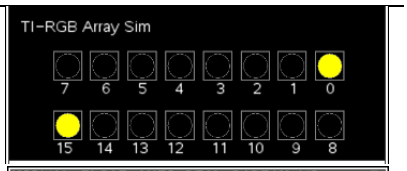
Unit 7: The TI-RGB Array — Skill Builder 3: Sequencing

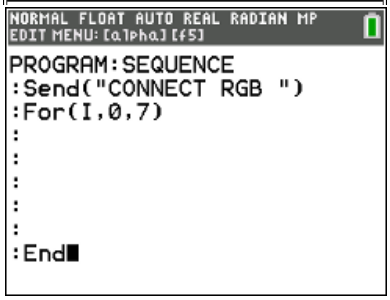
In this lesson, you will make the top row of LEDs light up from right to left and the bottom row light up from left to right at the same time.

Objectives:

- Use any key to end the program.
- Use a **For(** loop to control both rows of LEDs at once.

Take a close look at the numbers on the LEDs on the TI-RGB Array. The top row goes from 0 to 7 (right to left) and the bottom row goes from 15 to 8 (left to right). If we use a For I, 0, 7 statement to light up the top row, then what expression can we use to light up the bottom row in the opposite direction?





1. Start a new program – we called it **SEQUENCE -** and add the statements

   Send("CONNECT RGB ")

   For(I, 0, 7)

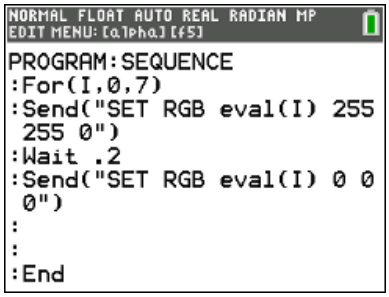
   Press [enter] several times to leave room for statements here

   End

   to control the top row of LEDs

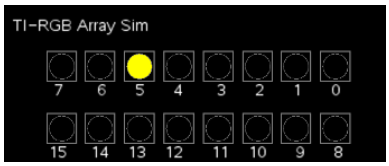
2. In the **For(** loop, write the three statements to turn on light # *I* (we are using yellow), wait a moment, then turn it off:

   Send("SET RGB eval(I) 255 255 0")
   Wait .2
   Send("SET RGB eval(I) 0 0 0")



Teacher Tip: We are developing this program from the inside out: get something working then build upon it: bottom-up design.

3. Test the program. You should see the top row of LEDs light up from right to left once.



We need to make the bottom row light up going from left (15) to right (8) in the same loop.

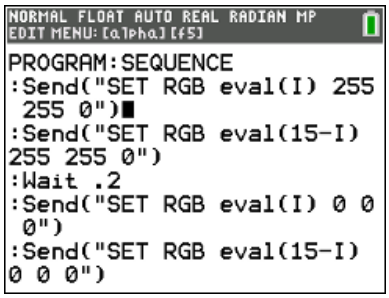
When *I* varies from 0 to 7, what *expression* varies from 15 to 8?

Answer: 15 – *I* *(do the mathematics!)*

4. Right after lighting up Led # I also light up LED # 15-I with the statement

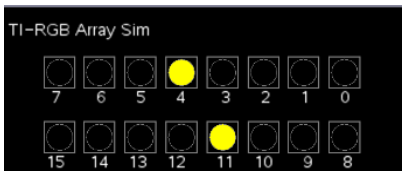
   Send("SET RGB eval(15 - I) 255 255 0")

   and turn it off after the **Wait** statement, too.



Teacher Tip: Here's an opportunity to teach about the *linear relationship* between the points (0,15) and (7,8). Use point-slope form to develop the equation of the line
y = 15 - x.

5. Run the program again. You should see two LED's light up, the top row going from right to left and the bottom row going from left to right. The program ends after one pass.



Next, create a loop to make the program continue until a key is pressed.

Teacher Tip: Students will 'wrap' the code in a loop structure by adding a statement before and another statement after the loop body.

Use an infinite **While** loop this time. Inside the loop check for a keypress to **Stop** the program.

Place the statement **While 1** above the **For(** statement.

The number **1** is the TI-84's representation of a True condition. 0 represents False.

```
NORMAL FLOAT AUTO REAL RADIAN MP
EDIT MENU: [alpha] [f5]
PROGRAM:SEQUENCE
:Send("CONNECT RGB ")
:While 1
:For(I,0,7)
:Send("SET RGB eval(I) 255
 255 0")
:Send("SET RGB eval(15-I)
255 255 0")
:Wait .2
:Send("SET RGB eval(I) 0 0
```

Teacher Tip: Infinite loops are 'dangerous'. On the CE, pressing ON will immediately 'break' any running program and show the error message "BREAK". This project handles infinite loops with an exit condition inside the loop.

6. Below the **End** statement that indicates the end of the **For(** loop, add another **End** statement to terminate the **While** loop.

```
NORMAL FLOAT AUTO REAL RADIAN MP
EDIT MENU: [alpha] [f5]
PROGRAM:SEQUENCE
255 255 0")
:Wait .2
:Send("SET RGB eval(I) 0 0
 0")
:Send("SET RGB eval(15-I)
0 0 0")
:End
:End
```

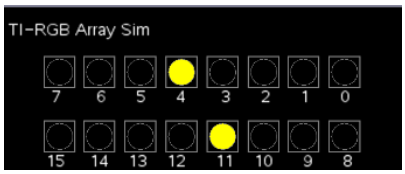
7. After all the LED's are turned off, add the **If** statement to check for a keypress and stop the program:

   If getKey≠0 : Stop

   The colon character is on the decimal point key ([alpha] [.]) and allows you to write two statements on the same line.
   Stop is on **prgm>Ctl.**

```
NORMAL FLOAT AUTO REAL RADIAN MP
EDIT MENU: [alpha] [f5]
PROGRAM:SEQUENCE
:Wait .2
:Send("SET RGB eval(I) 0 0
 0")
:Send("SET RGB eval(15-I)
0 0 0")
:If getKey≠0:Stop
:End
:End
:
```

8. Run the program and notice the effect: does it appear that the lights are moving 'around' the board? When the program is running press any key to end it.



Teacher Tip: Challenge students to complete the following scenarios:

- Can you make only one LED move 'around' the board? Start at 0, go to 7, *then* go from 15 to 8. Use two methods: 1) using two For loops and 2) using one For loop. Like a light moving around a movie theatre marquee.
- Turn off the previous LED before turning 'this' one on. Gives a different visual effect.
- Make a group (use an argument to the program) of LED's fading away as they move around the board. The first is the brightest and each one in the group is dimmer like Kitt's hood light from Knight Rider or Cylon 'eyes' from Battlestar Gallactica.
- Change the 'direction' of the movement from counterclockwise to clockwise.

©2019 Texas Instruments Incorporated