



Unit 7: The TI-RGB Array

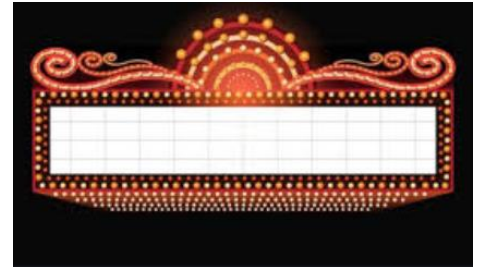
Skill Builder 3: 順番に

このレッスンでは、ループ内で2つのLEDを同時に制御して、マーキー効果(劇場などの入り口の上に取り付けられる看板をマーキーといい、文字をスクロールするとき使います)を作成する方法を学習します。

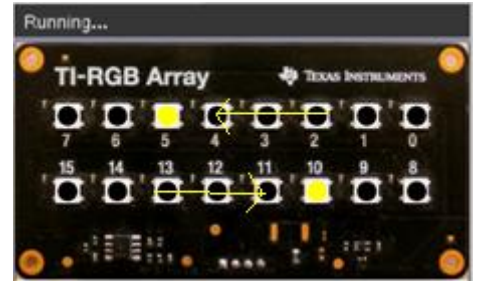
目標

- forループを使って単一の変化するLEDを点灯
- 数式を使って同時に別のLEDを制御

映画館の入り口にある頭上の看板には、蟻が行進しているように順番に点滅するライトの線があります。ライトを順番にオンまたはオフにすることによりTI-RGB Arrayに同様の効果を作成できます。



このレッスンのプログラムでは、一度に2つのLEDが点灯します。1つは上段が右から左(0から7)に、もう1つは下段が左から右(15から8)に点灯します。上段の行の数列と下段の行の数列の関係は何でしょう。



(demo3.1.gif)

1. 新規のPython Hub Projectを開始します。

`rgb_array()`コンストラクターを使って変数を作成し、前のレッスンのように`esc`キーループを使います。

変数`cb`を再び使いますが、独自の変数を自由に選択できます。

次のステートメントを追加します。

```
while get_key() != "esc":
    block
```

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

```
1.4 1.5 1.6 ▶ *Unit 7 Py...ray RAD [X]
* u7sb3.py 1/15
# Unit 7 SB3 - Sequencing the LEDs
#=====
from ti_hub import *
from math import *
from time import sleep
from ti_plottlib import text_at,cls
from ti_system import get_key
#=====
cb=rgb_array()
while get_key() != "esc":
    block
```

2. `for`ループを使って、上段の行を右から左(0から7)へ順番に点灯させます。

```
for top in range (8):
    cb.set(top, 255,255,0) (これは黄色)
```

変数`top`を使うのは、これが上段の行を制御するためです。

`range(8)`は0から7までの数値を処理することに注意します。

プログラムをテストしてみましょう。

```
1.4 1.5 1.6 ▶ *Unit 7 Py...ray RAD [X]
* u7sb3.py 12/20
#=====
from ti_hub import *
from math import *
from time import sleep
from ti_plottlib import text_at,cls
from ti_system import get_key
#=====
cb=rgb_array()
while get_key()!="esc":
    for top in range(8):
        cb.set(top,255,255,0)
```

Teacher Tip: この時点でLEDは点灯しますが、前のLEDは消灯しません。それは後で起こります。

3. 8つのLEDがすべて非常に速く点灯し、プログラムの最後に上段の行のLEDがすべてオンになります。

つぎに、下段の行を扱います(変数はbottomとします)。下段の行は15から8に移動する必要があります。下段と上段の関係式はどうなるでしょう。

bottom = ???
cb.set(bottom, 255, 255, 0)

```

1.4 1.5 1.6 *Unit 7 Py...ray RAD 14/20
* u7sb3.py
from math import *
from time import sleep
from tiplotlib import text_at,cls
from ti_system import get_key
#=====
cb=rgb_array()
while get_key()!="esc":
    for top in range(8):
        cb.set(top,255,255,0)
        bottom = ???
        cb.set(bottom,255,255,0)
    
```

Teacher Tip: bottom = 15 – top.

4. プログラムを実行します。16個のLEDすべてが非常に速く点灯します。2つのステートメントを追加します。1つは速度を落とすための **sleep()**ステートメントで、もう1つはループブロックの下部にあるすべてのLEDをオフにするためのステートメントです。

sleep(.25)
cb.all_off()

```

1.4 1.5 1.6 *Unit 7 Py...ray RAD 16/20
* u7sb3.py
from tiplotlib import text_at,cls
from ti_system import get_key
#=====
cb=rgb_array()
while get_key()!="esc":
    for top in range(8):
        cb.set(top,255,255,0)
        bottom = ???
        cb.set(bottom,255,255,0)
        sleep(.25)
        cb.all_off()
    
```

5. プログラムを再度、試行します。 **sleep()**値を調整し、おそらく **all_off()**の後に別の**sleep()**を追加します。



(demo 3.1.gif)

Teacher Tip: すべてのLEDがオフになっていますが、ループの次の反復で1つがオンになるため、オフ状態が長時間表示されることはありません。さらに興味深い課題は、2つ以上の連続したLEDを一度に点灯させることです。