

1-1 PROGRAMMER UN ALGORITHME LIE A UNE NOUVELLE FONCTIONNALITE.

Le **pluviomètre** est un instrument utilisé pour mesurer les précipitations. Son fonctionnement consiste à canaliser les précipitations et mesurer le volume d'eau. L'unité de mesure utilisée pour mesurer les précipitations est le mm (millimètre) ou L/m².

1-1-1 Observer la vidéo du pluviomètre

Le programme du "pluviomètre".



Le "pluviomètre" stocke les données dans un fichier CSV.

Dans le tableau ci-dessous, le résultat des données produites.

Temps (min)	Valeur du capteur ILS
1	0
2	1
3	1
4	0
5	0
6	1
7	1
8	1
9	1
10	1

1-1-2 En observant ces données, pouvons-nous dire quelle quantité d'eau est tombée ?

Si besoin, indiquer-le(s) information(s) qu'il manque ?

Mon avis : _____

Avis du groupe : _____

1-1-3 Problématique : Comment programmer le pluviomètre pour mesurer précisément la quantité d'eau tombée ?

- Ce que je ferais

Trois élèves ont réfléchi à ce problème et proposent un algorithme.

- Écouter le fichier "[algorithme du pluviomètre.mp3](#)" et répondre aux deux questions.
- Combien de variables ces élèves veulent ils utiliser ? **2 variables**

- Quels noms proposent-ils pour ces variables ? **état d'avant** et **eau**

1-1-4 Compléter l'organigramme de droite à partir du fichier l'audio.

1-1-5 Comparer cet organigramme avec celui des autres élèves du groupe et corriger le si nécessaire.

1-1-6 Répondre à ces deux questions :

Quelle partie de cet algorithme ne sera fait qu'une seule fois, au démarrage du pluviomètre ?

→ Sur l'algorithme de droite, entourer au crayon de couleur rouge, la partie concernée.

au démarrage

Quelle partie de cet algorithme sera lu plusieurs fois ?

toujours

→ Sur l'algorithme de droite entourer au crayon de couleur marron, la partie concernée.

1-1-7 Compléter les cases vides du programme ci-dessous

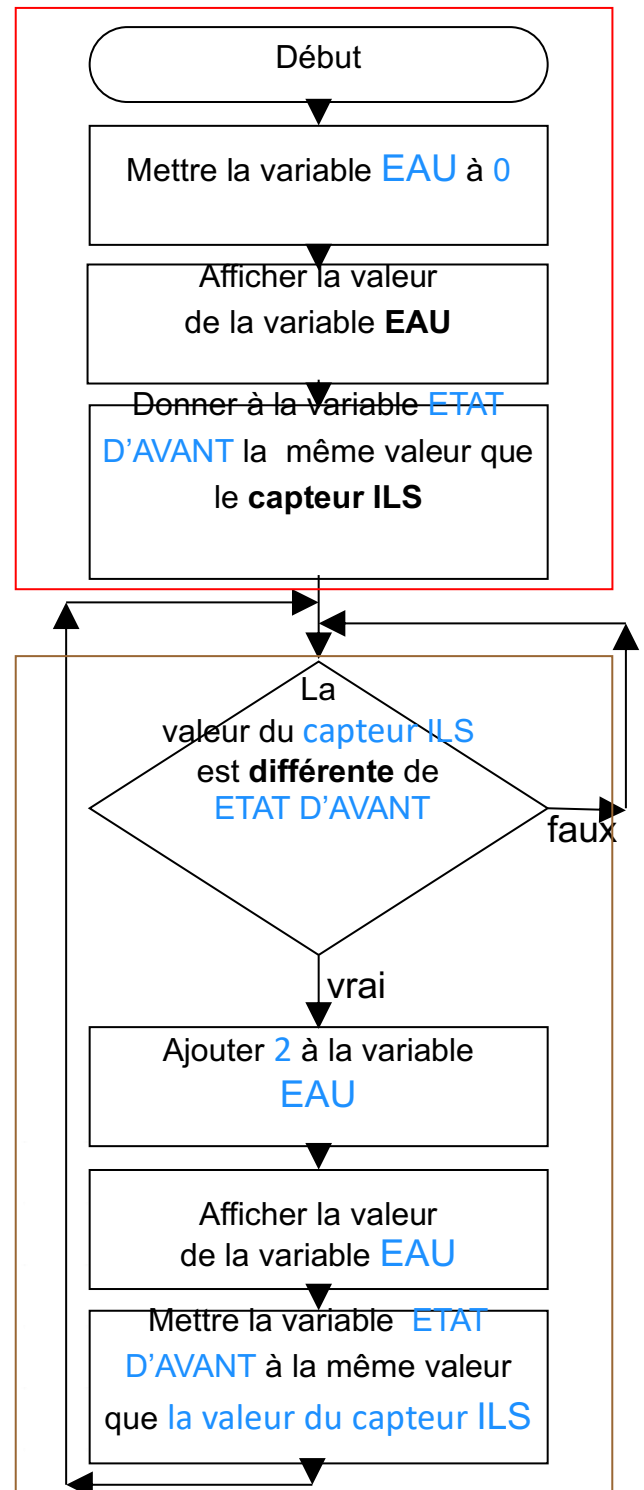


```

au démarrage
définir 4digit à 4-Digit Display at P8 and P14
définir EAU à 0
4digit show number EAU
définir ETAT AVANT à lire la broche numérique P15
  
```

```

toujours
si ETAT AVANT ≠ lire la broche numérique P15 alors
  modifier EAU de 2
  4digit show number EAU
  définir ETAT AVANT à lire la broche numérique P15
  
```



1-1-8 Brancher la maquette sur le PC

1-1-9 Lancer le logiciel CHROME (obligatoire, ne pas prendre un autre navigateur). Aller sur la page <https://makecode.microbit.org/>

1-1-10 Créer le programme et tester-le.

1-1-11 Dans l'application micro : bit, traduire ton programme en langage JavaScript. Observer ce que cela donne (cliquer en haut sur « javascript »)

→ Pour ceux qui veulent aller plus loin

Compléter le programme pour qu'il corresponde à l'algorithme suivant :

Si j'appuie sur le bouton A de la carte, l'afficheur à 4 chiffres affiche la vitesse du vent

Si j'appuie sur B, l'afficheur à 4 chiffres affiche la luminosité.

Si j'appuie en même temps que A et B, je visualise la quantité d'eau tombée

Aide : penser à retirer le bloc dans la boucle "pour toujours"...



1-1-12 Bilan :

D'une manière générale, quelles sont toutes les étapes par lesquelles on doit passer avant de commencer à programmer ?

Avant de commencer la programmation, la conception de l'algorithme constitue une étape préalable indispensable. Cette structuration simplifie ensuite l'écriture du code.

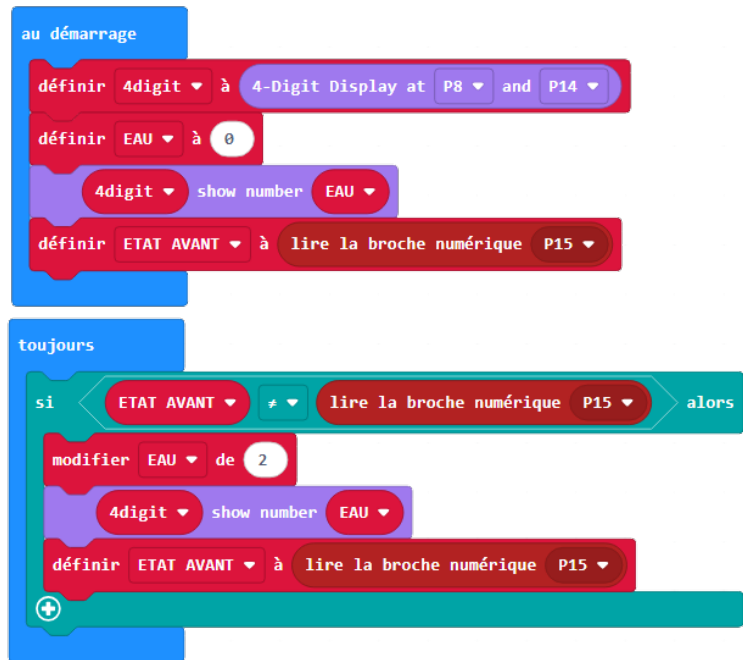
La réflexion s'articule autour de trois questions principales :

- Quelles actions ne sont réalisées qu'une seule fois au démarrage ? Cette étape correspond à l'initialisation.
- Quelles actions doivent être exécutées en continu ? Il s'agit du fonctionnement en boucle.
- Le programme nécessite-t-il le stockage temporaire de données ? Si des valeurs doivent être conservées, la création de variables est requise.

Une fois ces éléments identifiés, l'élaboration d'un organigramme précède le choix du langage de programmation (par bloc, Python, etc.) et le codage effectif.

1-2 COMPARER UN PROGRAMME GRAPHIQUE PAR BLOC ET LA PROGRAMMATION TEXTUELLE.

Voici un programme du pluviomètre réalisé en programmation par blocs.



Mais le programme aurait pu être réalisé dans un autre langage, comme par exemple le langage Python (qui sera vu peut-être au lycée). Voici un exemple :

```
display4 = grove.create_display(pin8, pin14)    # on définit l'afficheur Grove 4 digits

EAU = 0
display4.show(EAU)    # on affiche la variable EAU

ETAT_AVANT = pin15.read_digital()

while True:
    if ETAT_AVANT != pin15.read_digital():
        EAU += 2
        display4.show(EAU)
        ETAT_AVANT = pin15.read_digital()

    sleep(10)    # petite pause de 10 millisecondes pour éviter de surcharger la carte microbit
```

Observer les ressemblances entre ces deux "mêmes" programmes réalisés dans deux langages différents puis répondre aux questions :

Dans la programmation en Python :

- Quel caractère est utilisé au début d'un "commentaire" ? (PS : ce commentaire est destiné à un autre programmeur qui voudrait relire ce programme) **Il y a #**
- Comment est écrit « Broche 15 » ? **pin15**
- Comment est écrit le signe "différent" ? **!=**
- Comment s'écrit "si" ? **if**
- Quelle ligne de commande a été ajoutée dans le programme Python et qui n'existe pas dans le programme par bloc ?

Il y a : sleep(100) # petite pause pour éviter de surcharger le micro: bit