



#### Unit 5: Roverセンサ

#### Skill Builder 2: Off the Wall (壁を離れて)

このレッスンでは、TI-Innovator Roverの超音波Ranger(レンジャー)を使って障害物を回避します。

#### 目標

- Roverの前にある物体までの距離読み取り
- 距離が小さいとき、向きを変えて運転続行

Roverがドライブしているとき、プログラムは超音波Rangerを使って前方の障害物を検出し、Roverに割り込んで別のルートをとることができる。このプロジェクトでは、Roverは2つの向かい合った壁の間を行ったり来たりします。



1. Rover Codingテンプレートを使って新規のPythonプログラムを開始します。

次のコマンドを使って、Roverを動かします。

**rv.forward(100)**

それは10m(約39フィート)ですが、途中で障害物があります。

```

1.3 1.4 1.5 *Unit5 Pyt...ors RAD 10/26
# Rover Turns around
#-----
import ti_rover as rv
from math import *
import ti_plotlib as plt
from ti_system import *
from time import *
#-----
rv.forward(100)

```

2. Roverの前の距離を読みます。

**dist = rv.ranger\_measurement()**

続けられる限り、続けます。しかし、Roverが障害物に近づき過ぎるときは、向きを変えてください。「近づき過ぎる」とはどのくらいでしょう。それはあなたが決定してください。

```

1.3 1.4 1.5 *Unit5 Pyt...ors RAD 11/22
# Rover Turns around
#-----
import ti_rover as rv
from math import *
import ti_plotlib as plt
from ti_system import *
from time import *
#-----
rv.forward(100)
dist = rv.ranger_measurement()

```

3. whileループを追加して、障害物が遠くにある限り、距離の監視を続けます。

**while dist > ? :**

(疑問符(?)を小さい数字に置き換えます。)

**dist = rv.ranger\_measurement()**

今のところ、ループが終了したらRoverを停止します。

**rv.stop()**

rv.stop()は、menu > TI Rover > Drive(メニュー>TI Rover>ドライブ)にあります。

Roverを壁(40フィート(約12.19m)未満)に向けてプログラムをテストし、プログラムを実行します。Roverは壁にぶつかる前に停止する必要があります。

```

1.2 1.3 1.4 *Unit5 Pyt...ors RAD 16/16
from time import *
#-----
rv.forward(100)
dist = rv.ranger_measurement()
while dist > ? :
    dist = rv.ranger_measurement()
rv.stop()

```



4. Roverが障害物に遭遇すると、Roverは向きを変える必要があります。Roverが停止したら、次を使ってRoverを回転させます。

**rv.left(180)** または **rv.right(180)**

```

1.3 1.4 1.5 *Unit5 Pyt...ors RAD
*u5sb2.py 14/22
from ti_system import *
from time import *
#=====
rv.forward(100)
dist = rv.ranger_measurement()
while dist > ? :
    ++dist=rv.ranger_measurement()
rv.stop()
rv.left(180)

```

5. 向きを変えた後、キーが押されるまで反対方向(forward(前方))に進みます。Roverが動いていることを忘れないでください。そのため、**esc**を押すのは難しいかもしれません。次のステートメントを置きます。

**while get\_key() != "esc":**

これは、コードの先頭にあります。

ステートメントを編集して、任意のキーを使ってプログラムを停止できるようにします。

**while get\_key() == "":**

(==は等しいを表し、""は何もないことを表します。)

```

1.3 1.4 1.5 *Unit5 Pyt...ors RAD
*u5sb2.py 15/23
from ti_system import *
from time import *
#=====
while get_key() == "":
    ++rv.forward(100)
    ++dist = rv.ranger_measurement()
    ++while dist > ? :
        +++dist=rv.ranger_measurement()
    ++rv.stop()
    ++rv.left(180)

```

このwhileステートメントの下にあるすべてのステートメントをインデント(字下げ)して、**while block**にします。ステートメントのグループをインデントする最も簡単な方法は、**shift+下矢印**を使ってそれらのステートメントすべてを選択し、**tab**を押します。

6. プログラムを実行します。Roverは壁に近づき、立ち止まり、向きを変えます(十分なスペースがある場合)。待ってください...何かがおかしい。Roverが動けなくなりました!

向きを変えるのにRoverは1,2秒かかりますが、プログラムはすぐにループの先頭に戻り、Roverに前進するように指示してから、壁を再び検出して停止します。Roverは向きを変え、前進し、停止します。コマンドがほぼ同時に入ってくるので、Roverは圧倒されます。モーターが苦勞しているのが聞こえますが、Roverは動きません。すべてが速すぎて、何かが行われなければRoverは神経衰弱を起こします...まもなく!

```

1.3 1.4 1.5 *Unit5 Pyt...ors RAD
*u5sb2.py 15/23
from ti_system import *
from time import *
#=====
while get_key() == "":
    ++rv.forward(100)
    ++dist = rv.ranger_measurement()
    ++while dist > ? :
        +++dist=rv.ranger_measurement()
    ++rv.stop()
    ++rv.left(180)

```



7. 任意のキーを押してプログラムを終了します。

Roverはまだしばらく苦勞するかもしれません。その場合は電卓からRoverのプラグを抜き、電源をオフにしてから再接続し、再度オンにします。

プログラムの問題を修正するには、次のステートメントを追加して、Roverが向きを変え終わるまでTI-Nspire CX IIを待機させます。

### **rv.wait\_until\_done()**

このステートメントを**rv.left(180)**ステートメントのすぐ下に置きます。

プログラムを再び実行すると、Roverは2つの向かい合った壁の間を楽しく移動します。

Roverは向きを変えるのに約20cm必要です。Roverの後部は船首よりも長くなっています。**while dist > 0.2:** これを適切に処理します。

```
1.2 1.3 1.4 *Unit5 Pyt.ors RAD 17/17
*u5sb2.py
from time import *
#=====
while get_key()!="":
    rv.forward(100)
    dist = rv.ranger_measurement()
    while dist > 0.2 :
        dist=rv.ranger_measurement()
    rv.stop()
    rv.left(180)
    rv.wait_until_done()
```