

THE ON-RAMP TO ROBOTICS WITH TI-INNOVATOR™ SYSTEM (TI-NSPIRE CX)
Overview:

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.

Goals:

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 speed = distance/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 <i>b</i>	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 <i>b</i> .
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()	<i>k</i> =getKey()	Returns the last key pressed as a string to the variable named <i>k</i> . e.g. the number 1 key is returned as “1” and the escape key is returned as “esc” when the value of <i>k</i> is accessed.

THE ON-RAMP TO ROBOTICS WITH TI-INNOVATOR™ SYSTEM (TI-NSPIRE CX)
Setup Rover:

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”.

Supplies:

- Rover with calculator attached.
- Obstacle setup for “Escape the Room” Challenge

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 Activity
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.
- 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.

```
Send "CONNECT RV"
```

```
Send "READ RV.RANGER"
```

```
Get d
```

```
DispAt 3, "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 the name of the most recently pressed key. This command can be combined along with a While/EndWhile structure to make a continuous loop that can be exited using any key of choice. In the example program below, the [esc] key is used to exit the program. The main idea of this structure is to use the getKey() command to monitor the keyboard.
- Each time the While loop is executed the expression $k \neq \text{"esc"}$ is evaluated. Initially k is set equal to the empty string “”, so when the expression is evaluated, it returns true and the While structure loops again. When getKey() returns “esc” due to the [esc] key press, the expression $k \neq \text{"esc"}$ 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.

THE ON-RAMP TO ROBOTICS WITH TI-INNOVATOR™ SYSTEM (TI-NSPIRE CX)

```

k:=""
n:=1
While k≠"esc"
  DispAt 3,"n=",n
  n:=n+1
  k:=getKey()
EndWhile

```

- 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.

Sample Code:

```

Define c1()=
Prgm
Send "CONNECT RV"
k:=""
While k≠"esc"
  Send "READ RV.RANGER"
  Get d
  DispAt 3,"Distance =",d," M"
  Send "SET SOUND eval(d*1000) TIME 1"
  Wait 1
  k:=getKey()
EndWhile
EndPrgm

```

*Please see the associated sample program, c1, in the file named "Unit 2 Skill Builder 2 Example Programs.tns" as a reference if help is needed.

THE ON-RAMP TO ROBOTICS WITH TI-INNOVATOR™ SYSTEM (TI-NSPIRE CX)

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 .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?

Guidance during challenge 2:

- This program will use two lists(arrays), the first named *time* and the other *dist*. The variables *time* and *dist* are initialized as blank lists in the beginning section of the program, using { }. Lists use an index variable to reference locations within the list. The index variable is placed in square braces after the list name. In the example, *dist*[3]:=42, 42 will be stored in the third location of the list named *dist*. Loops are often used to populate a list with data.
- The array index can be the loop variable in a For..EndFor loop or a counter variable in a While..EndWhile loop e.g. *c:=c+1* will increment an index variable named *c* by one each iteration of the loop. It is important to note the first index 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 calculator page, press the [var] key and select the name *dist* and press the [enter] key. You should see the numbers 1 through 10 in a list with curly braces on each end and separated with commas. *dist* {1,2,3,4,5,6,7,8,9,10}

Sample Code:

```
Define c2a()=
Prgm
c:=1
dist:={}
While c<11
  dist[c]:=c
  c:=c+1
EndWhile
EndPrgm
```

*Please see the associated sample program, c2a, in the file named "Unit 2 Skill Builder 2 Example Programs.tns" as a reference if help is needed.

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

THE ON-RAMP TO ROBOTICS WITH TI-INNOVATOR™ SYSTEM (TI-NSPIRE CX)

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.

Sample Code:

```
Define c2b()=
Prgm
d:=10
n:=1
dist:={}
time:={}
Send "CONNECT RV"
Send "RV FORWARD 10 SPEED .14 M/S"
While d>0.15
  Send "READ RV.RANGER"
  Get d
  dist[n]:=d
  Send "READ RV.WAYPOINT.TIME"
  Get t
  time[n]:=t
  n:=n+1
EndWhile
Send "RV STOP"
EndPrgm
```

*Please see the associated sample program, c2b, in the file named "Unit 2 Skill Builder 2 Example Programs.tns" as a reference if help is needed.

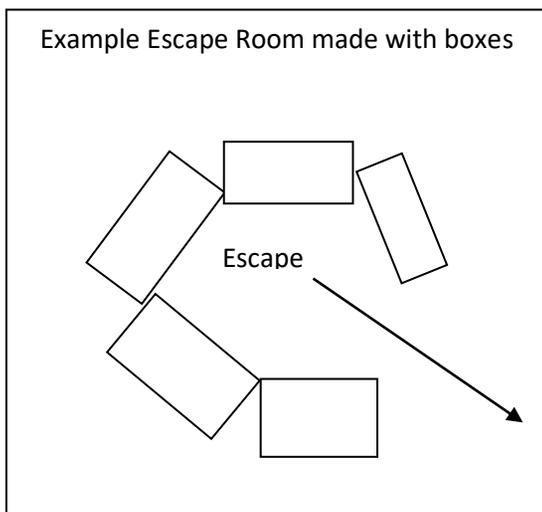
THE ON-RAMP TO ROBOTICS WITH TI-INNOVATOR™ SYSTEM (TI-NSPIRE CX)

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

1. turns a full 360° circle
2. measures the distance to the nearest obstacle every 30° arc and updates a variable with the heading that points in the direction 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.

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 30° arc intervals (12 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, and the other to store the heading for that maximum distance.
- 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 distance variable and store the present heading in the escape path variable.
- Repeat this process twelve times to find the heading with the maximum distance (escape path). Once Rover completes the 12 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.



Sample Code:

```

Define c3()=
Prgm
Send "CONNECT RV"
maxd:=0
escape:=0
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>maxd Then
    maxd:=d
    escape:=h
  EndIf
DispAt 3,"Escape=" ,escape," max d= ",maxd
    
```

THE ON-RAMP TO ROBOTICS WITH TI-INNOVATOR™ SYSTEM (TI-NSPIRE CX)

```
EndFor
Send "RV TO ANGLE eval(escape)"
Send "RV FORWARD 2 M"
EndPrgm
```

*Please see the associated sample program, c3, in the file named "Unit 2 Skill Builder 2 Example Programs.tns" as a reference if help is needed.