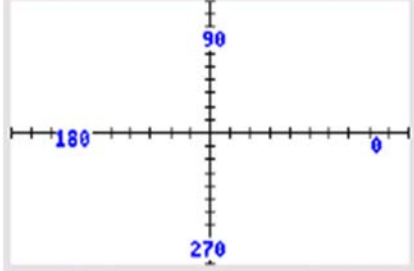
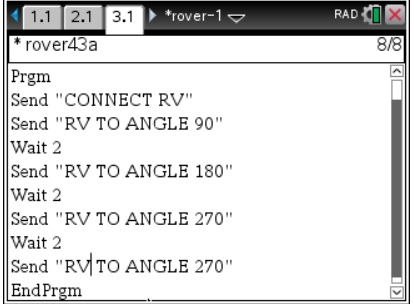




Lektion 4: Setz den Rover in Bewegung!	Übung 3: Optionen, Farb- und Zeiteinstellung
<p>In der dritten Übung dieser Lektion wirst du lernen, wie man den Rover in eine bestimmte Richtung dreht, wie man den zeitlichen Ablauf steuert und wie man mit der COLOR LED-am Rover arbeitet.</p>	<p>Lernziele:</p> <ul style="list-style-type: none"> • Die TO ANGLE-Anweisung verwenden • Die RV.COLOR LED am Rover einsetzen • Den zeitlichen Ablauf am Rechner und am Rover kontrollieren
<p>Diese Übung spricht drei weitere Möglichkeiten des Arbeitens mit dem Rover an:</p> <ul style="list-style-type: none"> • Einsatz der TO ANGLE-Anweisung (die sich von LEFT und RIGHT unterscheidet) • Einschalten der RV.COLOR LED am Rover (bezeichnet mit 'Color' an der linken oberen Ecke neben der Anzeige der Batterieladung) • Synchronisierung des Programms mit den Bewegungen des Rovers mit Wait 	
<p>TO ANGLE</p> <p>Die Anweisung Send "RV TO ANGLE <Zahl>" wird verwendet, um den Rover in eine bestimmte Richtung zu drehen. Wenn du mit einem Befehl zum Rover verbindest, dann ist seine Richtung auf 0° eingestellt. Das ist in der „Mathematikwelt“ in Richtung Osten (positive Richtung der x-Achse). Nord, West und Süd sind in dieser „Welt“ 90°, 180° und 270°. Vergleiche mit dem rechts abgebildeten Diagramm.</p> <p>Unabhängig von seiner momentanen Richtung bringt die Anweisung Send "RV TO ANGLE 0" den Rover wieder in seine Ausgangsrichtung, die er beim Start mit "CONNECT RV" eingenommen hat.</p> <p>Das Standardmaß für den Winkel ist DEGREES, aber du kannst es auf RADIANS oder GRADS (auszuwählen über das RV Settings-Menü) abändern.</p>	 <p style="text-align: center;"><i>Richtungen</i></p>
<p>Versuche dies:</p> <p>Send "RV TO ANGLE 90" Wait 2 Send "RV TO ANGLE 180" Wait 2 Send "RV TO ANGLE 270" Wait 2 Send "RV TO ANGLE 360"</p> <p>Hat sich der Rover so verhalten, wie du es erwartet hast?</p>	



Synchronisierung des Programms mit dem Rover

Programme am Rechner sind schon *“fertig”*, wenn sich der Rover noch bewegt. Das rührt daher, dass die Steuerkommandos im TI-Innovator™ Hub schneller gespeichert werden als sie der Rover ausführen kann. Die Steuerkommandos werden in einer „Warteschlange“ gereiht und ausgeführt sobald der Rover dazu bereit ist.

Jetzt werden wir ein Programm schreiben, das den Rover auf einem Zufallsweg bewegt und dabei die **RV.COLOR** LED aufleuchten lässt. Wir werden auch die **TO ANGLE**-Anweisung gemeinsam mit **eval()** verwenden, um den Rover in die entsprechenden Richtungen zu drehen.

Der Beginn des Programms:

1. Starte das Programm mit **CONNECT RV**.
2. Fahre fort mit einer **For**-Schleife (*menu > Control > For...EndFor*), um das Zufallsmuster für den Weg des Rovers zu erzeugen. In der Schleife kannst du immer weitere benötigte Zeilen hinzufügen. (Hier werden **h** und **i** als lokale Variable eingeführt, um den Speicher „sauber“ zu halten.)

```

* rover43
Define rover43()=
Prgm
© fährt einen Irrweg
Local h,i
Send "CONNECT RV"
Text "Zum Start drücke enter!"
For i,1,10
EndFor
EndPrgm
  
```

3. Schreibe **FORWARD** in den Schleifenkörper für eine Vorwärtsbewegung.
Send “RV FORWARD 1”
4. Um dem Rover eine *Zufallsrichtung* zu geben, nimm die Anweisung:
h:=randInt(0,360)
5. Füge eine **TO ANGLE**-Anweisung an, um in die Richtung **h** zu drehen:
Send “RV TO ANGLE eval(h)”
6. Starte nun das Programm und du wirst sehen, dass
 - sich der Rover in einem Zufallsmuster (Irrweg) bewegt,
 - und das Programm fast sofort beendet ist, sich der Rover aber weiterhin bewegt.

```

* rover43
Send "CONNECT RV"
Text "Zum Start enter drücken!"
For i,1,10
  Send "RV FORWARD 1"

  h:=randInt(0,360)

  Send "RV TO ANGLE eval(h)"
{}
EndFor
  
```

RV.COLOR

Da die Farb-LED des TI-Innovator™ Hub im Rover verborgen ist, haben wir eine andere Farb-LED am Dach des Rovers angebracht. Der Name dieser LED ist **RV.COLOR** und sie wird wie die LED am TI-Innovator™ Hub gesteuert. Du kannst jede der vier Anweisungen aus dem *menu > Hub > Rover(RV) > RV Color* verwenden, das rechts gezeigt wird.

Send “SET RV.COLOR 255 255 255” erzeugt weißes Licht.

1: Aktionen	
2: Syntax	1: Send "SET
3: Variabl	2: Send "READ
1: Drive RV	
2: Read RV Sensors	
3: RV Settings	
4: Read RV Path	
5: RV Color	1: RV.COLOR
6: RV Setup	2: RV.COLOR.RED
7: RV Control	3: RV.COLOR.GREEN
8: Send "CONNECT RV"	4: RV.COLOR.BLUE
9: Send "DISCONNECT RV"	



Einbau von RV.COLOR in dein Programm

- Setze die **RV.COLOR**-Anweisung an den Beginn der Schleife unmittelbar vor die **FORWARD 1**-Anweisung. Die Farbwerte sind dir überlassen.
- Starte das Programm nochmals und beobachte, was geschieht. Beachte, dass die LED fast unmittelbar aufleuchtet und weiter leuchtet.

```

* rover43
Text "Zum Start drücke enter!"
For i,1,n
  Send "SET RV.COLOR 128 246 100"
  Send "RV FORWARD 1"

  h:=randInt(0,360)
  Send "RV TO ANGLE eval(h)"
EndFor
EndPrgm

```

Wir wollen versuchen, dass die LED nur während der Vorwärtsfahrt des Rovers aufleuchtet. Dazu müssen wir den Rechner immer warten (**Wait**) lassen, bis eine Wegstrecke erledigt ist und dann während der Wendung die LED ausschalten.

Wir müssen eine **Wait**-Anweisung zur Steuerung des Ein- und Ausschaltens der LED einfügen.

Hinweis: Jetzt ist, wenn du Zeit hast, eine gute Gelegenheit zum Experimentieren. Im Gegensatz zu den Drive-Befehlen wird die **COLOR**-Anweisung sofort ausgeführt sobald sie vom TI-Innovator™ Hub empfangen wird. Daher laufen Bewegung und Licht nicht synchron. Das bleibt dem der Programmierer überlassen.

- Wie lange braucht der Rover für **FORWARD 1**? Etwa eine Sekunde? Schreibe **Wait 1** nach der **FORWARD 1**-Anweisung ins Programm. Du kannst den Wert anpassen, wenn er in Deiner Umgebung nicht passt. Denk daran, dass du **Wait** im *menu > Hub* Menü findest.
- Führe das Programm aus und teste es.

```

* rover43
Text "Zum Start drücke enter!"
For i,1,10
  Send "SET RV.COLOR 128 246 100"
  Send "RV FORWARD 1"
  Wait 1

  h:=randInt(0,360)
  Send "RV TO ANGLE eval(h)"
EndFor

```

Beachte, dass die LED eingeschaltet bleibt. Wir müssen sie noch ausschalten, sobald der Rover seine Fahrt beendet hat.

- Wie lange braucht der Rover für eine Wendung? Etwa eine Sekunde? Füge eine weitere **Wait 1**-Anweisung hinzu.
- Nach der **Wait 1**-Anweisung kannst du die LED abschalten, indem du den Befehl **Send "SET RV.COLOR 0 0 0"** anfügst.
- Teste dein Programm. Hat sich das Licht zur richtigen Zeit ausgeschaltet?

```

* rover43
Text "Zum Start drücke enter!"
For i,1,10
  Send "SET RV.COLOR 128 246 100"
  Send "RV FORWARD 1"
  Wait 1
  Send "SET RV.COLOR 0 0 0"
  h:=randInt(0,360)
  Send "RV TO ANGLE eval(h)"
EndFor

```

Wir müssen auch warten während sich der Rover in eine neue Richtung dreht, um sicher zu sein, dass das Licht ausgeschaltet ist.

- Schreibe ein **Wait** hinter die **TO ANGLE**-Anweisung. Dieses **Wait** sollte lange genug dauern, um jede Drehung zwischen 0 und 360 Grad zu ermöglichen.

```

* rover43
Text "Zum Start drücke enter!"
For i,1,10
  Send "SET RV.COLOR 128 246 100"
  Send "RV FORWARD 1"
  Wait 1
  Send "SET RV.COLOR 0 0 0"
  h:=randInt(0,360)
  Send "RV TO ANGLE eval(h)"
  Wait 3
EndFor
EndPrgm

```



Erweiterung

Kannst Du an jeder Ecke eine *andere* Farbe leuchten lassen? Tipp: Verwende **eval**(*frei gewählt*) und berücksichtige diesen Wert für **h**, beachte aber den dafür zulässigen Wertebereich.

Eine zusätzliche Herausforderung:

Definiere die Wartezeit für die Drehung als eine Funktion des Drehwinkels, der in der **TO ANGLE**-Anweisung festgelegt wird.

Hinweis: Fahrbefehle werden sofort an den TI-Innovator™ Hub geschickt und werden dort in einer Warteschlange gereiht bis der Rover der Reihe nach für sie bereit ist.

Die **Wait**-Kommandos sind Anweisungen *an den Rechner*, zu warte,n bis der Rover die laufende Aufgabe ausgeführt hat. Lange **Wait**-Anweisungen sind möglich, aber der Rover macht dann während seiner Fahrt eine scheinbare Pause bis er die nächste Anweisung erhält.

Für die Erweiterung können die Schüler mit einer Kombination der Laufvariablen *i* und der Richtung *h* verschiedene Farben erzeugen. Der Bereich für r, g und b reicht von 0 bis 255.