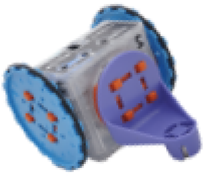
Sample Challenges

RoboPlay

Challenge Competition

April 2023



Barobo

UCDAVIS
C–STEM Center

Sample Challenges

RoboPlay™

Table of Contents

Grade Level Mats

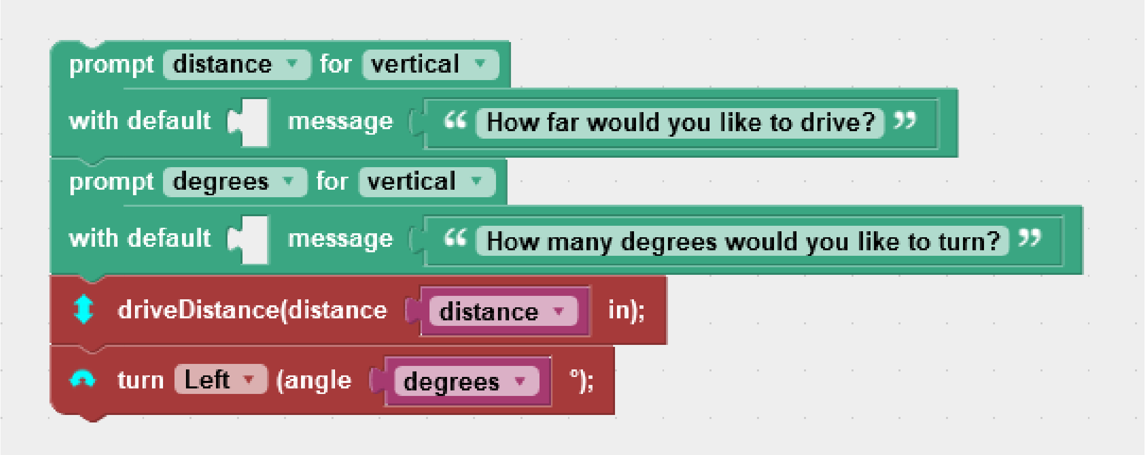
Grades 3-4:	RoboTown for grades 2-5
Grades 5-6:	Either RoboTown for grades 2-5 or grades 6-12
Grades 7-8:	RoboTown for grades 6-12
Grades 9-12:	RoboExploration for grades 6-12

About RoboPlay Challenge: http://c-stem.ucdavis.edu/roboplay/

These commands shall be used for challenges for grades 7-12 with random input as shown in the sample challenge “Programming with User Input.”

Sample prompt() Code in RoboBlocky

Sample code using **prompt**() in RoboBlocky is shown below:



```
prompt distance for vertical
with default message “ How far would you like to drive? ”
prompt degrees for vertical
with default message “ How many degrees would you like to turn? ”
driveDistance(distance distance in);
turn Left (angle degrees °);
```

The user’s input is stored in the variables “distance” and degrees.”

Sample scanf() Code in Ch

The equivalent Ch code using scanf() is shown below:

```
#include <linkbot.h>
CLinkbotI robot;
double radius = 1.75;
double trackwidth = 3.69;
double distance, degrees;

printf("How far would you like to drive?\n");
scanf("%lf", &distance);
printf("How many degrees would you like to turn?\n");
scanf("%lf", &degrees);
robot.driveDistance(distance, radius);
robot.turnLeft(degrees, radius, trackwidth);
```

Grades 3-4 (Division E)

Number Line Math Facts 70 Points

Background

RoboTown is a great place to practice arithmetic. Using the number line at the bottom of RoboTown, program the linkbot to move the correct amounts, using loops and playing notes when asked. When reversing, students may either turn around or move backwards. Use a single program for this challenge.

Setup

➢ Start in the bottom left corner of Robotown, facing east, so that the bottom of the wheels are touching the line at 0.

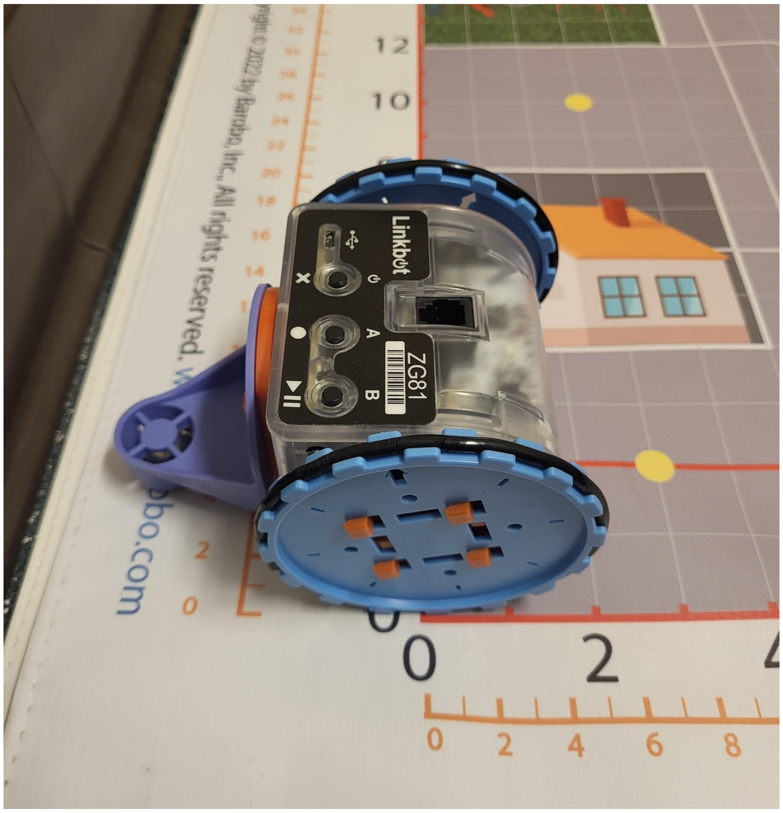
Objectives

1. Use a loop to move forward 8 x 4 inches (8 inches 4 times), pausing 1 second between each movement. At the end, pause for 3 seconds and play the G4 note.
2. Use a loop to reverse 30 inches, pausing for 1 second every 6 inches. At the end, pause for 3 seconds and play an A4 note the number of times the robot had to travel 6 inches.
3. Return to 0, pause for 2 seconds, then play the B4 note a number of times equal to the distance the robot traveled.
4. Move forward 3 x 5 inches, reverse 1/3rd of the distance the robot traveled, then play the C4 note.

#	Description of Scoring Criteria	Points
1	Move 8 X4, pause, and play the note..	20
2	Reverse 30 inches, pause, and play the note.	20
3	Return to 0, pause, and play the note.	10
4	Move 3 x 5, reverse 1/3rd, and play the note.	20

Run	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Score																			

Linkbot Starting Position



Grades 5-6 (Division D)

Treasure Island Navigation 70 Points

Background

Linkbot's GPS system has malfunctioned! Now, we must navigate using our math skills to return to the beach safely. Move around treasure island to exact coordinates, send your SOS signals, and search for buried treasure.

Setup

➢ Linkbot begins at point A (28, 26), in the center of the mat facing east.

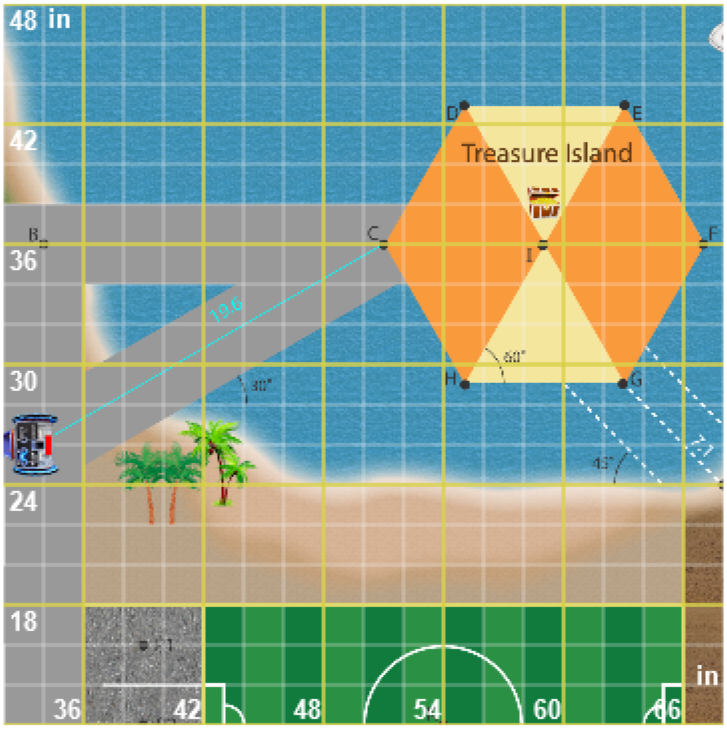
Objectives

1. Drive to coordinate (42, 42), pause for 1 second, and send your SOS by playing 9 A4 notes. The first 3 notes should be eighth notes, the next 3 notes should be quarter notes, and the final 3 notes should be eighth notes again.
2. Drive to (($20\frac{8}{1}$ x $3\frac{2}{4}$), 42), pause for 1 second, and send your SOS.
3. Drive to ($55\,\frac{9}{3}$, 22), pause for 1 second, and send your SOS.
4. Drives towards (36, 22), 5 inches at a time, pausing for 1 second after each movement to look for treasure.

#	Description of Scoring Criteria	Points
1	Drive to (42, 42), send SOS	20
2	Drive to (($20\frac{8}{1}$ x $3\frac{2}{4}$), 42), send SOS	20
3	Drive to ($55\frac{9}{3}$, 22), send SOS	10
4	Drive towards (36, 22), pause, and send SOS.	20

Run	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Score																			

Linkbot Starting Location



Grades 7-8 (Division C)

Interpreting Graphs and Charts 60 points

Background

In this challenge, use the graph of a Linkbot's movement to drive around RoboTown to specific distances and coordinates, pause, then blink the LED.

Setup

➢ Linkbot begins at (0, 2) in the bottom left corner of the mat facing north.

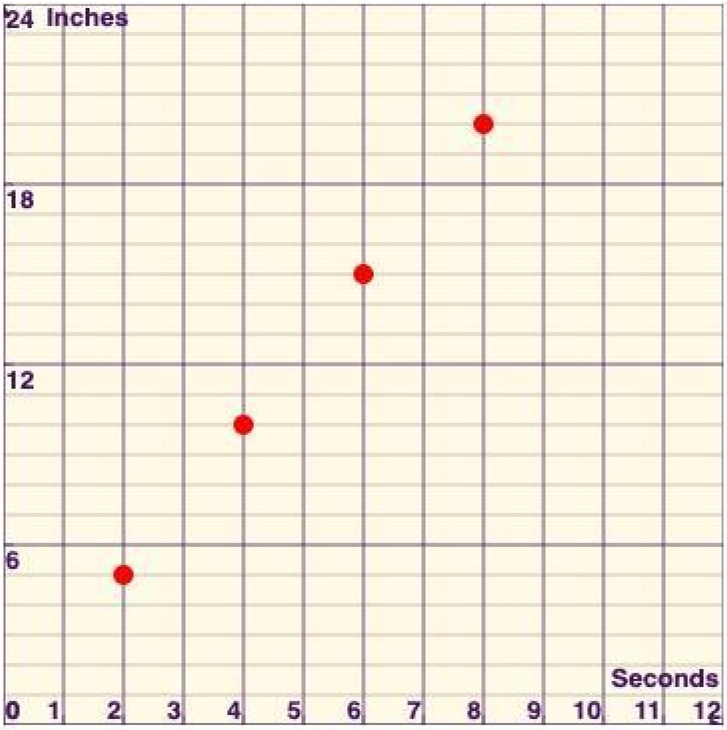
Objectives

1. Move the Linkbot for 8 seconds with the motion shown on the graph on the next page (Use a driveDistance() block, not a driveTime block(), to move the Linkbot.). Then pause for 1 second, and blink the LED red.
2. Move back to the starting position (0, 2) in 4 seconds (Use a driveTime() block, not a driveDistance() block). Then pause for 1 second and blink the LED green.

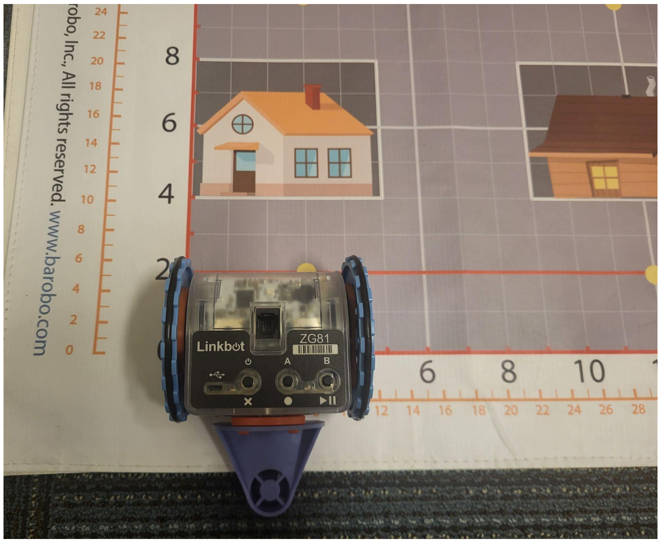
#	Description of Scoring Criteria	Points
1	Move the correct distance, speed, and time forward, pause, and blink the LED.	30
2	Move the Linkbot back to the original position with the correct distance, speed, and time, , pause, and blink the LED.	30

Run	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
Score																			

Graph 1 - Linkbot Motion in Distance versus Time



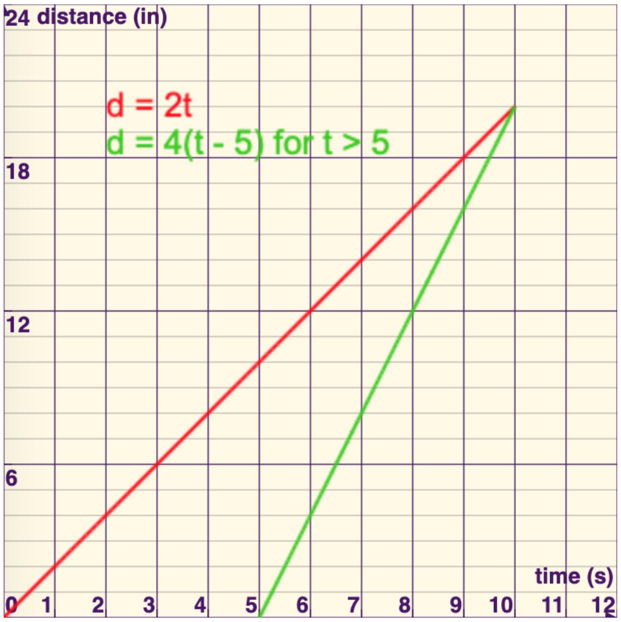
Linkbot Starting Position



Grades 9-12 (Division B)

Linear System of Equations for Motion 70 Points

Background



The graph on the right shows the times and distances that two Linkbots traveled. Your task is to recreate the motion represented in this graph using two Linkbots, making sure that each starts and stops at the correct times at the correct speeds.

Setup

- Linkbot 1 (the red line) starts with its wheels touching the red line in the bottom left corner of the mat, at position (2, 0), facing north.
- Linkbot 2 (the green line) starts with its wheel touching the red line at position (10, 0), facing north.

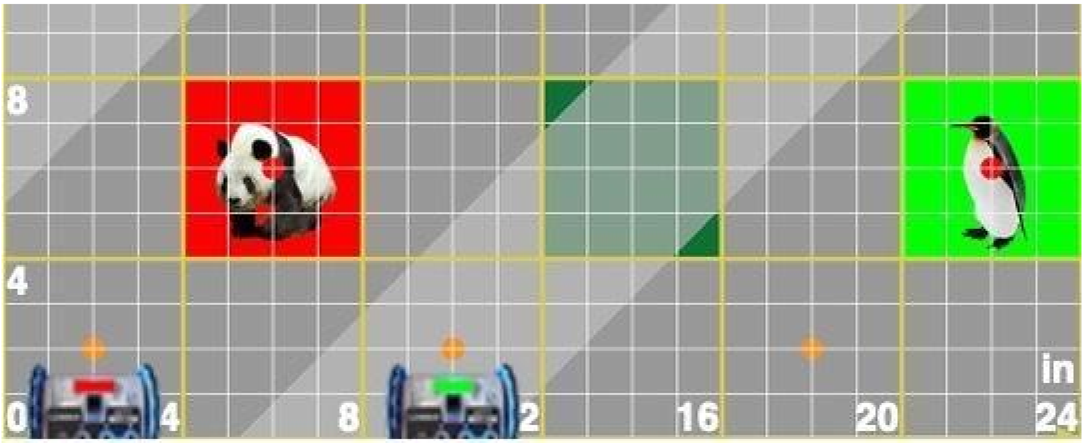
Objectives

1. Linkbot 1 moves the correct time, speed, and distance, as represented on the chart.
2. Linkbot 2 moves the correct time, speed, and distance, as represented on the chart.

#	Description of Scoring Criteria	Points
1	Linkbot 1 and 2 start at the correct times.	30
2	Linkbot 1 and 2 move at the correct speeds.	30
3	Linkbot 1 and 2 stop at the correct distance.	10

Run	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Score																			

Linkbot Starting Positions



Grade 9-12 (Division A)

Fly to Mars 60 Points

Background

The Linkbot Starship is preparing to take its maiden trip to Mars, and you are the lead engineer on the base. Calculate the distance and orientation angle to fly to Mars.

Setup

- ➢ Linkbot begins at (50, 30), in the center of Earth, facing the asteroid.

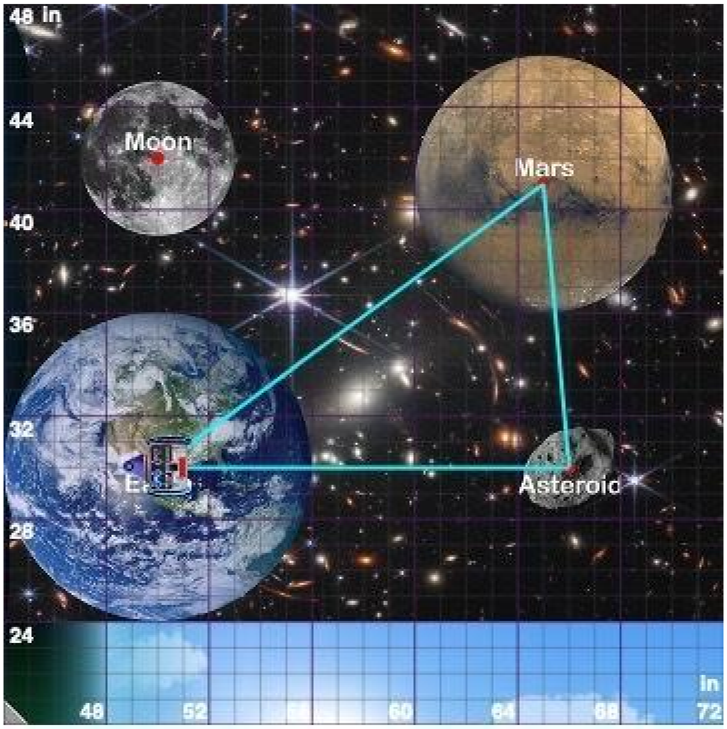
Objectives

1. Calculate the distance between the Earth and Mars using the information on the graph shown on the next page and the RoboExploration mat.
2. Calculate the angle the Starship needs to rotate to fly from the Earth to Mars.
3. Countdown to takeoff from 10 by blinking the LED any color each second for 10 seconds.
4. Rotate the Starship the correct amount of degrees, and drive to Mars (Note: you cannot use the command drivexy() or drivexyTo()).

#	Description of Scoring Criteria	Points
1	Drive the correct distance.	20
2	Rotate to the correct angle.	20
3	Countdown with LEDs and fly to Mars.	20

Run	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Score																			

Linkbot Starting Position



Grade 7-12 (No Division)

Programming with User Input 60 Points

Background

The Linkbot is stranded on Treasure Island and needs to send its rescue signal at certain spots on the island to be rescued. Create a program that is able to navigate to three corners of Treasure Island randomly chosen by the judge at the start of the program. Use the commands prompt() in RoboBlocky or scanf() in Ch to input spots on Treasure Island. Then have your robot drive to each spot, pause, and beep its horn.

Setup

- Linkbot starts in the center of Treasure Island at (60, 36) facing east.
- Program your Linkbot to drive to 3 of the following locations chosen randomly at the start of your program: point C (36, 51), point D (55, 43), point E (63, 43), point F (67, 36), point G (63, 29), or point H (55, 29).

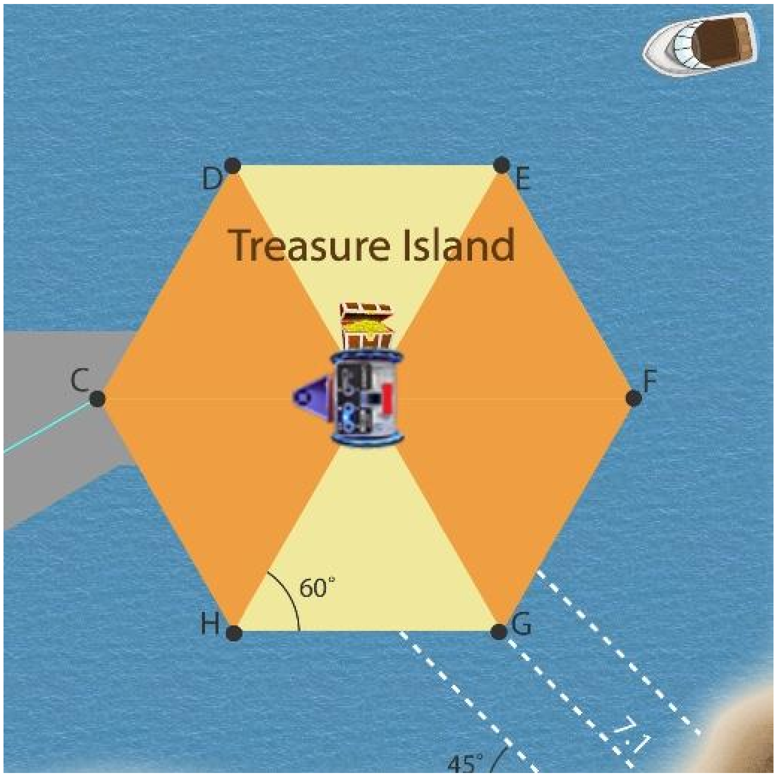
Objectives

1. Drive to the first point chosen randomly, pause for 1 second, and play the C5 note.
2. Drive to the second point chosen randomly, pause for 1 second, and play the C5 note.
3. Drive to the third point chosen randomly, pause for 1 second, and play the C5 note.

#	Description of Scoring Criteria	Points
1	Drive to the first point, pause, and beep.	20
2	Drive to the second point, pause, and beep.	20
3	Drive to the third point, pause, and beep.	20

Run	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Score																			

Linkbot Starting Position



Note on Random Values:

- Input random values into the program at the beginning of each run using the prompt() or scanf() commands.
- Random values change at the start of every run. Refer to the Table Judge, who will display and announce the relevant values for each run. Judges will use a random number generator to determine the random values for each run of the program and give the generated random values to the team.
- Enter your random values only after pressing "Run". Step away from the computer after inputting the values.
- Do not strategically abort your challenge to get better random values. Judges may ban teams that abort challenges from participating in the remainder of the competition period.

Notes

UCDAVIS

C–STEM Center

http://c-stem.ucdavis.edu

California Department of EDUCATION

NSF

Barobo

SoftIntegration®

UCDAVIS

COLLEGE OF ENGINEERING

About RoboPlay Challenge: http://c-stem.ucdavis.edu/roboplay/