



Unit 5: Roverセンサ

Application: The Winding Road(曲がりくねった道)

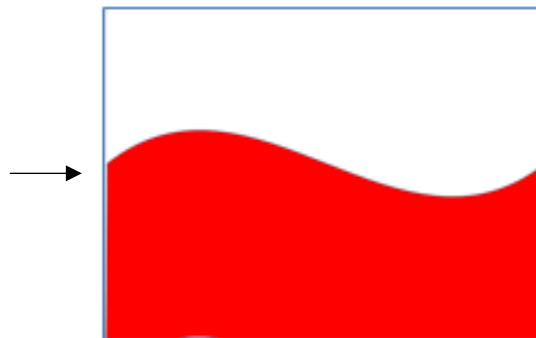
この応用では、Roverのカラーセンサを使って紙の上の曲がった経路をたどります。*この応用には、以下に示すような紙の曲がった経路が必要です。

目標

- カラーセンサを使って紙の曲がった経路を検出して追跡

Teacher Tip: 下の赤と白の曲線の経路は、Teacher Docsに含まれているUnit5 Color Test Pages.pdfに含まれています。

次のような曲がった経路をたどるようにRoverを軌道に乗せるプログラムを作成します。



Roverはページの左端から始まり、紙を横切る湾曲した経路をたどって右に移動します。

Roverが赤を「見る」と少し左に曲がり、少し前に進みます。Roverが白を「見る」と少し右に曲がり、少し前に進みます。

回転角度と移動距離を試して、Roverがさまざまな色にどのように反応するかを確認してください。

上の画像のようにページが赤と白の場合は`red_measurement()`を使って、用紙の各面にどのような値が与えられているかを確認できます。黒などの別の色を使う場合は、`gray_measurement()`(または`green_`、または`blue_`)を使えます。

Teacher Tip: 学生は自分のルートを作ることができます。経路に鋭い角があってはなりません。経路は、色の変化を検出しやすくする2つの非常に対照的な色によって決定されます。Roverには光センサが1つしかないため、テープのような狭い色のストリップを使うのはより困難です。したがって、Roverが左または右に行き過ぎているかどうかを判断できません。上に示したような対照的な経路で、Roverは軌道に乗るためにどちらの方向に曲がるかを知っています。

- これは、色測定関数が生成する値を決定するための、前のレッスンのオリジナルの短いtestプログラムです。

2つの異なる色の値の変動が最大になり、各色を個別に見たときに値の一貫性が最大になる色測定タイプ(`color_`, `red_`, `green_`, `blue_`, `gray_`)を選択します。

前のレッスンで行ったように、`c`の代わりに`color`などのより説明的な変数を使うこともできます。

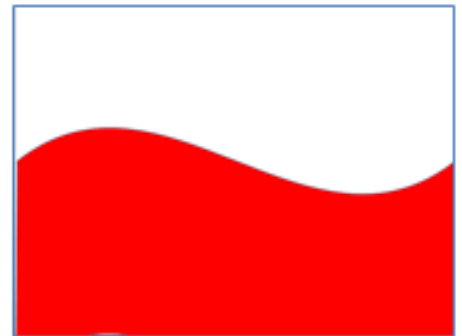
`color= rv.color_measurement()`

```

=====
import ti_rover as rv
from math import *
import ti_pltlib as plt
from ti_system import *
from time import *
=====
while get_key() != "esc":
    c = rv.color_measurement()
    plt.text_at(7, str(c), "left")
=====
    
```

- 経路追跡プログラム :

- 曲がった経路の近くのこの紙の左端または右端でRoverを開始
- 色を確認
- 色が赤の場合、少し白側に向ける
- それ以外の場合、少し赤い側に向ける
- 少し前に進む
- 経路の最後に到達するまで手順bから繰り返す



Teacher Tip: カラーセンサは、特定の色に対してつねに正確に同じ値を提供するとは限りません。そのため、このレッスンでは赤/白の曲線に`red_measurement()`を使うのが最適です。対照的な色を使い(白黒は適切に機能します)、色の値の大きな変化を探して、いつ回転するかを決定します。最初に色をテストし、2つの間のどのしきい値が環境に最適かを確認します。