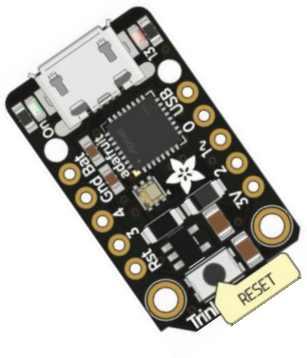
LYCÉE RÉGIONAL d'Enseignement Général et Technologique Arthur VAROQUAUX	python ISN. Microcontrôleurs et cartes programmables	Term S

Premier test : pilotage d'une DEL

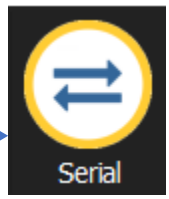


Réaliser un programme permettant de faire clignoter une DEL rouge toutes les 350 millisecondes.
Modifier votre programme pour qu'à chaque tour le délai diminue de **10 ms** et s'il devient inférieur ou égal à 20 ms le réinitialiser à **350 ms** ***(Aide** : utiliser une variable)*

Attention, dans la fonction **sleep** du module **time** les valeurs doivent être rentrées en seconde. Une pause de 200 ms s'obtient à l'aide de l'instruction **time.sleep(0.2)**

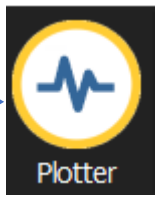
Deuxième test : DELs et bouton poussoir

Réaliser un programme pour qu'une Del rouge clignote toutes les demies secondes et qu'une Del verte clignote **10 fois** *(avec un intervalle de 100 ms)* si l'on appuie sur un bouton poussoir.
Compléter votre programme afin que le message **« poussoir appuyé »** s'affiche lors d'un appui sur le bouton. **Vérifier** en utilisant la **console série**



Troisième test : DELs et capteur analogique

Réaliser un programme permettant de lire la tension aux bornes d'un potentiomètre toutes les 1/2 secondes. ***Afficher*** la valeur dans le moniteur série. (Vous pouvez visualiser la valeur dans la console série ou le **traceur série** si vous le souhaitez)
Compléter votre programme :



- Si la tension dépasse **1,5 V** allumer une Del rouge et afficher dans le moniteur série le message ***« DEL rouge allumee »***, sinon allumer une Del verte et afficher le message ***« DEL verte allumee ».***

Quatrième test : pilotage de servomoteurs

Réaliser un programme simple permettant de faire varier alternativement toute les secondes la position d'un servo moteur de **30° à 90°**.

Réaliser un programme simple permettant faire varier la vitesse d'un servo moteur à rotation continue. On veut pouvoir répéter indéfiniment les instructions suivantes :

- Vitesse maximale dans 1 sens pendant 2 secondes
- Arrêt pendant 1 seconde
- Vitesse maximale dans sens inverse pendant 2 secondes
- Arrêt pendant 1 seconde

Pour les **servo moteurs continus** utilisés (**HSR 1425CR**): paramétrer les valeurs **min_pulse=1000 µs** et **max_pulse=2100 µs**)

Pour les **servo moteurs**
(**Tiny-S**): paramétrer les valeurs **min_pulse=900 µs** et **max_pulse=2300 µs et actuation_range=140**)
(**Nano-S**): paramétrer les valeurs **min_pulse=700 µs** et **max_pulse=2300 µs et actuation_range=140**)

Pour aller plus loin :

Facile :

Réaliser un programme permettant de faire varier la vitesse du moteur continu progressivement de 50% de la vitesse max à -50% puis de -50 à 50%. Vous choisirez un délai adapté.

Réaliser un programme permettant de faire varier progressivement l'angle d'un servomoteur afin qu'il réalise des allers-retours entre 2 positions que vous aurez déterminées. Vous choisirez un délai adapté.

Plus difficile :

On veut contrôler la position d'un servo moteur à l'aide d'un capteur optique (ici une photorésistance)

Réaliser un programme permettant de lire la tension aux bornes du capteur optique et **d'afficher** la valeur dans le moniteur série.

- **Déplacer** votre main au-dessus du capteur et noter les valeurs de tension pour 2 positions différentes de votre main (position **basse** et position **haute**).

On souhaite que pour une position donnée de la main le servo moteur tourne à 50% de sa vitesse maximale dans un sens, puis bascule progressivement jusqu'à 50% de la vitesse max dans l'autre sens lorsque votre main arrive à la seconde position.

Proposer une solution !

Aide :

Si pour la photorésistance à la lumière on lit une tension de 1,00V et 1,50 V à l'ombre

On veut :

- **Pour la tension de 1,00 V une vitesse de +50%**
- **Pour la tension de 1,50V une vitesse de -50%**

Attention, prévoir une butée, si la vitesse dépasse 50%, la bloquer à 50% (idem dans l'autre sens)

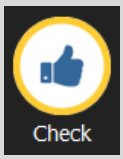
Adapter votre programme afin cette fois ci de piloter un servomoteur à angle avec votre main (On souhaite que pour une position donnée de la main le servo moteur soit à 30 ° puis bascule progressivement jusqu'à 150 ° lorsque votre main arrive à la seconde position)

Aide

Modules : ***Quelques exemples de modules et de fonctions, ou classes associées***

- **digitalio** contient **(DigitalInOut , Direction , Pull)**
- **analogio** contient **(AnalogIn , AnalogOut)**
- **pulseio** contient **(PWOut)**
- **adafruit_motor** contient **le sous module servo** qui contient **(Servo , ContinuousServo)**

Attention MUeditor utilise la syntaxe de python3



- Toujours mettre un espace avant et après les opérateurs et après une virgule. Le nombre de caractères par ligne est limité…etc. Dans la barre de menu cliquer sur ***Check*** pour vérifier votre programme.
- Pour l'affichage dans le moniteur série, lors de l'utilisation de la fonction **print**. Mettre des parenthèses.

  Exemple **print("coucou ")**
- Pour afficher dans le traceur série, il faut envoyer les données sous forme de tuple

  Exemple **print((a , b , c))**

  Pour tracer une valeur lue en continue, par exemple une tension :

  Exemple **print((U ,))**
- Pour réaliser un test sur une **variable booléenne** il faudra utiliser :

  If capteur.value **is True** *à la place de* **if** capteur.value **= = True**

Quelques tutoriels pour l'utilisation de **MUeditor :** https://codewith.mu/en/tutorials/

Connexions des composants sur les différentes cartes

	MetroM4 express	Trinket	Metro M0express
	pattes	**pattes**	**pattes**
Bouton poussoir	**D0**	**D0**	**D0**
Potentiomètre	**A0**	**A0 (D1)**	**A0**
photorésistance	**A1**	**A0***	**A1**
Delrouge	**D2**	**D2**	**D2**
Delverte	**D3**	**D3**	**D3**
Servomoteurs	**D4**	**D4**	**D4**

*les cartes **Trinket** n'ont qu'une entrée analogique, il faudra modifier la position d'un fil de connexion pour envoyer sur l'entrée **A0** le signal du potentiomètre ou celui de la photorésistance. **Appeler le professeur !**