Unit 5: Rover's Sensors — Skill Builder 1: Introducing Ranger

In this lesson, you will learn to read the distance from the front of the TI-Innovator Rover to an obstacle. This lesson monitors that distance and displays information in two different ways. Another sensor is on the bottom of Rover and it can 'see' colors.

Objectives:

- Read Rover's Ranger measurement
- Display it on the screen using **print()**
- Display it on the screen using **text_at()**
- Determine the unit of measure

The two small cylinders on the front of Rover are not headlights. They are an Ultrasonic Ranger. A silent (to us) tone is sent out by one of the two sensors and the other one 'hears' the echo when the sound bounces off an obstacle. The internal software then calculates the distance to the obstacle using the speed of the sound and the time that it takes for the echo to return to the sensor: **D = V * T**

All that happens in the blink of an eye!

Teacher Tip: Sometimes referred to as a 'motion detector', an Ultrasonic Ranger is more of a 'rangefinder' that determines distance, not motion.

1. Begin a new Python Rover Coding project. For this lesson, use the convenient 'press **[clear]** to exit' loop:

   while not escape( ):

   block

   You will find this command on several menus, but since you are using **Rover** commands, look in **[math] ti_rover… Commands**.

2. Add three statements in the **while** block to
   a) read the distance in front of Rover:

   d = rv.ranger_measurement()

   find this on **[math] ti_rover… I/O Inputs…**

   b) print it on the screen:

   print("distance= ",d)

   c) wait before reading the next distance:

   sleep(.5)

Run the program and move your hand toward and away from the front of Rover. Point Rover at a wall, the ceiling, and the floor. Watch the values that are displayed. *Note: Rover does not move…yet.*

Can you determine what distance unit (feet, meters, etc.) is being used?

Teacher Tip: The Ranger's distance unit is meters.

3. Rather than have the printed numbers scroll down the screen, improve the display of the distance values by using the **disp_at()** function. If you worked through the previous TI-Innovator™ Hub lessons (Units 1, 2, and 3), you have used that command before.

   Add **disp_clr()** before the **while** loop and **disp_at()** in place of **print()** in the loop. Both functions are found on **[math] ti_rover… Commands**

   Recall that **disp_at( )** has three arguments: line number, string, and alignment. Use **str(d)** to convert the value of variable **d** to a string.
   str( ) is found on **<Fns…> Type.**
   Remember to use **[2nd] [3]** for the # sign to comment a line.

   Teacher Tip: Students can use either "left" or "right" alignment, too. Be careful, since new data may not properly erase all the old data. Putting **disp_clr()** in the loop will erase all previous data but may cause some flickering.

4. You now have a digital tape measure. In the next lesson you will use this information to keep Rover from crashing into something.