

Serre Automatisée



Avant de commencer le TP, il faut vérifier que le système est branché correctement.

Ecrire Programme Test1

```
toujours
si bouton A est pressé alors
  afficher texte "A"
sinon
  si bouton B est pressé alors
    afficher texte "B"
  afficher texte "0"
```

Que fait ce programme ?

Ecrire Programme Test2

```
toujours
si bouton A est pressé alors
  afficher texte "A"
  écrire sur la broche P0 la valeur 1
sinon
  si bouton B est pressé alors
    afficher texte "B"
    écrire sur la broche P0 la valeur 0
  afficher texte "0"
```

Que fait ce programme ?

Que s'est-il passé sur ce composant (relais) ?

Sur quel port est branché le composant (Relais) ?



Serre automatisée

Ce TP utilise deux maquettes

8

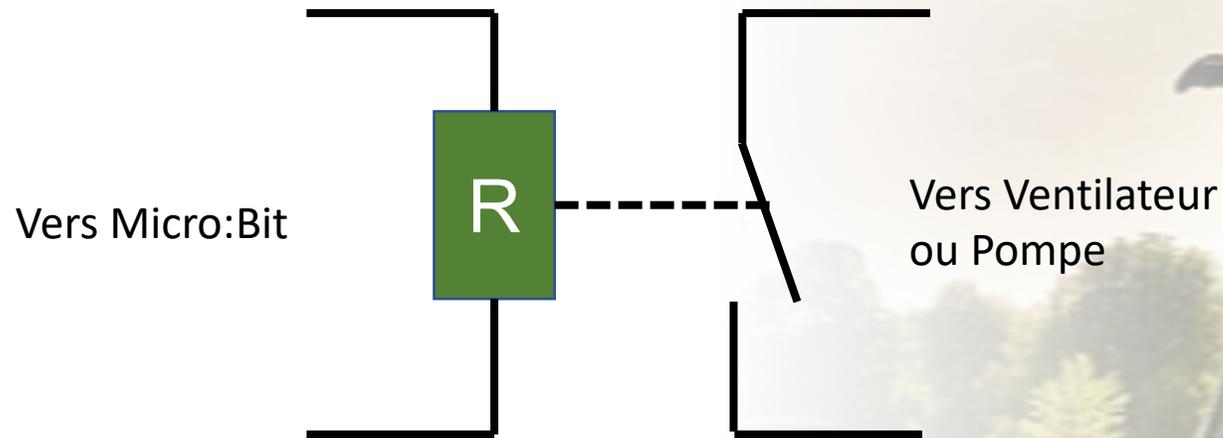
Vous ferez une analyse fonctionnelle de la maquette de la serre en répondant à différentes questions telles que : A qui rend-elle service ? Sur quoi agit-elle ? Etc

...

6

Vous montrerez par un schéma électrique simple comment est branchée la pompe sur le microbit.

Symbole du relais à utiliser



Serre automatisée

Ce TP utilise deux maquettes

Maquette Ventilation

6

Vous devez réguler la température dans un espace confiné grâce à un programme adéquat. Pour extraire l'air trop chaud, un ventilateur se met en marche. **(Voir Annexe 1)**

2

Le système permettra d'afficher la température dans deux unités différentes de votre choix en fonction de l'appui sur le bouton A ou B

Maquette Arrosage

6

Il est demandé d'installer un contrôle de l'arrosage par électrovanne dans une serre automatisée. L'arrosage sera contrôlé par un taux d'humidité adéquat. **Voir Annexe 2**

6

Les données de température et d'humidité du sol seront transmises à un ordinateur distant par Bluetooth.

Voir Annexe 3

2

Vous calculerez le débit et volume d'eau consommée en fonction de paramètres que vous aurez récupérés sur la maquette (diamètre du tuyau, ...).

10

Vous proposerez une séquence en 5 séances pour réaliser l'analyse de chaque composant entrant dans le fonctionnement de la maquette de la serre.

Vous montrerez au cours de cette séquence, aux élèves de cycle 3, que les constituants principaux de la serre se retrouvent dans leur principe dans d'autres systèmes. (ex: moteur électrique, pompes, capteurs de température, etc., ...)

Annexe 1

Structure du programme permettant la régulation de la température.



Annexe 2

Version avec
Capteur Spécial

```
Rechercher...  
Base  
Entrée  
Musique  
LED  
Radio  
Boucles  
Logique  
Variables  
Maths  
GatorSoil  
Grove  
Extensions
```

```
toujours  
définir Hum à get moisture on pin P0 in soilMoisture using power pin P0  
tracer le graphe de Hum  
à 0.7  
si Hum ≤ 0.5 alors  
écrire sur la broche P1 la valeur 1  
sinon  
écrire sur la broche P1 la valeur 0
```

Version avec
deux clous

```
au démarrage  
montrer nombre lire la broche analogique P0  
définir MesureHum à lire la broche analogique P0
```

```
toujours  
définir MesureHum à lire la broche analogique P0  
tracer le graphe de MesureHum  
à 1023  
si MesureHum ≤ 900 alors  
écrire sur la broche P2 la valeur 1  
sinon  
écrire sur la broche P2 la valeur 0
```

Comment brancher les deux clous avant de les enfoncez dans le sol ?

Annexe 3

Programme d'émission des données

Programme à charger dans un micro:bit qui enverra les données.
Astuce : définir le même groupe radio pour que les deux micro:bit puissent communiquer l'un avec l'autre.

La partie à modifier est encadrée en rouge

The code is written in Scratch and consists of two main blocks: 'au démarrage' (when started) and 'toujours' (forever loop). The 'au démarrage' block contains a 'radio définir groupe' (radio set group) block with the value '99'. The 'toujours' block contains three 'envoyer la valeur' (send value) blocks, each with a variable name ('x', 'y', 'z') and a sensor block for 'accélération (mg)' (acceleration in mg). The entire 'toujours' block is highlighted with a red border, indicating it is incorrect.

Cette partie du programme n'est pas correcte

Programme de réception des données

Programme à charger dans un micro:bit qui recevra les données.
Astuce : définir le même groupe radio pour que les deux micro:bit puissent communiquer l'un avec l'autre.

La partie à modifier est encadrée en rouge

The code is written in Scratch and consists of two main blocks: 'au démarrage' (when started) and 'toujours' (forever loop). The 'au démarrage' block contains a 'radio définir groupe' (radio set group) block with the value '99'. The 'toujours' block contains a 'quand une donnée est reçue par radio' (when data received by radio) block with 'name' and 'value' inputs. This block is followed by three conditional blocks: 'si name = "x" alors' (if name is 'x' then), 'sinon si name = "y" alors' (else if name is 'y' then), and 'sinon si name = "z" alors' (else if name is 'z' then). Each conditional block contains a 'définir x à value' (set x to value) block. The entire 'quand une donnée est reçue par radio' block and its three conditional blocks are highlighted with a red border, indicating they are incorrect. To the right of the code, there is a green-bordered box containing the text 'Cette partie du programme est correcte' and a 'toujours' block with two 'série écrire valeur' (serial write value) blocks for 'Hum' and 'Temp'.

Cette partie du programme n'est pas correcte