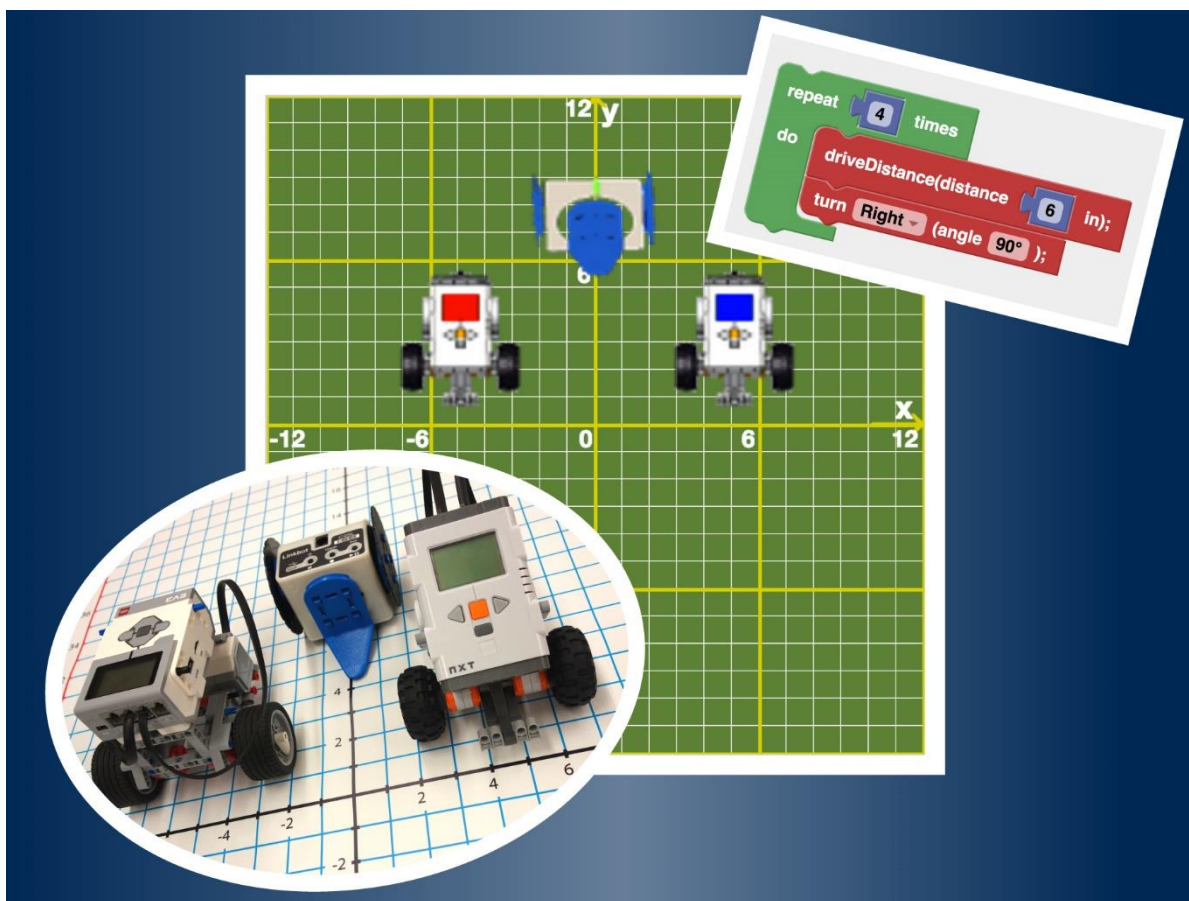


Learning Coding and Math with RoboBlockly

Teaching Resource for Robotics Level 1 Activities



Harry H. Cheng

UC Davis Center for Integrated Computing and STEM Education (C-STEM)

<http://c-stem.ucdavis.edu>

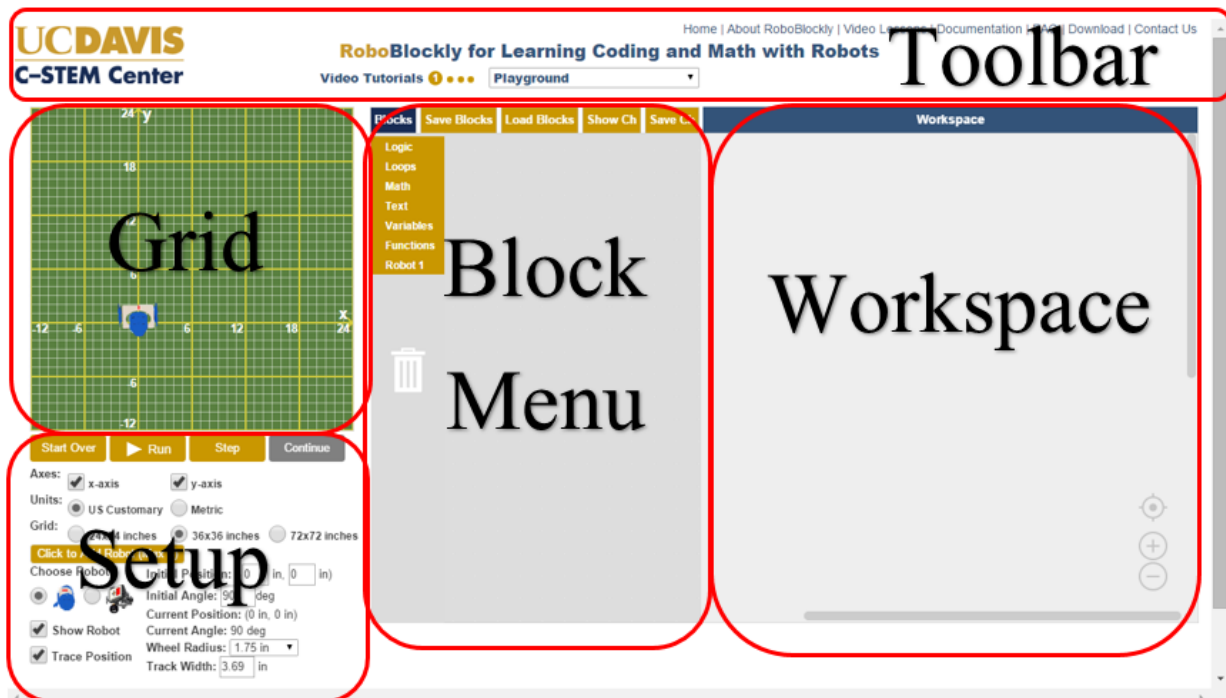
<http://roboblockly.ucdavis.edu>

Learning Coding and Math with RoboBlockly

Teaching Resource for Robotics Level 1 Activities

Harry H. Cheng

UC Davis Center for Integrated Computing and STEM Education (C-STEM)
University of California-Davis



Copyright © 2015, UC Davis C-STEM Center, All rights reserved. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written consent of the author, including, but not limited to, in any network or other electronic storage or transmission, or broadcast for distance learning.

Acknowledgements:

This work was supported in part by the National Science Foundation under grant numbers CNS-1132709, IIS-1208690, IIS-1256780, and by the CaMSP Program of the California Department of Education.

Additional Contributor(s):

Kayce Mastrup

How to Use RoboBlockly Activities in your Classroom

What is the purpose of RoboBlockly?

In RoboBlockly, students program a robot using a user-friendly drag-and-drop method. Students will complete basic problem solving to move a robot or multiple robots.

RoboBlockly is built to allow students to work at their own pace, but in general each problem should take approximately 5 minutes to complete each activity. We encourage you to give students additional time if needed or make it clear that they don't need to finish the entire set of activities during one class session.

General Computer Usage Requirements

Technology Requirements: Any modern browser on computers, laptops, tables, or smartphones with any type of operating system.

It is important to know that every browser functions differently. We encourage you to test RoboBlockly on the computers you will have students using *before* implementation. Please test out the following: audio and video streaming quality, default browser specific mechanisms for saving blocks and saving Ch code, etc. all so you are better able to support your students. Make sure that pop ups have been enabled on all computers.

You may wish to provide headphones or ask students to bring headphones to allow students to independently watch tutorial videos.

Prepare yourself

Go through the activities yourself so that you are familiar with what your students will be experiencing. The Teacher Resource Packet contains all the activities and solutions for the pathway. Please note that the activities build on previous activities in each pathways such that students may need to complete some or all activities prior to the activity selected.

- 1) Determine the purpose for students using RoboBlockly:
 - To support student learning in Math,
 - To support student learning in Computer Programming,
 - To support student learning in Robotics.

- 2) Based on your purpose, determine what additional resources your students will need for instruction. We do not recommend using RoboBlockly to introduce a mathematical concept but to rather use it for skill building or as a culminating performance task.

Use as skill building: We recommend that you provide your students with a worksheet that includes important related definitions, work space, leading questions, etc. and encourage your students to refer to their class notes which cover these topics.

Use as a culminating performance task: Carefully select which activity directly relates to the content you have taught, making note that the previous activities may be necessary to complete to build prior knowledge.

Prepare your students

Help students get excited about RoboBlockly by inspiring students and discussing how computer science impacts every part of our lives. As a class, list things that use code in everyday life, or discuss different ways technology impacts our lives etc.

When using RoboBlockly in class, first demonstrate to students how to navigate and use the RoboBlockly website. There are five Video Tutorials, along with a self-guided interactive non-video tutorial which should be used to help familiarize your students with the different functionalities of RoboBlockly. Helping students understand the functionality of RoboBlockly and which elements can be manipulated in which manners is very important to ensuring your students have full access to the content.

Pre-Requisite Skills

Math

- Number Sense – performing all mathematical operations on positive whole numbers within 100.
- Geometry – basic area and perimeter problems relating to simple geometric shapes (rectangles, polygons)
- Length – relating addition and subtraction to length.

Computer

- Basic computer skills:
 - Drag and drop using a mouse
 - Key boarding
 - Navigating a web browser
 - Zoom In/Out in a browser
 - Disabling or enabling pop-up windows
 - Adjusting volume for videos

Extension

Using hardwired robots, Linkbot Controller, RoboSim or Robot Controller to execute programs built in RoboBlockly.

All can be downloaded from the UC Davis C-STEM Center’s webpage: <http://c-stem.ucdavis.edu/downloads/>

Learning Coding and Math with RoboBlockly

Teaching Resource for Robotics Level 1 Activities

Table of Contents: Activity to Block Alignment

Activity	Blocks Used
1. Drive Distance: Moving Forward	driveDistance
2. Drive Distance: Moving Forward	driveDistance
3. Drive Time: Moving Forward	driveTime
4. Moving Forwards then Backwards Using Different Blocks	driveTime driveDistance
5. Turning Left or Right in the Coordinate Plane	driveDistance turnRight or turnLeft
6. Turning Left/Right – A Real World Example	driveDistance turnRight or turnLeft
7. Multiple Robots: Moving Forward	driveDistance
8. Multiple Robots: Moving Forward and Turning	driveDistance driveTime turnRight

Table of Contents: Textbook to Activity Alignment

<i>Learning Robot Programming with Linkbot for the Absolute Beginner 5th Edition</i>	Activities							
	1	2	3	4	5	6	7	8
Section 5.4 Move a Distance for a Two-Wheel Robot	X	X		X	X	X	X	X
Section 5.6 Turn Left and Turn Right					X	X		X
Section 9.2 Move a Two-Wheel Robot with the Specified Time			X					X
Section 12.1 Control Multiple Linkbots Using the Robot Control Pane							X	X
Section 12.2 Control Multiple Linkbots Using a Program							X	X
Section 12.5 Move Multiple Linkbots with Specified Distances or Joint Angle							X	X

Activity #1 Drive Distance: Moving Forward

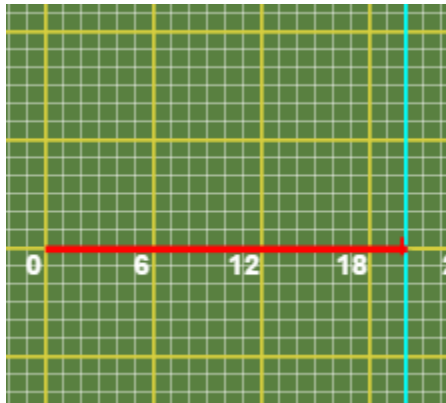
Objective: Students will use the driveDistance block to move the robot a specified distance.

RoboBlockly Student Activity:

Video Lesson	R1. Drive a Distance: driveDistance()
Initial Student Prompt	<p>Drive Distance</p>  <p>This block drives the robot forward. The number in the blue block determines how far the robot will travel.</p>
Pre-Placed Blocks	
Problem Statement	Hello, I am Linkbot/Lego Mindstorm. Please help me move forward a distance of 20 units.
Wrong Prompt	I did not move a total of 18 units. Please try again. Use the hint or watch the video if you are having trouble.
Hint	 <p>The preplaced block drives the robot 5 inches forward because the number in the blue box is 5. You want to drive the robot 20 inches.</p> <p>You can uncheck <input type="checkbox"/> Show Robot to more easily view the distance the robot travels, to help you count.</p>
Possible Solution in Ch	<pre>#include <linkbot.h> CLinkbotI robot; double radius = 1.75; robot.driveDistance(20, radius);</pre>

Activity #1 Drive Distance: Moving Forward

Picture of solution in RoboBlockly



Problem Statement:

Hello, I am Linkbot/Lego Mindstorm. Please help me move forward a distance of 20 units.

```
driveDistance(distance 20 in);
```

Location of solution for “Load Blocks” tab in RoboBlockly

C-STEM Studio -> Teaching Resources -> TeachLinkbot->RoboBlocklySolution->r1.xml

C-STEM text alignment: *Robot Programming with Linkbot for the Absolute Beginner, 5th edition*

a) Section 5.4 Move a Distance for a Two-Wheel Robot. (driveDistance block)

Activity #2 Drive Distance: Moving Forward

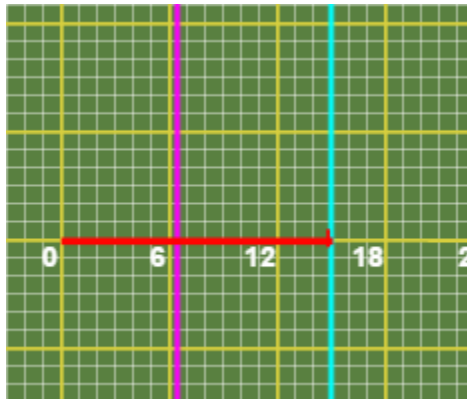
Objective: Students will use the driveDistance block to move the robot a specified distance.

RoboBlockly Activity:

<i>Initial Student Prompt</i>	You will need to move the robot forward again, but this time there will be no preplaced block. You will need to draw the driveDistance() block from the block menu to the workspace.
<i>Pre-Placed Blocks</i>	None.
<i>Problem Statement</i>	Use two different driveDistance() blocks to move the robot forward. First move the robot 6.5 inches, then move the robot 8.5 inches. You need to drag the block over to workspace and attach them like puzzle pieces.
<i>Wrong Prompt</i>	You did not move the robot the two correct distances. Please try again. Use the hint if you are feeling stuck.
<i>Hint</i>	The value in the block needs to be changed to equal the amount you want the robot to move. There are no preplaced blocks in this problem, so you need to click and drag over two driveDistance() blocks and change their values.
<i>Possible Solution in Ch</i>	<pre>#include <linkbot.h> CLinkbotI robot; double radius = 1.75; robot.driveDistance(6.5, radius); robot.driveDistance(8.5, radius);</pre>

Activity #2 Drive Distance: Moving Forward

Picture of solution in RoboBlockly



Problem Statement:

Use two different driveDistance() blocks to move the robot forward. First move the robot 6.5 inches, then move the robot 8.5 inches. You need to drag the block over to workspace and attach them like puzzle pieces.

```
driveDistance(distance 6.5 in);
```

```
driveDistance(distance 8.5 in);
```

Location of Solution for “Load Blocks”

C-STEM Studio -> Teaching Resources -> TeachLinkbot->RoboBlocklySolution->r2.xml


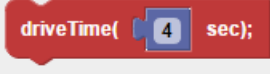
C-STEM text alignment: [Robot Programming with Linkbot for the Absolute Beginner, 5th edition](#)

a) Section 5.4 Move a Distance for a Two-Wheel Robot. (driveDistance block)

Activity #3 Drive Time: Moving Forward

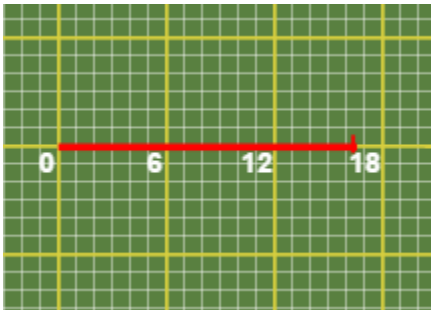
Objective: Students will apply their understanding of multiplication and use a robot to model the product of two numbers.

RoboBlockly Activity:

<i>Video Lesson</i>	R3. Drive for a Certain Time: driveTime()
<i>Initial Student Prompt</i>	<p>Drive Time</p>  <p>This block drives the robot for an amount of seconds specified by the number in the blue block.</p>
<i>Pre-Placed Blocks</i>	
<i>Problem Statement</i>	Please help me drive forward for 12 seconds long.
<i>Wrong Prompt</i>	Drive the robot for 12 seconds long. Please try again. Use the hint if you are feeling stuck.
<i>Hint</i>	 <p>This block drives for 4 sec long because the number in the blue box is 4. You need to change the number in the blue box.</p>
<i>Possible Solution in Ch</i>	<pre>#include <linkbot.h> Linkbot robot; robot.driveTime(12);</pre>

Activity #3 Drive Time: Moving Forward

Picture of solution in RoboBlockly



Problem Statement:

Please help me drive forward for 12 seconds long.

```
driveTime(time 12 sec);
```

Location of Solution for “Load Blocks”

C-STEM Studio -> Teaching Resources -> TeachLinkbot->RoboBlocklySolution->r3.xml

C-STEM text alignment: *Robot Programming with Linkbot for the Absolute Beginner, 5th edition*

a) Section 9.2 Move a Two-Wheel Robot with the Specified Time (driveTime block)

Activity #4 Moving Forward then Backwards Using Different Blocks

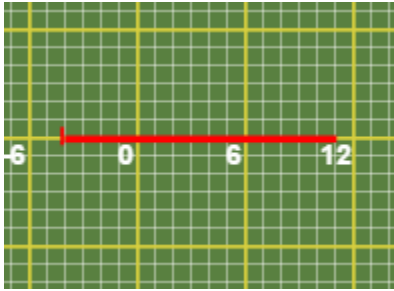
Objective: Students will use two different drive blocks to move the robot forward then backwards.

RoboBlockly Activity:

<p><i>Initial Student Prompt</i></p>	<p>You have now learned how to move the robot forwards using two different blocks:</p> <div style="text-align: center;">  and  </div> <p>You can make the robot move backwards by putting a minus sign in front of your number in either of the blocks.</p> <div style="text-align: center;">  </div>
<p><i>Pre-Placed Blocks</i></p>	<p>None.</p>
<p><i>Problem Statement</i></p>	<p>Use the driveTime() and driveDistance() blocks to move the robot forward then backwards. First drive the robot forward for 8 sec. Then drive the robot backwards 15 inches.</p>
<p><i>Wrong Prompt</i></p>	<p>You did not move the robot the correct distance. Try again. Please see the hint if you need help.</p>
<p><i>Hint</i></p>	<p>Remember the driveTime() block moves the robot for a certain time. To move the robot backwards you have to put a minus “-” sign in front of the number in the block.</p> <div style="text-align: center;">  </div>
<p><i>Possible Solution in Ch</i></p>	<pre>#include <linkbot.h> CLinkbotI robot; robot.driveTime(8); robot.driveDistance(-15);</pre>

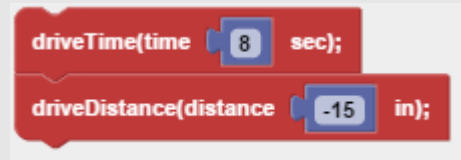
Activity #4 Moving Forward then Backwards Using Different Blocks

Picture of solution in RoboBlockly



Problem Statement:

Use the driveTime() and driveDistance() blocks to move the robot forward then backwards. First drive the robot forward for 8 sec. Then drive the robot backwards 15 inches.



Location of Solution for “Load Blocks”

C-STEM Studio -> Teaching Resources -> TeachLinkbot->RoboBlocklySolution->r4.xml


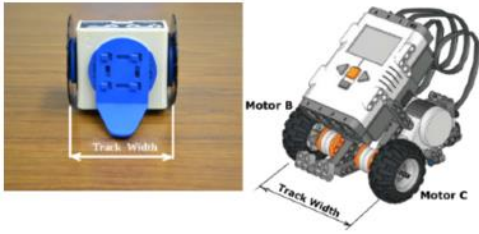

C-STEM text alignment: *Robot Programming with Linkbot for the Absolute Beginner, 5th edition*

- a) Section 5.4 Move a Distance for a Two-Wheel Robot. (driveDistance block)
- b) Section 9.2 Move a Two-Wheel Robot with the Specified Time (driveTime block)

Activity #5 Turning Left or Right in the Coordinate Plane

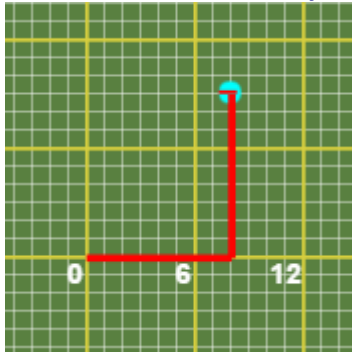
Objective: Students will use the turn and drive blocks to move the robot to different locations.

RoboBlockly Activity:

<i>Video Lesson</i>	R5. Turn Left and Turn Right: turnLeft() and turnRight()
<i>Student Initial Prompt</i>	<p>Turn Left/Right</p>  <p>This block changes the direction of the robot using degrees.</p> <p>The track width is the distance between the two wheels as shown below.</p> 
<i>Pre-Placed Blocks</i>	
<i>Problem Statement</i>	Change the values in the preplaced blocks to drive the robot to the dot. Each white line equals one unit.
<i>Wrong Prompt</i>	You did not drive the robot to the dot. Please try again. Use the hint if you are stuck.
<i>Hint</i>	Count the number of white lines to determine how far to drive the robot before turning. You then need to move the robot up (vertical), do you need to change the way the robot turns? Then count the white lines to determine how far to drive the robot.
<i>Possible Solution 1 in C++</i>	<pre>#include <linkbot.h> CLinkbotI robot; double radius = 1.75; double trackwidth = 3.69; robot.driveDistance(8, radius); robot.turnLeft(90, radius, trackwidth); robot.driveDistance(9, radius);</pre>

Activity #5 Turning Left or Right in the Coordinate Plane

Picture of solution in RoboBlockly



Problem Statement:

Change the values in the preplaced blocks to drive the robot to the dot. Each white line equals one unit.



Location of Solution for “Load Blocks”

C-STEM Studio -> Teaching Resources -> TeachLinkbot->RoboBlocklySolution->r5.xml

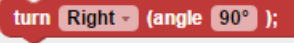
C-STEM text alignment: *Robot Programming with Linkbot for the Absolute Beginner, 5th edition*

- Section 5.4 Move a Distance for a Two-Wheel Robot. (driveDistance block)
- Section 5.6 Turn Left and Turn Right (turn block)

Activity #6 Turning Left/Right – A Real World Example

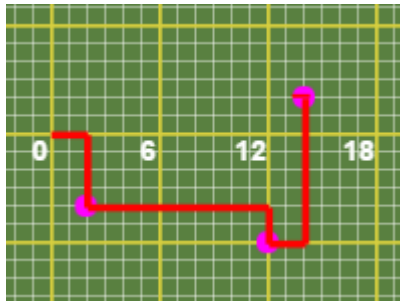
Objective: Students will program the robot to complete a real world example.

RoboBlockly Activity:

<i>Initial Student Prompt</i>	Turn Left/Right Use the  to have the robot model the following real-world example.
<i>Pre-Placed Blocks</i>	None.
<i>Problem Statement</i>	The robot has gotten a job delivering newspapers. Each dot represents a house that the robot must deliver a paper to. Use driveDistance() and turnRight/Left() blocks to drive the robot on the newspaper route.
<i>Wrong Prompt</i>	You did not make it to all the correct stopping points.
<i>Hint</i>	Remember each white line represents one unit. Use the lines to help you determine of far to drive the robot before turning. You can complete the route in any order you want, just make sure to stop at all the dots.
<i>Possible Solution in Ch</i>	<pre>#include <linkbot.h> CLinkbotI robot; double radius = 1.75; double trackwidth = 3.69; robot.driveDistance(2, radius); robot.turnRight(90, radius, trackwidth); robot.driveDistance(4, radius); robot.turnLeft(90, radius, trackwidth); robot.driveDistance(10, radius); robot.turnRight(90, radius, trackwidth); robot.driveDistance(2, radius); robot.turnLeft(90, radius, trackwidth); robot.driveDistance(2, radius); robot.turnLeft(90, radius, trackwidth); robot.driveDistance(8, radius);</pre>

Activity #6 Turning Left/Right – A Real World Example

Picture of solution in RoboBlockly



Problem Statement:

The robot has gotten a job delivering newspapers. Each dot represents a house that the robot must deliver a paper to. Use `driveDistance()` and `turnRight/Left()` blocks to drive the robot on the newspaper route.



Location of Solution for “Load Blocks”

C-STEM Studio -> Teaching Resources -> TeachLinkbot->RoboBlocklySolution->r6.xml


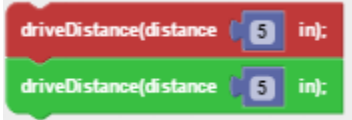
C-STEM text alignment: *Robot Programming with Linkbot for the Absolute Beginner, 5th edition*

- Section 5.4 Move a Distance for a Two-Wheel Robot. (`driveDistance` block)
- Section 5.6 Turn Left and Turn Right (`turn` block)

Activity #7 Multiple Robots: Moving Forward

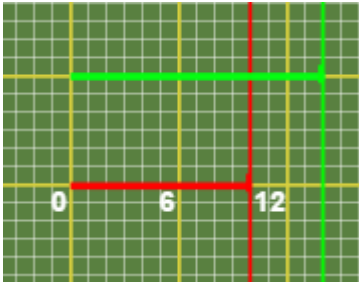
Objective: Students will move multiple robots using the blocks they've learned thus far.

RoboBlockly Activity:

<i>Video Lesson</i>	R8. Program Multiple Robots of Linkbot and Lego Mindstorms
<i>Initial Student Prompt</i>	<p>More Robots You can run up to four robots in RoboBlockly at one time.</p> <p>More robots can be added by clicking  in the web page. For this problem two robots have already been added.</p> <p>Each robot will have a different color block that represents their coding.</p>
<i>Pre-Placed Blocks</i>	
<i>Problem Statement</i>	Move Robot 1: 10 inches forward and move Robot 2: 14 inches forward.
<i>Wrong Prompt</i>	You did not move one or both of the robots the correct distance. Please try again.
<i>Hint</i>	Remember, the number in the blue block determines how far the robot will travel.
<i>Possible Solution in Ch</i>	<pre>#include <linkbot.h> CLinkbot robot1; double radius1 = 1.75; CLinkbot robot2; double radius2 = 1.75; robot1.driveDistance(10); robot2.driveDistance(14);</pre>

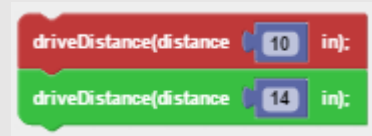
Activity #7 Multiple Robots: Moving Forward

Picture of solution in RoboBlockly



Problem Statement:

Move Robot 1: 10 inches forward and move Robot 2: 14 inches forward.



Location of Solution for “Load Blocks”

C-STEM Studio -> Teaching Resources -> TeachLinkbot->RoboBlocklySolution->r7.xml

C-STEM text alignment: *Robot Programming with Linkbot for the Absolute Beginner, 5th edition*

- a) Section 12.1 Control Multiple Linkbots Using the Robot Control Panel
- b) Section 12.2 Control Multiple Linkbots Using a Program
- c) Section 5.4 Move a Distance for a Two-Wheel Robot. (driveDistance block)

Activity #8 Multiple Robots: Moving Forward and Turning

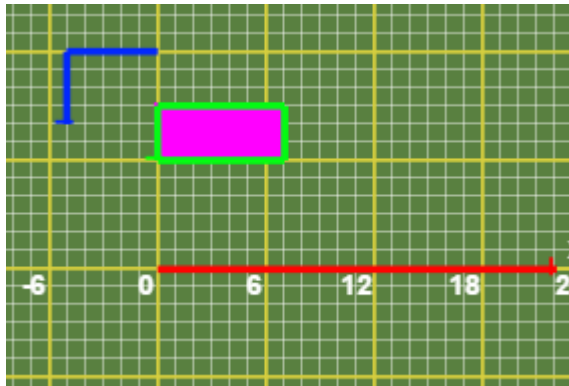
Objective: Students will program robots to move according to a given set of parameters.

RoboBlockly Activity:

<i>Initial Student Prompt</i>	You will need to follow the directions to add additional robots and program them to run accordingly.
<i>Pre-Placed Blocks</i>	None.
<i>Problem Statement</i>	Add two additional robots. Robot 1: Drive forwards for 16 sec. Robot 2: Drive around the pink rectangle. Robot 3: Move backwards 5 units then turn right and move 4 units.
<i>Wrong Prompt</i>	You did not program one or more of the robots correctly. Please try again. Use the hint if you are feeling stuck.
<i>Hint</i>	First you must add two additional robots by clicking:  2 times. Then each robot has different colored blocks, they are placed and programmed on the same workspace. Make sure to connect all your blocks and follow the directions carefully.
<i>Possible Solution in Ch</i>	<pre>#include <linkbot.h> CLinkbotI robot1; CLinkbotI robot2; double radius2 = 1.75; double trackwidth2 = 3.69; CLinkbotI robot3; double radius3 = 1.75; double trackwidth3 = 3.69; robot1.driveTime(16); robot2.driveDistance(7, radius2); robot2.turnLeft(90, radius2, trackwidth2); robot2.driveDistance(3, radius2); robot2.turnLeft(90, radius2, trackwidth2); robot2.driveDistance(7, radius2); robot2.turnLeft(90, radius2, trackwidth2); robot2.driveDistance(3, radius2); robot3.driveDistance(-5, radius3); robot3.turnRight(90, radius3, trackwidth3); robot3.driveDistance(4, radius3);</pre>

Activity #8 Multiple Robots: Moving Forward and Turning

Picture of solution in RoboBlockly



Problem Statement:

Add two additional robots.

Robot 1: Drive forwards for 16 sec.

Robot 2: Drive around the pink rectangle.

Robot 3: Move backwards 5 units then turn right and move 4 units.



Location of Solution for “Load Blocks”

C-STEM Studio -> Teaching Resources -> TeachLinkbot->RoboBlocklySolution->r8.xml

C-STEM text alignment: *Robot Programming with Linkbot for the Absolute Beginner, 5th edition*

- Section 12.1 Control Multiple Linkbots Using the Robot Control Panel
- Section 12.2 Control Multiple Linkbots Using a Program
- Section 5.4 Move a Distance for a Two-Wheel Robot. (driveDistance block)
- Section 5.6 Turn Left and Turn Right (turn block)
- Section 9.2 Move a Two-Wheel Robot with the Specified Time (driveTime block)