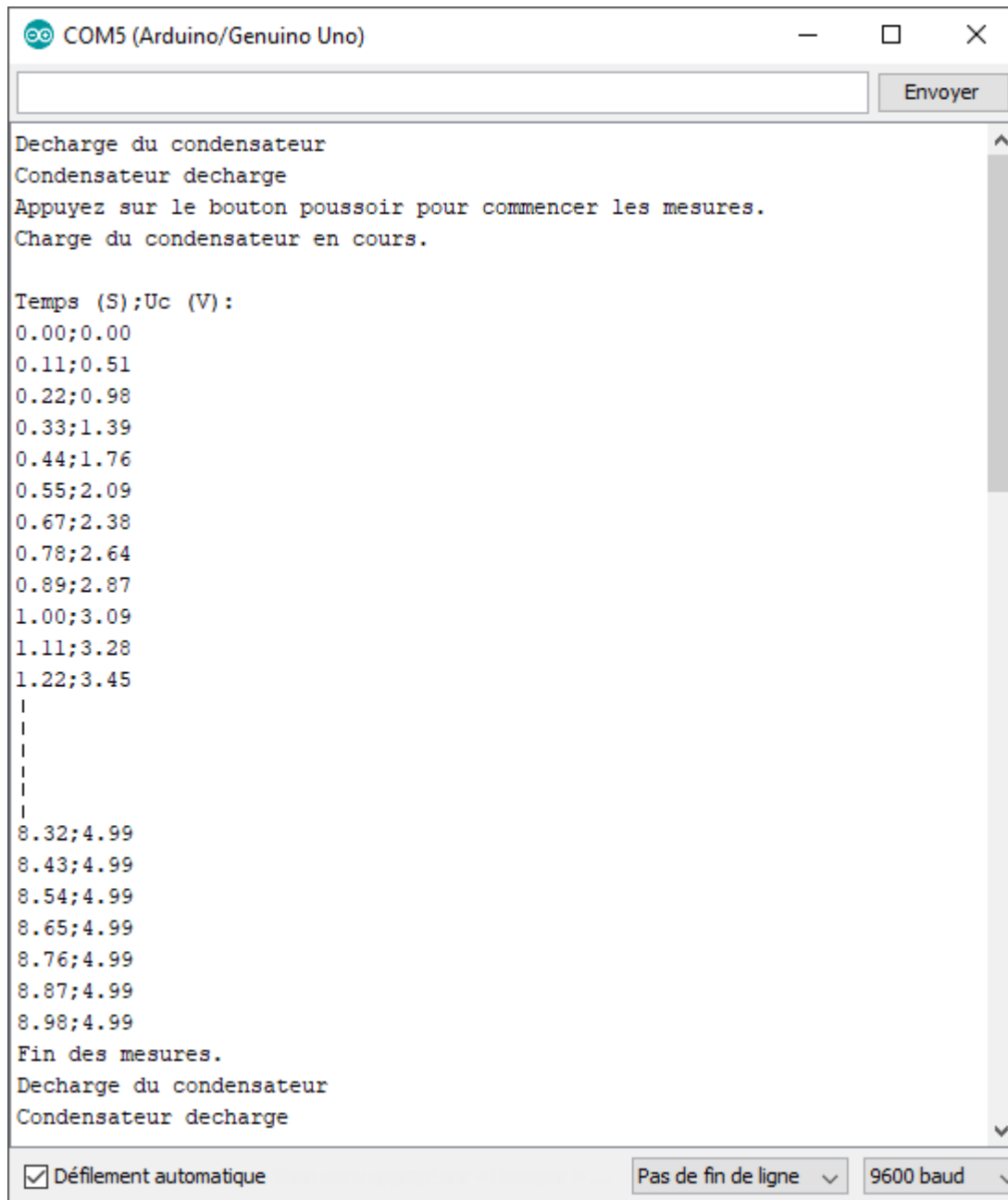
La fonction « **micros()** » renvoie le nombre de microsecondes, sous la forme d'un nombre de type unsigned long, depuis que la carte Arduino a démarré le programme en cours. Ce nombre repasse à 0 après

approximativement 70 minutes.

Syntaxe :

```
variable_unsigned_long = micros();
```

**Résultats dans le moniteur série :**



The screenshot shows the 'COM5 (Arduino/Genuino Uno)' serial monitor window. The text displayed is as follows:

```
Decharge du condensateur
Condensateur decharge
Appuyez sur le bouton poussoir pour commencer les mesures.
Charge du condensateur en cours.

Temps (S);Uc (V):
0.00;0.00
0.11;0.51
0.22;0.98
0.33;1.39
0.44;1.76
0.55;2.09
0.67;2.38
0.78;2.64
0.89;2.87
1.00;3.09
1.11;3.28
1.22;3.45
|
|
|
|
|
8.32;4.99
8.43;4.99
8.54;4.99
8.65;4.99
8.76;4.99
8.87;4.99
8.98;4.99
Fin des mesures.
Decharge du condensateur
Condensateur decharge
```

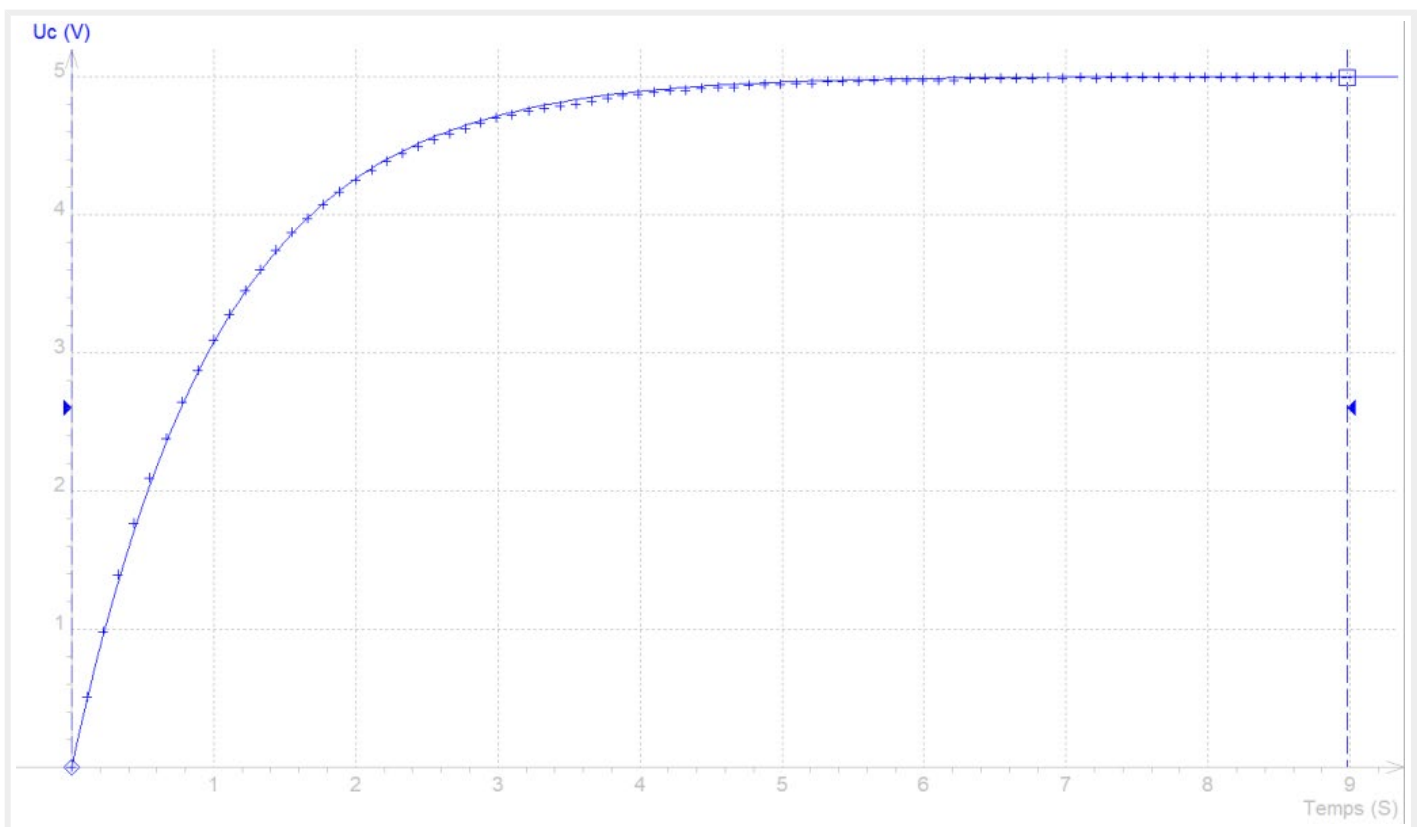
At the bottom of the window, the settings are: ☒ Défilement automatique, Pas de fin de ligne, and 9600 baud.

## **. Exploitation des mesures :**

Pour exploiter les mesures, il suffit de sélectionner et de copier toutes les données à partir du moniteur série et d'ouvrir un [nouveau fichier](#) dans Regressi à partir du "Presse-papier".

Grandeurs		
Paramètres Variables Expre		
Trier Ajouter Sup. colonne Sup. ligne In		
i	Temps S	Uc V
0	0,000	0,000
1	0,1100	0,5100
2	0,2200	0,9800
3	0,3300	1,390
4	0,4400	1,760
5	0,5500	2,090
6	0,6700	2,380
7	0,7800	2,640
8	0,8900	2,870
9	1,000	3,090
10	1,110	3,280
11	1,220	3,450
12	1,330	3,600
13	1,440	3,740
14	1,550	3,870
15	1,660	3,970
16	1,770	4,070

On peut alors tracer le graphe représentant la tension aux bornes du condensateur en fonction du temps :



Et effectuer une modélisation suivant :

$$u_c(t) = E.(1 - e^{-\frac{t}{R.C}})$$

Expression du modèle	Résultats de la modélisation
$U_c(\text{Temps}) = 5 * (1 - \exp(-\text{Temps}/\tau))$	Ecart expérience-modèle 0,38 % sur $U_c(\text{Temps})$ Ecart quad. $U_c = 17,13 \text{ mV}$  $\tau = 1,048 \pm 0,005$

La valeur théorique de  $\tau$  est :

$$\tau = RC = 10 \cdot 10^3 \times 100 \cdot 10^{-6} = 1 \text{ S}$$

Par la modélisation, la détermination de  $\tau$  donne une valeur très proche de la valeur théorique.

On peut également déterminer  $\tau$  à l'aide de la tangente à l'origine ou par la mesure du temps pour lequel le condensateur est chargé à 63 % ( $0,63 \times 5 = 3,15 \text{ V}$ ):

