



Unit 7: The TI-RGB Array

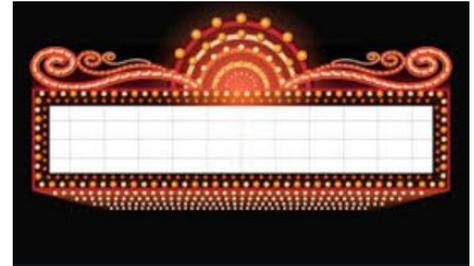
Skill Builder 3: Sequencing

In this lesson, you will learn to control two LEDs at once in a loop to create a 'marquee' effect.

Objectives:

- Use a **for** loop to light up a single varying LED.
- Use a mathematics expression to control another LED at the same time.

The overhead sign outside of some movie theatres has a border of lights flashing in sequence like ants marching. You can create a similar effect on the TI-RGB Array by turning on and off lights *in sequence*.



Your program in this lesson will light up two LEDs at a time, one on the top row going from right-to-left (0 to 7) and one on the bottom row going from left-to right (15 to 8). What is the relationship between the top row sequence and the bottom row sequence?



1. Begin a new Python Hub Project and import the TI-RGB Array module.

Make a variable using the `var=rgb_array()` constructor. We use the variable `r` again, but you are free to choose your own variable.

Add the statement:

while not escape():

◆◆

found at `[math] ti_system...`





- Use a **for** loop to light up the top row in sequence from right to left (0 to 7).

```
◆◆ for t in range(8):
    ◆◆◆◆ r.set(t, 255,255,0) (This is yellow.)
```

Use the variable **t** because this controls the top row.
Remember that **range(8)** processes the numbers from 0 to 7.
Test your program now.

```
EDITOR: RGBC
PROGRAM LINE 0008
# Hub Project
from ti_system import *
from time import *
from rgb_arr import *
r=rgb_array()
while not escape():
    for t in range(8):
        r.set(t,255,255,0)_
```

Teacher Tip: At this point, the LEDs light up but the previous ones do not turn off. That's coming up later.

- All 8 LEDs light up very quickly and, at the end of the program, all the top row LEDs are on.
Next deal with the **bottom** row. The bottom row must go from 15 to 8.
What is the relationship between bottom (**b**) and top (**t**)?

```
◆◆◆◆ b = ???
◆◆◆◆ r.set(b, 255, 255, 0)
```

```
EDITOR: RGBC
PROGRAM LINE 0011
# Hub Project
from ti_system import *
from time import *
from rgb_arr import *
r=rgb_array()
while not escape():
    for t in range(8):
        r.set(t,255,255,0)
        b=???
        r.set(b,255,255,0)
    _
```

Teacher Tip: bottom = 15 – top.

- Did you write **b = 15 – t** ?
<Run> the program now. All 16 LEDs light up very quickly.
Add two statements: a **sleep()** statement to slow things down and a statement to turn **all** LEDs off at the bottom of the loop block.

```
◆◆◆◆ sleep(.25)
◆◆◆◆ r.all_off()
```

```
EDITOR: RGBC
PROGRAM LINE 0009
from ti_system import *
from time import *
from rgb_arr import *
r=rgb_array()
while not escape():
    for t in range(8):
        r.set(t,255,255,0)
        b=15-t
        r.set(b,255,255,0)
        sleep(.25)
        r.all_off()
```

- Try your program again. Adjust the **sleep()** value and perhaps add another **sleep()** after **r.all_off()**.



Teacher Tip: All the LEDs are turned off, but the next iteration of the loop turns one on, so you do not see the 'off' state for very long. A more interesting challenge is to have two or more *consecutive* LEDs lit at one time.