



Unité 5: Les capteurs du Rover

Compétence 2 : Capteur de distance et mouvement

Dans la première leçon de l'unité 5, vous avez testé le capteur de distance du Rover pour voir comment le capteur lit et affiche une valeur. Cette leçon vous a donné l'occasion de progresser afin de contrôler le mouvement du mobile.

Objectifs :

- Utiliser l'instruction READ RV.RANGER pour déterminer la distance à un obstacle.
- Contrôler le mouvement du Rover lorsque celui-ci se rapproche trop d'un obstacle.
- Gérer le programme du Rover dans la calculatrice.

Ecrivons un programme pour que le Rover puisse aller et venir entre deux murs. Nous commencerons par faire avancer le Rover FORWARD, ensuite nous lirons la valeur lue par le capteur Ranger et lorsque le Rover se rapprochera trop du mur, celui-ci fera demi-tour et avancera de nouveau FORWARD.

La grande idée

```

Dans une boucle For (qui finira par se terminer)
  Démarrer le Rover en mouvement
  Alors que la distance est supérieure à environ 3 cm
    continuer à surveiller le capteur Ranger
  Mettre fin à la boucle While
  STOP, tournez à DROITE 180
Terminer la boucle For

```

1. Commencer un nouveau programme
2. Ajouter une boucle **For** contenant 10 itérations.
3. Ajouter l'instruction **FORWARD 100 UNITS** afin de faire avancer le robot sur 10 mètres (100 * 0.1m par unité)

```

1.1 | *Doc | RAD | 4/5
* rover52
Define rover52()=
Prgm
Send "CONNECT RV"
For i,1,10
  Send "RV FORWARD 100 UNITS "
EndFor
EndPrgm

```

Conseil de l'enseignant : Il y a plus de code à entrer dans le corps de la boucle For. À ce stade du programme, il semble que le Rover se déplace FORWARD 100 dix fois. Ce n'est pas le cas dans le programme final car le Robot s'arrêtera et fera demi-tour. La commande FORWARD entre dans la « file d'attente des instructions » du TI-Innovator™ Hub et le TI-Innovator Hub commence à traiter cette file d'attente immédiatement.

Comme dans la leçon précédente, nous utiliserons la variable *dist* pour représenter la **distance** du Ranger à un obstacle.

4. Initialiser *dist* à 1, et insérer une boucle **While *dist* > 0.25**.

Noter les deux instruction **End** dans le code : un **EndFor** et un **EndWhile**. Ceci est appelé une boucle imbriquée.

```

1.1 | *Doc | RAD | 6/8
* rover52
Define rover52()=
Prgm
Send "CONNECT RV"
For i,1,10
  Send "RV FORWARD 100 UNITS "
  dist:=1
  While dist>0.25
EndWhile
EndFor

```



10 Minutes de Code

TI-NSPIRE™ CX AVEC LE TI-INNOVATOR™ ROVER

5. Nous insérons ensuite le code dans le corps de la boucle **While**.
6. Ajouter l'instruction **Send "READ RV.RANGER"**.
7. Utiliser l'instruction **Get** pour enregistrer la valeur lue dans la variable **dist**.

UNITE 5 : COMPETENCE 2

NOTES DU PROFESSEUR

```

1.1 | *Doc | RAD | 7/9
* rover52
Send "CONNECT RV"
For i,1,10
  Send "RV FORWARD 100 UNITS "
  dist:=1
  While dist>0.25
    Send "READ RV.RANGER"
    Get dist
  EndWhile
EndFor
EndPrgm

```

Cela termine la boucle **While**. Le Rover avance de 10 mètres et la boucle **While** surveille la distance. Si on le souhaite, on ajoute une instruction **DispAt** à cette boucle **While** pour afficher la distance actuelle afin de s'assurer que les choses fonctionnent correctement.

Lorsque la boucle **While** se termine, cela indique que le mobile est trop proche d'un obstacle. Nous allons dire au Rover d'arrêter de se déplacer vers l'avant et de faire demi-tour. Cependant, nous n'avons pas besoin de dire au Rover de se déplacer à nouveau.

8. Après le **End** de la boucle **While** mais avant le **End** de la boucle **For**, ajouter l'instruction RV **STOP** ainsi que la commande **RV RIGHT 180**.
9. Ajouter l'instruction **Wait 2** pour donner au Rover le temps de se retourner avant de se déplacer à nouveau. (Ne pas oublier que la commande **FORWARD** est au début de la boucle For.)
10. Testez le programme. Lorsque le Rover se rapproche d'un obstacle, il doit faire demi-tour et aller dans la direction opposée. Ajuster les valeurs (distance minimum et temps d'attente), selon les besoins, pour la surface de déplacement. Si le Rover se rapproche trop du mur, l'arrière du Rover pourrait heurter le mur en se retournant.
- 11.

```

1.1 | *Doc | RAD | 11/12
* rover52
dist:=1
While dist>0.25
  Send "READ RV.RANGER"
  Get dist
EndWhile
Send "RV STOP "
Send "RV RIGHT 180"
Wait 2
EndFor
EndPrgm

```

Conseil de l'enseignant : Un programme complet (Indenté pour plus de clarté) :

```

Send "CONNECT RV"
For i,1,10
  Send("RV FORWARD 100")
  1→d
  While d>0.25
    Send "READ RV.RANGER"
    Get d
  EndWhile
  Send "RV STOP "
  Send "RV RIGHT 180"
  Wait 2
EndFor

```