



Attendus de fin de cycle

Ecrire, mettre au point et exécuter un programme.

Connaissances et
compétences associées

Ecrire, mettre au point (tester, corriger) et exécuter un programme commandant un système réel et vérifier le comportement attendu.
- Notion d'algorithme et de programme.

Notion d'algorithme et de programme

Un **algorithme** est une suite d'instructions bien précise pour résoudre un problème en un nombre fini d'étapes.

Exemple : Un robot évitant un obstacle.



Algorithme sous forme de texte

~ 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.



Représentation graphique de l'algorithme par blocs.

```
17 //AI_NPU1);
18 //((0*(analogRead(A7))>1070*1);
19 //while(1){
20 //  distance = ultrasonic_3.distanceCm();
21 //  if((distance) < (10)){
22 //    motor.move(1,0);
23 //    delay(1000*1);
24 //    motor.move(4,100);
25 //    delay(1000*0.45);
26 //    motor.move(1,100);
27 //    delay(1000*0.6);
28 //    motor.move(1,0);
29 //    delay(1000*0.6);
30 //    motor.move(3,100);
31 //    delay(1000*0.45);
32 //    motor.move(1,100);
33 //  }
34 //}
```

Un **programme** est la traduction de l'algorithme dans le langage du système.

Programme : lignes de codes en langage C.

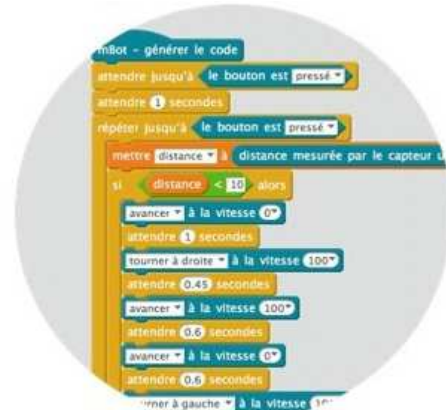
Etape 1 - Ecriture d'un algorithme :

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.

Etape 2 - Construction d'une représentation graphique de l'algorithme :



Etape 3 - Traduction de la représentation graphique en langage de programmation

```
17 //AI_NPU1);
18 //((0*(analogRead(A7))>1070*1);
19 //while(1){
20 //  distance = ultrasonic_3.distanceCm();
21 //  if((distance) < (10)){
22 //    motor.move(1,0);
23 //    delay(1000*1);
24 //    motor.move(4,100);
25 //    delay(1000*0.45);
26 //    motor.move(1,100);
27 //    delay(1000*0.6);
28 //    motor.move(1,0);
29 //    delay(1000*0.6);
30 //    motor.move(3,100);
31 //    delay(1000*0.45);
32 //    motor.move(1,100);
33 //  }
34 //}
```