

Connaissance : Notions d'algorithme et de programme

Les **objets connectés** sont souvent programmés pour fonctionner **automatiquement**. Chaque fonction numérique de l'objet connecté peut être assimilée à un « problème » à résoudre. La **résolution d'un problème** par un programmeur peut s'effectuer en **trois étapes** :

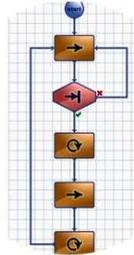
-1- Ecriture d'un algorithme : Suites logique d'opérations ou d'instructions, souvent rédigées sur feuille de papier en utilisant le langage naturel et des mots clés : si, alors, tant que, jusqu'à ...

Exemple : Un robot évitant un obstacle.

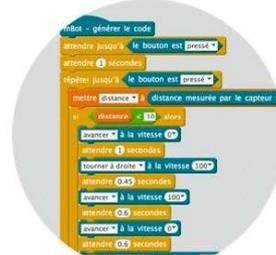


*~ Si le robot détecte un obstacle avec son capteur de pare-choc, alors tourner à gauche de 90°, avancer de 10cm puis tourner à droite de 90°.
~ Sinon avancer indéfiniment.*

-2- Construction à l'aide d'un logiciel d'une représentation graphique de l'algorithme



Algorithme : organigramme de programmation



Logiciel de représentation graphique par bloc (ou briques) comme Scratch

-3- Traduction de la représentation graphique en langage de programmation qui lui-même sera converti en langage machine (code binaire) que le microprocesseur peut exécuter

```

17 // ... / (ANALOG)
18 // ... lire() (0*(analogRead(A7)) ..
19 // ... delay(1000*1);
20 // ... while() (0*(analogRead(A7)>1070-1,
21 // {
22 //     distance = ultrasonic_3.distanceCm();
23 //     motor.move(1,0);
24 //     delay(1000*1);
25 //     motor.move(1,100);
26 //     delay(1000*0.45);
27 //     motor.move(1,100);
28 //     delay(1000*0.8);
29 //     motor.move(1,100);
30 //     delay(1000*0.45);
31 //     motor.move(1,100);
32 //     delay(1000*1);
33 // }
34 // ... }

```

Programme : lignes de codes en langage C

Pour résoudre un problème, le programmeur commence par écrire un **algorithme** dans lequel il donne des ordres en fonction de conditions (état des capteurs). Puis il construit sur un ordinateur une **représentation graphique de l'algorithme** (Algorithme ou par bloc avec Scratch). Le logiciel va ensuite traduire la représentation graphique en **ligne de code (le programme)** que le système va exécuter.

Connaissance : Instructions conditionnelles, déclenchement d'une action par un évènement

Instructions conditionnelles : Si – Alors – Sinon : Dans un algorithme, les instructions peuvent être soumises à une condition pour s'exécuter.



```

quand la touche flèche haut est pressée
si distance mesurée par le capteur ultrasons
avancer à la vitesse 0%
sinon
avancer à la vitesse 100%

```

Exemple : S'arrêter devant un piéton à une certaine distance
- Si le capteur d'obstacle du robot détecte un piéton à une certaine distance, **alors** arrêter les moteurs.
- **Sinon** avancer à la vitesse de 100.

Déclenchement d'une action par un évènement :

Les actions peuvent être déclenchées par un évènement, par exemple :

- La **variation d'une grandeur physique** (Changement de luminosité, de chaleur, de couleur...)
- Le **déplacement d'un objet mesuré par un capteur** du système.

Exemple : Dans une maison, s'il fait nuit, alors allumer les lumières.

- L'évènement est la **variation de la luminosité**.
- Le **capteur** (LDR) mesure la **quantité de lumière** et envoie cette valeur pour stockage dans une variable (« varA » ou « luminosité »).
- Puis l'algorithme **compare** cette variable avec un **seuil** (fixé ici à 150) correspondant à la **nuit**.
- Si la valeur mesurée est en **dessous** du seuil, alors on donne l'**ordre** (action) d'**allumer** les lumières.

```

quand espace est cliqué
Mesurer la luminosité capteur de lumière LDR
mettre luminosité à capteur de lumière LDR
si luminosité < 150 alors
Allumer les lumières
sinon
Eteindre les lumières

```



Dans un **algorithme**, l'exécution des instructions peut être **conditionnée** par l'apparition d'un **évènement**. Dans ce cas, l'instruction s'exécute **SI** l'évènement a lieu. **SINON** une instruction différente se réalisera.

Connaissance : Séquences d'instructions, boucles

L'**algorithme** réalisé par le programmeur va permettre de répondre au problème posé (pour rendre les **objets connectés plus « intelligents »** par exemple). Il y a plusieurs « degrés de complexité » de programmation. Les **instructions** peuvent être simplement indiquées et **exécutées** une seule fois ou **répétées en boucle**. Les instructions peuvent aussi être **conditionnées** par l'apparition d'un évènement **détecté par un capteur**.

Séquences d'instructions :

Les actions d'un système (exemple : robot) peuvent être déclenchées en **séquences d'instructions** sans conditions préalables : avancer, tourner à gauche, à droite, reculer... Les ordres sont enchaînés les uns à la suite des autres.



```

robot - générer le code
joue la note C4* beat un demi
envoyer à un mBot le message Aloha
attendre 1 secondes
avancer à la vitesse 100%
attendre 1 secondes
tourner à droite à la vitesse 60%
attendre 2 secondes
arrêter le son

```



```

robot - générer le code
répéter 4 fois
avancer à la vitesse 100%
attendre 1 secondes
tourner à droite à la vitesse 100%
attendre 1.5 secondes
avancer à la vitesse 60%

```

```

robot - générer le code
répéter indéfiniment
avancer à la vitesse 100%
attendre 1 secondes
tourner à droite à la vitesse 100%
attendre 1.5 secondes
avancer à la vitesse 60%

```

Les **instructions** d'un **algorithme** peuvent être déclenchées en **séquences** : les ordres étant enchaînés les uns à la suite des autres sans conditions préalables (avancer, tourner...) et/ou **répétées en boucle** un nombre de fois précis, indéfiniment ou en fonction des évènements détectés par les capteurs.