Coding with Linkbot and Mindstorms

User's Guide

UC Davis

C-STEM Center

Table of Contents

1 Introduction

"Coding with Linkbot and Mindstorms" is an online learning tool that is currently designed to teach coding and mathematics using a visual programming interface. Students are able to simplify coding by using puzzle blocks in place of typing code manually. Students are then able to run the code to control a virtual robot or directly generate Ch code from the blocks.

This Ch code can then be used with a separate C-STEM program called RoboSim to run a separate simulation. Furthermore, when the Linkbot or Mindstorms hardware is connected, and the code is loaded, the generated Ch code can also move the robot itself. Currently, there are ten coding and ten math lessons available on "Coding with Linkbot and Mindstorms," each introducing a new concept in coding and math respectively as the user progresses through them. Several tutorial videos are also available introducing the website and explaining how the different parts of the website work.

Section 2: "How to Get Started" describes the requirements needed to run the website and how to use all features available on the website. Section 3: "Playground" describes the Playground resource and how it can be used. Section 4: "Documentation on Linkbot/Mindstorms Blocks" describes each of the blocks that are available for use in the "Coding with Linkbot and Mindstorms" lessons.

2 How to get started

Supported Browser(s):

Currently the only supported browser is:

Google® Chrome™

To begin, go to the "Coding with Linkbot and Mindstorms" web page at computing.ucdavis.edu/blockly to begin the first lesson.

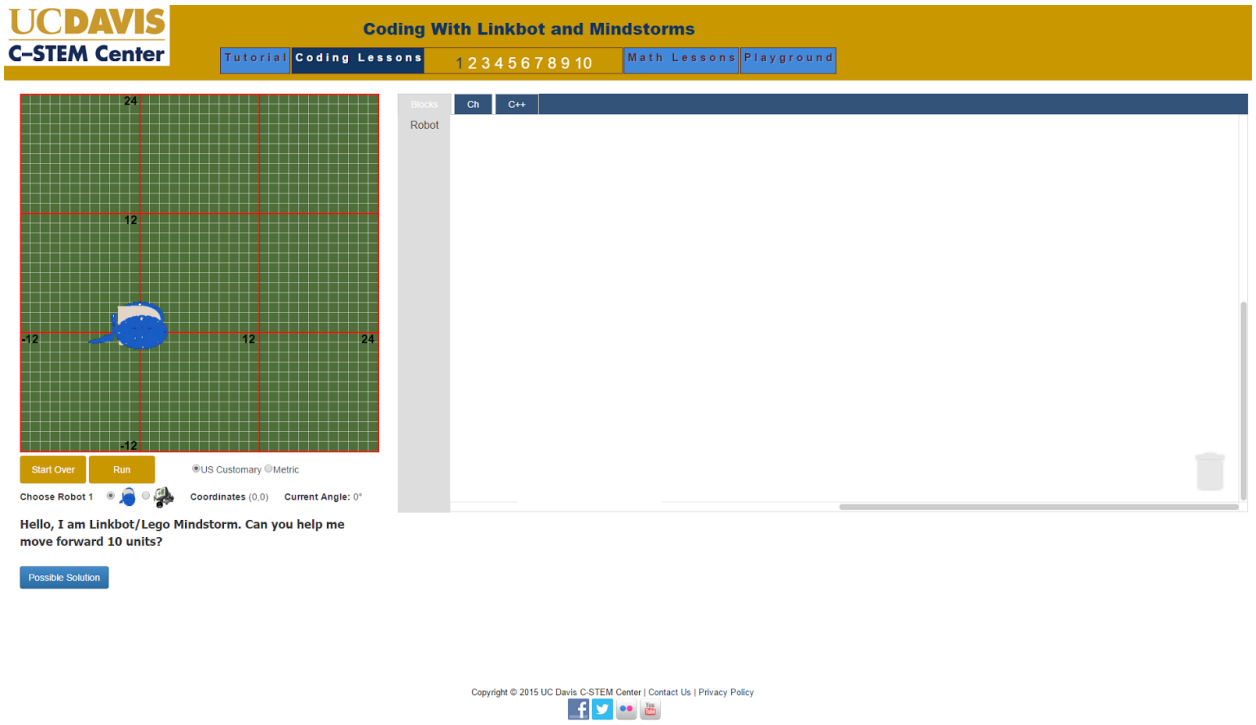


Figure 1: Home page of "Coding with Linkbot and Mindstorms"

From there, the website tutorials can be accessed by clicking the tab to the left of the Coding Lessons tab. By clicking this tab, you will be able to access a sub tab with links to several brief videos that are meant to guide users on how to use the website.

By selecting the Coding Lessons or Math Lessons tabs, it will open up sub tabs that allow you to access the ten coding and math lessons respectively. There are no

restrictions as to the order in which you must complete these lessons. Any lesson may be revisited, and the user may choose to skip ahead to another lesson if desired. By clicking the Playground tab, the user will be able to access the submenu which includes the link to the Playground, which will be described later. To the left of this menu, we have included a direct link to the University of California, Davis C-STEM center website that can be accessed by clicking the UC Davis C-STEM logo.

There are four main parts to the website. The first, was the top most toolbar described above. The second part of this website would be the left side portion below the toolbar, which contains the grid. The grid will display the user's robot of choice (ie. Linkbot or Mindstorms, or both) and its real-time simulated movement based on the blocks used. The distance between the red lines are 12 units, and the distance between each white line is 1 unit. The numbers -12, 0, 12, and 24 are marked along the x- and y-axis at appropriate locations to mark these distances. Directly below the graph are the buttons "Start Over" and "Run." Clicking the "Start Over" button will delete all blocks currently placed in the workspace (which will be described in more detail next), while clicking "Run" will allow the user to run the code that is generated from the blocks that are placed in the workspace. After "Run" has been clicked, it will change into the "Reset" button, which will allow the user to make changes to the blocks in the workspace and rerun the simulation with the new code. To the right of these buttons, the user can view the robot's current position (coordinates) and its current orientation angle. Below this, there are buttons that allow the user to select which robot he or she would prefer to use, as well as the lesson prompt. A possible solution is provided if the user clicks the blue "Possible Solution" button, which can be hidden again by clicking on the button again.

The third part of this website is the right side portion below the toolbar. This is the workspace section of the website. On the left side of this workspace section, there are tabs that lead to various blocks. In the example above, they can be found by clicking Robot. In further lessons, there will be tabs available for logic, math, etc. blocks as well. These blocks can be dragged into the white space for use. Extra or unnecessary blocks can be dragged to the trash can at the bottom right, or back to the left side for deletion. The

tabs at the workspace labeled Ch and C++ can be clicked to show the generated Ch and C++ codes respectively for the blocks used in the workspace.

The fourth part of this website is the bottom portion of the website, below the graph and the workspace as shown in Figure 2.

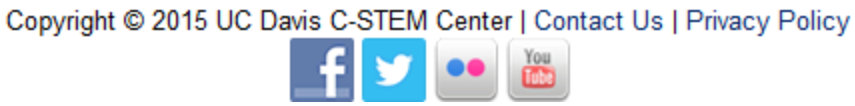


Figure 2: Related links located at the bottom of the webpage

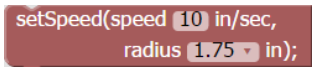
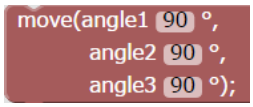
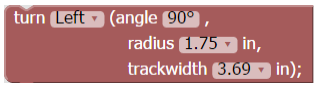
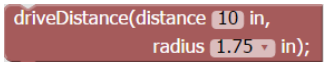
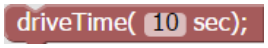
Here a variety of links can be found that will direct you to the UC Davis C-STEM Center's Facebook, Twitter, Flickr, and YouTube pages.

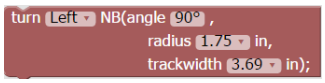
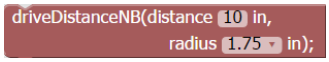
3 Playground

The playground is located at the right of the navigation bar at the top of the website. It contains all available blocks for the user to access unlike the lessons which contain only the blocks necessary to solve each problem. It has the capability to add up to four robots. The user can still specify whether these robots are linkbot or mindstorm as well as their initial position and orientation. Users will not be limited by lessons in the playground (except for graph boundaries).

Here, users can freely experiment, practice, and generate any code they can imagine.

4 Documentation on Linkbot/Mindstorms Blocks

Block	Image	Description
setSpeed (speed, radius)		Sets the speed of a robot with a specific wheel radius.
delaySeconds (seconds)		Delays the next action of a robot by some amount of seconds.
move(angle1, angle2, angle3)		Moves a robot's joints relative to their current position using three angles.
moveWait()		Pauses the program until all joints of a robot have finished moving.
traceOn()		Turns the tracing for a robot on.
traceOff()		Turns the tracing for a robot off.
turnLeft(angle, radius, trackwidth) turnRight(angle, radius, trackwidth)		Turns a robot right/left by a specified amount of degrees.
driveDistance (distance, radius)		Drives a robot by a specified distance.
driveTime(seconds)		Drives a robot for a specified amount of time.
driveAngle(angle)		Rotate the wheels of a robot a specified amount of degrees.

resetToZero()		Resets a robot to its zero position.
turnLeftNB(angle, radius, trackwidth) turnRightNB(angle, radius, trackwidth)		Turns a robot right/left by a specified amount of degrees. (Non-blocking version)
driveDistanceNB (distance, radius)		Drives a robot by a specified distance. (Non-blocking version)
driveTimeNB (seconds)		Drives a robot for a specified amount of time. (Non-blocking version)
driveAngleNB (angle)		Rotate the wheels of a robot a specified amount of degrees. (Non-blocking version)
resetToZeroNB()		Resets a robot to its zero position. (Non-blocking version)

Note: In the files for these blocks, there are copies of each function depending on the unit measurement (customary/metric), number of robot (up to 4) and type of robot (linkbot/mindstorms)

5 Example Lesson Walkthrough (Coding Lesson 3)

"Can you help me move forwards 10 units and then backwards 10 units using the turn block? " - Robot

First, we need to click on the robot tab along the left side of the workspace and drag out a drive distance block. Setting the distance to 10 will allow the robot to move forward 10 units.

Second, we need to drag out the turn block and attach it to the bottom of the drive distance block and set the angle of turn to 180 degrees . This will turn the robot 180 degrees.

Finally, we need to drag out another drive distance block (or right click an existing one and select duplicate) and place it under the turn block. Make sure the distance is also set to 10. This will then move the robot 10 units in the new direction it is facing, which moves it back to its original position. The lesson is complete!