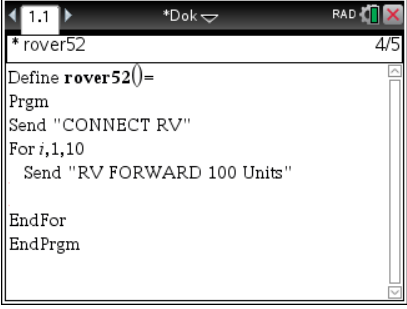




Lektion 5: Die Sensoren des Rovers	Übung 2: Der Ranger und Bewegung
<p>In der ersten Übung dieser Lektion hast du den Ranger des Rovers getestet, um zu sehen, wie man den Sensor einsetzt und einen Wert anzeigen lässt. Jetzt werden wir diesen Test weiterführen, indem wir die Fahrt des Rovers kontrollieren.</p>	<p><b>Lernziele:</b></p> <ul style="list-style-type: none"> <li>• Die <b>READ RV.RANGER</b>-Anweisung verwenden, um die Entfernung von einem Hindernis festzustellen</li> <li>• Die Fahrt des Rovers steuern, wenn er zu nahe an ein Hindernis gerät</li> <li>• Den zeitlichen Ablauf der Roverfahrt im Rahmen des Programms steuern</li> </ul>
<p>Wir wollen ein Programm schreiben, das den Rover zwischen zwei Wänden nach vorwärts und rückwärts fahren lässt. Wir beginnen mit der Vorwärtsbewegung (FORWARD), lesen den Sensor des Rangers ab, und sobald der Rover zu nahe an die Wand gerät, bleibt der Rover stehen, kehrt um und bewegt sich wieder vorwärts (FORWARD).</p>	
<p><b>Das ist die große Idee:</b></p> <ul style="list-style-type: none"> <li>• Beginne eine For-Schleife (die am Ende der Fahrt verlassen wird)</li> <li>• Starte mit der Vorwärtsbewegung des Rovers (FORWARD)</li> <li>• Solange der Abstand größer ist als etwa 25 cm             <ul style="list-style-type: none"> <li>• beobachte weiterhin den Sensor des Rangers</li> </ul> </li> <li>• Verlasse die While-Schleife</li> <li>• STOP, drehe den Rover um 180° nach rechts</li> </ul> <p>Beende die For-Schleife</p>	
<ol style="list-style-type: none"> <li>1. Beginne das Programm in der üblichen Weise.</li> <li>2. Füge eine <b>For</b>-Schleife ein, die zehn Mal durchlaufen wird.</li> <li>3. Fahre fort mit der <b>FORWARD 100 UNITS</b>-Anweisung für eine Vorwärtsbewegung um 10 Meter (100 * 0,1m pro Einheit).</li> </ol>	 <pre> 1.1 *Dok RAD 4/5 * rover52 Define rover52()= Prgm Send "CONNECT RV" For i,1,10   Send "RV FORWARD 100 Units" EndFor EndPrgm </pre>
<p><b>Hinweis:</b> Da kommt noch mehr Code in die Schleife. Momentan sieht es aus, als sollte der Rover 10-mal FORWARD 100 ausführen. Im fertigen Programm wird das nicht der Fall sein, denn der Ranger wird den Rover anhalten und wenden lassen. Die FORWARD-Anweisung geht in die „Fahr-Warteschlange“ im TI-Innovator™ Hub und dieser beginnt sofort mit der Abarbeitung dieser Schlange.</p>	



Wie in der vorigen Übung verwenden wir die Variable **entf**, um die **Entfernung** des Rovers vom Hindernis zu messen.

4. Setze **entf** auf 1 und füge eine **While entf > 0.25**-Schleife an.

Beachte die beiden **End**-Anweisungen im Programm: ein **EndFor** und ein **EndWhile**. Das wird als eine „geschachtelte Schleife“ bezeichnet.

```
* rover52 7/8
Define rover52()=
Prgm
Send "CONNECT RV"
For i,1,10
  Send "RV FORWARD 100 Units"
  entf:=1
  While entf>0.25
EndWhile
EndFor
```

5. Dann schreiben wir den Programmcode in den Körper der **While**-Schleife.
6. Zuerst die Anweisung **Send "READ RV.RANGER"**
7. Und dann die **Get**-Anweisung, um den Wert in der Variablen **entf** zu speichern.

```
* rover52 6/9
Send "CONNECT RV"
For i,1,10
  Send "RV FORWARD 100 Units"
  entf:=1
  While entf>0.25
    Send "READ RV.RANGER"
    Get entf
  EndWhile
EndFor
EndPrgm
```

Damit ist die **While**-Schleife vollständig. Der Rover bewegt sich 10 m vorwärts und die **While**-Schleife überwacht den Abstand. Wenn gewünscht, kannst du mit einer **DispAt**-Anweisung in der Schleife den aktuellen Abstand anzeigen lassen, um sicher zu gehen, dass alles richtig abläuft.

Wenn die **While**-Schleife endet, dann bedeutet dies, dass der Rover dem Hindernis zu nahe gekommen ist. Wir sagen **STOP** mit der Vorwärtsbewegung und lassen ihn umkehren. Wir müssen aber nicht wieder befehlen, vorwärts zu fahren.

8. *Nach* dem **End While** aber noch *vor* dem **End For**, werden die **RV STOP**- und die **RV RIGHT 180**-Anweisung eingefügt.
9. Nimm die **Wait 2**-Anweisung und gib damit dem Rover ausreichend Zeit für das Umkehren, bevor er sich wieder nach vorwärts bewegt. (Erinnere dich, dass die **FORWARD**-Anweisung schon am Beginn der **For**-Schleife steht.)
10. Teste nun dein Programm. Wenn der Rover nahe beim Hindernis angelangt ist, soll er umkehren und in die Gegenrichtung fahren. Passe allfällig die Werte für den Minimalabstand und **Wait** an deine Fahrunterlage an. Wenn der Rover zu nahe an die Wand gerät, könnte sein Heck beim Umdrehen an die Wand stoßen.

```
* rover52 11/12
entf:=1
While entf>0.25
  Send "READ RV.RANGER"
  Get entf
EndWhile
Send "RV STOP "
Send "RV RIGHT 180"
Wait 2
EndFor
EndPrgm
```

**Hinweis:** Das komplette Programm (der Deutlichkeit halber):

```
Send "CONNECT RV"
For i,1,10
  Send "RV FORWARD 100"
  entf:=1
  While entf>0.25
    Send "READ RV.RANGER"
    Get entf
  EndWhile
  Send "RV STOP "
  Send "RV RIGHT 180"
  Wait 2
EndFor
```