

Unit 1: Skill Builder 2 - Turning

Objectives:

Students will write a program on their calculator to turn the Rover left and right at various angles. They are challenged to make their Rover slowly turn like the hands of clock and to report the time of day on their calculator.

Students will:

1. write a TI BASIC program to turn Rover left or right at different angles.
2. incorporate the For...EndFor control structure into a program.
3. incorporate the DispAt I/O statement into a program.
4. make a model of a clock using the Rover.

Background:

The Rover turns by rotating its wheels in opposite directions at the same speed at the same time. This type of turn is called a spin because it spins in a circle that has a center at the midpoint of the two wheels. This midpoint is also the location of the marker tip when a dry-erase marker is inserted into the pen holder. When the Rover performs a turn, the program needs to inform the motors the direction and size of the spin. The direction is determined by LEFT or RIGHT, this direction is from the view of as if Rover had a driver's seat. The size is determined by the angle, this is a value in degrees, radians, or gradians. A full spin is 360 degrees. This number is from the base 60 sexagesimal system used by the Sumerians in ancient Babylon. Similarly, a full spin is 2π radians. This number comes from the fact that the angular width of an arc of one radius in length along the circumference of any circle is defined as one radian. Also, a full spin is 400 gradians. The grad is defined in the metric system as 1/100 of a circle quadrant. The Rover can accept all three units when an angle is given using the key words of DEGREES, RADIANS and GRADS.

Rover Command	Example	Behavior
RV LEFT <i>angle</i> SPEED <i>speed</i> UNIT	RV LEFT	Rover makes a left hand* 90° spin
RV RIGHT <i>angle</i> SPEED <i>speed</i> UNIT	RV RIGHT	Rover makes a right hand* 90° spin
	RV LEFT 45	Rover makes a 45° left hand spin
	RV RIGHT 3.14 RADIANS **	Rover makes a π radians (180°) right hand spin
	RV RIGHT 200 GRAD ***	Rover makes a 200 gradians (180°) right hand spin
	RV LEFT 180 SPEED .14 M/S ****	Rover makes a slow 180° left hand turn
	RV RIGHT 180 SPEED .23 M/S ****	Rover makes a fast 180° right hand turn

* The LEFT and RIGHT turns are made with a frame of reference from Rover's driver's seat.

** Radians is an angular unit of measure used in mathematics. There are 2π RADIANS in 360° DEGREES.

*** Gradians is an angular unit of measure also used in mathematics. There are 100 GRADIANS in a quarter circle; hence 400 grads in a full circle.

**** The maximum speed is .23 M/S and the minimum is .14 M/S.

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 as a driving surface.

Materials:

- Masking tape
- Drive mats

Prerequisites:

- It is recommended that the “On Road to Robotics” activities are done in sequence, starting with Unit 1 – Skill Builder 1.
- The “challenge” format of these activities will be more manageable for students if they have some foundation in programming prior to starting the lessons. If students are new to programming, it is strongly recommended that they visit *10 Minutes of Code for the TI-Innovator Hub* for an introduction to programming in TI-BASIC at www.education.ti.com/en/activities/ti-codes
- Students should have a general understanding of the TI-BASIC editor and how to run a program, and the following skills:
 - Basic navigation on the calculator.
 - Saving and opening files.
 - Editing new and existing programs.
 - Running programs.
 - Editing program features.
 - See example program c1 in [U1SB1 Example Programs.tns](#)

Student Activity:

Challenge 1: Write a program named “c1” that causes the Rover to spin 360 degrees to the right.

Teacher Notes:

Teacher Guidance during Challenge 1:

- This first challenge should be done just after the RV RIGHT and RV LEFT commands are introduced and before introducing the angle argument for the commands. Do not, yet, inform the students that an angle can be entered into the command.
- As the students explore using the RIGHT and LEFT turns, challenge them to determine what angle the Rover is turning. Next, ask how many angle turns are needed to spin a circle?
- Discuss and challenge students that are ahead how they could do the same program turning to the left.
- See program **c1** in [U1SB2 Example Programs.tns](#)
- For the last task, students can modify the angle speed as well as a wait command in the program to modify the time it to complete a circle.

```
Define c1()=
Prgm
Send "CONNECT RV"
Send "RV RIGHT "
Send "RV RIGHT "
Send "RV RIGHT "
Send "RV RIGHT "
EndPrgm
```

Challenge 2: Write a program named “c2” that will display the total angle turned at the end of each turn step while turning in a circle.

Teacher Guidance during challenge 2:

Review the usage of the DispAt input/output (I/O) statement with students.

- Menu> I/O> DispAt
- Enter the line number, 1 (top of screen) to 8 (bottom of screen)
- Enter the display string in quotes “ “
- An example: DispAt 1, "Turn Angle is 90"

Review the usage of the Wait command with students.

- Menu> Hub> Wait
- Enter the time in seconds to wait.
- Example: Wait 2
- The Rover and the calculator perform at different speeds. When a drive command is sent to the Rover, it

may require several seconds for the Rover to drive that command. The program must wait for the Rover to finish before doing another command. The Wait command is used to tell the calculator to do nothing and just wait for the Rover to finish driving.

- Have the students explore the effect of using the Wait command in different places in their program and also with different times. It can be tricky to keep the DispAt in sync with the Rover's motion in order to achieve Challenge 2. Please refer to program c2 [U1SB2 Example Programs.tns](#) for example solution for this challenge.
- Discuss and challenge students who are ahead how they could do the same program in radians or gradians.

```
Define c2()=  
Prgm  
Send "CONNECT RV"  
Send "RV RIGHT "  
DispAt 1,"Turn to 90"  
Wait 3  
Send "RV RIGHT "  
DispAt 1,"Turn to 180"  
Wait 3  
Send "RV RIGHT "  
DispAt 1,"Turn to 270"  
Wait 3  
Send "RV RIGHT "  
DispAt 1,"Turn to 360"  
Wait 3  
EndPrgm
```

Challenge 3: Write a program named “c3” and use a For...EndFor loop to turn three circles to the right and then three circles to the left and display the total angle turned by the Rover on the calculator screen.

Teacher Guidance during Challenge 3:

Review the usage of the For...EndFor statement with students.

- Menu> Control> For...EndFor
- Notice the two commas For ,, these are delimiting punctuation characters for the three arguments of; index variable name, start value, and end value required in the command. An example of a For...EndFor that does something 5 times is:

```
For n,1,5
... do something five times
EndFor
```

- Help the students understand the difference between using the previous program which explicitly calls out each command and the use of the For...EndFor loop which reuses one set of commands repeatedly. Also, inform students of the use of the index variable. This variable may be used to calculate the total angle turned by the Rover using the expression $n*90$, where n is incremented each time the For...EndFor loop is iterated. See example program **c3** in [U1SB2 Example Programs.tns](#).

```
Define c3()=
Prgm
Send "CONNECT RV"
For n,1,12
Send "RV RIGHT "
DispAt 1,"Right Turn Angle = ",n*90
Wait 2
EndFor
For n,1,12
Send "RV LEFT "
DispAt 1,"Left Turn Angle = ",n*90
Wait 2
EndFor
EndPrgm
```

Challenge 4: Write a program named “**c4**” that has Rover model the hour hand on a clock. Include the time that Rover is pointing to on the display of the calculator

Teacher Guidance during Challenge 4:

- This challenge is the final challenge of this activity and requires that students incorporate all of the skills learned so far. Encourage student to look back through programs c1, c2, and c3 as references for creating this new program. Students may be curious of the fact that there are 24 hours in a day. This number, like 360 degrees in a circle, comes from the Babylonian sexagesimal system since $360/15=24$.

- See example program **c4** in [U1SB2 Example Programs.tns](#).

```
Define c4()=  
Prgm  
Send "CONNECT RV"  
For n,1,12  
Send "RV RIGHT 30"  
DispAt 2,"The Time is:",n  
Wait 2  
EndFor
```

- Extension: Use a loop to have students program the Rover to rotate at least two full circles. Use a variable and the mod() function to display the time. Example: $\text{mod}(11,12)+1 = 12$, $\text{mod}(12,12)+1 = 1$, $\text{mod}(13,12)+1 = 2$, $\text{mod}(14,12)+1 = 3$ $\text{mod}(t,12)+1$

```
Define c4extend()=  
Prgm  
Send "CONNECT RV"  
For n,0,23  
Send "RV RIGHT 30"  
DispAt 2,"The Time is:",mod(n,12)+1  
Wait 2  
EndFor
```