



Dans la première leçon de l'unité 5, vous avez testé le capteur de distance du Rover afin de voir comment le capteur lit et affiche une valeur. Cette leçon poursuit les essais pour contrôler le mouvement du robot.

#### Objectifs :

- Utiliser l'instruction **READ RV.RANGER** pour mesurer la distance à un obstacle
- Contrôler le mouvement du Rover quand le Rover se rapproche trop d'un obstacle
- Gérer le temps lors de la programmation du Robot dans le programme de la calculatrice

Écrivons un programme pour que le Rover puisse aller et venir entre deux « murs ». Nous allons mettre le Rover en mouvement, lire le capteur à ultrasons Ranger et lorsque le Rover se rapproche trop du mur, faire arrêter le Rover, puis faire demi-tour et avancer.

#### 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**( celle-ci devra réaliser 10 itérations.
3. Ajoutez une commande **FORWARD 100** pour avancer de 10 mètres (100 \* 0.1m par unité).
4. Ne pas oublier **End** de la boucle **For**(.

```

NORMAL FIXE2 AUTO RÉEL RAD MP
EDIT MENU: [α]Pho.] [f5]
PROGRAM:ROVERS2
:Effécran
:Send("CONNECT RV")
:For(I,1,10)
:Send("RV FORWARD 100")
:

```

**Conseil de l'enseignant** : A cet endroit du programme, il apparait que le Rover de déplacera vers l'avant de 100 unités FORWARD 100 et 10 fois. Ce n'est pas le cas dans le programme final où le capteur Ranger arrêtera le Rover qui retournera sur ses pas.

Comme dans la leçon précédente, nous utilisons la variable **D** pour représenter la distance entre le capteur Ranger et un obstacle.

5. Initialiser **D** à 1, et commencer la boucle tant que **While D > 0.25**.

Noter les deux **End** dans le code ; un **End** est pour la boucle **For** et un autre **End** est pour la boucle **While**.

```

NORMAL FIXE2 AUTO RÉEL RAD MP
EDIT MENU: [α]Pho.] [f5]
PROGRAM:ROVERS2
:Effécran
:Send("CONNECT RV")
:For(I,1,10)
:Send("RV FORWARD 100")
:1→D
:While D>0.25
:End
:End
:

```



## 10 Minutes de Code

### TI-83 PREMIUM CE AVEC LE TI-INNOVATOR™ ROVER

- Nous insérons ensuite le code dans le corps de la boucle **While**. Si l'on n'a pas assez de lignes vides dans le programme, placer le curseur au début d'une ligne où le code doit être inséré. Appuyer sur [Insérer] (**2nd** **suppr**), et ensuite appuyer sur **enter**. Une autre méthode pour ajouter une ligne sous la ligne en cours, consiste à appuyer sur **alpha** **F5**, puis à sélectionner **Insert Ligne** ↑.
- Ajouter l'instruction **Send("READ RV.RANGER")**.
- Utiliser l'instruction **Get**( pour sauvegarder le contenu de la variable **D**.

Cela termine la boucle **While**. Le robot avance de 10 mètres et la boucle **While** surveille la distance. Si on le souhaite, ajouter une instruction **Output**( à cette boucle **While** pour afficher la distance actuelle afin de vous assurer que le programme fonctionnent correctement.

Lorsque la boucle **While** se termine, cela indique que le robot est trop proche d'un obstacle. Nous allons demander 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.

- Après les instructions **End** de la boucle **While** mais avant le **End** de la boucle **For** ajouter l'instruction **RV STOP** puis la commande **RV RIGHT 180**.
- Ajouter une instruction **Wait 2** pour donner au Rover le temps de tourner à droite de 180° avant de repartir « **FORWARD 100** » une nouvelle fois. (Se souvenir que l'instruction **FORWARD** est au début de la boucle **For**.)
- Quitter l'éditeur de programme, et tester votre programme. Lorsque le Rover se rapproche d'un obstacle, il doit faire demi-tour et aller dans la direction opposée. Ajuster les valeurs (distance minimale et temps d'attente), selon les besoins correspondants à la situation de déplacement du robot (espace). Si le Rover se rapproche trop du mur, l'arrière du Rover pourrait heurter le mur en se retournant.

## UNITE 5: COMPETENCE 2

### NOTES DU PROFESSEUR

```
NORMAL FIXE2 AUTO REEL RAD MP
EDIT MENU: [a:1pha.] [f5]

PROGRAM: ROVERS2
: Effécran
: Send("CONNECT RV")
: For(I,1,10)
: Send("RV FORWARD 100")
: 1→D
: While D>0.25
: End
: End
:
```

```
NORMAL FIXE2 AUTO REEL RAD MP
EDIT MENU: [a:1pha.] [f5]

PROGRAM: ROVERS2
: Effécran
: Send("CONNECT RV")
: For(I,1,10)
: Send("RV FORWARD 100")
: 1→D
: While D>0.25
: Send("READ RV.RANGER")
: Get(D)
: End
```

```
NORMAL FIXE2 AUTO REEL RAD MP
EDIT MENU: [a:1pha.] [f5]

PROGRAM: ROVERS2
: 1→D
: While D>0.25
: Send("READ RV.RANGER")
: Get(D)
: End
: Send("RV STOP ")
: Send("RV RIGHT 180")
: Wait 2
: End
```

**Conseil de l'enseignant** : Le programme complet (indenté pour plus de clarté):

```
EffEcran
Send("CONNECT RV")
For(I,1,10)
  Send("RV FORWARD 100")
  1→D
  While D>0.25
    Send("READ RV.RANGER")
```



## 10 Minutes de Code

TI-83 PREMIUM CE AVEC LE TI-INNOVATOR™ ROVER

UNITE 5: COMPETENCE 2

NOTES DU PROFESSEUR

```
Get(D)
End
Send("RV STOP ")
Send("RV RIGHT 180")
Wait 2
End
```