



In de eerste les van module 5, heeft u de afstandsmeter van de Rover getest om te zien hoe u de sensor moet lezen en een waarde weergeeft. Deze les brengt dit testen verder om de beweging van de Rover aan te sturen.

Doelen:

- De opdracht READ RV.RANGER gebruiken om de afstand tot een obstakel te bepalen.
- De beweging van de Rover aansturen wanneer de Rover te dicht bij een obstakel komt.
- De timing van de beweging van de Rover regelen binnen het programma van de rekenmachine

We gaan een programma schrijven om de Rover heen en weer te laten bewegen tussen twee muren. We beginnen de Rover FORWARD te laten bewegen, de afstandsmeter uit te lezen en, wanneer de Rover te dicht bij de muur komt, de Rover te laten omkeren en FORWARD te bewegen.

Het idee in grote lijnen

In een For-lus (die uiteindelijk afloopt)

Begin met de Rover die FORWARD beweegt

Zolang (while) de afstand meer dan ongeveer 3 cm is

ga door met het controleren van de afstandsmeter

Beëindig (end) de While-lus

STOP, draai RIGHT 180

Beëindig (end) de For-lus

1. Begin het programma op de gebruikelijke manier.
2. Voeg een **For**(-lus toe die 10 keer herhaald zal worden.
3. Voeg een opdracht **FORWARD 100** toe om 10 meter vooruit te bewegen (100 * 0,1m per unit) toe.
4. Denk er aan om de **For**(-lus structuur af te sluiten met **End**.

```
NORMAL FLOAT AUTO REAL RADIAN MP
EDIT MENU: [a]lpha [f5]
PROGRAM: ROVER52
:ClrHome
:Send("CONNECT RV")
:For(I,1,10)
:Send("RV FORWARD 100")
:
```

Docenten Tip: Op deze plaats in het programma lijkt het alsof de Rover 10 keer 100 FORWARD zal bewegen. Dit is echter niet het geval in het uiteindelijke programma omdat de afstandsmeter (Ranger) ervoor zorgt dat de Rover stopt en zich omdraait.

Net zoals in de vorige les gebruiken we ook hier de variabele **D** om de afstand van de afstandsmeter tot een obstakel te representeren.

5. Initialiseer **D** met de waarde 1 en voeg een lus **While D > 0.25** in.

Merk op dat er in de programmacode twee keer een **End** staat: een **End** voor de **For**-lus en een **End** voor de **While**-lus.

```
NORMAL FLOAT AUTO REAL RADIAN MP
EDIT MENU: [a]lpha [f5]
PROGRAM: ROVER52
:ClrHome
:Send("CONNECT RV")
:For(I,1,10)
:Send("RV FORWARD 100")
:1→D
:While D>0.25
:End
:End
:
```



10 minuten programmeren

TI-84 PLUS CE MET DE TI-INNOVATOR™ ROVER

- We voegen vervolgens de programmacode in in de kern van de While-lus. Als je niet genoeg lege regels in je programma hebt, plaats de cursor dan aan het begin van een regel waar je de programmacode wilt invoegen. Druk op `[ins]` (`[2nd]` `[del]`) en druk vervolgens op `[enter]`. Je kunt ook dit alternatief gebruiken: om een regel met programmacode boven de huidige regel toe te voegen: druk op `[alpha]` F5 en selecteer **Insert Line Above**.
- Voeg de opdracht **Send("READ RV.RANGER")** toe.
- Gebruik de opdracht **Get** (om de waarde op te slaan in de variabele **D**).

MODULE 5: OEFENBLAD 2

DOCENTENHANDLEIDING

```
NORMAL FLOAT AUTO REAL RADIAN MP
EDIT MENU: [alpha][f5]
PROGRAM: ROVER52
:ClrHome
:Send("CONNECT RV")
:For(I,1,10)
:Send("RV FORWARD 100")
:1→D
:While D>0.25
:End
:End
:
```

```
NORMAL FLOAT AUTO REAL RADIAN MP
EDIT MENU: [alpha][f5]
PROGRAM: ROVER52
:ClrHome
:Send("CONNECT RV")
:For(I,1,10)
:Send("RV FORWARD 100")
:1→D
:While D>0.25
:Send("READ RV.RANGER")
:Get(D)
:End
```

Dit maakt de **While**-lus af. De Rover beweegt 10 meter FORWARD en de **While**-lus houdt de afstand bij. Voeg desgewenst een **Output** (-opdracht toe aan deze **While**-lus om de actuele afstand te tonen om er zeker van te zijn dat alles naar behoren werkt.

Als de **While**-lus eindigt geeft dit aan dat de Rover te dichtbij een obstakel is. We gaan de Rover vertellen dat hij moet STOPpen met FORWARD bewegen en om moet draaien. We hoeven de Rover echter niet te vertellen om opnieuw FORWARD te gaan bewegen.

- Voeg *na* de **End** van de **While**-lus, maar *voor* de **End** van de **For**-lus, de opdracht RV **STOP** en vervolgens de opdracht **RV RIGHT 180** toe.
- Voeg de opdracht **Wait 2** toe om de Rover tijd te geven om om te draaien voordat hij opnieuw **FORWARD** gaat bewegen. (Bedenk dat de opdracht **FORWARD** aan het begin van de **For**-lus staat.)
- Verlaat en test je programma. Als de Rover dicht bij een obstakel komt, zou hij moeten omdraaien en in de tegengestelde richting bewegen. Pas, als dat nodig is, de waarden (minimumafstand en **Wait** tijd) aan voor jouw situatie (het oppervlak waarop de Rover beweegt). Als de Rover te dicht bij de muur komt dan kan de achterkant van de Rover tegen de muur botsen bij het keren.

```
NORMAL FLOAT AUTO REAL RADIAN MP
EDIT MENU: [alpha][f5]
PROGRAM: ROVER52
:1→D
:While D>0.25
:Send("READ RV.RANGER")
:Get(D)
:End
:Send("RV STOP ")
:Send("RV RIGHT 180")
:Wait 2
:End
```

Teacher Tip: Een volledig programma (inspringen is alleen voor de duidelijkheid):

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



10 minuten programmeren

TI-84 PLUS CE MET DE TI-INNOVATOR™ ROVER

MODULE 5: OEFENBLAD 2

DOCENTENHANDLEIDING

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