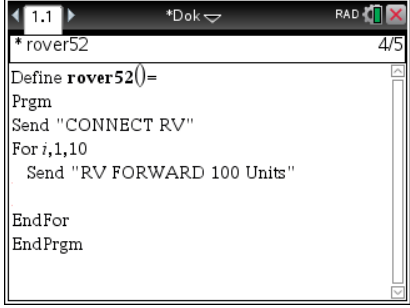
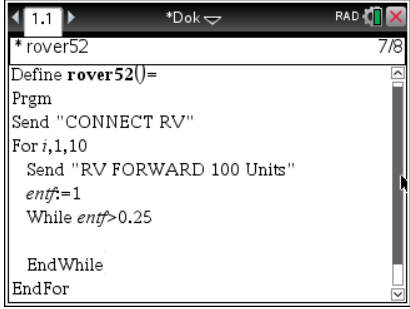




Lektion 5: Die Sensoren des Rovers	Übung 2: Der Ranger und Bewegung
<p>In der ersten Übung dieser Lektion hast du den Ranger des Rovers getestet, um zu sehen, wie man den Sensor einsetzt und einen Wert anzeigen lässt. Jetzt werden wir diesen Test weiterführen, indem wir die Fahrt des Rovers kontrollieren.</p>	<p>Lernziele:</p> <ul style="list-style-type: none"> Die READ RV.RANGER-Anweisung verwenden, um die Entfernung von einem Hindernis festzustellen Die Fahrt des Rovers steuern, wenn er zu nahe an ein Hindernis gerät Den zeitlichen Ablauf der Roverfahrt im Rahmen des Programms steuern
<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 nach (FORWARD).</p>	
<p>Das ist die große Idee:</p> <p>Beginne eine For-Schleife (die am Ende der Fahrt verlassen wird)</p> <p> Starte mit der Vorwärtsbewegung des Rovers (FORWARD)</p> <p> Solange der Abstand größer ist als etwa 25 cm</p> <p> beobachte weiterhin den Sensor des Rangers</p> <p> Verlasse die While-Schleife</p> <p> STOP, drehe den Rover um 180° nach rechts</p> <p>Beende die For-Schleife</p>	
<ol style="list-style-type: none"> 1. Beginne das Programm in der üblichen Weise. 2. Füge eine For-Schleife ein, die zehn Mal durchlaufen wird. 3. Fahre fort mit der FORWARD 100 UNITS-Anweisung für eine Vorwärtsbewegung um 10 Meter (100 * 0,1m pro Einheit). 	 <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>Wie in der vorigen Übung verwenden wir die Variable entf, um die Entfernung des Rovers vom Hindernis zu messen.</p> <ol style="list-style-type: none"> 4. Setze entf auf 1 und füge eine While entf > 0.25-Schleife an. <p>Beachte die beiden End-Anweisungen im Programm: ein EndFor und ein EndWhile. Das wird als eine „geschachtelte Schleife“ bezeichnet.</p>	 <pre> 1.1 *Dok RAD 7/8 *rover52 Define rover52()= Prgm Send "CONNECT RV" For i,1,10 Send "RV FORWARD 100 Units" entf=1 While entf>0.25 EndWhile EndFor </pre>



10 Minuten Coding

TI-NSPIRE™ CX MIT DEM TI-INNOVATOR™ ROVER

LEKTION 5: ÜBUNG 2

SCHÜLERTÄTIGKEIT

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.

```
1.1 *Dok RAD 6/9
*rover52
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.

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