

**Unit 2: Skill Builder 2 - Distance Ranger**
**Objectives:**

Students are presented with three challenges that utilize the ultrasonic ranger mounted on the front of Rover. The first challenge introduces the ranger and how to generate a sound proportional with measured distance. The next challenge has the student drive the Rover toward a wall and uses a position vs. time graph to find the speed of the Rover. In the final challenge, students use their Rover and coding skills to escape from a room of obstacles.

Students will:

1. Use the built-in speaker on the TI-Innovator Hub to make a tone.
2. Use the ultrasonic ranger to make measurements and store for later graphing on calculator.
3. Use the ultrasonic ranger to avoid obstacles.

**Background:**

Driverless cars need to be able to “know” how close surrounding cars and obstacles are to avoid collisions and to assist in autonomous navigation. There are many different technologies used to accomplish this task. One method uses high frequency sound waves referred to as ultrasound. The prefix ultra- means “beyond”, thus, ultrasound is beyond the human hearing range. So, you will not hear the sound coming from the ultrasonic ranger mounted on the front of Rover. The ultrasonic ranger appears to be two “eyes” on Rover. One of the “eyes” is the emitter of the sound wave, while the other is the receiver of the sound wave. The ranger works by sending out a pulse of waves with the emitter and then listening for reflected pulse, or echo, with the detector. This distance is determined by using the formula of  $\text{speed} = \text{distance}/\text{time}$ . In this case, the speed is that of the sound (~344 m/s) and the time is the period between the emitted and detected pulse. Knowing the speed and the time, the distance to the object can be quickly calculated on the Innovator Hub. This same method, called echolocation, is used by some mammals and a few aves (birds) to make sense of their surroundings. These animals can produce a high frequency click that reflects off of nearby obstacles and is then heard by the two ears. The animal's brain then makes a kind of image of the surroundings using the echo information. Dolphins in particular make excellent use of echolocation to be extremely efficient hunters and are able to detect their prey that may be hiding below sand on the bottom of the ocean. The ultrasonic ranger on Rover can be used to help your program make sense of the surroundings and avoid obstacles.

Command	Example	Behavior
READ RANGER	Send("READ RV.RANGER")	Reads the distance to an obstacle in front of the Rover and returns that reading to the calculator in meters.
READ RV.WAYPOINT.TIME	Send("READ RV.WAYPOINT.TIME")	Reads the elapsed time Rover has been moving since the beginning of the first drive command.
READ RV.WAYPOINT.HEADING	Send("READ RV.WAYPOINT.HEADING")	Reads the present angular heading of the Rover in degrees as measured by the internal gyroscope.
RV TO ANGLE heading	Send("RV TO ANGLE 105")	Turns Rover 105° counter-clockwise from the initial heading at the start of program execution.
RV.STOP	Send("RV STOP")	Stops the Rover immediately.
Get variable	Get(B)	The Get command retrieves the value returned to the calculator from the Hub after a READ command is issued. In this example, the brightness measurement is stored in the variable named B.
SET SOUND frequency TIME duration	Send("SET SOUND 440 TIME 10")	Plays a tone of a given frequency in Hz. The tone will play for the duration in seconds given after the TIME keyword. In this example, a pitch of 440 Hz will sound for 10 seconds.
getKey	getKey→K	Returns the last key pressed as a two-digit value, row number counting from the top of the keyboard and

		column number from the left, to the variable named K. e.g. the number 1 key is returned as 92 (9 <sup>th</sup> row and 2 <sup>nd</sup> column) and the clear key is returned as 45 when the value of K is accessed.
While b ≠ 10....End	While b ≠ 10....End	Repeats commands within structure until the variable b equals 10.
<b>num</b> → <b>dim(list)</b>	<b>0</b> → <b>dim(L<sub>2</sub>)</b>	Sets the initial element in L <sub>2</sub> to the value 0
<b>num</b> → <b>L<sub>2</sub>(index)</b>	<b>D</b> → <b>L<sub>2</sub>(C)</b>	Puts the value of variable D into L <sub>2</sub> at index C

### TI-Innovator™ Rover Set-up:

Students may work in groups of two or three. Choose an area to work that has at least 2 meters of clear uniform floor space. Carpeted flooring is less desirable than tile. If needed, driving mats may be used for driving surface. The “Escape the Room” course may be constructed using several obstacles that offer only one way out of the “room”.

### Materials:

- Obstacle setup for Challenge 3: “Escape the Room”

### Student Activity:

**Challenge 1:** Write a program for Rover that reads the ranger and then scales that distance into an audible tone and plays it on the Hub speaker. Test your program by moving your hand in front of the Rover’s ranger sensor.

### Teacher Notes:

#### Teacher Guidance during Challenge 1:

*Review the Rover commands needed for this activity.*

- Students should practice making tones on their Innovator Hub and Rover with different frequencies and durations.
  - If students are not familiar with the speaker on the Hub, review 10 Minutes of Code with TI-Innovator Unit 2- SB3: Loop through the musical notes.
- A one line program may be used to explore the speaker.  

```
Send("SET SOUND 261 TIME 5")
```
- Students should practice making and displaying distance measurements on their Innovator Hub. Students practice writing a small program to display the distance their Rover is to a wall or obstacle.
- Three lines of code are required to display a distance measurement on Rover (after the Rover is connected).  

```
Send("CONNECT RV")
Send("READ RV.RANGER")
Get(D)
Disp "DISTANCE=",D
```
- Once the students are familiar with using the speaker and ranger, introduce to them the getKey command. This command polls the keyboard and returns a two-digit value. The first digit is the row on the keyboard

counting from the top beginning at 1. The second digit is the column on the keyboard counting from the left beginning at 1. The getKet command can be combined along with a While/End structure to make a continuous loop that can be exited using any key of choice. In the example program below, the clear key (2-digit value of 45) is used to exit the program. This structure uses the getKey command to monitor the keyboard.

- Each time the While loop is executed the expression  $K \neq 45$  (the two-digit value for the clear key) is evaluated. Initially K is set equal to zero, so when the expression is evaluated, it returns true and the While structure loops again. When getKey returns 45 due to pressing the clear key, the expression  $K \neq 45$  returns false and the While loop exits. This is a handy tool for the programmer's toolbox, it may be used anytime a process needs to occur indefinitely and then quit with an easy key stroke.

```
0→K
1→N
While K≠45
  Disp "N=",N
  N+1→N
  getKey→K
End
```

- Once your students have practiced the SOUND, RV.RANGER and the getKey command, they are prepared to complete challenge 1. The program below continuously reads the distance to an object using the ranger, scales the distance with an appropriate factor, in this case 1000, and then uses the evaluation of the product to set the frequency of the SOUND output. For example, if the distance to the object is .7 m, when multiplied by 1000, the product is 700. 700 Hz is an audible frequency.
- Please see the associated sample program, U2C1SB2.8xp as a reference if help is needed.

**Sample Code:**

```
PROGRAM:C1SB2
Send("CONNECT RV")
0→K
While K≠45
  Send("READ RV.RANGER")
  Get(D)
  Disp "DISTANCE (M)=",D
  Send("SET SOUND eval(D*1000) TIME 1")
```

```
Wait 1
getKey→K
End
```

**Challenge 2:** Write a program that:

1. drives Rover slowly toward a wall.
2. record the distance and elapsed time to the wall every half second.
3. stop Rover when it is 0.15 meter from the wall.
4. make a graph of distance vs. time for the drive path.

*What do the slope and Y-intercept of the graph represent about the drive path?*

**Teacher Guidance during Challenge 2:**

- This program will use two lists(arrays), the first named  $L_1$  and the other  $L_2$ . The variable  $L_1$  will be used for time values and  $L_2$  will be used for distance values.  $L_1$  and  $L_2$  are initialized as blank lists in the beginning section of the program by setting their dimension (length) to zero. Lists use an index variable to reference positions within the list. The index variable is placed in parentheses after the list name. In the example,  $42 \rightarrow L_2(3)$ , 42 will be stored in the third location of the list named  $L_2$ . Loops are often used to populate a list with data.
- The list (array) index can be the loop variable in a For..End loop or a counter variable in a While..End loop e.g.  $C+1 \rightarrow C$  will increment an index variable named C by one each iteration of the loop. It is important to note the first index position of an array is 1(one). Students should practice filling an array with ten elements using a loop and verifying the result.
- After running this program, while on the home screen, press 2<sup>nd</sup>  $L_2$  enter key. You should see the numbers 1 through 10 in a list with curly braces on each end and separated with commas.

$L_2$  {1,2,3,4,5,6,7,8,9,10}

- Please see the associated sample program, U2C2ASB2.8xp as a reference if help is needed.

**Sample Code:**

```
PROGRAM: C2ASB2
1→C
0→dim(L2)
While C<11
  C→L2(C)
  C+1→C
End
```

- Once students understand how to use an array, the next step is to fill two separate arrays named  $L_1$  (time) and  $L_2$  (distance) with measurements from the Rover.  $L_1$  (time array) is filled with values from the Rovers clock that is accessed with the "READ RV.WAYPOINT.TIME" command and  $L_2$  (distance array) will be filled with measurements from the ranger using the Send "READ RANGER" command. The two arrays will be filled using a While..End loop that runs while the ranger value is greater than .15 meters.
- The End statement for the While loop is followed by an RV.STOP command to immediately stop Rover

just before hitting the wall. Once the program ends, students can graph the position vs. time data on their calculator. Challenge students to calculate the slope of this line to find the speed of Rover. Students may wish to try repeating with a different speed programmed into Rover and then checking the effect on the slope of the line.

- Note: SPEED and M/S are available from the RV Settings menu. RV.WAYPOINT.TIME is available from the RV Path menu.
- Please see the associated sample program, U2C2BSB2.8xp as a reference if help is needed.

**Sample Code:**

```
PROGRAM: C2BSB2
10→D
1→N
0→dim(L1)
0→dim(L2)
Send("CONNECT RV")
Send("RV FORWARD 10 SPEED .14 M/S")
While D>0.15
  Send("READ RV.RANGER")
  Get(D)
  D→L2(N)
  Send("READ RV.WAYPOINT.TIME")
  Get(T)
  T→L1(N)
  N+1→N
End
Send("RV STOP")
```

**Challenge 3 (Escape the Room):** Write a program that:

1. turns a full 360° circle
2. measures the distance to the nearest obstacle every 15° arc and updates a variable with the heading that points in the direction

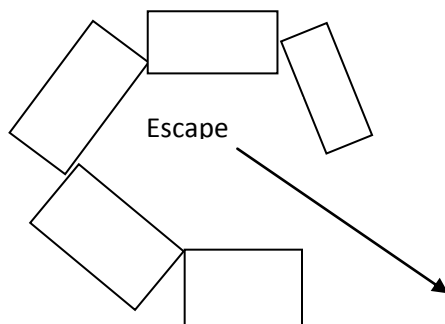
**Teacher Guidance during Challenge 3:**

- This challenge requires many of the skills practiced in this and the previous skill builders. The goal is to turn the Rover in a complete circle at 15° arc intervals (24 steps) while using the ranger to measure distance and the gyroscope to measure heading angle. Two variables are required, one to store the maximum obstacle distance (M), and the other to store the heading for that maximum distance (E).
- Each time a measurement is made, the distance is compared with the maximum distance variable. If the measured distance is greater than the maximum distance, store the new distance in the maximum

of the escape path (greatest distance).

3. after all directions have been measured, turn Rover to the correct heading and drive forward and escape the room.

Example Escape Room made with boxes



distance variable (M) and store the present heading in the escape path variable (E).

- Repeat this process twelve times to find the heading with the maximum distance (escape path). Once Rover completes the 24 measurements, Rover is turned to the escape path heading and drives forward to “escape the room”.
- The gyroscope measures angular displacement counter-clockwise from the initial direction.
- The setup for the escape room can be made by placing cardboard boxes on a smooth floor in such a way that there is a gap that Rover can drive through.
- Please see the associated sample program, U2C3SB2.8xp as a reference if help is needed.

#### Sample Code:

```
PROGRAM: C3SB2
Send ("CONNECT RV")
0→M
0→E
For (N,1,24)
  Send ("RV LEFT 15")
  Wait 1
  Send ("READ RV.RANGER")
  Get (D)
  Send ("READ RV.WAYPOINT.HEADING")
  Get (H)
  If D>M
    Then
      D→M
      H→E
  End
Disp "ESCAPE ANGLE=",E,"MAX DISTANCE=",M
End
Send ("RV TO ANGLE eval(E)")
Send ("RV FORWARD 2 M")
```