



#### Unit 7: The TI-RGB Array

#### Application: Smart Lights

In this application, you will control the number of LEDs lit on the TI-RGB Array using the TI-Innovator Hub's brightness sensor.

#### Objectives:

- Use the brightness sensor to control the TI-RGB Array
- Adjust the brightness range to suit the TI-RGB Array
- Make sure that all 16 LEDs are impacted by the brightness

#### Smart Lights

As the room darkens, the lights in the room get brighter. Imagine a 'smart home' with no light switches! Write a program that monitors the brightness and turns on more or less LEDs, as necessary.

1. As usual, begin this Python Hub Project using the **rgb\_array()** constructor and the **while** loop to terminate the program with **esc**.

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

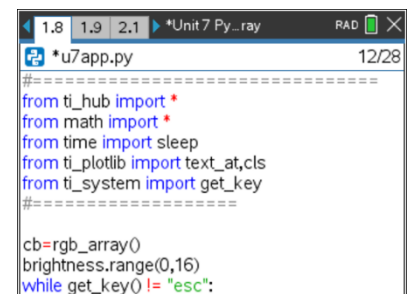
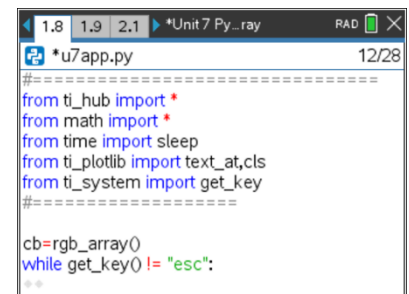
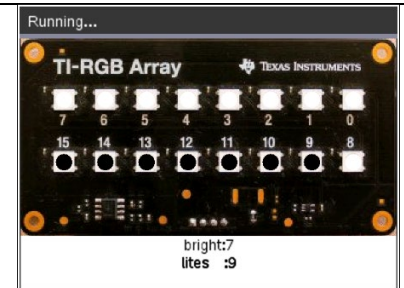
2. Before the **while** loop, set the **brightness.range()** to match the number of LEDs on the TI-RGB Array board that could be lit: 0 to 16.

Press **menu > TI Hub > Hub Built-in Devices > Brightness Input > range(min,max)** for the statement:

```
brightness.range(0,16)
```

Use **0,16** because this is the range of the number of LEDs to light up on the board.

The maximum value the sensor will produce is 16. Is the minimum 0?



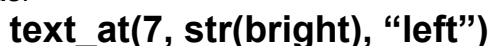


## UNIT 7: APPLICATION

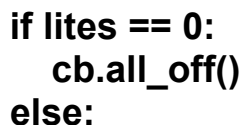
```
bright = brightness.measurement()
```

```
bright = int(bright)
```

```
bright = int(brightness.measurement())
```



**lites** = ? ? ?





## 10 Minutes of Code - Python

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

7. We want *all* the LEDs to be affected by the brightness so we will use a **for** loop to control the state of every LED every time. The **lites** variable is a deciding factor when turning a LED on or off:

**for i in range(1,17):**

*(Remember that the value 17 is not processed by the loop so i takes on the values from 1 to 16 representing the 16 LEDs.)*

8. Complete the program by adding an **if...else...** statement to tell the TI-Innovator Hub which LEDs are on and which ones are off.

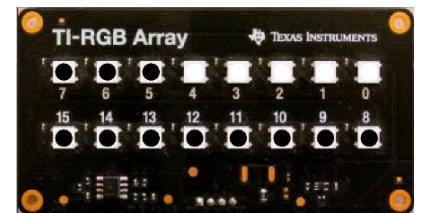
Hint: If **lites** is 1, then you want to turn on LED 0. When **lites** is 16, you want to turn on all LEDs (#0 to #15). Use the color (255,255,255) to get a bright white light.

Remember to turn **all** the LEDs **off** at the end of the program.

## UNIT 7: APPLICATION

### STUDENT ACTIVITY

```
*u7app.py
bright=brightness.measurement()
bright=int(bright)
text_at(7,str(bright),'left')
lites= ??
if lites==0:
    cb.all_off()
else:
    for i in range(1,17):
```



bright:11

(demoAPP.gif)