THE EDUCATION FUND
CELEBRATING 30 YEARS

2015-2016

ideas with IMPACT

idea packet

Sponsored by:

FPL

Ford Motor Company Fund

Mathbotics:

Learning Math Through Robotics

MATHBOTICS

By

Marcelle Farley

Table of Contents

Pages

Build Your Own Robot Arm Student Handout: How To Build Your Own Robot Arm?

You are a member of a team of three or four students, all working together to design and build a robot arm out of the following materials which are provided to you.

The robot arm must be at least 18 inches in length and be able to pick up an empty Styrofoam cup.

Your team must agree on a design for the robot arm and identify what materials will be used.

Your team should draw a sketch of their agreed upon design prior to construction.

Part of the teamwork process is sharing ideas and determining which design your team will go with.

Trial and error are part of the design process. There is no "right" answer to the problem - your team's creativity will likely generate an arm that is unique from the others designed in your class.

Resources/Materials:

3" wide and approx. 22" long strips of cardboard-- 5 or so

Binder clips (different sizes)-- 8 or more

Brads-- @10

Clothespins-- 6

Craft sticks--10-15

Fishing line-- 3-4 feet

Hangers-- 1 or 2

Paper clips (diff. Sizes)-- 10-15

Pencils-- 3-4

Rubber bands (different sizes)--15

Tape-- clear and masking (partial rolls should be fine)

Twine-- 3-4 feet

Various size scraps of cardboard--10 assorted

Build Your Own Robot Arm Student Handout: Robot Arm Exercise Questions

Did you use all the materials provided to you? Why, or why not?

Which item was most critical to your robot arm design?

How did working as a team of four help in the design process?

Were there any drawbacks to designing as a team?

What did you learn from the designs developed by other teams?

Name three industries that make use of robots in manufacturing:

Some steps to follow to organize your students for robotics are:

• Research curriculum and resources and decide what you want to teach and how to integrate robotics in your required curriculum.

• Student teams for classroom purposes should be set up in groups of 2-4 students per robot. Some competitions, such as FIRST LEGO League, limit team size to 10 students. 4

• Define the role of each member of the team.

For instance:

(1) engineer (builder);

(2) software specialist (programmer);

(3) information specialist (gets the information and materials for the team to move forward or communicates the directions);

(4) project manager (keeps the team on task and facilitates problem solving).

• Identify what technology and materials are required. Such items as one computer per robot, one robot per team, ample space and storage, and backup components in case of breakdown should be considered. Also, make sure you have adequate system backup by using a separate disk or memory stick for each team.

• Write a budget and get funding. Consider robots, programming software, materials, etc. when planning.

As of this printing, the LEGO costs were: LEGO MIND STORMS Education EV3 set was $349 on Amazon. Shop around for other deals

• At the end of each lesson, share common problems and solutions, collect any written worksheets, provide test(s), if available, and move on to next lesson.

• At the end of instructional period, clean up LEGO parts and return them to sorting trays, save all programs for each group, and assign any homework questions.

• You may decide to assign projects for a 2-3 week process. Be sure to provide a rubric of what is expected and plenty of time to complete. At the end of the project time, have each group present their solutions and how they arrived at them.

Team Activity Sheet

Project Title: ____

Name of team: ____

Team Members:

Engineer: ______________________________________

Programmer: ______________________________________

Information Specialist: ______________________________________

Project manager: ______________________________________

What materials did you use?

____

____

____

What problems did you encounter?

____

____

____

____

____

____

____

What were your solutions?

____

____

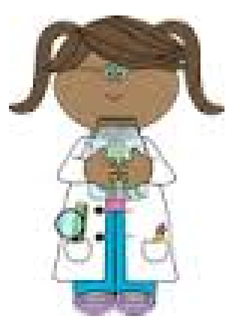
____

____

____

____

____

Engineer • Responsible for building missions • Design robot • Draw plans 	Programmer • Write code for robot • Test code on robot • Explain code to group 
Information Specialist • Research ideas for robot • communicate ideas • Check Plan for accuracy 	Project Manager • Keeps team on task • Ensure all rules are being followed • Collect materials 

Lab1 – Getting Your EV3 To Move and Respond to Its Environment

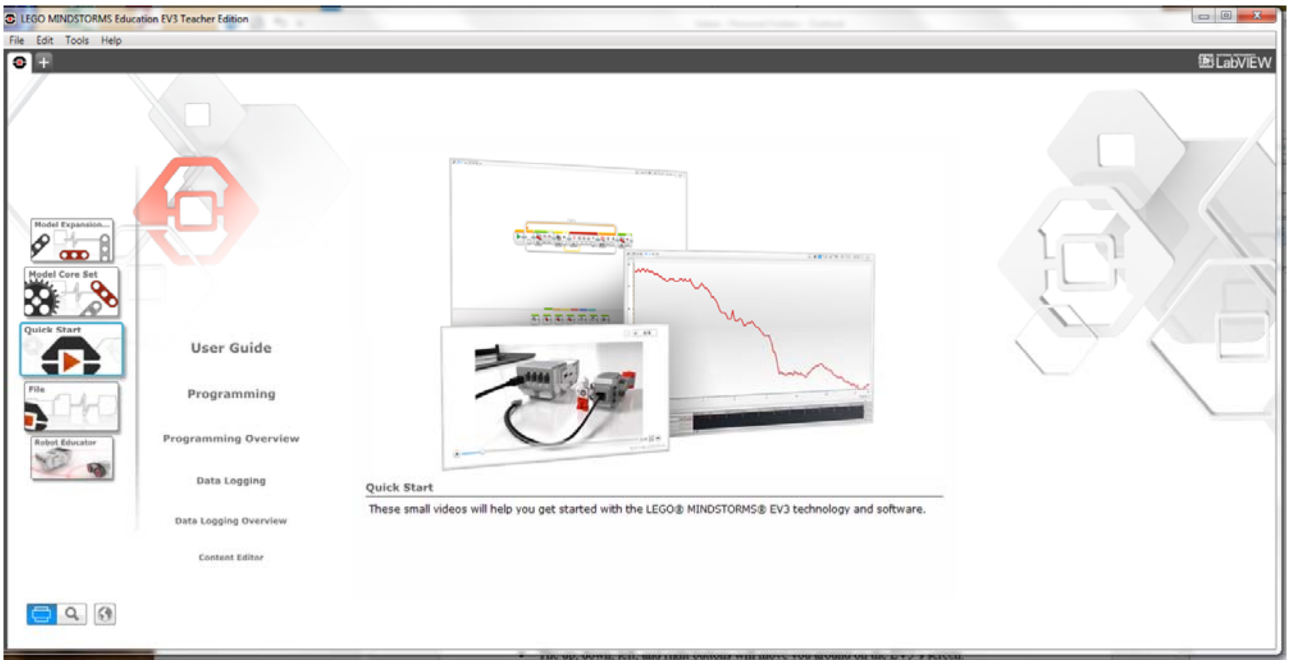
In Lab1 you will create a simple move program that provides a constant source of power to the two motors connected to Ports B and C until a touch sensor is pressed.

Using the Move Block in a Program

1____ Holding the robot in front of you with the ball skid pointing toward you, make sure the left motor is connected to output B and the right motor is connected to output port C.

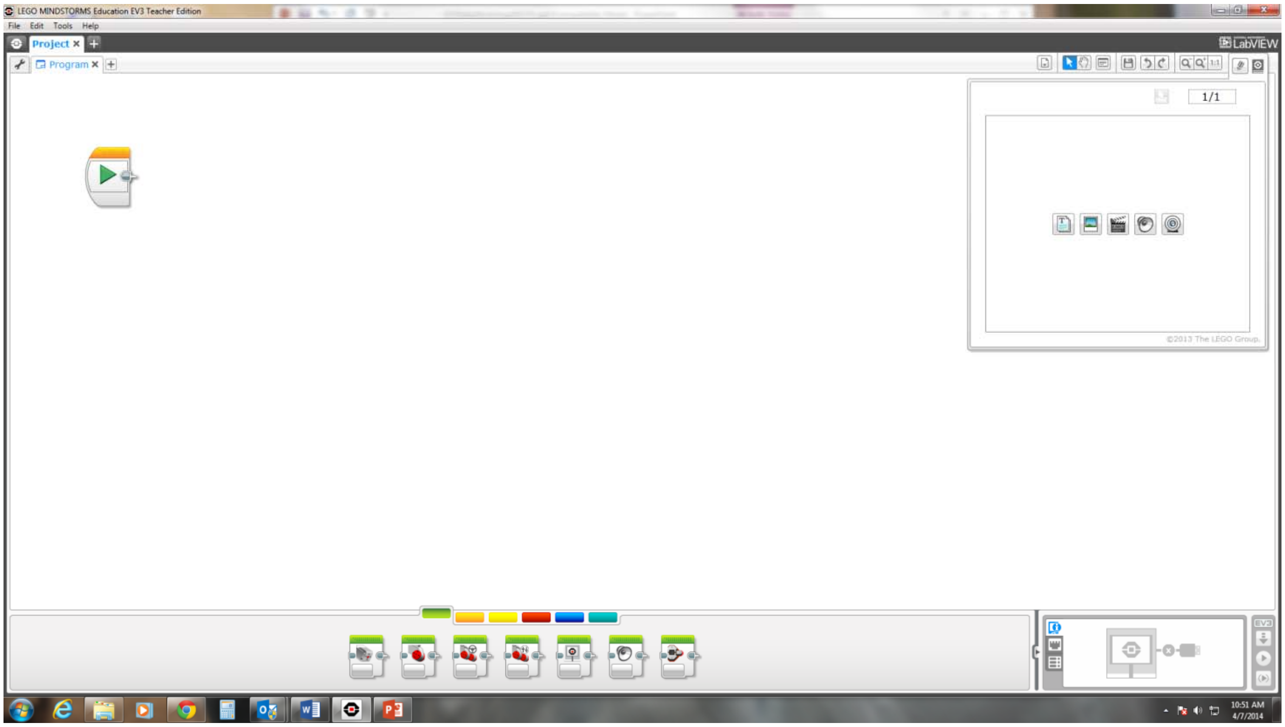
2____ Also make sure the touch sensor is connected to input port 1.

3____ Now go to your computer, and if the Mindstorms EV3 software is running, you should see a window that looks like the following.



This is called the lobby. We will cover what is available here in the homework material at the end of this handout. To get going on writing your first program, open the file menu in the upper left corner of the window, and select New Project. Then select Program from the sub-menu that pops up.

4____ Your window should now look like this:



Click on the rightmost icon under the word LabVIEW in the upper right-hand corner of the window. This will close the documentation window that is open there to provide more space for writing your programs.

The basic component with which you will work as you program your EV3 is the **Project.** **Projects** are saved as a single file with the extension .ev3. You will typically write several **Programs** for your EV3 as part of a **Project**. So a **Project** usually consists of several **Programs**, and a **Project** can contain other kinds of files as well:

- Programs (.ev3p)
- Images (.rgf)
- Sounds (.rsf)
- Text (.rtf)

Looking at the Mindstorms EV3 window on your computer screen, you will see a tab in the upper left hand corner just below the menu bar with the label **Project**. Below that you will see a tab with the label **Program**. Double click the label **Program** and change it to a name that you will remember. Be sure to press Enter after you have typed in the new program name. This will be your first program.

Now open the File menu and select Save Project As… Don't worry about what directory comes up, but type a new name for your project in the file name box, and click Save. Notice that the project tab now has the name that you have given for your project.

5____ You write a program for the EV3 by arranging **Program Blocks** in the order in which you want those blocks executed. All of the program blocks are at the bottom of the window by the six colored bars. This area contains the **Programming Palettes**. Move your cursor over each of the colored bars, and you will see that a tooltip pops up to tell you what types of blocks are represented by each color. You should see: Action, Flow

Control, Sensor, Data Operations, Advanced, and My Blocks. In the exercises today we will only use blocks from the green (Action) and orange (Flow Control) areas.

6____ If it is not already selected, click on the green (Action) bar. Seven programming blocks will pop up beneath the colored bars. Move your cursor over each and note what the tooltip tells you about each block. You should see things like: Medium Motor, Large Motor, Move Steering, etc.

7____ Now put your cursor over the third block (**Move Steering**), and while holding down the left mouse button drag the block up into the programming area and connect it to the block with the green arrow that is already there. As you drag near the block with the green arrow, you will see a gray box appear. When that happens, you can release the mouse button, and your block will be added to your program. Remember where you found the **Move Steering** block, since you will use it in other programs that you will write as you go through these exercises.

8____ Let's look at that **Move Steering** block in more detail. The upper right hand corner of the block should have the text B + C. This tells us that this block is set up to control the motors that are connected to ports B and C. The **Move Steering** block always controls two motors, and you select which motors by clicking on each of the letters and choosing which two motors you want associated with the block. Try it. Make sure that you select the same two motors in the program as are connected to the wheels on your EV3.

9____ Along the bottom of the block are 5 parameters. The leftmost one sets the mode of the block. Click on it with your mouse (it should look like a # sign with an arrow around it), and a menu will popup showing the 5 modes for the **Move Steering** block.

- **Off** turns off the motors associated with the block.
- **On** turns the motors on.
- **On for Seconds** turns the motors on for the specified number of seconds.
- Each of the motors has a rotation sensor built into it. **On for Degrees** turns on the motors until the axle connected to that motor has turned the specified number of degrees.
- Similarly, **On for Rotations** turns on the motors until the axle connected to the motors has turned the specified number of rotations. Note that **On for Degrees** and **On for Rotations** accomplish the same thing with 360 degrees equivalent to one rotation.

Set the mode to **On for Rotations**.

10____Now click on the next parameter, which is under the arrow pointed straight up. A slider will pop up. You use this to control whether the robot goes straight or turns. When set to 0, the same power is applied to each motor so that the robot goes straight. As you move the slider left or right, more power will be applied to one motor than the other causing the robot to turn. If you move the slider all the way to the left or right, one motor goes one way and the other motor goes the other way causing the robot to spin. Set the steering parameter so that the robot goes straight.

11____Now click on the third parameter. A vertical slider pops up. This parameter controls the power that is applied to each motor as well as the direction in which the motors will turn. Minimum power is 0 and maximum power is 100. Moving the slider up to larger positive numbers increases the power to the motors in one direction, and moving the slider down to larger negative numbers increases the power in the opposite direction. Set the power to +30. (Note that you can type in a number instead of using the slider.)

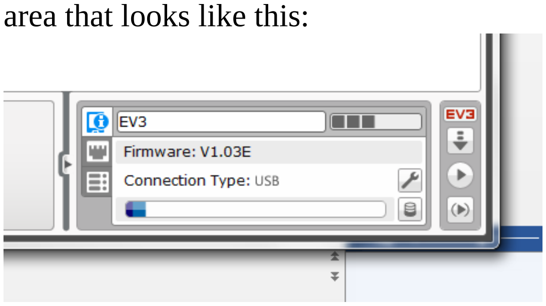
12____The fourth parameter is a number that indicates the number of rotations to turn the motors. Select that parameter, type in 1.5, and press the Enter key. This tells the block to apply power until the motors have turned one and a half rotations.

13____The fifth parameter tells the EV3 what to do after the conditions of the block have been met. When you click on that parameter, you get two choices. The check mark indicates that the motor should brake after the 1.5 rotations, while the X indicates that the motors should coast after the 1.5 rotations. Select the brake option.

Your program should look like this:

14____Now it is time to download your program to the EV3 and try it out. If your EV3 is off, push the center button to turn it back on. Connect the USB cable from your computer to the PC port on the EV3.

Look at the lower right hand corner of the Mindstorms EV3 window on the computer, and if your EV3 is on and connected to the computer with the USB cable, you will see an area that looks like this:



This is the master control area for communicating between your computer and your EV3. You'll learn more about this area later, but for now click on the down arrow under the red EV3. This will download all the components of your current project to your EV3, and the EV3 will play a little tune.

15____Now look at the screen on your EV3. Use the left and right buttons to move to the second tab. You will see a list of all the projects that have been downloaded to your EV3. Use the up and down buttons to find the project that you just downloaded.

Remember the name you gave it back in step 4 when you saved the project? When the cursor is on that project, push the center button. That will open the project to show you the programs that have been downloaded as part of that project. Use the up and down buttons to move the cursor to the program you want to run. Unhook the USB cable, set the robot down, and push the center button. (Be sure it doesn't dive off the table! ☺) What did it do?

If all went according to plan, the robot should have driven straight and stopped after a short drive. If you watch the wheels closely, it stops after the wheels have made one and a half revolutions.

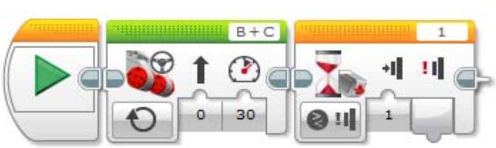
16____Now let's make the program a bit more interesting by using the touch sensor to stop the robot. First change the mode of the **Move Steering** block to **On**. Note that the only parameters that are left are the steering and power.

17____Move your cursor to the **Programming Palettes** area, and click on the orange bar. The set of **Flow Control** program blocks comes up. Use your mouse to add the second program block in the list (It's tooltip says **Wait**.) to your program after the **Move Steering** block. Remember where you got this **Wait** block, as you will also use it in several more places in the rest of the Labs.

Now let's look at the parameters of this **Wait** block in more detail. If you click on the first parameter at the bottom left of the wait block, you will see that you can configure the **Wait** block for all the sensors in our EV3 kits and several more. Slide your cursor to the **Touch Sensor**, and a pop-up menu comes up with options for **Compare** and **Change**. Move your cursor to **Compare**, and then another pop-up menu with **State** comes up. Click that option, and the **Wait** block changes to a Wait block configured for a **Touch Sensor**.

18____Now click on the second parameter in the **Wait** block, and three choices come up for comparing the state of the **Touch Sensor**: a release of the **Touch Sensor** button (the 0), a press of the button (the 1), and a bump of the button (the 2), which is a press followed by a release. Choose the 1, which is the press option.

19____Now you need to do one more thing before this new program is complete. Note the number 1 in the upper right hand corner of the **Wait** block. This indicates the port to which your **Touch Sensor** is connected. If you have been following all the directions correctly, your **Touch Sensor** should be connected to port 1. Since your touch sensor is plugged into port 1 on the EV3 and the programming block says 1, you are good to go. Your program should now look like:



20____Now download this new program to your EV3 and run it. The robot will start moving, and when you press the red button on Touch Sensor, what happens?

21____Note the * after the name of your project in the project tab. That * means that you have changed something in the project but haven't saved it yet. Go to the File menu, and select Save Project to save this revised version of your project.

CONGRATULATIONS! You have just created and run your first two Mindstorms EV3 programs.

Lab2: Write A Program To Move Forward And Detect A Black Line

1____ Go back to your computer and you should find that the project you did for Lab 1 is still visible in the Mindstorms EV3 window. Click on the + sign to the right of the tab with the name of the program you created in Lab 1 and then click on **New Program**. A new program tab will be created. Double click that tab and change the name of the program to one that you will remember.

2____ Click on the program tab to the left of the one you just created, and the program from Lab 1 will be displayed on the screen. Use your mouse and scribe a rectangle around the blocks of that program. You will notice that if you have done it correctly, a blue line will show around each of the three blocks. Now type CTL-c to copy those blocks to the clipboard.

3____ Click on the new program tab that you created above in Step 1, and type CTL-v to paste the blocks into this window. Since you don't need two of the **Start** blocks (the block with the green arrow), click on the **Start** block that is not connected to other blocks, and type the delete key.

The last few steps have demonstrated how to delete blocks if you don't want them in your program and how you can select multiple blocks and copy and paste them to other programs.

4____ Now the only thing left to do is modify the **Wait** block to use the **Color Sensor** instead of the **Touch Sensor**. First click on the lower left section of the **Wait** block where the mode is set. Once again the list of sensors pops up. Move your cursor over the **Color Sensor** option then over the **Compare** option, and then click on the **Reflected Light Intensity** option.

5____ Click on the port number in the upper right corner of the block and select the port number to which your **Color Sensor** is attached. Unless you have changed something, it should be port 3.

6____ Now click on the second parameter of the **Wait** block. It is the one below the < sign. This will cause a list of comparison operators to pop up. We want the **Wait** block to make a comparison between what the light sensor is reading as the robot moves to some fixed value that we will set in a moment. But what comparison is appropriate? For now, just trust that the < operator is correct.

7____ The next parameter sets the value to which we want to compare. This is called the **Threshold Value**. It is just a number that you type in once you have selected that parameter field. The default value is 50, and that will do fine for our first attempt with the **Color Sensor**. The last parameter is an output value that gives the value that the light sensor is reading. That will be left for later workshops. Your program should now look like:

8____ You should have a mat with your EV3 robot kit that contains a black oval line with some green rectangular shapes inside the oval. Download your project to your EV3. Remember that when you download, all the parts of the project are downloaded. In your case that will be both of the programs that you have written for this exercise. Locate this new program starting with the second tab on the EV3 screen, place your robot inside the black oval on the mat, and run the program. What happens?

Let's discuss what is going on with this program. When you start the program, the **Move Steering** block starts the robot moving, and the point of control moves to the **Wait** block. This block says to read the value of reflected light until the value read becomes less than 50, at which point the point of control is moved to the next block. Since there is no next block, the program turns the robot off. You should notice that it doesn't stop very quickly. You are always better off to tell the robot exactly what you want it to do.

9____ So add another **Move Steering** block after the **Wait** block, and set its mode to **Off**. Now download the project and run your program again. Now what happens?

Now your robot will stop more quickly because you have told it exactly what you want it to do when the light sensor sees the reflected light value fall below 50. Light values read by the **Color Sensor** range from 0 for the blackest of blacks to 100 for the whitest of whites. Now it should start becoming clear why we chose the < comparison operator and set what we call the **Threshold Value** to 50. In this exercise, we usually start the robot in a white area, and we are waiting for it to cross a black area and then stop. Since white values are closer to 100, and black areas are closer to 0, we use 50 as our **Threshold Value**, since it is about half way between the value the robot sees when it starts in white and the value it sees when we want it to stop in black.

Is it clear why we use the < operator in this case? How would you change your program if you started in black and wanted to stop when the robot comes to a white area? Try it!

Perhaps you pointed your robot so that it had to cross a green patch before encountering a black area. What happened in that case? Most likely it stopped at the green and never made it to the black. So that raises some questions about whether the **Color Sensor** can be used to distinguish colors. In order to look into this, we need to be able to understand what values the **Color Sensor** is reading as it moves over different colors.

10____We're going to use the robot to tell us what values it is reading with its **Color Sensor** as it moves over different colors on the mat. Use the left and right buttons to move to the third tab on the EV3 screen. The first item below the tab should say **Port View**. Using this view we can see what the robot is sensing at each of its ports. Press the center button to select the **Port View**. The screen will now show that you are looking at Port 1, which has a **Touch Sensor** connected to it. Click the right button, and Port 2 will be displayed, showing that the **Gyro Sensor** is connected to it. Click the right button again, and Port 3 will be displayed showing that a **Color Sensor** in **Reflected Light** mode is connected to it. Continue to click the right button, and the display will move through the remaining input ports and then the 4 output ports at the top of the screen. Continue to click the right button until you are back at Port 3.

11____Now with Port 3 being displayed, set your robot on the mat with the **Color Sensor** pointing at black, green, and white one after the other. Record the readings below:

Black_______________ Green______________ White_____________

12____You should be able to make some slight alterations to the program we have been working on to come up with two different variants:

- Develop a program that will stop the robot when it encounters either green or black.
- Develop a program that will skip over green and stop when it encounters black. (This one might be a little tricky, since the values for green and black are probably pretty close together.)

13____Once again you will see the * to the right of the project tab, meaning that you have made changes to the project but haven't yet saved those changes. Go to the File menu and select Save Project.

Lab 3 – Your Color Sensor Final Exam

Assume you have made the following light readings with your EV3 robot:
- White: 68
- Green: 38
- Black: 25

You have written a program like the program in Lab 2, which uses the Wait block configured as **Color Sensor – Compare – Reflected Light Intensity**. In that block you have chosen < as the comparison operator. You now have to set the **Threshold Value** in the **Wait** block. Discuss each of the following cases with your team and write what you think will happen for each of these **Threshold Value** choices in the spaces provided:
- You choose a value greater than 68:____
- You choose a value between 38 and 68:____
- You choose a value between 25 and 38:____
- You choose a value less than 25: ____

If you have time, you can check your answers by programming your robot for each case and running the program to see what happens. We'll talk about your answers as a class when everyone has completed Lab 3.

Lab 4 – Putting it All Together

You have enough fundamentals at this point to put several blocks together to accomplish a bit more impressive task. Develop a third program in your project that will get the robot to do the following once you set the robot inside the black oval and execute the program:
- The robot spins until the red button on the **Touch Sensor** is pressed.
- Then the robot moves forward until it hits a green or black area.
- Then the robot backs up and turns a bit in the process.

Try this on your own, and if you need help, the detailed steps are given below.

1____ Click the + sign to the right of the second program tab and select **New Program**. Double click the new program tab and change the name of the program to one that is meaningful to you. Be sure to press the Enter key after typing in your new program name.

2____ Add a **Move Steering** block to your program. Choose **On** for its mode, set the steering to 100 or -100 (That will make it spin.), and set the power to 10. You want it spinning kind of slowly so that you can press the button on the **Touch Sensor**. Make sure that ports B+C are set in the upper right hand corner.

3____ Add a **Wait** block and configure its mode to **Touch Sensor – Compare – State**. Set the second parameter to 1, indicating a wait until the touch sensor is pressed. Make sure the port number is correct.

4____ Add a second **Move Steering** block. Configure this one to **On**, go straight, and power of 30.

5____ Add a second **Wait** block and configure its mode to **Color – Compare – Reflected Light Intensity**. The comparison operator should be set to <, and the **Threshold Value** should be set to a value between the reading for green and the reading for white. Try using 20. This should make the **Color Sensor** signal when it reaches either green or black.

6____ Now add a third **Move Steering** block. Configure it to **On for Rotations**. Set the steering so that it will move with a slight curve. Set the power to -30, which will make it move in the opposite direction from the previous two **Move Steering** blocks. Set the number of rotations to 1.5. The resulting program should look like:

7____ Download the project, and run the new program to see how you did.

8____ Go to the file menu and select **Save Project**, and you are done with all the Labs associated with Workshop 1. Congratulations!!

Resources

http://www.lego.com/en-us/mindstorms/about-ev3

http://www.k-12robotics.org/uploads/5/6/3/3/5633548/pdf_introduction_to_robotics.pdf

http://www.slideshare.net/cybertronicinstituto/ev3-student-worksheets

THE EDUCATION FUND
CELEBRATING 30 YEARS

APPLY FOR AN
IMPACT II ADAPTER GRANT!

M-DCPS teachers, media specialists, counselors or assistant principals may request funds to implement an IMPACT II idea, teaching strategy or project from the Idea EXPO workshops and/or curriculum ideas profiled annually in the ***Ideas with IMPACT*** catalogs from 1990 to the current year, 2015-16. Most catalogs can be viewed at The Education Fund website at www.educationfund.org under the heading, "Publications."

- Open to all K-12 M-DCPS teachers, counselors, media specialists
- Quick and easy reporting requirements
- Grants range from $150 - $400
- Grant recipients recognized at an Awards Reception

To apply, you must contact the teacher who developed the idea before submitting your application. Contact can be made by attending a workshop given by the disseminator, communicating via email or telephone, by visiting the disseminator in their classroom, or by having the disseminator visit your classroom.

Project funds are to be spent within the current school year or an extension may be requested. An expense report with receipts is required by May 2, 2016.

APPLICATION DEADLINE:
December 11, 2015

Apply online at www.educationfund.org

For more information, contact:

Edwina Lau, Program Director
305.558.4544, ext. 113
elau@educationfund.org

THE EDUCATION FUND
CELEBRATING 30 YEARS

Contributors with IMPACT

Platinum Star

FLORIDA MATCHING GRANTS PROGRAM

Gold Star

TD Bank
America's Most Convenient Bank®

Ford Motor Company Fund

Humana Foundation

verizon

ASSURANT®

P L Dodge Foundation

FPL

FLORIDA 123 ABH SUPPORT EDUCATION

Silver Star

PEREZ TRADING COMPANY

Rod and Lucy Petrey

Raj Rawal and Anne Marie Miller

Robert Russel Memorial Foundation

Bronze Star

original IMPRESSIONS℠
Your Marketing Communications Partner®
creative | e-business | print | fulfillment | signs

The Jack Chester Foundation

PNB

Boeing

Miami Sportfishing Tournament/ Gary M. Pappas