



Unit 7: The TI-RGB Array

Skill Builder 1: Light Them Up

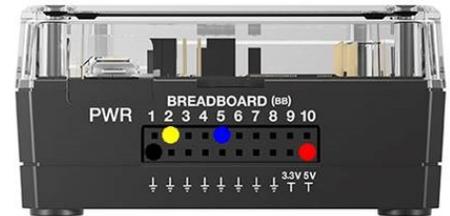
In this lesson, you will learn to control the sixteen color LEDs on the TI-RGB Array as a group and individually.

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-Nspire CX.

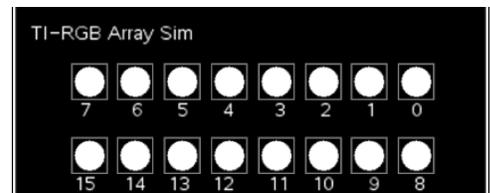
Your first program will make all 16 LEDs on the TI-RGB Array blink 10 times. There are two instructions that control the Array:

Send “CONNECT RGB”

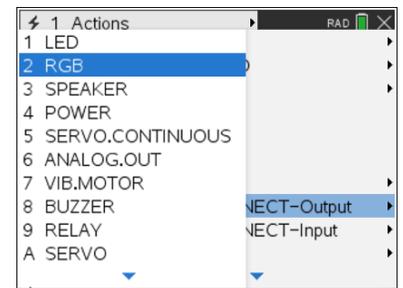
Send “SET RGB ALL *r g b*” or Send “SET RGB # *r g b*”

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

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



1. Start a new program - we call it **blink()** - and add the statement **Send “CONNECT RGB”** by pressing **menu > Hub >**
Send “CONNECT-Output > RGB and add the closing quotation mark:
Send “CONNECT RGB”





10 Minutes of Code

TI-NSPIRE™ CX WITH TI-INNOVATOR™ HUB AND TI-RGB ARRAY™

UNIT 7: SKILL BUILDER 1

STUDENT ACTIVITY

2. Add a **For** loop by pressing **menu > Control > For...EndFor**
The loop parameters are: a variable (we used *i*) varying from 1 to 10 or an upper 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 in the next Skill Builder.

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 **menu>Hub>Send "SET... > RGB** and complete the rest of both statements manually.

Add the two **Wait** statement using **menu>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.

4. Press ctrl-R to run the program and watch the LED's on the RGB Array blink.
Modify the program to display your favorite color.

```

1.1 *Doc RAD 2/4
* blink
Define blink()=
Prgm
Send "CONNECT RGB"
For i, 1, 10
EndFor
EndPrgm

```

```

1.1 *Doc RAD 6/7
* blink
Define blink()=
Prgm
Send "CONNECT RGB"
For i, 1, 10
  Send "SET RGB ALL 255 255 255"
  Wait 0.5
  Send "SET RGB ALL 0 0 0"
  Wait 0.5
EndFor
EndPrgm

```

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.

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.

5. To 'wrap' the **Send** and **Wait** statements in another **For** loop, first select the four statements by placing the insertion cursor at the beginning of the block and press [shift] [downarrow]. Then...

```

1.6 1.7 1.8 *10MoC ...Sim RAD 9/9
* blink
©Local i,j
connectrgb()
Send "CONNECT RGB"
For i,1,10
  Send "SET RGB ALL 255 255 255"
  Wait 0.1
  Send "SET RGB ALL 0 0 0"
  Wait 0.1
EndFor
EndPrgm

```



10 Minutes of Code

TI-NSPIRE™ CX WITH TI-INNOVATOR™ HUB AND TI-RGB ARRAY™

UNIT 7: SKILL BUILDER 1

STUDENT ACTIVITY

- With the block selected, add the **For** loop by pressing **menu > Control > For...EndFor**. The selected text will be inside the new **For** and **EndFor** statements.
- Complete the **For** statement. Note that the LEDs are numbered 0 to 15.
- Change **ALL** to **eval(j)** found on **menu>Hub**

```

For i, 1, 5           use a smaller loop to go faster
  For j, 0, 15
    Send "SET RGB eval(j) 255 255 0"   yellow
    Wait .1           make this smaller to go faster, too
    Send "SET RGB eval(j) 0 0 0"
    Wait .1           make this smaller to go faster, too
  EndFor
EndFor

```

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

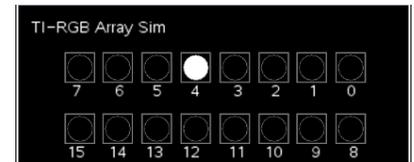
We also made the outer loop smaller and decreased the Wait value to 0.1 to speed things up.

Run the program to see the effect.

```

1.6 1.7 1.8 *RGB Arr...ub) RAD
"blink" stored successfully
connectrgb()
Send "connect rgb"
For i,1,5
  For j,0,15
    Send "set rgb eval(j) 255 255 255"
    Wait 0.1
    Send "set rgb eval(j) 0 0 0"
    Wait 0.1
  EndFor
EndFor

```



a 'moving' light