

CONTRÔLE D'ACCÈS EN ZONE INONDÉE.

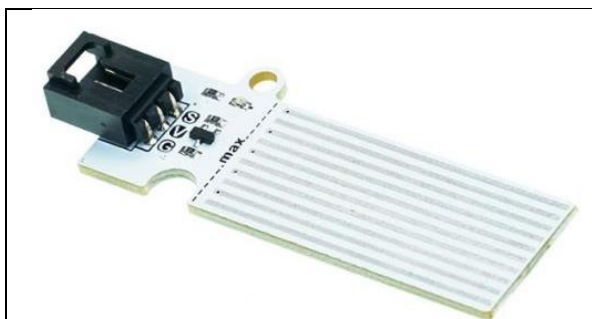
Contexte.

Entre l'île de Noirmoutier et le continent se trouve une route qui est recouverte par la mer à marée haute et qui ne peut donc être utilisée qu'à certains moments. Cette route s'appelle le passage du Gois.



Ta mission est de concevoir et de réaliser le contrôle de l'accès à la route. Tu disposes d'un capteur de niveau d'eau qui va mesurer la hauteur de l'eau présente sur la chaussée. Si ce niveau dépasse un certain seuil, tu devras fermer une barrière commandée par un servomoteur.

Capteurs et actionneurs utilisés.



Un détecteur de niveau d'eau.

Fonctionne en mesurant la résistance de l'eau présente sur les lignes métalliques. Cette résistance est transformée en une tension qui peut être mesurée par le Micro:bit.

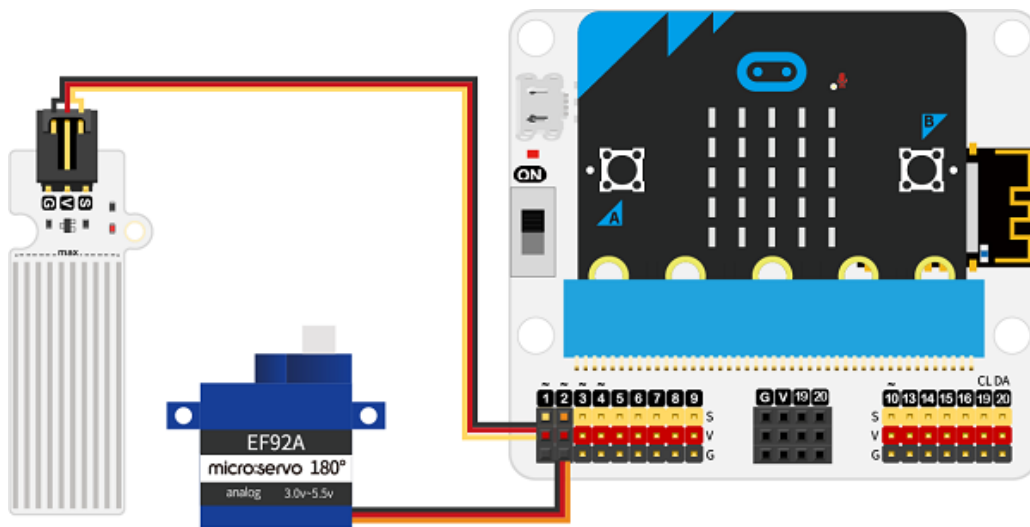


Un servo 180°.

Moteur dont l'angle de rotation est contrôlé par une impulsion électrique dont la largeur détermine l'angle de rotation.

Ces éléments se connectent sur une carte d'extension iot:bit (ou Octopus:bit) sur laquelle un Micro:bit v2 doit être placé. Un câble de connexion avec connecteurs à 3 fiches doit être utilisé pour le détecteur, le servo possède déjà un câble (voir ci-dessous). Le Micro:bit est raccordé à ton PC par un câble USB.

Câblage.



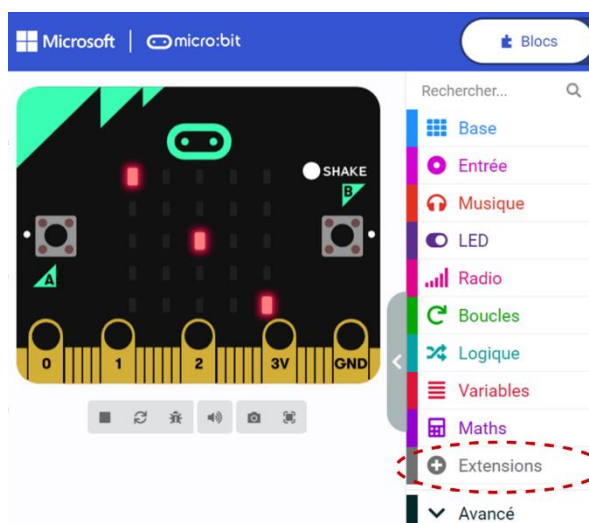
Ce schéma représente le câblage **avec la carte d'extension iot:bit**. Il faut bien respecter les couleurs des connexions. Le détecteur va sur la broche P1, le servo sur la broche P2.

Programme.

Créer un nouveau projet et lui donner un nom, par exemple [Barrière pour inondations](#)

Le programme va utiliser des familles de blocs qui ne sont pas présentes par défaut, si cela n'a pas déjà été fait, il faut les installer.

Chargement des extensions nécessaire au projet.



Cliquer sur Extensions dans la liste des blocs.

Puis faire une recherche avec le terme [iot-environment-kit](#) et choisir :



Ensuite, cliquer à nouveau sur Extensions, faire une recherche avec le terme [servo](#) et choisir :



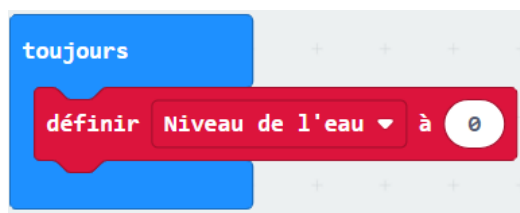
Après ces opérations la liste des familles de blocs devrait comporter les nouveaux membres suivants :



Tu peux maintenant commencer le programme.

Il se composera d'un bloc **toujours** dans lequel nous allons insérer les blocs du programme qui se répéteront indéfiniment.

Tu vas créer une variable que tu appelleras [Niveau de l'eau](#) et insérer le bloc **définir** pour contrôler cette variable :

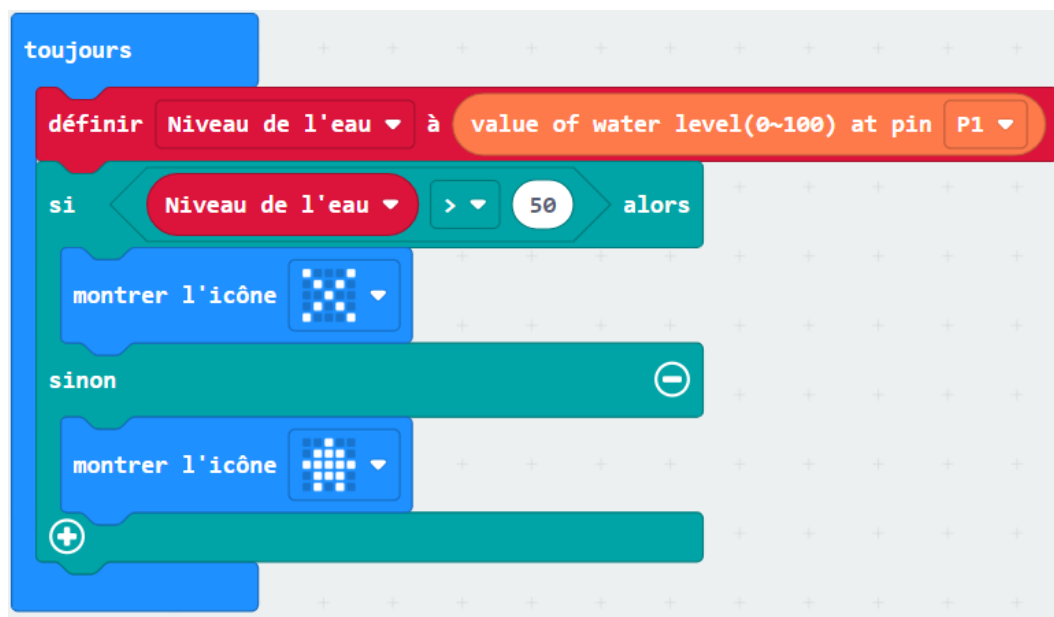


Cette variable devra contenir la mesure venant du détecteur de niveau d'eau (value of water level) pour cela tu dois insérer un élément de la famille Octopus qui permet d'obtenir cette valeur sous la forme d'un nombre compris entre 0 et 100.



L'indication P1 indique le nom de la broche sur laquelle tu as connecté le détecteur.

Il faut maintenant ajouter un test sur la variable **Niveau de l'eau** permettant dans un premier temps d'afficher par des icônes si le niveau dépasse un seuil donné.

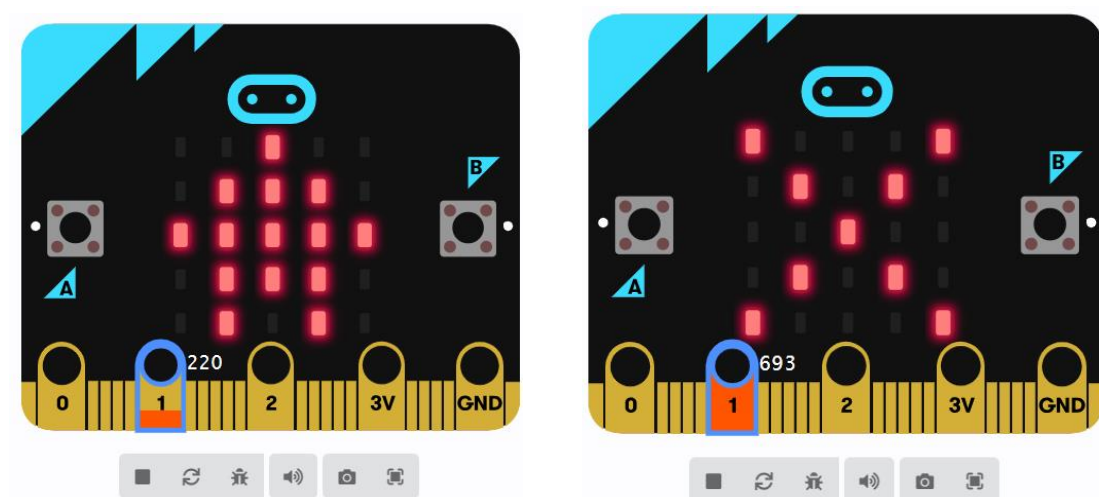


Le test est un bloc **si...sinon** dans la famille Logique dans ce bloc tu remplaces la condition **vrai** par un bloc de comparaison (dans la famille logique), puis tu choisis l'opérateur **>**, tu mets comme membre de gauche de la comparaison la variable **Niveau de l'eau** puis la constante **50** dans l'autre membre.

Tu ajoutes enfin un affichage d'icône (dans la famille de base) représentant une **croix** dans la première partie du **si** et un affichage d'icône (dans la famille de base) représentant un **personnage** dans la partie **sinon**.

Cette instruction signifie que si le niveau d'eau est supérieur à 50% (100% correspondant au capteur complètement immergé) alors la première partie du bloc **si** est exécutée (et l'icône croix affichée) et que si le niveau est inférieur à 50% c'est la seconde partie du bloc (**sinon**) qui est exécutée (et l'icône personnage affichée).

Test en mode simulation.



Pour simuler le détecteur, on fait bouger la souris en cliquant sur son bouton gauche. Les nombres qui s'affichent à droite de la broche 1 correspondent à la valeur que reçoit le programme. Nous pouvons faire varier les valeurs entre 0 et 1023.

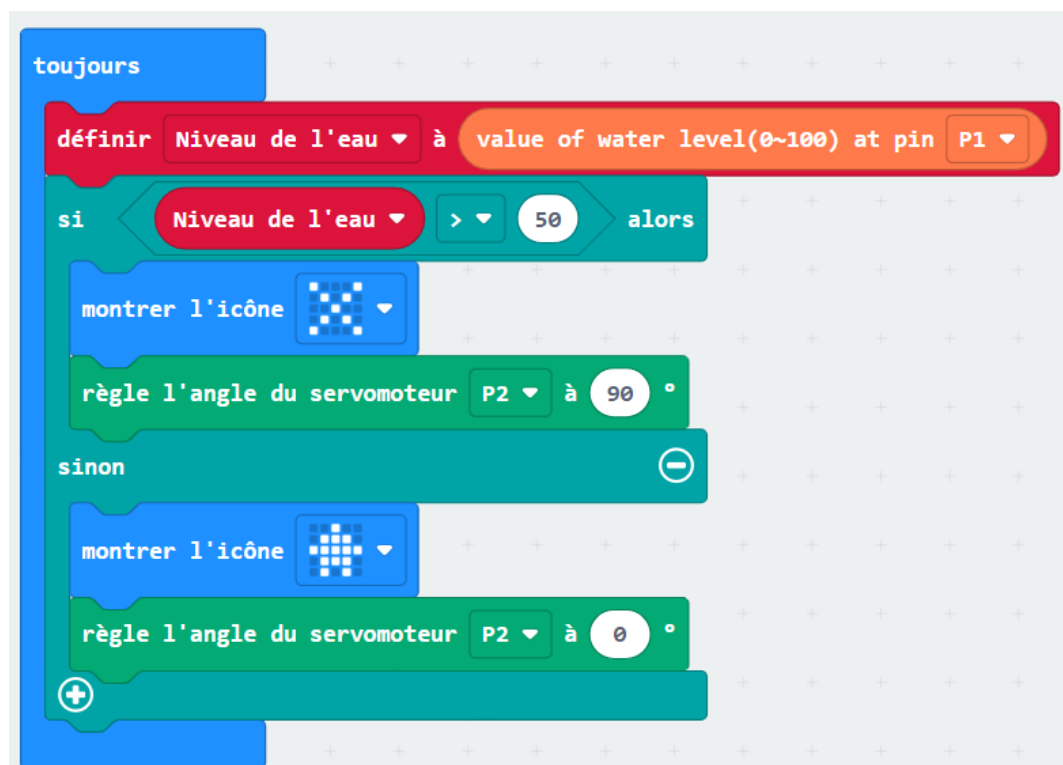
Dans la réalité le capteur fournit des valeurs entre 0 et 700.



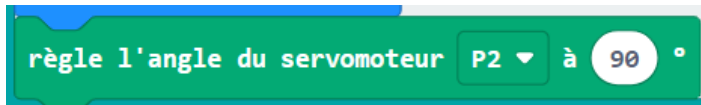
50% correspondant donc à 350.

Les valeurs au-dessus de 700 ne sont pas produites par le capteur.

Suite du programme : commande de la barrière.

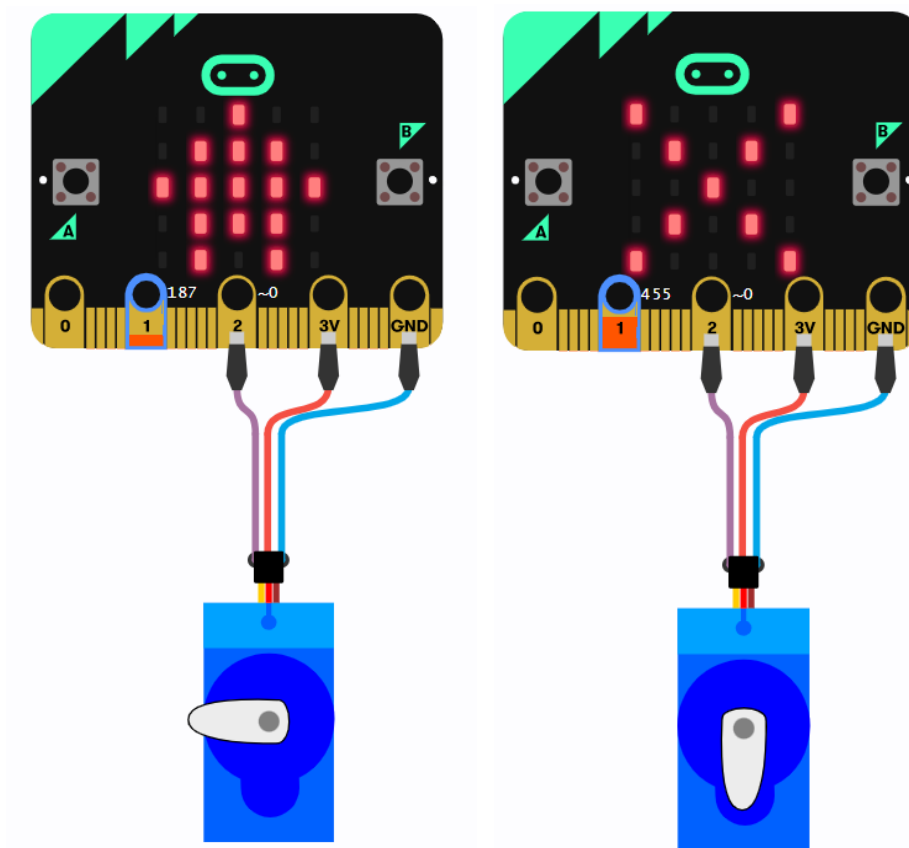


Un bloc de commande de l'angle de rotation du servomoteur est ajouté dans chaque branche du bloc **si**.



Ce bloc contient deux paramètres : la broche sur laquelle est connecté le servo (ici P2) et l'angle de rotation (qui peut varier entre -90° et $+90^\circ$)

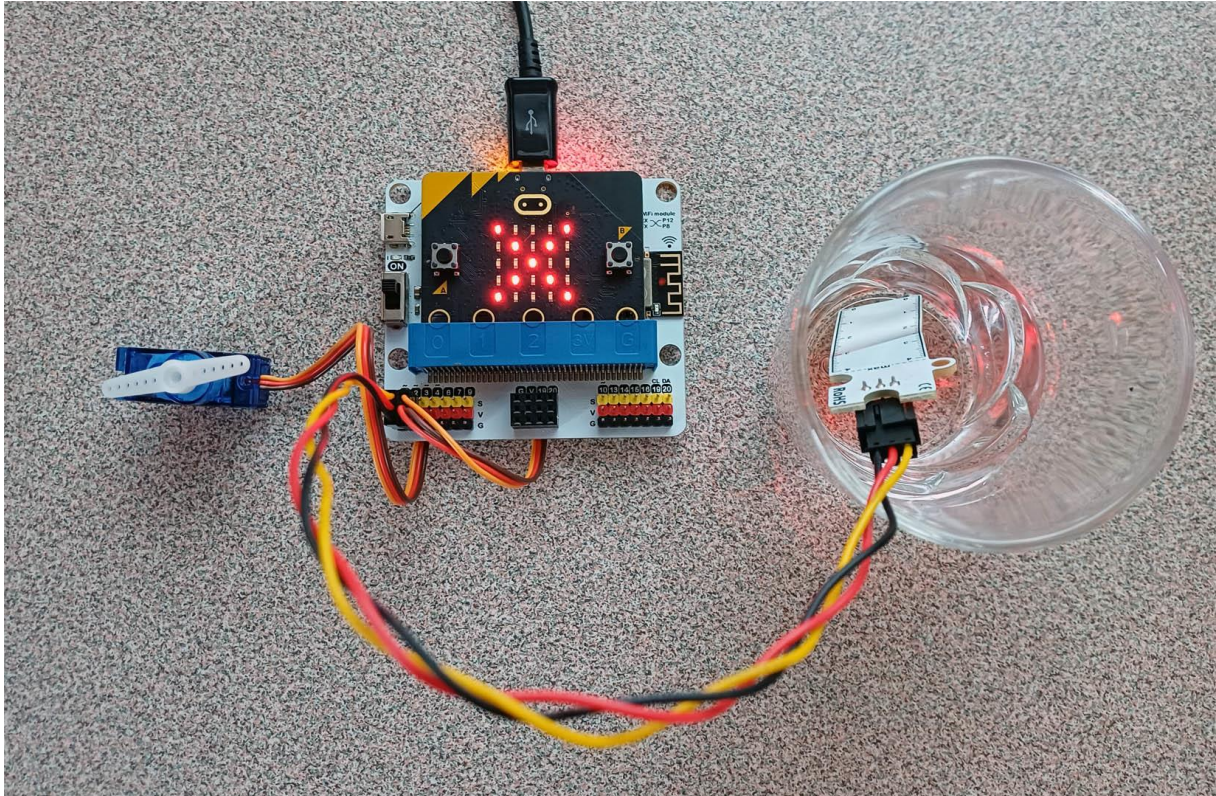
Test en mode simulation.



Tu procèderas comme décrit plus haut en faisant varier avec la souris la valeur sur la broche P1 entre 0 et 700 ($350 = 50\%$). Ici la simulation montre le résultat avec les valeurs 187 et 455.

Test en situation réelle.

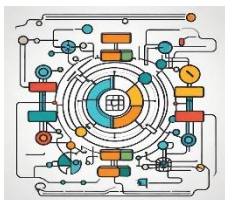
La valeur du niveau d'eau provoquant le déclenchement de la fermeture de la barrière (50 dans l'exemple) dépend de la conductivité de l'eau, si cela ne fonctionne pas avec 50 il faut essayer avec une valeur plus petite.



L'angle pour la fermeture du servo doit aussi être affiné par essais et erreurs. Une fermeture correspondra peut-être à 87° ou 92°, cela dépend du servo.

Améliorations.

- Ajouter un signal rouge (barrière fermée)/vert (barrière ouverte) en connectant des LEDs sur les broches P13 et P14 [difficulté : *].
- Faire descendre la barrière lentement [difficulté : **].
- Ajouter une sonnerie quand la barrière descend (en utilisant un bloc de la famille musique) [difficulté : **].



Pour les curieuses et les curieux.

Comment fonctionne le capteur de niveau d'eau ?



Pour mesurer le niveau de l'eau, le capteur utilise une propriété électrique des corps : leur résistance. Lorsque le capteur est sec, sa résistance est très grande, plus il est enfoncé dans l'eau, plus sa résistance diminue.

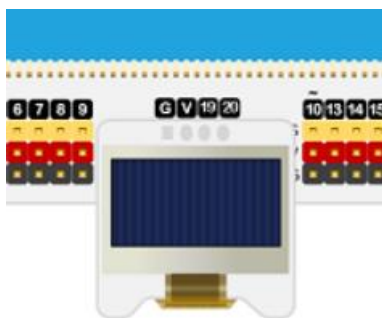
La valeur de la résistance de l'eau dépend de sa pureté. De l'eau de pluie aura une résistance élevée, de l'eau de mer, à cause du sel, aura une résistance faible. Donc le capteur va te donner une réponse qui ne dépendra pas que de la hauteur de l'eau, mais aussi des impuretés qu'elle contient.

Dans ton programme le niveau d'eau peut être mesuré par l'instruction :

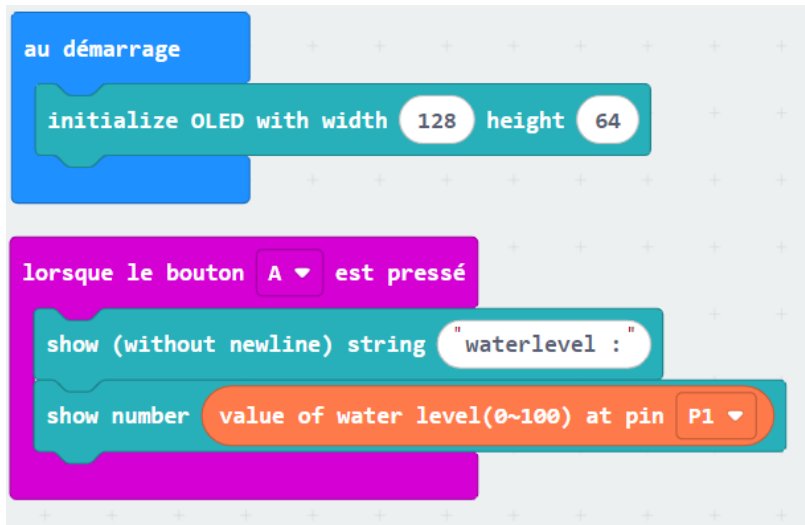
```
value of water level(0~100) at pin P1 ▼
```

Elle te donne une valeur comprise entre 0 et 100. Vu ce qui vient d'être expliqué, il faudra faire des essais si tu souhaites connaître la hauteur de l'eau en fonction de la valeur retournée. Par exemple, lors de mes premiers essais la valeur retournée était 50 pour le capteur complètement immergé.

Pour faire des essais, tu vas connecter l'afficheur OLED comme indiqué ci-dessous :



Cet afficheur, qui est du même type que les écrans des smartphones va être utilisé pour afficher la valeur retournée par le capteur avec le petit programme suivant :



Lorsque tu appuieras sur le bouton A, le programme affichera la valeur retournée par l'instruction `value of water level`.

Exerce-toi à lire les deux valeurs pour différente valeur de la hauteur immergée (l'arrière du capteur est gradué) avec de l'eau du robinet, puis avec cette même eau plus du sel. Note ces valeurs et compare-les.

Lexique anglais-français.

Smarties : ville intelligente.

Value of water level at pin P1 : valeur du niveau d'eau sur la broche P1.

https://www.electfreaks.com/learn-en/microbitKit/smart_city_kit/microbit-Smart-City-Kit-case-01.html