



Unit 4: ドライブ機能

Skill Builder 2: コーナーの光

このレッスンでは、Roverの2つの興味深い機能について学習します。カラーLEDと、Roverが次のタスクの準備ができるのを待つことです。

目標

- RoverのカラーLED制御
- 新規タスクを開始前に、Roverが現在のタスクを完了するのを待つ
- 適切なタイミングでLED点灯

バッテリー強度メーター(4つの緑色のライト)の近くのRoverの上に、TI-Innovator™ Hubと同じようにカラーLEDがあります。ただし、TI-Innovator HubのLEDは表示されません。このLEDはTI-Innovator Hub LEDのように機能します。形式(赤, 緑, 青)の値が必要です。

新規のRover Codingテンプレート指定のプログラムで、次のコマンドを試してください(右図参照)。

menu > TI Rover > Outputs > color_rgb(r, g, b)

(メニュー>TI Rover>出力>color_rgb(r, g, b))

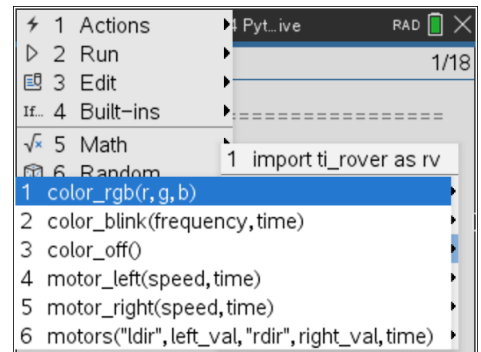
これは、プログラムに次のように貼り付けられます

rv.color_rgb(red, green, blue).

3つのカラーチャンネル(それぞれ0~255)の値を指定し、プログラムを実行して、LEDが指定の色で点灯することを確認しましょう。

Teacher Tip: Roverには他に2つの付属部品があります。前面の超音波Rangerと下部のカラーセンサです。

1. 前レッスンでは、forループを使ってRoverドライブを正方形のパターンで作成しました。このレッスンでは、頂点でのみカラーLEDを赤く点灯させます。



2. ユニット4スキルビルダー1からsquareプログラムのコピーを作成します。

menu > Actions > Create copy...

(メニュー>アクション>コピー作成...)

プログラムのturnステートメント(左または右)の直前で、LEDを赤に変えます。turnステートメントの後、LEDをオフにします。

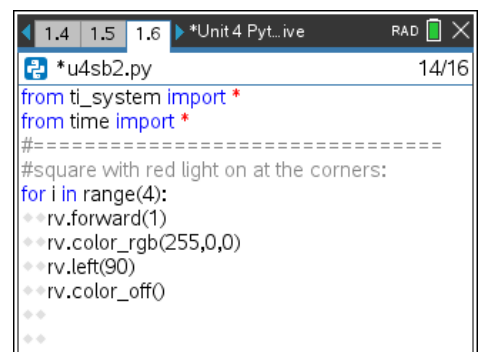
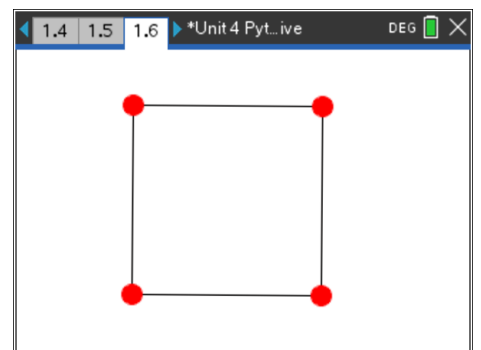
rv.color_rgb(255,0,0)

rv.left(90)

rv.color_off()

2つのカラーステートメントは、**menu > TI Rover > Outputs**(メニュー>TI Rover>出力)にあります。

プログラムを実行します。期待どおり動きますか。



Teacher Tip: プログラムはRoverがドライブコマンドを処理できるよりも速く実行するため、LEDはほんの一瞬だけ点灯します。これは、ドライブコマンドがTI-Innovator Hub内のdrive queue(ドライブキュー)に保存されているため、1つずつ処理するためです。ただし、LEDコマンドはキューコマンドの一部ではありません。そのため、すべてのコマンドがTI-Innovator Hubに非常に迅速に送信されます。TI-Innovator Hubは、コマンドのリストをドライブコマンド(queued(キューに入れられた))と他のコマンド(not queued(キューに入れられていない))に分割します。

このレッスンの残りの部分では、キューに入れられていないコマンドをdrive queueと同期する方法について説明します。

- TI-Nspire CX IIIはRoverよりも高速に動作しているため、この試行は正しく機能しません。TI-Nspire CX IIIは、すべての命令をTI-Innovator Hubにできるだけ速く送信します。つぎに、TI-Innovator Hubは運転指示を保存し、マッピングアプリのルートをとどるドライバとして一度に1つずつ処理します。ドライブコマンドはキュー(リスト)に保存され、各命令が完了するまでに時間がかかるため、一度に1つずつ処理されます。

ただし、`rv.color`コマンドはドライブコマンドではありません。TI-Innovator Hubがこの命令を受信すると、すぐに処理します(ドライブコマンドとは関係ありません)。そのため、運転開始直後にLEDが4回速く点滅します。

Teacher Tip: キュー(queue)の概念は、学生にとっては新しいかもしれません。スマートフォンやGPSデバイスのmaps(マップ)アプリでは、ルートはキューに保存され、ドライバーがルートの次の重要なポイント(ウェイポイント)に到達すると、指示が処理されます。

ドライブ命令のみがドライブキューに保存され、他の命令はキューとは独立して処理されるため、ドライブコマンドと同期していません。`rv.wait_until_done()`は、ドライブコマンドと他のコマンドの同期に役立ちます。

Note: キュー(queue)とは、データ構造の1つで、要素が入ってきた順に1列に並び、先に入れた要素から順に取り出すという規則で出し入れするものです。

- 幸いなことに、Roverが方向転換する準備完了まで待つようにプログラムに指示するステートメントがあります。`rv.forward()`ステートメントの後に、次のステートメントを追加します。

`rv.wait_until_done()`

これは、`menu > TI Rover > Commands`(メニュー>TI Rover>コマンド)にあります。

この命令は、Roverが`rv.forward()`コマンドで終了したという信号をRoverから受信するまで待機するようにTI-Nspire CX IIIに指示します。その後、赤色LEDが点灯し、Roverが回転します。

さあプログラムをテストしてみましょう。

Drive Queueとその他

	A	B	C	D
1	drive	other		
2	forward 1	LED red		
3	left 90	LED off		
4	forward 1	LED red		
5	left 90	LED off		

```

#square with red light on at the corners:
for i in range(4):
    rv.forward(1)
    rv.wait_until_done()
    rv.color_rgb(255,0,0)
    rv.left()
    
```

5. ターンの開始時、LEDがすばやく点滅することに注意します。回転も完了するまでに時間がかかるため、TI-Nspire CX IIに、回転が行われている間、LEDをオフにする前に再度待機するように指示する必要があります。

`rv.left()`ステートメントの後、LEDがオフになる前に、別の `rv.wait_until_done()`ステートメントを追加して、ターン全体を通してLEDがオンのままになるようにします。

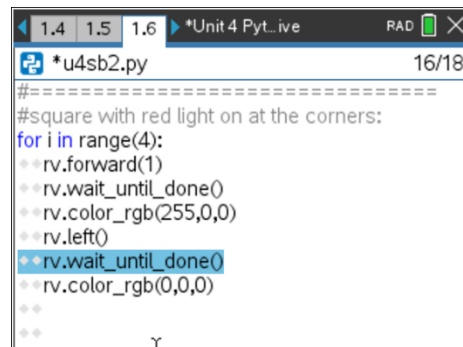
プログラムを再実行してください。

正方形の辺に沿ってLEDを別の色で点灯させることはできますか。

Hint: あと1つだけステートメントが必要です。

Teacher Tip: この `wait_until_done()`関数は、TI-Basicでは使えません。

正方形の辺に沿って異なる色でLEDを点灯させるには、追加の `rv.color()`ステートメントを `rv.forward()`ステートメントの直前に置きます。



```

1.4 1.5 1.6 *Unit4 Pyt_ive RAD 16/18
*u4sb2.py
=====
#square with red light on at the corners:
for i in range(4):
    rv.forward(1)
    rv.wait_until_done()
    rv.color_rgb(255,0,0)
    rv.left()
    rv.wait_until_done()
    rv.color_rgb(0,0,0)
    **
    **
    **
    
```