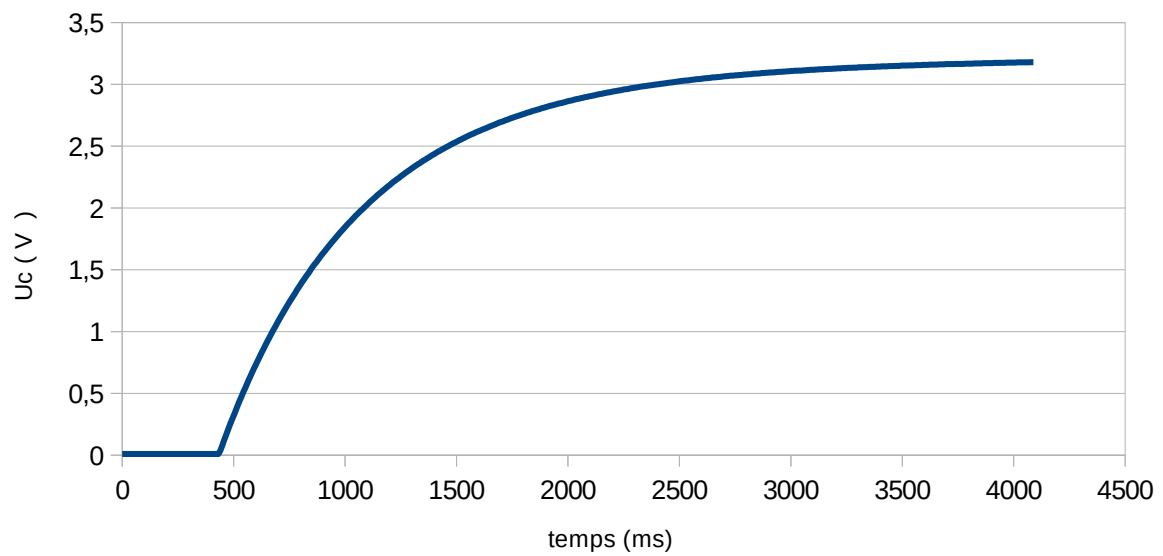
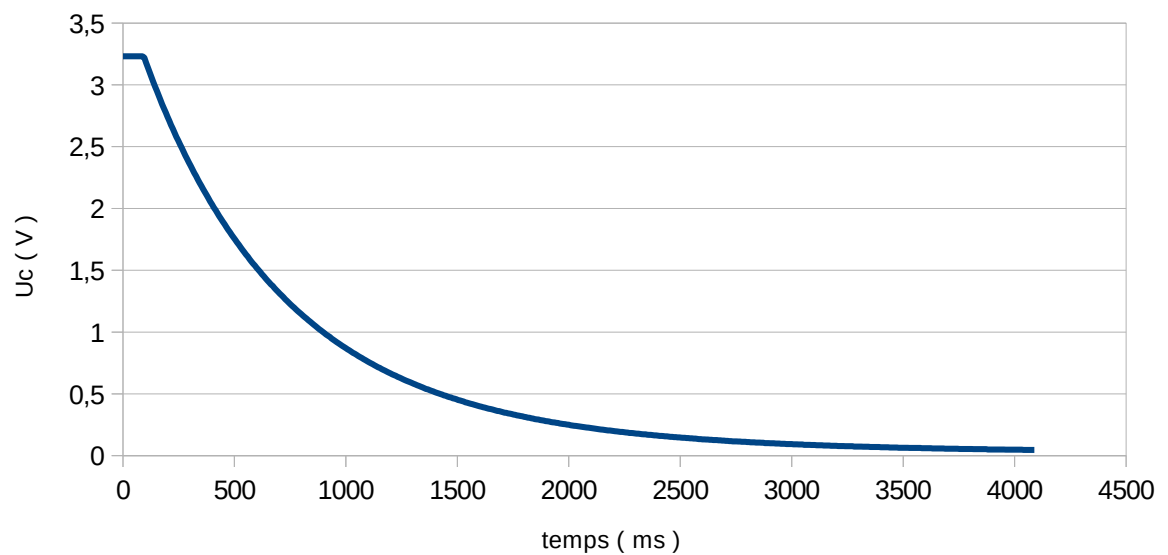


Charge d'un condensateur 4,7μF à travers une résistance de 100kΩ



Décharge d'un condensateur 4,7μF à travers une résistance de 100kΩ



Niveau code en Python c'est pas ce qu'il y a de plus difficile :

```
ESP32_ADS1015_V2.py ×
1 from Capteur_ADS1015 import *
2 from time import ticks_ms, sleep_ms
3
4 nb_data = 24
5
6 t0 = ticks_ms() # Relever le compteur de millisecondes
7
8 data = capteur_ads ( nb_data, rate=4, canal0=0, canal1=None )
9
10 t1 = ticks_ms() # Relever le compteur de millisecondes
11
12 delta_t = t1-t0 # Calculer le temps total des mesures
13
14 print("Durée de la mesure : {:d} ms".format(delta_t))
15 print("Période d'échantillonnage : {:.2f} ms".format(delta_t/nb_data))
16
17 for i in range ( nb_data ):
18     print("{:d},{:d}".format(i,data[i])) # Afficher les mesures
```

Code pour effectuer :

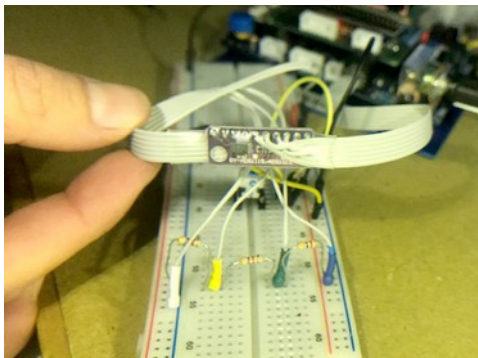
24 mesures (c'était pour un test. Pour le condensateur j'ai utilisé 512 mesures)

rate = 4 → 128 échantillons par seconde [0 ; 7] → de [8 à 860] échantillons par seconde

Canal0 → entrée analogique N°0 (4 entrées 0 → 3)

Canal1 = None → Mesure de potentiel électrique par rapport à 0V

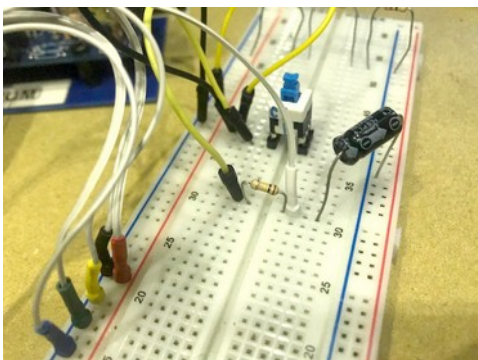
(Il y a la possibilité d'associer les entrées deux à deux pour faire de la mesure différentielle)



Convertisseur Analogique Numérique intégré dans la nappe de mesure.

Montage avec pont diviseur de tension constitué de 3 résistances égales 4,7 kΩ. Ici, les quatre entrées numériques permettent de mesurer U0, U1, U2 et U3 tel que $U_0+U_1+U_2+U_3 = U_{max}$

```
Console ×
>>> ads.read(4, 0)
1
>>> ads.read(4, 1)
552
>>> ads.read(4, 2)
1099
>>> ads.read(4, 3)
1649
>>>
```



Montage pour mesurer la charge / décharge du condensateur sur l'entrée A0 (canal0).