©2020 Texas Instruments Incorporated

10 Minutes of Code - Python

TI-NSPIRE[™] CX II wITH THE TI-INNOVATOR[™] ROVER

Unit 5: Roverセンサ

このレッスンでは、Roverのカラーセンサを使って、色が検出 されたときに方向を変える方法を学びます。 このレッスンでは、色紙または大きな色の形状物が必要で す。

Roverの下部にはカラーセンサがあります。カラーセンサの下の床に光が 輝いているのが見えます。光はRoverがその下の色を見るのを助けます。 最初にtestプログラムを作成して、カラーセンサが生成する値の種類を確 認します。つぎに、さまざまな色に反応するプログラムを作成します。画 用紙のような色紙を用意するか、右図の長方形のような色の形を印刷する だけです。それらは、Roverが「見る」のに十分な大きさが必要です。

色測定が利用可能です(右図参照)。関数color_measurement()は、1から9 までの値を返します。ここで、
1=red(赤), 2=green(緑), 3=blue(青), 4=cyan(シアン), 5=magenta(マゼンタ),
6=yellow(黄色), 7=black(黒), 8=white(白), 9 = gray(灰色)
他の4つの測定値は、メニューに示されているように0~255の範囲で、示された色の量を返します。

menu > TI-Rover > Inputs(メニュー>TI-Rover>入力)では、5つの異なる

Teacher Tip: カラーセンサを使ういくつかのサンプルプロジェクトについては, Teacher Docsに含まれているPDFドキュメント「Unit5 Color Test Pages」を参照し てください。 さまざまな色で、学生は床に形を置いてルートを設計し、ある形から別の形にドライ ブすることができます。単純なルートは、頂点にスポットを置いて作成された正方形 です。各色は、プログラムに応じて異なるアクションを表すことができます。

 これは、Rover Codingテンプレートを使って、色測定関数が生成する 値を決定する短いtestプログラムです。

while get_key() != "esc":

c = rv.color_measurement()

plt.text_at(7, str(c), "left")

さまざまな色の表面で5つの色測定(color_, red_, green_, blue_, gray_)をすべて試し、表示される値を観察します。

1.4	1.5	1.6	•	*Doc		RAD 📘 🗙	
🔁 *u	5sb3	.py				12/14	
#====	====	====	====	=====		=====	
import ti_rover as rv							
from math import *							
import ti_plotlib as plt							
from ti_system import *							
from time import *							
#====	====	====	====		======		
while get_key() != "esc":							
<pre>**c = rv.color_measurement()</pre>							
<pre>• plt.text_at(7,str(c),"left")</pre>							

2	color_measurement()	1-9
3	red_measurement()	0-255
1	green_measurement()	0-255
5	blue_measurement()	0-255
5	grav measurement()	0-255

• カラーセンサで色を検出して反応

Skill Builder 3: 色を見つける

目標





UNIT 5: SKILL BUILDER 3 TEACHER NOTES

10 Minutes of Code - Python **TI-NSPIRE[™] CX II** WITH THE **TI-INNOVATOR[™] ROVER**

2. 床の色に応じて、Roverが床からの色の変化を「見る」ことができる ように、Roverの前にいくつかの色付きの紙(付箋やカラフルな画用紙 など)を置きます。最初にこれらのパッチ(一部分)をテストして、 Roverが「見る」ものを確認します。

Roverが色の違いに反応するようにするプログラムを作成します。右 図のサンプルでは、赤い境界線がRoverをページ上に維持する必要が あります。 Roverが赤を「見る」とき、振り返ります。

Roverが緑色を「見る」ときは、右または左に曲がります。

プログラムにステートメントを追加して、ライトまたはカラーLEDを 制御することもできます。

3. 開始したカラーtestプログラムを編集します。ループの先頭に次のス テートメントを追加して、Roverの前進を開始します。

rv.forward(10)

つぎに、Roverの下の色を監視します。変数cをcolorに変更すると、 プログラムが読みやすくなります。

color = rv.color_measurement()

色が赤の場合はRoverを180°回転させ、色が緑の場合はRoverを90°回 転させます。これには2つのifステートメントが必要です。

plt.text_at(7, str(c), "left")を削除します。

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、コントラストの高い色を使い(白黒が適切に機能します)、 red_, green_, blue_, またはgray_ measurements()を使って色の値の大きな変化を 探し、いつ方向転換するかを決定するのが最適です。

4. 2つのifステートメントを追加し、?の代わりにcolorサンプルのテスト から取得した値を使います。

if color == ?:

block

if color == ?:

block

(ifステートメントのプレースホルダー(仮に確保した場所)blockは表 示されません。右図のredとgreenという単語は単なるプレースホルダ 一です。)

> Teacher Tip: ステートメントの使用 red=1 areen=2 プログラムの最初にあるということは, ifステートメントで変数redとgreenを使える ことを意味します。If color== red:





UNIT 5: SKILL BUILDER 3 TEACHER NOTES

4 10 Minutes of Code - Python

TI-NSPIRE[™] CX II wITH THE TI-INNOVATOR[™] ROVER

5. 2つのifブロックはよく似ています。赤ー右(red-right)と緑ー左(greenleft)を覚えておいてください。

各blockは次のようになります。

- a) Roverを停止
- b) 180°回転
- c) 少し前に進み、色付きのスポットから離れる
- d) Roverがこれらの3つのタスクを完了するまで待つようにTI-Nspire CX IIに指示

先に進む前に自分で試してみましょう。

6. ifブロックはこれに似ていますか。

if color == 1: (赤。色が赤でないとき,別の数字を使って ください。) rv.stop() rv.right(180)

rv.forward(1) rv.wait_until_done()

Tip: プログラムの先頭で変数red = 1を割り当てると, ifステートメン トにif color == red:と書くことができます。これにより, 意図が明確 になります。

UNIT 5: SKILL BUILDER 3 TEACHER NOTES



