Lektion 5: Die Sensoren des Rovers **Übung 2: Ranger und Bewegung**

In der ersten Übung der Lektion 5 wurde der Ranger vorgestellt und wie man mit ihme Entfernungen bestimmt und darstellt. Nun erfolgt die Erweiterung, in der der Ranger die Bewegung des Rovers beeinflusst.

Lernziele:

- Verwendung des Befehls **READ RV.RANGER,** um die Entfernung zu einem Hindernis zu bestimmen
- Steuern der Bewegung des Rovers, wenn er zu dicht an das Hindernis herankommt
- Berücksichtigen des Timings bei der Bewegung des Rovers im Programm

---

Es soll ein Programm geschrieben werden, mit dem der Rover hin und her zwischen zwei „Mauern“ fährt, die sich genau gegenüberstehen. Der Rover fährt erst vorwärts und bestimmt dann die Entfernung. Gerät er zu dicht an die „Mauer“, so hält der Rover an, dreht sich um 180° und fährt wieder vorwärts bis zur nächsten „Mauer“.

Daraus ergibt sich der grundsätzliche Ablaufplan

Kernstück ist die zentrale For-Schleife, die dafür sorgt, dass der Rover nach einiger Zeit wieder anhält.

For - Schleife

Der Rover startet vorwärts (FORWARD)

While die Entfernung größer ist als 0,25 m

Ermittele die Entfernung

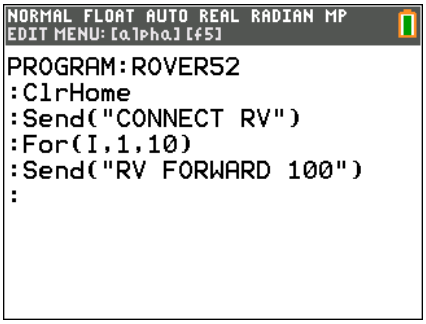
End der While - Schleife

STOP

Drehe um 180° (RIGHT 180)

End der For - Schleife

Das Programm

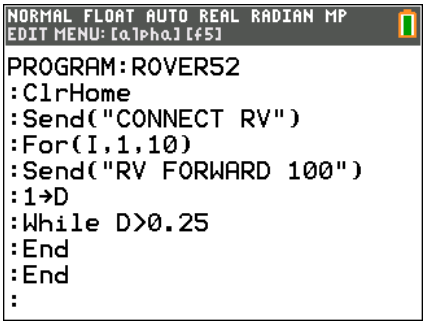


1. Das Programm enthält am Anfang die üblichen Befehle.
2. Eine **For(** - Schleife wird hinzugefügt, die 10mal durchlaufen wird..
3. Der Befehl **FORWARD 100** wird hinzugefügt, so dass der Rover maximal 10 Meter (100 * 0.1m pro Entfernungseinheit) fährt.
4. Der Befehl **End** der **For(** - Schleife darf nicht vergessen werden!

Hinweis: Im Moment sieht es so aus, als würde der Rover zehnmal jeweils 10 m vorausfahren. Das wird im endgültigen Programm natürlich nicht so sein, da der Ranger den Rover anhalten und er dann drehen wird.

Wie in der vorhergehenden Übung wird wieder die Variable ***D*** für die Entfernung (**d**istance) des Rangers vom Hindernis verwendet.

5. Bestimmung des Anfangswertes von ***D*** zu ***1.***
6. Starten der **While D > 0.25** - Schleife.

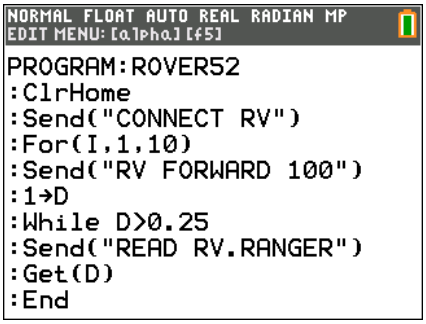


Der Programmcode enthält zwei **End**s: einer ist das **End** der **For** - Schleife und der andere das **End** der **While** - Schleife.

7. Nun kommen die Befehle, die sich innerhalb der **While** – Schleife befinden. Hat man nicht genug Leerzeilen bis zum **End** eingefügt, kann man den Cursor an den Anfang der Zeile setzen, wo man einen weiteren Befehl einfügen möchte. Mit [ins] ([2nd] [del]) und dann [enter] fügt man eine neue Leerzeile ein. Alternativ dazu kann man eine Leerzeile auch einfügen über [alpha] F5 und dann **Insert Line Above**.

8. Der Befehl **Send("READ RV.RANGER")** wird hinzugefügt.

9. Mit dem Befehl **Get(** wird die Entfernung in der Variablen ***D*** gespeichert.



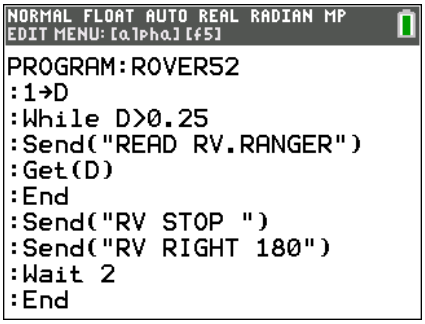
Die **While** – Schleife ist nun vollständig. Der Rover fährt 10 m geradeaus und in der **While** – Schleife wird die Entfernung gemessen. Wenn man will, kann man sich die Entfernung auf dem Display mit einem Befehl **Output(** innerhalb der **While** – Schleife anzeigen lassen um zu überprüfen, dass alles richtig funktioniert.

Endet die **While** – Schleife, so ist das ein Hinweis darauf, dass sich der Rover zu dicht vor einem Hindernis befindet. Jetzt muss dem Rover mitgeteilt werden, dass er anhalten (**STOP**) und sich um 180° (**RIGHT 180**) drehen soll. Aber man muss dem Rover nicht erneut mitteilen, dass er vorwärtsfahren soll, da der erneute Durchgang bei der **For** – Schleife ja mit diesem Befehl anfängt

10. *Zwischen* dem **End** der **While** – Schleife und dem **End** der **For** – Schleife werden also die Befehle
    RV STOP und dann **RV RIGHT 180** eingefügt.

11. Der zusätzliche Befehl **Wait 2** lässt dem Rover genügend Zeit, um sich einmal umzudrehen, bevor er wieder vorwärtsfährt, da nun ein erneuter Durchlauf der **For** – Schleife anfängt.

12. Das Programm ist fertig und kann getestet werden. Nähert sich der Rover nun einem Hindernis, so sollte er sich um 180° drehen und in die entgegengesetzte Richtung zurückfahren. Die Werte für die Minimaldistanz und die Wartezeit beim Drehen müssen eventuell noch angepasst werden. Ist die Minimaldistanz zu klein, so kann der Rover mit seinem Heck an die „Mauer“ treffen und die Drehung beeinflussen.



Hinweis:

Zur Verdeutlichung noch einmal das vollständige Programm:

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