

—

Careers

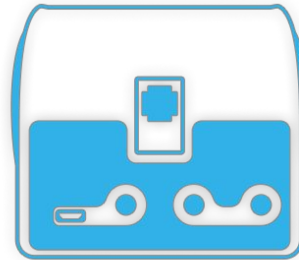
—

Coding & Robotics Update

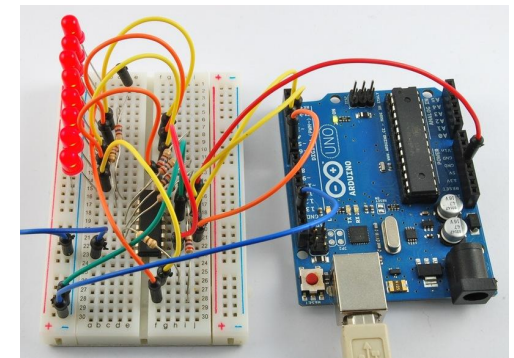
UPDATE

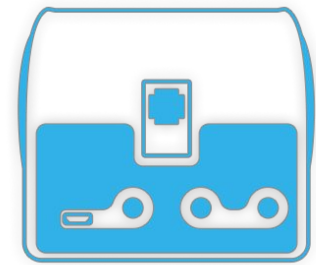
- ❑ School Power Grant 2014/15
- ❑ C-STEM Training
- ❑ Ch Program Language
 - ❑ Computational Programs
 - ❑ Robotic Programs
 - ❑ Arduino Programs

```
1  /* File: grosspay.ch
2  Calculate the gross pay, deducted tax, and net pay.
3  Plot the gross pay versus time.
4  The formula for the gross pay depends on
5  if(0 <= t <= 8)
6      grosspay = 20t
7  else if (8 <= t <= 24)
8      grosspay = 30*t - 80 */
9  #include <chplot.h>
10 /* declare variables */
11 double t, grosspay, netpay, tax;
12 CPlot plot;
13
14 /* get the user input for the hours worked */
15 printf("Welcome to the Wage Calculator\n");
16 printf("Please enter the hours worked\n");
17 scanf("%lf", &t);
18
19 /* process the payment */
20 if(0 <= t && t <= 8) { /* when 0 <= t <= 8 */
21     grosspay = 20*t;
22 }
23 else if(8 < t && t <= 24) { /* when 8 < t <= 24 */
24     grosspay = 30*t - 80;
25 }
26 else { /* when t<0 or t > 24 */
27     printf("Invalid time\n");
28     exit(-1);
29 }
30
31 tax = 0.3*grosspay;
32 netpay = grosspay - tax;
33
34 /* display the payment as output */
35 printf(" gross pay: $%.2lf\n", grosspay);
36 printf("deducted tax: $%.2lf\n", tax);
37 printf(" net pay: $%.2lf\n", netpay);
38
39 /* plot the relation of gross pay versus time */
40 plot.title("Gross pay versus time");
41 plot.label(PLOT_AXIS_X, "time (hours)");
```

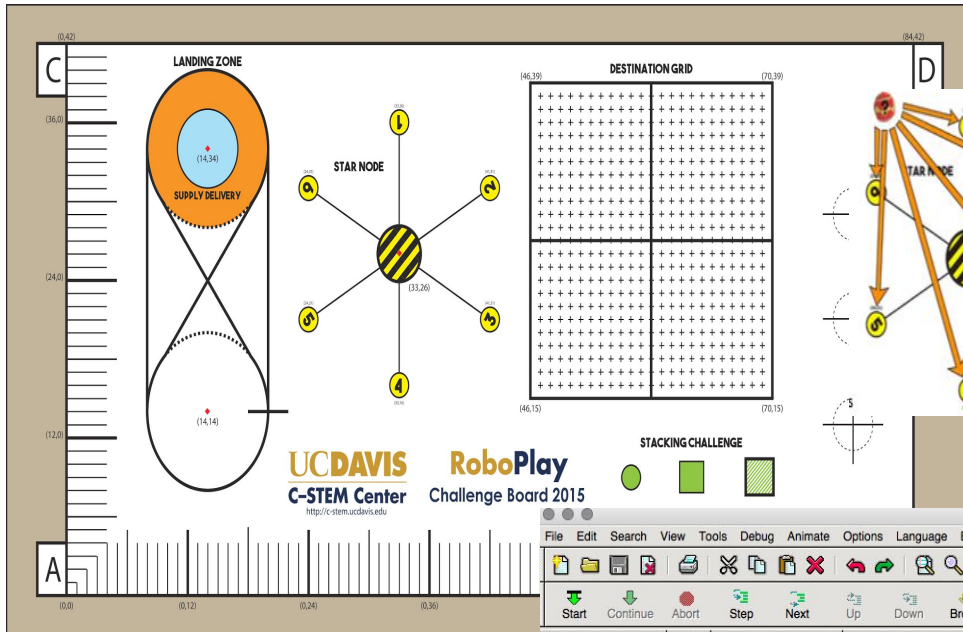


Linkbot Labs



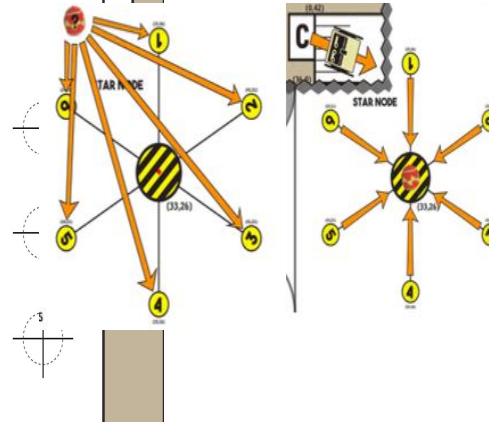


Linkbot Labs



UC DAVIS
C-STEM Center
<http://c-stem.ucdavis.edu>

RoboPlay
Challenge Board 2015



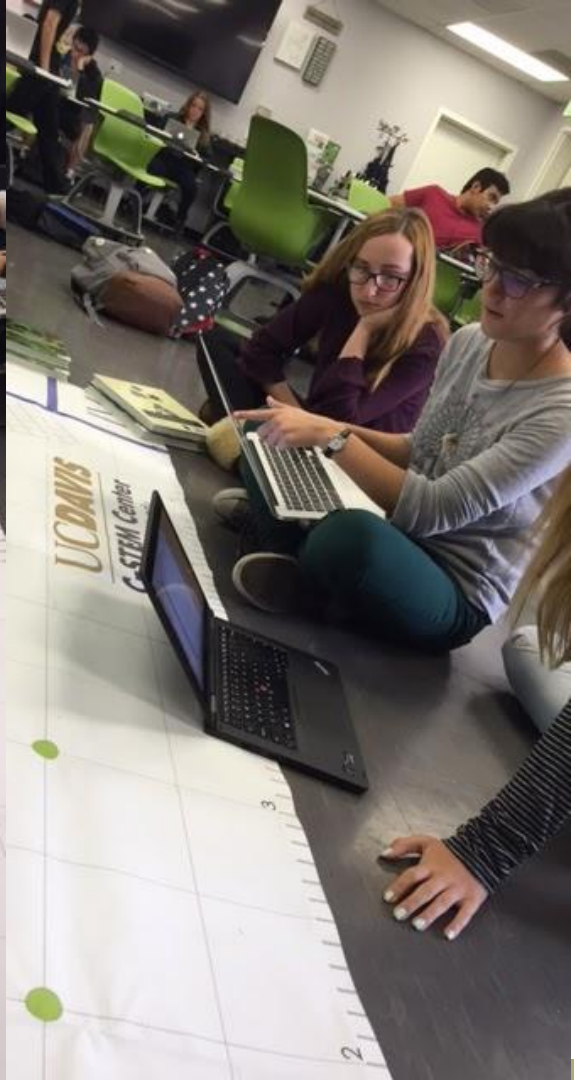
bounce MS 2014.ch - ChIDE - Professional Edition

File Edit Search View Tools Debug Animate Options Language Buffers Help

Start Continue Abort Step Next Up Down Break Clear Parse Run Stop

print odds while loop.ch | gpa.ch | squares with for loop.ch | Disco MS 2014 Challenge.ch | robot slide.ch | bounce MS 2014.ch | hit the wall MS 2014.ch | racetrack MS 2014.ch | high five low five MS 2014.ch | steal the gold MS 2014.ch | /Users/carrie/Documents/C-STEMTeachingResources/C-STEM Training Week 2/bounce MS 2014.ch

```
1  - /*Name: Carrie Denton
2  File: bounce MS 2014.ch
3  Summary: This program tells the robot to drive forward the height of the board, turn around & go back to where it started*/
4
5  #include <linkbot.h>
6  CLinkbotI robot;
7
8
9
10 robot.setJointSpeeds (100, NaN, 100); //tells the robot to go 50% of its total speed abilities
11
12 robot.driveDistance (38, 1.75); //tells the robot to drive forward for 38 inches
13
14 robot.turnLeft (185, 1.75, 3.69); //tells the robot to turn around to the left, 185 degrees to all for carpet errors
15
16 robot.driveDistance (29, 1.75); //tells the robot to drive 40 inches which returns it to the II section of the mat, Bot fully in the boundary
17
18
19
```



CHALLENGE ROBOTIC SLIDE

1st Part

- pairs write code at ½ speed

2nd Part

- speed up the dance
- Repeat dance 4 times

3rd Part

- multiple groups run their code together to mimic the country western line dance "The Electric Slide"

3. Robotic Slide

(40 Points Possible)

Basic Task Description

Move the bot to complete a series of steps simulating the 'Electric Slide' dance.

Starting Position/Set up

Bot is placed in the center of the board facing the \overline{CD} line.

Objective

Note: Each 'step' is represented by moving the bot a distance equal to the bot width.

1. Bot rotates 90 degrees clockwise, takes two steps, rotates 90 degrees counterclockwise and pauses.
2. Bot rotates 90 degrees counterclockwise, takes two steps, rotates 90 degrees clockwise and pauses.
3. Bot moves one step back, pauses and blinks red light once.
4. Bot moves one step forward, pauses, rotates 90 degrees counterclockwise and blinks blue light once.

Scoring

#	Description	Points
1	Step one completed.	10
2	Step two completed.	10
3	Step three completed.	10
4	Step four completed.	10
5	Does not finish facing the \overline{AC} line ($\pm 25^\circ$).	-5

Move Sequence	Starting Position	Ending Position
1. 90° Clockwise Two steps right 90° Counterclockwise		
2. 90° Counterclockwise Two steps left 90° Clockwise		
3. One step back Pause Blink light red once		
4. One step forward Pause 90° Counterclockwise Blink light blue once		

Team Scores

DEMO

VIDEO

PROBLEM SOLVING PROCESS

- ❑ Code is written correctly, but does not result in correct execution of steps in problem statement

- ❑ Real World Errors

- ❑ turning of robot

- ❑ flex of the wheels

- ❑ speed of robots

think

