



Unit 7: The RGB Array

Skill Builder 1: Light Them Up

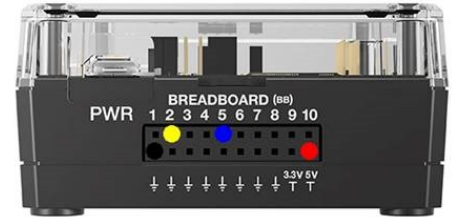
In this lesson, you will learn to control the sixteen color LEDs on the TI-RGB Array as a group and individually.

Objectives:

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



back



BB 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 using the cable wires that plug into breadboard (BB) ports on the TI-Innovator. Follow the wiring instructions on the back of the board to connect it to the TI-Innovator and then connect the TI-Innovator to your TI-84 CE.

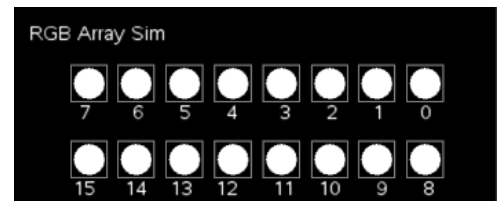
Your first program will make all 16 LEDs on the RGB Array blink 10 times. There are two instructions that control the Array:

Send("CONNECT RGB")

Send("SET RGB ALL *r g b*") or **Send("SET RGB # *r g b*")**

is an LED number (0 to 15) or **eval(var or expression)**

r, g, b are in the range 0 to 255.



Teacher Tip:

"CONNECT RGB" sets up Innovator to control the RGB Array.

"SET RGB ALL *r g b*" controls all 16 LEDs at once with the color *r-g-b*.

"SET RGB *L r g b*" lights up LED #*L* with the color *r-g-b*.

At the end of a program it is wise to turn off all LEDs with **"SET RGB ALL 0 0 0"**

There is one more command using the RGB Array:

Send("READ RGB")

Get(*M*)

Reads the current in mA (milliamperes) that the board is currently using. This command can be useful when studying electricity and determining relationships such as mA per LED, color and mA, brightness and mA, etc.



1. Start a new program - we call it **BLINK** - and add the statement **Send "CONNECT RGB"** by pressing **prgm > HUB > Send "CONNECT-Output > RGB** and add the closing quotation mark:

Send("CONNECT RGB")

```
NORMAL FLOAT AUTO REAL RADIAN MP
EDIT MENU: [alpha] [F5]

PROGRAM:BLINK
:Send("CONNECT RGB ")
:
```

Teacher Tip: Remember the Program Editor's editing features:

[del] to delete the object the cursor is on

[clear] to clear a line of code all at once

[2nd] [del] to begin insert mode (cursor changes appearance)

[alpha] [graph] (F5) for additional editor tools including 'Undo Clear'

2. Add a **For(** loop by pressing **prgm > For(**
The loop parameters are: a variable (we used **I**) varying from 1 to 10 or a value of your choice:

For(I,1,10)

But remember that when running the program, you must wait for the program to end before doing anything else. There will be a better way of ending programs, at any time, and it will be introduced in the next Skill Builder.

```
NORMAL FLOAT AUTO REAL RADIAN MP
EDIT MENU: [alpha] [F5]

PROGRAM:BLINK
:Send("CONNECT RGB ")
:For(I,1,10)
:
```

3. In the loop body add two **Send(** statements: one to turn all the LEDs on and one to turn them all off. Follow each with a **Wait** statement:

Send("SET RGB ALL 255 255 255") *white*

Wait 0.5

Send("SET RGB ALL 0 0 0") *off*

Wait 0.5

For the **Send** statements press **prgm>HUB>Send("SET... > RGB** and complete the rest of both statements manually.

Add the two **Wait** statement using **prgm>HUB>Wait** and provide a time value.

As with the COLOR LED on the Innovator and RV.COLOR on Rover, there are three color values: red, green, and blue, each varying from 0 to 255. We used white LED's but you can create any color you like.

```
NORMAL FLOAT AUTO REAL RADIAN MP
EDIT MENU: [alpha] [F5]

PROGRAM:BLINK
:Send("CONNECT RGB ")
:For(I,1,10)
:Send("SET RGB ALL 255 255 255")
:Wait .5
:Send("SET RGB ALL 0 0 0")
:Wait .5
:
```

Teacher Tip: Skill Builder 2 uses **getKey()** to control a loop allowing the user to press a key to end the program rather than waiting for a For loop to finish. A running program is awkward to 'Break': on the handheld, press [On] to break the program.



4. Add an **End** statement (**prgm > End**) to complete the **For**(loop.

```

NORMAL FLOAT AUTO REAL RADIAN MP
EDIT MENU: [alpha] [f5]

PROGRAM:BLINK
:Send("CONNECT RGB ")
:For(I,1,10)
:Send("SET RGB ALL 255 255
255")
:Wait .5
:Send("SET RGB ALL 0 0 0")

:Wait .5
:End
    
```

5. Run the program and watch the LEDs blink.
Modify the program to display your favorite color.

Each of the sixteen LED's can be controlled individually. Use the LED number (0 to 15) in place of **ALL** as in **Send("SET RGB 5 255 0 0")** turns LED #5 to red.

If using a variable, then remember to use the **eval()** function found on the **Hub** menu. The next step illustrates this technique.

6. To control the LEDs one at a time, use an *inner For*(loop and modify the **Send** commands to make use of the inner loop variable in place of **ALL** in the **Send** statements. (*indentation is for clarity only*):

```

For(I, 1, 5)           smaller loop to go faster
  For(J, 0, 15)
    Send("SET RGB eval(J) 255 255 255")
    Wait .1           make this small to go faster, too
    Send("SET RGB eval(J) 0 0 0")
    Wait .1           make this small to go faster, too
  End
End
End
    
```

eval() is found on **prgm > HUB**

We also made the outer loop (**For(I, 1, 5)**) smaller and decreased the **Wait** value to 0.1 to speed things up.
Run the program to see the effect.

