10 Minutes of Code - Python

TI-NSPIRE[™] CX II WITH THE TI-INNOVATOR[™] HUB AND TI-RGB ARRAY[™]

| ©2020 Texas Instruments Incorporated | |
|--------------------------------------|--|
|--------------------------------------|--|

| Unit 7. The TFROD Array | |
|-------------------------------------|---|
| この応用では、TI-Innovator Hubの明るさセンサを使って、 | 目標 |
| TI-RGB Arrayで点灯するLEDの数を制御します。 | • 明るさセンサを使ってTI-RGB Arrayを制御 |
| | TI-RGB Arrayに合わせて明るさ範囲を調整 |
| | 16個のLEDすべてが明るさの影響を受ける ことの確認 |

スマートライト

部屋が暗くなると、部屋の明かりが明るくなります。電灯のスイッチがないス マートホームを想像してみましょう。明るさをモニター(監視)し、必要に応じ て多かれ少なかれLEDをオンにするプログラムを作成します。

> Teacher Tip: このプロジェクトの課題の1つは、すべてのLEDを終了条件(すべてオン およびすべてオフ)を含む、ある明るさの値に反応させることです。照明条件に合わ せて、brightness.range()の値を調整する必要がある場合があります。スマートフォ ンの懐中電灯は明るさを制御するのに役立ちます。

1. いつものように, rgb_array()コンストラクターとwhileループを使って Python Hub Projectを開始し, esc でプログラムを終了します。

> cb = rgb_array() while get_key() != "esc":

2. whileループの前に, brightness.range()を, 点灯する可能性のあるTI-RGB Arrayボード上のLEDの数(0~16)と一致するように設定します。

menu > TI Hub > Hub Built-in Devices > Brightness Input > range(min,max)(メニュー>TI Hub>Hub内蔵デバイス>明るさ入力>範囲(最 小,最大))を押して、次のステートメントを選択します。

brightness.range(0,16)

0,16を使います。これはボード上で点灯するLEDの数の範囲なためです。 センサが生成する最大値は16です。最小値は0ですか。

> Teacher Tip: センサが値0を達成するのは困難です。次のステップでは、明るさの値 を整数に変換します。



Running.

TI-RGB Array

.... bright:7 lites :9

-



-U

10 Minutes of Code - Python

TI-NSPIRE[™] CX II WITH THE TI-INNOVATOR[™] HUB AND TI-RGB ARRAY[™]

 whileブロックで, brightness.measurement()を読み取ることから始め, 値を変数(bright)に格納します。

bright = brightness.measurement()

この関数は、浮動小数点数(float, decimal)を生成します。次を使って整数値 に変換します。

bright = int(bright)

または、2つのステートメントを1つの操作に結合します。

bright = int(brightness.measurement())

Teacher Tip: 明確さまたは効率? 好みと経験の問題です!

プログラムをテストするには、text_at()ステートメントを追加します。これは、menu > TI Hub > Commands(メニュー>TI Hub>コマンド)にあります。

text_at(7, str(bright), "left")

text_at()関数は表示に文字列を必要とするため, str(bright)が必要である ことを思い出してください。str()と入力するか, menu > Built-ins > Type(メニュー>組み込み>タイプ)から取得できます。

プログラムを実行して、17個の値(0~16)がすべて表示されることを確認し ます。そうでない場合はrange()を調整します。懐中電灯やスマートフォン の懐中電灯機能などの人工光源を使ってみてください。

Teacher Tip: 照明条件を変えると完璧を達成するのが難しくなることがありますが、練習すれば学生は物事を理解します。

0が最も暗い値で16が最も明るいので、点灯するLEDの数を逆にします。
 Bright=0の場合、6個のLEDが点灯し、bright=16の場合は0個のLEDが点灯します。

bright.(明るい)という観点からlites(ライト)の表現を書きます。

lites = ? ? ?

6. すべてのLEDがオフになっている可能性があります。

if lites == 0: cb.all_off() else:



| 1.8 1.9 2.1 ▶ *Unit 7 Pyray | CAPS RAD 🚺 🗙 | | | | |
|---|--------------|--|--|--|--|
| 🛃 u7app.py | 15/32 | | | | |
| from ti_plotlib import text_at,cls | | | | | |
| from ti_system import get_key | | | | | |
| #====================================== | | | | | |
| cb=rgb_array() | | | | | |
| brightness.range(0,16) | | | | | |
| while get_key() != "esc": | | | | | |
| bright=int(bright) | | | | | |
| <pre>structure at(7 str(bright) "left")</pre> | | | | | |
| •• | | | | | |
| | | | | | |

| 1.8 1.9 2.1 ▶ *Unit 7 Pyray | RAD 📘 | × |
|--|-------|----|
| 🛃 *u7app.py | 18/ | 29 |
| #====================================== | | |
| cb=rgb_array() brightness.range(0,16) while get_key() != "esc": ** * bright=brightness.measurement() | | |
| <pre>bright=int(bright) text_at(7,str(bright),"left")</pre> | | |
| ** | | |
| ••lites= ? ? | | |

| 1.8 | 1.9 | 2.1 | ▶ *Unit 7 Pyray | RAD 📋 🗙 |
|---------------------|---------------------------|---------------------------|---|---------|
| 🔁 *u | 7app | .py | | 22/29 |
| while | get_k | ey() | != "esc": | |
| brig brig tex | ght=b ght=ir t_at(7 | right ht(bri 7,str(| ness.measuremen ght) (bright),"left") | tO |
| • • lites | s= ? ' | ? | | |
| ** | | - | | |
| e if lit | es== | 0: | 、 、 | |
| els | e: | _0П(|) | |

UNIT 7: APPLICATION TEACHER NOTES

10 Minutes of Code - Python

TI-NSPIRE[™] CX II WITH THE TI-INNOVATOR[™] HUB AND TI-RGB ARRAY[™]

すべてのLEDが明るさの影響を受けるようにしたいので、forループを使って毎回すべてのLEDの状態を制御します。lites変数はLEDをオンまたはオフにするときの決定要因です。

for i in range(1,17):

(値17はループによって処理されないため、16個のLEDを表す1から16までの値を使うことに注意します。)

8. **if...else...**ステートメントを追加してプログラムを完了し, TI-Innovator HubにどのLEDがオンでどのLEDがオフであるかを通知します。

ヒント:litesが1の場合,LED 0をオンにします。litesが16の場合,すべてのLED(#0から#15)をオンにします。色(255,255,255)を使って,明るい白 色光を取得します。

プログラムの最後にすべて(all)のLEDをoffにすることを忘れないでください。

Teacher Tip: 完全なプログラム

```
cb = rgb_arry()
brightness.range (0,16)
while get_key() !="esc":
    bright = brightness.measurement()
    bright = int(bright)
    text_at (7,str(bright),"left")
    lites = 16 - bright
    if lites == 0:
        cb.all_off()
    else
        for i in range (1.17) :
            if i <= lites:
               cb.set(i-1,255,255,255)
            else:
               cb.set(i-1,0,0,0)
</pre>
```

cb.all_off()

Tip: 電源をTI-Innovator Hubに接続し, cb==rgb_array(ランプとして)を使ってLEDを 大幅に明るくします。

より大きな課題の場合:各LEDが点灯しているかどうかに関係なく,必要に応じてプログラムで次のLEDを徐々に明るくしたり暗くしたりします。brightとlitesにはフロートを使い,小数部を使ってLEDを部分的に点灯させます。

UNIT 7: APPLICATION TEACHER NOTES





(demoAPP.gif)