Unit 7: The TI-RGB Array	Skill Builder 2: A Rainbow of Color

In this lesson, you will generate random colors on each of the LEDs on the TI-RGB Array.

Objectives:

- Use getKey( ) to end a program
- Use randInt( ) to light a random LED in a random color

We will use the random number generator **randInt( )** to make random LEDs light up in random colors.

1. Start a new program - we called it **rainbow()** - and add

   Send "CONNECT RGB"

   Using **menu > HUB > Send "CONNECT-Output…** and select **RGB** from the next menu.

   Type the closing quote and parenthesis.

2. Add a **While… EndWhile** loop using **menu > Control > While…EndWhile**

3. The loop will end when the **[esc]** key is pressed. The condition is:
   While **getkey(0) ≠ "esc"**

   ≠ is found on **ctrl =**
   type the quotes using [ctrl] [*] then the letters *esc*.

Teacher Tip: If you prefer 'any key' to terminate the program use:
   While getkey(0) = " "   with *nothing* between the quotes.

Caution: rapidly flashing lights may be disturbing for some students, so it is wise to use Wait statements to slow things down a bit. You may also want to ask students if anyone has issues with this, prior to starting the lesson.

4. In the loop body write four statements to assign random integers to four variables representing the LED number and the red, green, and blue values.
   led   := **randInt(0, 15)**
   red   := **randInt(0, 255)**
   green := **randInt(0, 255)**
   blue   := **randInt(0, 255)**

5. Add the **Send** statement to control one of the TI-RGB Array LEDs:
   Send "SET RGB eval(led) eval(red) eval(green) eval(blue)"
   and a **Wait** statement to slow the lights down:
   Wait .25

6. Press **ctrl-R** to run the program to see a wide variety of colors blinking.

   Press **[esc]** to stop the program. Notice that the LEDs are still lit even though the program has ended. Fix this by turning them off just after the **EndWhile** statement by adding the statement:

   EndWhile
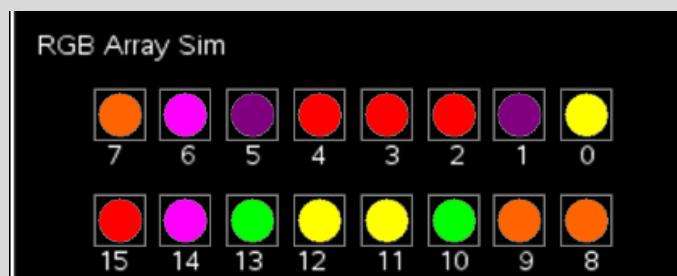   **Send "SET RGB ALL 0 0 0"**
   EndPrgm

> **Teacher Tip:** The colors generated seem bland. That's because there are 256^3 = 16777216 colors to choose from. A different approach is to define specific colors, like ROYGBIV.
>
> In a Lists & Spreadsheet app, define three lists: **listred, listgreen**, and **listblue.** In each row, enter the values that make up the rainbow's colors (indigo is left for you to determine):
>
> | | A listred | B listgreen | C listblue | D |
> |---|---|---|---|---|
> | 1 | 255 | 0 | 0 | red |
> | 2 | 255 | 100 | 0 | orange |
> | 3 | 255 | 255 | 0 | yellow |
> | 4 | 0 | 255 | 0 | green |
> | 5 | 0 | 0 | 255 | blue |
> | 6 | _ | _ | _ | indigo |
> | 7 | 255 | 0 | 255 | violet |
>
> *A1* 255
>
> Modify the **rainbow( )** program to select random colors from these lists:
> **color:=randInt(1,7)** *in place of the three randint(0,255) statements*
> **Send "SET RGB eval(led) eval(listred[color]) eval(listgreen[color] eval(listblue[color])"**
>
> *Much more colorful!*