



In this third lesson for Unit 4, you will learn about turning to a specific angle heading, timing, and working with the COLOR LED on the Rover.

Objectives:

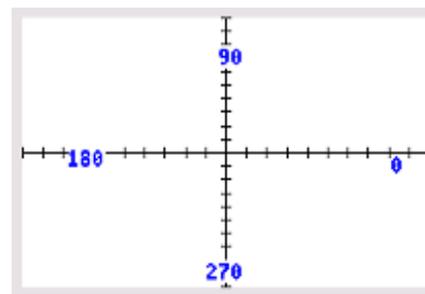
- Use the **TO ANGLE** command
- Use the **RV.COLOR** LED on the Rover
- Control timing on the handheld and the Rover

This lesson addresses three more features of the Rover:

- the **TO ANGLE** command (which is different from **LEFT** and **RIGHT**)
- lighting up the **RV.COLOR** LED onboard the Rover (labeled 'Color' on the top-front-left corner next to the battery level indicators)
- synchronizing your program with the Rover's movements using **Wait**

TO ANGLE

The command **Send "RV TO ANGLE <number>"** is used to turn the Rover to a particular 'heading'. When you send a command to connect the Rover, its heading is set to 0 degrees, which, in the mathematical world, is 'East' (facing from the origin toward the positive x-axis). In this world, North is 90 degrees, West is 180 degrees, and South is 270 degrees. See the diagram to the right.



Headings

Regardless of where the Rover is currently heading, the command **Send "RV TO ANGLE 0"** causes the Rover to turn to the direction that it was heading when the **"CONNECT RV"** command was issued.

The default angle measure is **DEGREES** but you can also specify **RADIANS** or **GRADS** (selected from the **RV Settings** menu).

Try this:

- Send "RV TO ANGLE 90"**
- Wait 2**
- Send "RV TO ANGLE 180"**
- Wait 2**
- Send "RV TO ANGLE 270"**
- Wait 2**
- Send "RV TO ANGLE 360"**

```

3.1 4.1 4.2 *Rover - 1...ams RAD 8/8
rover43a
Prgm
Send "CONNECT RV"
Send "RV TO ANGLE 90"
Wait 2
Send "RV TO ANGLE 180"
Wait 2
Send "RV TO ANGLE 270"
Wait 2
Send "RV TO ANGLE 360"
EndPrgm

```

Did the Rover do what you expected?



Syncing Your Program With Rover

Programs on the handheld are 'Done' before the Rover finishes moving. This is because the driving commands are stored in the TI-Innovator™ Hub faster than the Rover can process them. Driving commands are stored in a 'queue' and are processed by the TI-Innovator Hub when the Rover is ready for them.

In this activity, we'll write a program to move in a random pattern and light up the **RV.COLOR** LED on the Rover while the Rover is moving. We'll also make use of the **TO ANGLE** command and incorporate **eval()** to turn to the appropriate heading.

Setting Up the Program:

1. Start the program with the **CONNECT RV** command.
2. Add a **For** loop (**menu > Control > For...EndFor**) to have the Rover move in a random pattern. You can always add lines to the loop body if you need them.
3. In the loop body, add the statement to move **FORWARD**.
Send "RV FORWARD 1"
4. To have the Rover turn to a *random* heading, first add the command:
h:=randInt(0,360)
5. Add the **TO ANGLE** statement to turn in the direction **h**.
Send "RV TO ANGLE eval(h)"
6. Run the program now to see that:
 - The Rover moves in a random pattern.
 - The program is 'Done' almost immediately while the Rover still has to move.

```

* rover43
Denine rover43
Prgm
  makes a random sketch (scribble)
Local h,i
Send "CONNECT RV"
Text "Press enter to start."
For i,1,10
|
EndFor

```

```

* rover43
Send "CONNECT RV"
Text "Press enter to start."
For i,1,10
  Send "RV FORWARD 1"

  h:=randInt(0, 360)
  Send "RV TO ANGLE eval(h)"

```

RV.COLOR

Since the color LED on the TI-Innovator Hub is hidden inside the Rover, we are given another color LED on top of the Rover to control. The name of the Rover's color LED is **RV.COLOR** and works using the same controls as the onboard LED. You can use any of the four commands in the **menu > Hub > Rover (RV) > RV Color** menu shown to the right.

Send "SET RV.COLOR 255 255 255" produces a white light.

1: Drive RV	
2: Read RV Sensors	
3: RV Settings	
4: Read RV Path	
5: RV Color	1: RV.COLOR
6: RV Setup	2: RV.COLOR.RED
7: RV Control	3: RV.COLOR.GREEN
8: Send "CONNECT RV"	4: RV.COLOR.BLUE
9: Send "DISCONNECT RV"	



10 Minutes of Code

TI-NSPIRE™ CX WITH THE TI-INNOVATOR™ ROVER

UNIT 4: SKILL BUILDER 3

TEACHER NOTES

Adding RV.COLOR to Your Program

7. Add a **RV.COLOR** command to the loop body immediately before the **FORWARD 1** command. The color values are up to you.
8. Run the program again to see what happens. Notice that the LED lights up almost immediately and stays on.

```

3.1 4.1 4.2 *Rover - 1...ams RAD 12/15
* rover43
send CONNECT RV
Text "Press enter to start."
For i,1,10
  Send "SET RV.COLOR 128 246 100"
  Send "RV FORWARD 1"

h:=randInt(0,360)
Send "RV TO ANGLE eval(h)"

```

Now, let's try to get the LED to light up *only* while the Rover is moving FORWARD. This will require that we make the handheld **Wait** until each segment is completed and then turn the LED off while it is turning.

We need to add a **Wait** statement to our program to control when the LED is turned on and off.

Teacher Tip: This is a good place for experimentation if you have the time. Unlike the Drive commands, the COLOR instruction is processed as soon as it is accepted by the TI-Innovator Hub, so timing of the movement and the LED are not synchronized. That's left up to the programmers.

9. How long does it take Rover to move **FORWARD 1**? About 1 second? Add a **Wait 1** statement after the **FORWARD 1** statement. You can adjust the value if it is not suitable for your environment. Recall that **Wait** is found in the **menu > Hub** menu.
10. Run the program to test it.

```

3.1 4.1 4.2 *Rover - 1...ams RAD 9/16
* rover43
send CONNECT RV
Text "Press enter to start."
For i,1,10
  Send "SET RV.COLOR 128 246 100"
  Send "RV FORWARD 1"
  Wait 1

h:=randInt(0,360)
Send "RV TO ANGLE eval(h)"

```

Note that the LED still stays on. We have to turn the LED off after the Rover has finished moving.

11. How long does it take the Rover to turn? About 1 second? Add a **Wait 1** statement.
12. After the **Wait 1** statement, turn the LED off by adding **Send "SET RV.COLOR 0 0 0"**.

```

3.1 4.1 4.2 *Rover - 1...ams RAD 9/16
* rover43
send CONNECT RV
Text "Press enter to start."
For i,1,10
  Send "SET RV.COLOR 128 246 100"
  Send "RV FORWARD 1"
  Wait 1
  Send "SET RV.COLOR 0 0 0"
h:=randInt(0,360)
Send "RV TO ANGLE eval(h)"

```

13. Test your program. Did the Rover turn the light off at the right time?

We also need to wait while the Rover turns to a new heading to make sure the light has turned off.

14. Add a **Wait** statement *after* the **TO ANGLE** statement. This **Wait** should be long enough to handle any turn from 0 to 360 degrees.

```

3.1 4.1 4.2 *Rover - 1...ams RAD 12/13
* rover43
For i,1,10
  Send "SET RV.COLOR 128 246 100"
  Send "RV FORWARD 1"
  Wait 1
  Send "SET RV.COLOR 0 0 0"
  h:=randInt(0,360)
  Send "RV TO ANGLE eval(h)"
  Wait 3
EndFor
EndPrgm

```



Extension

Can you make a *different* color at each corner? Hint: Use **eval**(*something*), and incorporate the value of **h** but be careful about the range of values permitted.

Additional Challenge:

Make the turning wait time a function of the angle to be turned by observing the Rover's turning direction when using **TO ANGLE**.

Teacher Tip: Driving commands are sent to the TI-Innovator Hub immediately and are queued up in the TI-Innovator Hub until the Rover is ready for each one.

The **Wait** commands are instructions *to the handheld* to wait while the Rover is completing its current task. Large **Wait** values will work but the Rover will seem to pause during its travel until it receives the next instruction.

For the Extension, students can use a combination of *i*, the loop variable, and *h*, the heading, to produce different colors. Remember that the range of values for *r*, *g*, and *b* is 0...255.