



Lektion 6 : Verwendung der Module ti_hub & ti_rover

Übung 3 : Der Rover

In dieser dritten Übung der Lektion 6 soll gezeigt werden, wie man die Anweisungen aus dem Modul **ti_rover** in ein Programm einbindet.

Lernziele :

- Entdeckung des Moduls **ti_rover**.
- Schreiben eines Programmes, das die Sensoren des Rovers mit Fahrbefehlen verknüpft.
- Verwendung einer While-Schleife.

In dieser Lektion wird ein Programm erstellt, mit dem der ROVER 2m geradeaus fährt. Trifft er auf ein Hindernis, leuchtet die RGB-LED auf dem ROVER rot und der ROVER hält an. Während der Fahrt leuchtet die LED grün. Die Entfernung zum Hindernis wird mit dem Ultraschall-Entfernungssensor RANGER gemessen. Mit `clear` kann das Programm vorzeitig beendet werden. Ist die Fahrt beendet, leuchtet die LED kurz blau.

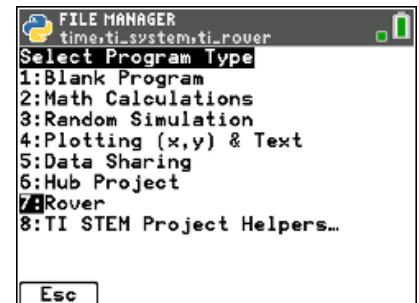


```

2m geradeaus fahren
  Solange wie die Bewegung nicht durch den Benutzer gestoppt wird
    a ← Entfernung zu einem Hindernis messen
    falls a < 0.2
      LED leuchtet rot und Rover hält an
    Andernfalls leuchtet die LED grün und der Rover fährt weiter
  Endet die Fahrt, so leuchtet die LED blau
  1s warten
  LED ausschalten

```

- Ein neues Programm mit dem Namen U6SB3 anlegen.
- Den Programm-Typ **7 : ti-rover** auswählen.
- Man erhält das untere Bild.





- Mit `disp_clr()` aus dem Menü `ti_system` den Bildschirm säubern.
- Ebenfalls im Menü `ti_system` findet man den Befehl `disp_cursor()`. Mit dem Argument 0 wird der Cursor unterdrückt.
- Der ROVER soll vorwärts fahren. Die zu fahrende Strecke ist in Einheiten zu **0,1m** anzugeben. Somit weist `rv.forward(20)` dem Roboter eine Vorwärtsbewegung über eine Distanz von **2 m** zu. Die Anweisung `rv.forward()` befindet sich im Menü **Modul**, dann **7 : ti_rover** und schließlich **2 : forward(distance)** im Menü **Drive**.
- Beim Starten der Bewegung leuchtet die LED grün.

```
EDITOR: U6SB3
PROGRAM LINE 0007
# Rover
from time import *
from ti_system import *
import ti_rover as rv
disp_clr()
disp_cursor(0)
_
```

Lehertipp : Mehrere Befehle aus dem Modul `ti_system` findet man auch in anderen Modulen.

```
EDITOR: U6SB3
PROGRAM LINE 0010
# Rover
from time import *
from ti_system import *
import ti_rover as rv
disp_clr()
disp_cursor(0)
disp_at(6, "Abbruch mit [clear]",
"center")
rv.forward(20)
rv.color_rgb(0,255,0)
_
```

- Für den Abbruch der Bewegung ist es sinnvoll, eine **While**-Schleife aus dem Modul `ti_system` in das Programm einzufügen.
- Anlegen einer Variablen `a`, in der die Entfernungsmessung des RANGERS abgespeichert wird. Hinter der Zeichenfolge `a=` wird die Anweisung `rv.ranger_measurement()` aus dem Menü **Modul**, dann **7 : ti_rover**, dann **I/O** (Eingänge/Ausgänge) und dann **1 : Inputs** und zum Schluss **1 : rv.ranger_measurement()** eingefügt. Die Masseinheit für die Entfernung ist m. Um eine sichere Messung der Entfernung zu gewährleisten, wird vorher noch eine kurze Pause von 0,1s eingefügt.
- Mit einer **If**-Anweisung wird das weitere Verhalten gesteuert. Ist die Entfernung kleiner als 20cm, wird der ROVER angehalten und die LED leuchtet rot. Die Anweisung `rv.color_rgb(,)` befindet sich im Modul `ti_rover` im Menü **I/O** unter **2 : Outputs**. Die Anweisung `rv.stop()` ist eine Fahr-Anweisung und befindet sich im Menü **Drive**. Andernfalls leuchtet die LED grün und der ROVER fährt die in `forward()` eingestellte Strecke. Die Anweisung `rv.resume()` beendet alle weiteren Aktionen der Warteschlange für die ROVER-Befehle.
- Ohne Hindernis hält der ROVER nach 2m an. Mit Drücken auf `[clear]` wird das Programm beendet durch die folgenden Anweisungen.
- Die letzten Anweisungen :
 - Den ROVER anhalten durch `rv.stop()` (bei ROVER-Programmen unbedingt einfügen, damit der ROVER nicht immer weiter fährt).
 - Der Bildschirm wird gelöscht.
 - Die LED leuchtet für 1s blau.
 - Die LED ausschalten, da sie sonst mit der zuletzt eingestellten Farbe weiter leuchtet.

```
EDITOR: U6SB3
PROGRAM LINE 0020
while not escape():
sleep(0.1)
a=rv.ranger_measurement()
if a<0.2:
rv.color_rgb(255,0,0)
rv.stop()
else:
rv.color_rgb(0,255,0)
rv.resume()
rv.stop()
disp_clr()
rv.color_rgb(0,0,255)
sleep(1)
rv.color_off()_
```