

# 12.TP3 – Mesure d’une capacité à l’aide d’un microcontrôleur

**Notions abordées :**

- Modèle du circuit RC série : charge d’un condensateur par une source idéale de tension, décharge d’un condensateur, temps caractéristique.

**Compétences travaillées :**

- Réaliser un montage électrique pour étudier la charge et la décharge d’un condensateur dans un circuit RC.
- Déterminer le temps caractéristique d’un dipôle RC à l’aide d’un microcontrôleur.

**Matériel :**

**Salle informatique**

- Ordinateurs avec logiciel Mu ou Thonny
- Imprimante réseau

**Elève**

- Générateur de tension continu réglable
- Carte microbit
- Plaque d’essai pour microbit
- Résistances (non montées) : 1kΩ 4,7 kΩ
- Condensateurs (non montés) : 2 ou 3 entre 10 µF et 1000 µF

## I - But du TP

Le but de TP est de mesurer le temps de charge d’un dipôle RC.

- La démarche sera la suivante :
  - a => on décharge le condensateur ;
  - b => on déclenche un chronomètre
  - c => on démarre (en même temps que le déclenchement du chronomètre) la charge du condensateur ;
  - d => on attend que le condensateur soit chargé à 67,37 % ;
  - e => dès que c’est le cas, on arrête le chronomètre.
- Le matériel utilisé sera un microcontrôleur avec ses entrées et sorties : la carte microbit.
- Le microcontrôleur sera programmé en python.

## II - Les entrées/sorties analogiques de la carte microbit

### 1 - Rôle des entrées/sorties analogiques

Une entrée analogique permet de convertir la tension imposée par le circuit extérieur en un nombre binaire qui pourra être utilisé dans le programme.

Une sortie analogique permet de convertir un nombre binaire du programme en une tension délivrée par la carte.

Dans les deux cas, et pour la carte microbit :

- la tension est comprise entre 0 et la tension d’alimentation de la carte (en général 3,3 V) ;
- le nombre binaire est codé sur 10 bits, autrement dit, il correspond à un nombre entier compris entre 0 et 1023.

En résumé, voici le tableau de correspondance :

<i>Valeur utilisée dans le programme</i>	<i>Tension entre la masse et la borne d’entrée/sortie</i>
0	0 V
1023	3,3 V

### 2 - Instruction du langage

#### a - Lecture/écriture sur les entrées/sorties

La lecture sur l’entrée analogique N se fait à l’aide de la fonction `pinN.read_analog()` qui renvoie un nombre entier entre 0 et 1023 en fonction de la valeur de la tension entre la masse et la borne N.

L’écriture sur la sortie analogique N se fait à l’aide de la fonction `pinN.write_analog(i)` où i est un entier compris entre 0 et 1023. Cette fonction a pour effet d’imposer une tension comprise en 0 et 3,3 V entre la masse et la borne N en fonction de la valeur de i.

#### b - Valeur du temps

La fonction `running_time()` renvoie la valeur du temps écoulée entre le démarrage du programme et l’instant où elle est exécutée, en millisecondes.

### 3 - Sur la carte

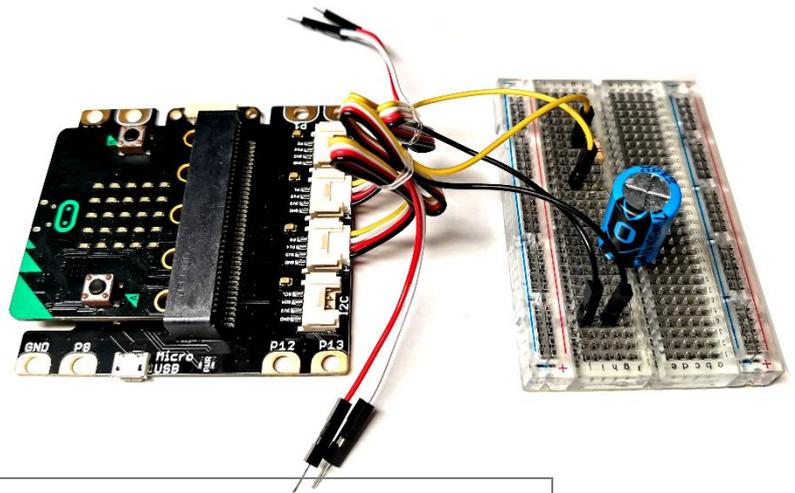
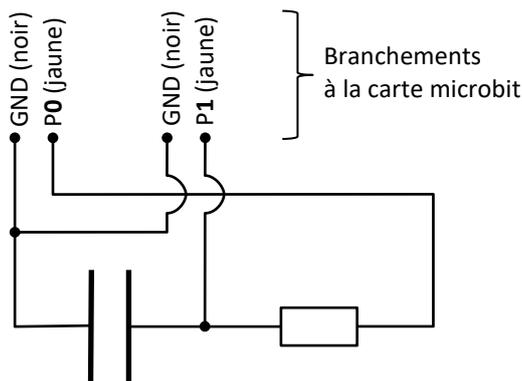
Sur la carte microbit, seules les bornes 0, 1, 2 et 3 sont des entrées/sorties analogiques.

Nous utiliserons :

- la borne 0 comme sortie analogie, c’est-à-dire comme générateur ;
- la borne 1 comme entrée analogie, c’est-à dire comme voltmètre.

### III - Principe de la mesure du temps de charge

#### 1 - Circuit électrique pour l'étude



#### 2 - Programme python

```

1 import microbit as mb
2 # Attente du clic sur le bouton A
3 print("Cliquer sur le bouton A pour décharger le condensateur.")
4 mb.display.show(mb.Image.ARROW_W)
5 while mb.button_a.was_pressed() == False:
6     mb.sleep(50)
7 # Décharge du condensateur
8 mb.display.show(mb.Image.HAPPY)
9 print("Decharge du condensateur en cours...")
10 mb.pin0.write_analog(0)
11 while mb.pin1.read_analog() > 5:
12     mb.sleep(50)
13 print("Condensateur déchargé")
14 # Attente du clic sur le bouton B
15 print("Cliquer sur le bouton B pour lancer la charge du condensateur.")
16 mb.display.show(mb.Image.ARROW_E)
17 while mb.button_b.was_pressed() == False:
18     mb.sleep(50)
19 # Charge du condensateur et mesure du temps
20 mb.display.show(mb.Image.HAPPY)
21 print("Charge du condensateur en cours...")
22 t0=mb.running_time()
23 mb.pin0.write_analog(1023)
24 while mb.pin1.read_analog() < 647:
25     pass
26 t1=mb.running_time()
27 tau =
28 #
29 print(tau)
30 mb.display.scroll(tau)
31 mb.display.clear()

```

### IV - Travail à faire

#### 1 - Questions sur le programme

- 1) Identifier, dans le programme, les blocs d'instructions qui correspondent aux 5 étapes décrites dans le I.
- 2) Expliquer la ligne 5 du programme.
- 3) Expliquer la ligne 24 du programme, et particulièrement l'origine de la valeur 647.
- 4) Expliquer le rôle des variables t0 et t1 et compléter la ligne 27.
- 5) Compléter le commentaire de la ligne 28.

#### 2 - Manipulations



Réaliser le câblage du circuit avec  $C = 470 \mu\text{F}$  et  $R = 4,7 \text{ k}\Omega$ .



Lancer le programme.

- 1) Comparer la valeur expérimentale mesurée à la valeur théorique attendue.