

# Système d'arrosage automatique

## 31 – Programmation et capteurs

### Synthèse du cours – programmation SCRATCH

Travail sur la programmation et les capteurs d'humidité du sol pour le système d'arrosage automatique.

La première partie du cours a été consacrée à des généralités sur la programmation.

Cette introduction a été suivie par une grosse partie sur la résolution de l'exercice proposé en travail personnel sur la fiche 13 (Les processeurs dans la chaîne de traitement de l'information).

Et enfin, selon les classes, une troisième partie a été consacrée à la modification du programme précédent pour obtenir un exemple simple de l'algorithme pour le système d'arrosage automatique, avec une partie de réflexion sur les capteurs d'humidité.

#### 1 – Un peu de théorie sur la programmation

Un programme est un ensemble composé de **données** d'un côté, et de **code** de l'autre.

Les données peuvent être de trois types :

- **nombres**,
- **texte**,
- "**média**" (son et images).

Pour accéder aux données et les manipuler, le code utilise des **variables**, qui servent à **donner un nom** à une donnée.

Le code est composé d'une suite d'instructions qui peuvent être classées en 5 catégories :

- les **entrées** :  
(lecture dans un fichier, attente d'un événement ou d'un délai, acquisitions depuis un capteur, ...),
- les **sorties**  
(affichage de texte, images fixes ou animées, sons, commande d'un actionneur, envoi d'un signal, ...),
- les **opérations**  
(calculs mathématiques plus ou moins complexes),
- les **tests**  
(exécution de code après vérification d'une condition),
- les **boucles**  
(répétition d'une portion de code jusqu'à ce qu'une condition soit remplie, ou indéfiniment).

#### 2 – Une boucle simple

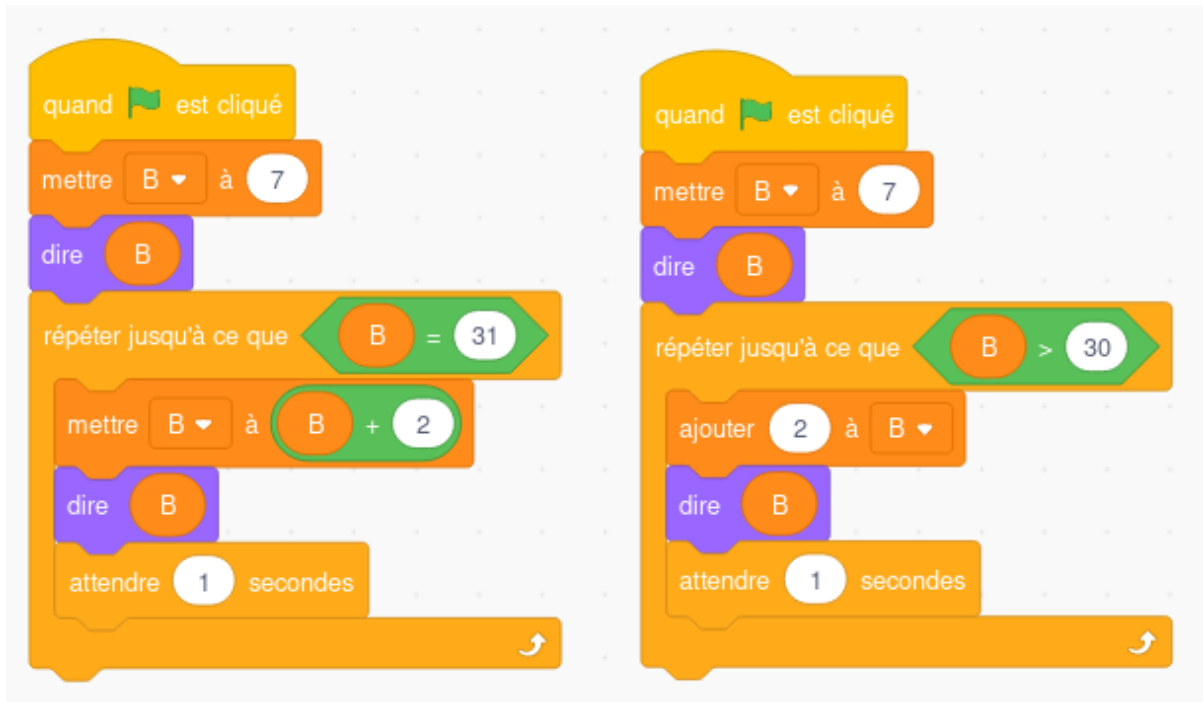
La deuxième partie de la séance a été consacrée à la résolution de l'exercice proposé à la fin de la fiche 13.

N'ayant pas la possibilité d'utiliser les ordinateurs, cette résolution a été faite en projetant au tableau à partir des instructions données par les élèves.

Globalement les élèves ont assez rapidement identifié le besoin d'utiliser une **boucle**, ainsi que d'un **test** pour la **condition de sortie de la boucle**.

Par contre, l'utilisation d'une **variable** pour **stocker la valeur courante** a été plus compliquée à apporter, et l'**incrément** de cette variable a aussi posé problème.

Le résultat exact n'a pas été le même pour chaque classe, mais les deux solutions suivantes regroupent les diverses options utilisées selon les classes :



Je pense que le principal problème pour les élèves est le manque de pratique (ou d'intérêt) et le manque de connaissance du langage SCRATCH.

### 3 – Le programme d'arrosage automatique

Pendant la troisième partie de la séance nous avons travaillé sur les capteurs d'humidité de la terre et la modification du programme précédent pour obtenir une version simple du programme d'arrosage automatique.

>> Rappels sur la mesure : nous avons revus quelques-uns des principes de la mesure vus avec la fiche 06 (Capteurs et mesure) : une mesure se fait par comparaison avec une valeur de référence (les graduations sur une règle graduée par exemple) et s'exprime soit avec une unité, soit comme grandeur sans dimension.

>> Rappels sur les capteurs d'humidité (fiche 11) :

Les mesures d'humidité se font en pourcentage d'humidité relative du milieu considéré (la terre dans notre cas), c'est à dire le rapport entre la quantité d'eau actuellement dans le milieu et la quantité maximum d'eau que le milieu peut contenir. On parle de taux d'humidité.

Une terre parfaitement sèche a un taux d'humidité de 0%, tandis que de la terre saturée en eau a un taux d'humidité de 100%.

Il s'agit donc d'une grandeur "sans dimension" pouvant aller de 0 à 100%HR (HR pour Humidité Relative).

Pour mesurer ce taux d'humidité, il faudrait utiliser un capteur capable de mesurer la quantité actuelle d'eau dans le sol.

En pratique, les capteurs utilisés mesurent l'effet de la présence de l'eau, et non la quantité d'eau présente. C'est le cas des capteurs capacitifs, qui fonctionnent sur le même principe que les écrans tactiles des smartphones actuels (voir l'expérience proposée sur la première page de la fiche 11).

La présence d'eau vient perturber le capteur qui nous donne une valeur liée à la quantité d'eau présente, mais pas directement la quantité d'eau présente.

Résultat, la valeur doit être calculée à partir de la mesure, et nécessite un **étalonnage** du capteur.

Cette opération d'étalonnage, dans le cas présent, consiste à mesurer les valeurs retournées par le capteur pour des cas connus.

Les cas connus les plus simples sont les cas extrêmes : 100% et 0% d'humidité de la terre.

- 100% d'humidité de la terre correspond à la terre saturée en eau (des flaques se forment en surface), que l'on peut simuler en plongeant simplement le capteur dans un verre d'eau. Cela nous donne la valeur maximale, correspondant à 100%.

- 0% d'humidité correspond à une terre parfaitement sèche. En pratique, il y a plus simple que de cuire de la terre pour faire évaporer toute l'eau ou d'attendre qu'elle s'évapore complètement : il suffit de laisser le capteur à l'air libre pour obtenir une valeur valide pour le 0%.

Une fois ces deux extrêmes obtenus, nous avons cherché la formule permettant d'obtenir un pourcentage d'humidité à partir d'une valeur lue depuis le capteur :

En notant  $V_{\min}$  et  $V_{\max}$  les valeurs extrêmes obtenues par étalonnage et " $V_{\text{cap}}$ " la valeur lue depuis le capteur on utilise la formule suivante :

$$V_{\%HR} = ((V_{\text{cap}} - V_{\min}) / (V_{\max} - V_{\min})) * 100$$

Cette formule est utilisée dans le programme suivant, qui utilise des blocs "spécifiques" créés pour simuler les parties manquantes dans SCRATCH : la lecture d'une valeur depuis un capteur externe, et la commande d'un actionneur. Il serait nécessaire de les créer réellement avec une extension spécifique développée pour servir d'interface avec le capteur d'humidité et la pompe.

J'ai ajouté en bas à droite la solution utilisée en cours pour "simuler" le capteur et la commande de la pompe, en utilisant la distance entre le petit chat et le pointeur de la souris comme valeur pour le retour du capteur d'humidité (les valeurs  $V_{\min}$  et  $V_{\max}$  (40 et 520) sont les valeurs d'étalonnage que nous avons obtenu en cours en utilisant cette solution), et des affichages "Pompe allumée" et "Pompe éteinte" pour la commande de la pompe.

Les principales différences avec le programme précédant sont l'utilisation d'une boucle infinie (le système d'arrosage ne s'arrête jamais) et l'utilisation d'un test "si ... alors ... sinon" pour déterminer l'action à réaliser en fonction de la valeur d'humidité de la terre.

