Unit 3: Brightness, if and while with the TI-Innovator™ Hub

Skill Builder 3: Brightness and Color

In this lesson you will use the brightness sensor to control the color LED.

Objectives:

- Use **brightness.range()** to change the brightness scale
- Use the brightness value to light up the color LED
- Investigate numeric transformations

Unlike the TI-Innovator Hub light (the red LED), the color LED can vary in brightness. You will now use the brightness sensor to control that LED.

1. Make another copy of your brightness metering program We copied BRIGHTA to BRIGHTC. Add **import color** at the top of your code found on **[math] ti_hub…**.

   Since the three color channels of the color LED only permit values from 0 to 255, set the brightness range from 0 to 255.

   brightness.range(0, 255)

   The brightness value **b** can now be used as values for the three color channels.

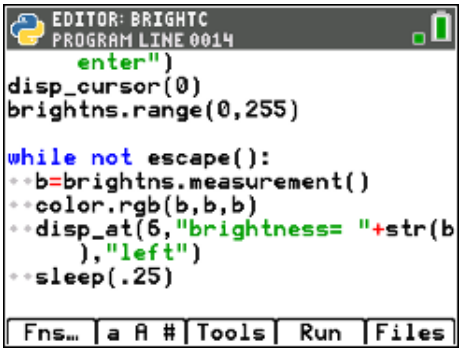
2. Add a statement after the **brightness.measurement()** statement to light up the color LED using the variable **b** for all three channels:

   color.rgb(b,b,b)

   from **[math] Color…**

   Use the variable **b** for all three color channels.

   <Run> your program.



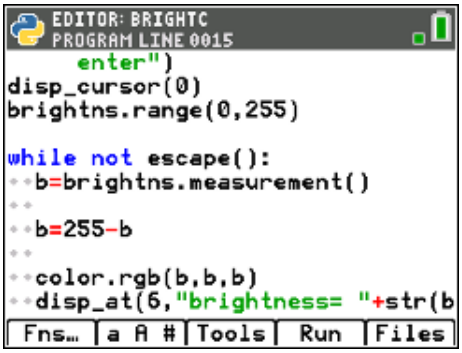
3. Notice that the LED gets *brighter* as the light level increases. This is backwards! The darker the room, the brighter the light should be. Change the variable **b** after the **brightns.measurement()** to make the LED bright for low values and dim for large values. (See **b=???** in the screen to the right.)

Try it yourself before proceeding to the next step.

4. Here's an expression that works:

b = 255 - b

When **b** is 0, the expression **255-b** is 255; when **b** is 255, the expression **255-b** is 0. The effect of the statement is to 'reverse' the values of **b.**



You may have to move (cut/paste) your **disp_at()** statement in the program to display the original value of **b** and not the transformed value or have two **disp_at()** statements to show both values.

Can you modify the program to produce other colors besides white?

When the program ends, the color LED may remain on. Add a statement at the end of the loop (not indented) to turn the color LED off.

Teacher Tip: New to programming? Statements like **b = 255 - b** are *mathematically* incorrect (not an 'equation') but are very useful in programming. The expression on the right is evaluated first, then the result of the calculation is assigned to the variable on the left. The value of b on the right is *different* from the value on the left. The equals sign (=) is not a statement of *equality* but rather a statement of *assignment*.

One way of using different colors is to only use two channels. Red and green make yellow: color.rgb(b, b, 0). Another is to use different expressions for each channel such as: to change between red and green using **color.rgb(b, 255-b, 0)**.

Just be carefule to not go outside the range 0..255 for each channel. One option for handling this is to use %256 (the 'mod' operator) in your expressions.