

Unit 7: The TI-RGB Array

Skill Builder 1: Light Them Up

In this lesson, you will learn to control the 16 LEDs on the TI-RGB Array both collectively (all at once) and individually (one at a time).

Note: The simulation that is used in this unit to replicate the lights on the TI-RGB Array is used for demonstration purposes only. This tool is not a TI product and is not available for purchase or distribution by TI.

*Caution: Rapidly flashing lights may be disturbing for some students, so it is wise to use **sleep()** statements to slow things down a bit.*

Objectives:

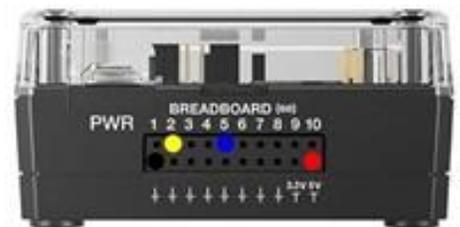
- Light up ALL LEDs and make them blink in unison using a loop
- Use another loop to light up and turn off the LEDs one at a time



TI-RGB Array



back



breadboard ports

The TI-RGB Array is a circuit board with 16 color LEDs and a controller chip and comes with a short 4-wire cable. It connects to the TI-Innovator Hub using the breadboard (BB) ports on the TI-Innovator Hub. Follow the wiring instructions on the back of the circuit board to connect it to the TI-Innovator Hub; connect the TI-Innovator Hub to your TI-84.

1. Start a new Python program using the Hub Project template.

Press **[math] ti_hub... Output devices...** and select the **TI-RGB Array**. This pastes **from rgb_arr import *** into your code.

Again, press **[math]** and select the new **TI-RGB Array...** menu item at the bottom of the list. Select the first option **var = rgb_array()**.

Then, in front of the = sign, type any variable name. We use **r**.

*This is Object-Oriented Programming at work: The variable (**r**) is an instance (object) of the **rgb_array** class. You will use methods of the class in the rest of these lessons.*

```

EDITOR: RGBA
PROGRAM LINE 0006
# Hub Project
from ti_system import *
from time import *
from rgb_arr import *
r=rgb_array()
    
```



- To get the lights to light up all at once, on the next line of your program type your variable name followed by a period or decimal point.

r

Press **[math] TI-RGB Array...** and select **var.set_all(r, g, b)** so that the method is appended to the variable as shown.

```
EDITOR: RGBA
PROGRAM LINE 0006
# Hub Project
from ti_system import *
from time import *
from rgb_arr import *
r=rgb_array()
r.set_all(.,.)
```

- Enter the three arguments, each one a value from 0 to 255, to control the color of the LEDs on the RGB Array.

```
EDITOR: RGBA
PROGRAM LINE 0006
# Hub Project
from ti_system import *
from time import *
from rgb_arr import *
r=rgb_array()
r.set_all(255,0,0)
```

- <Run>** your program to see all the LEDs light up in your selected color. Notice that the LEDs remain lit even after the program ends.



- To turn all the LEDs off at once use the statement

r.all_off()

found on the **[math] TI-RGB Array** menu.

Add a **sleep()** statement to keep the lights on for a few seconds otherwise you won't see anything happen.

When you **<Run>** the program now you should see the LED come on and then turn off.

```
EDITOR: RGBA
PROGRAM LINE 0009
# Hub Project
from ti_system import *
from time import *
from rgb_arr import *
r=rgb_array()
r.set_all(255,0,0)
sleep(2)
r.all_off()
-
```



- Put your LED control statements into a **for** loop to make them blink on and off several times. Add another **sleep()** after they are turned off. You may want to adjust the sleep times to speed things up a bit. Be sure to indent all statements in the loop block using **<Tools> Indent ▶** on each line.

Note: *Do not* include the *constructor* statement `r = rgb_array()` in the loop block. It only needs to be defined once!

<Run> your program to test it before you continue.

- If your program blinks all LEDs at once, then you are successful. Now let's control the LEDs one at a time with an inner loop.

```

EDITOR: RGBA
PROGRAM LINE 0006
# Hub Project
from ti_system import *
from time import *
from rgb_arr import *
r=rgb_array()

for i in range(10):
    r.set_all(255,0,0)
    sleep(2)
    r.all_off()
    sleep(1)
  
```



(demo1.1.gif)

- Below the **for i...** loop, add an *inner* loop: **for j in range(16):**
Be sure to change the loop variable from *i* to *j*.
Indent all four loop block statements so that they now apply to the *inner* loop.

Change the statement `r.set_all(r, g, b)` to:

`r.set(j, r, g, b)`

You can *either* erase the `.set_all()` statement and select the `.set()` statement from the menu *or* just edit the statement. `r.set(, , ,)` takes *four* arguments: the LED position and the three color values. Use the inner loop variable *j* as the `led_position` and enter your three color values.

Change the **sleep** values and (optionally) the outer loop **range()** argument to speed things up a bit.

- <Run>** your program. Now your 16 LEDs light up one at a time several times.

```

EDITOR: RGBA
PROGRAM LINE 0012
from ti_system import *
from time import *
from rgb_arr import *
r=rgb_array()
for i in range(10):
    for j in range(16):
        r.set(j,255,0,0)
        sleep(1)
        r.all_off()
        sleep(.1)
  
```



(demo1.2.gif)