Introductory Workshop EV3 Basics is a series of labs or exercises to introduce you to the EV3 robot and the Mindstorms EV3 programming language. This material walks you through the minimum set of things you will need to learn to get your EV3 robot moving and responding to its environment through its sensors.

There is another section following the last lab, which is Lab 4. This section walks you through some of the other capabilities within the programming environment that you will no doubt need as soon as you start trying more things with your robot. We encourage you to go through that material as homework and come to the next workshop with any questions that you might have.

Read the following material together as a team, and when you get to the numbered steps, do what is written for that step and check it off when complete. So jump right in, go through the EV3 Basics Labs, and have fun.

_____Be sure to check off Lab steps as you complete them.

Becoming Familiar with Your EV3

Pick up your EV3 and push the dark gray square center button. This will turn on your EV3. The red lights around the buttons will come on, and in about 30 seconds, those lights will turn green, and your EV3 will be ready to go.

Now let’s take a closer look at your EV3. There are six buttons as shown in the figure below.

- The up, down, left, and right buttons (3) will move you around on the EV3’s screen.
- The center button (2) turns on the EV3 and “executes” whatever is selected on the screen.
- The back button (1) stops any program that is running, and it also moves you backwards through the menus on the screen.
Look at where the motors and sensors are plugged into your EV3. Sensors are plugged into input ports, which are labelled 1, 2, 3, and 4. Motors are plugged into output ports, which are labelled A, B, C, and D. Finally notice that there is one more port next to the output ports, which is labelled PC. This is where you connect your EV3 to your PC with the USB cable provided in your kit.

If you look at the screen on your EV3, you will see that there are 4 tabs across the top of the screen. The tabs that we will use in these initial exercises are the second and third tabs. You move from one tab to the next using the left and right buttons. Try moving between the tabs on your EV3.

One of the most common questions at this point is, “How do I turn off my EV3?” Push the back button until you see the on/off symbol on the screen with an X and a check mark below it. Use the left and right buttons to move the cursor back and forth between the X and the check mark. To turn off the EV3, move the cursor to the check mark, and hit the center button. Otherwise, move to the X, and hit the center button.

Try turning off your EV3. It will take about 45 seconds before it turns itself off. In the meantime start reading Lab1 and performing each of the directions in turn.

**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.

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.

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.

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.

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.
- **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:

![Program Image]

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:

![Master Control Area Image]

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:

![Program Example](image)

20. Before we go ahead and download this new program to try it out, let’s learn about the very important part of programming called “commenting your code”!

Using your mouse, position your cursor above the green bar of the **Move Steering** block, and click the left mouse button. Now look at the set of tool icons in the upper right corner of your screen. You will see an arrow, which is probably currently selected, followed by a hand, followed by what looks like a block of text. Click on that third icon. A little box will pop up above the
green bar where you clicked with your mouse just a moment ago. When you move your mouse to that box, you will see that sizing handles pop up. Click on that box with the left mouse button to make the sizing handles hang around, and use them to make the box a reasonable size in which to place a comment. Now use your left mouse button to click inside the box, and you will see that a vertical bar appears. This is the text cursor. Type your comment into the box, and when you are finished click outside the comment box.

The purpose of these comment boxes is to leave a description of what you were trying to accomplish in this part of your program. When you come back to this in a few days, you will likely have forgotten what was going on here, and that comment will be very useful. You should encourage your teams to use these comment boxes, and you will find that the Robot Design judges will think even more highly of your team.

If you want to get rid of a comment, probably to replace it with a more accurate version, click on the comment box, and then hit the delete button.

21____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?

22____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.

Postscript to your first programs:
As a program executes, the point of control in the program moves from block to block. In most cases the point of control does not leave a block until all the conditions in that block have been met. For example, in the first program you did with just the single Move Steering block, that block was set to move the robot by one and a half rotation of the wheels. When control enters that block, the motors are started, and the point of control does not leave the block until the wheels have made one and a half rotation.

When we changed the program so that the robot would stop when the Touch Sensor was pushed, we introduced a block that does not behave quite that way. The first block was a Move Steering block in the On mode. Since there was no other condition to be met in that block, the point of control immediately moves to the next block once the motors are started. The next block was the Wait block that was configured to wait for a push of the Touch Sensor button. As with most of the other program blocks, the point of control does not leave the Wait block until it senses that the Touch Sensor button was pressed. Since there are no more program blocks to execute, the robot will stop. It might not have stopped as quickly as you would have expected. We’ll cover that in Lab 2.
Lab 2 – Using a Color Sensor

The EV3 Color Sensor has three different modes in which it can be used. In the mode we will use in this exercise, the Color Sensor emits light and reports the intensity of the reflected light it senses. Later workshops cover using the Color Sensor in its Color mode.

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.

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.
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. It will probably show that nothing is connected to Port 2. 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:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Black</td>
<td>Green</td>
<td>White</td>
</tr>
</tbody>
</table>

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

These are not the actual readings that your EV3 will make, but humor your instructor and go along with the exercise. 😊

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 (like 75): ____________________________
- You choose a value between 38 and 68 (like 45): ____________________________
- You choose a value between 25 and 38 (like 31): ____________________________
- You choose a value less than 25 (like 15): ____________________________
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!!
Homework for ORTOP’s Introductory FIRST® LEGO® League Workshop
More Things You Will Need to Know about the EV3 Programming System

In the workshop we covered just enough of the EV3 programming software so that you could complete the first exercises and begin to understand how to get your robot moving and responding to its sensors. As you work with your team, many questions will come up about how to do other common tasks. The purpose of this “homework” section is to provide you with more information about other tasks that you can do with the software.

It would be best if you read through this material while sitting in front of your computer with the EV3 software running. Then follow along so you see how to perform the various tasks that we will be describing.

First of all, let’s go back and look at the “lobby”. This is the first page that comes up when you start the EV3 software. If the software is already running, you can get back to the lobby by clicking the hexagonal icon in the top left corner of the screen right below the File menu option.

The Lobby Screen

Let’s look at what you can do from the lobby screen, which is shown on the next page:

- The first two icons are used to get building instructions and the corresponding programs for a number of interesting robots. Models with the first icon require both the Core and Expansion Sets, while models with the second icon require only the Core Set. These are interesting and fun, but they are not particularly relevant for FIRST® LEGO® League. The programs are quite advanced for youngsters who are just beginning to use the EV3 system.
- The information in the Quick Start area will be more interesting to the coaches and mentors than to the youngsters on the team.
- The file management area can be useful to FIRST® LEGO® League teams, since you can use this area to create new projects and open old ones. You can, however, do these same things from the File menu.
- The Robot Educator area may be useful for the Coaches and Mentors, but it is probably too advanced for the youngsters who are just starting.
The Programming Screen

Now let’s take a look at the screen where you write the programs that are part of your projects. You can close the lobby by opening one of your projects or creating a new project using the File menu.
In the workshop we pointed out the two sets of tabs that list the projects and the programs that are currently open in the EV3 software. The + sign at the end of the string of tabs will create a new project or program depending on which + sign you click. Here are some important things to note when dealing with projects and programs in this screen:

- **You change the name of a project** by using the Save Project As option in the File menu.
- **You change the name of a program** within a project by doubling clicking on the program tab and typing in a new name.
• If there is an * next to the project name, you have made changes to the project that have not yet been saved. Use the Save Project option in the File menu to save the project in the file system on your PC.
• Also note that there is an x next to each project and program name.
  o Clicking on the x next to a project name will remove that project from the EV3 programming screen. It does not delete the project from your PC’s file system. If you click the x next to an unsaved project, the system will ask if you want to save it first.
  o Similarly, clicking on the x next to a program name will remove that program from the list of programs that are currently open for the currently selected project. It does not delete the program from the project. We will show you how to do that a bit later.

Programming Toolbar

Let’s look next at the Programming Toolbar.

Content Editor:
Provides a number of tools with which to document your project. Not covered in this workshop.

Note: When you Save Project, all the files associated with the project are written to the PC.
We’ll take each of the icons on the toolbar in turn:

- **List Programs in Project**: Clicking on this icon will bring up a list of all programs in the project, whether they are currently open in the Programming Screen or not. For example, suppose your project has 3 programs in it. When you open that project with the File menu Open Project, all 3 programs will be shown in the program tabs area. Maybe you are focusing on one of the programs so you click the x on the other two programs to close them. If you click on the Program List icon, all 3 programs will be listed, and by clicking on one of the closed programs in the list, that program will now be opened as well.

- **Select and Pan**: At any time when you are using the Programming Screen the cursor is in one of two modes. In the Select mode, the cursor is in the form of an arrow, which you use to select various elements on the screen. In the Pan mode, the cursor takes the shape of a hand. Now when you click and move the mouse, you will be panning across your program. This is useful when your program has grown so big that it won’t fit on one screen.

- **Comments**: Good programming practice says that you should add comments as you develop your program. Make sure your cursor is in select mode, and click on an area of the screen where you want to add a comment box. Then click on the Comments icon in the programming toolbar. A comment box will pop up on the screen. Use the sizing handles to increase the size of the box appropriately, click inside the box, and type in your comment. You can delete a comment box by selecting it with your mouse and then hitting the delete key.

- **Save Project**: Clicking on this icon will save the current version of your project in your PC’s memory.

- **Undo and Redo**: These buttons are used to undo or redo the last actions you performed.

- **Zoom Out, Zoom In, and Reset Zoom**: Zoom out reduces the size of the images on the screen so that you can see more blocks in your program. Zoom in goes in the opposite direction by magnifying the images on the screen. Reset zoom brings the magnification level back to the default value.

- **Open Content Editor**: The content editor provides tools to document your project. This is beyond the scope of our initial workshops and will not be covered further here.

**Hardware Page**

Now let’s look at the area in the bottom right of the programming screen. This is called the Hardware Page. First, you should remember from the workshop that the down arrow on the right side of block below the red EV3 is the button you click to download your program from your PC to the EV3. The other two buttons below the download button (Download and Run, Run Selected Blocks) have only special utility, and we won’t cover them any more here.

Note that there are three tabs on the left side of the block. The topmost one provides information about your EV3 brick, the middle tab provides information about what is connected to each port on your EV3, and the last tab provides information about other EV3 bricks that might be accessible. We’ll cover each of these in turn. Note that you must have your EV3 connected to your PC in order to get useful information displayed.
The **Brick Information** tab gives information about the EV3 connected to the PC that is running the EV3 software.

- The text box at the top of the window shows the name that has been assigned to the EV3, which is EV3 by default.
- The box to the right of the name box shows the battery charge of the EV3.
- The boxes below the name box give the firmware version and the manner in which the EV3 is connected to the PC.
- Finally, the last box shows how much memory is being used on the EV3. We cover memory management in a later workshop.

**Note:** When you download, all the programs in the project are downloaded, not just the one you were just working on.
The **Port View** tab shows what devices are connected to the 4 output ports and the 4 input ports. The Port View box in the figure above shows:

- Port A has a medium motor.
- Ports B and C have large motors.
- Port D has nothing connected.
- Port 1 has a color sensor in reflected light mode.
- Port 2 has nothing connected.
- Port 3 has a touch sensor.
- Port 4 has a gyro sensor.

The **Available Bricks** tab gives information about all the EV3s that are accessible. Since *FIRST® LEGO®* League allows only one EV3 to be associated with the robot, this will not be useful to *FIRST® LEGO®* League teams.

**Project Properties Screen**

You get to this screen by clicking on the button that looks like a wrench on the left end of the string of program tabs. If you have a lot of programs open, that wrench icon will move off the screen to the left. You can also click on the List Programs icon on the Programming Toolbar that we discussed above. The first item on the drop down list is Project Properties. The Project Properties screen is important, since it is where you manage the characteristics of your projects. An example screen is shown in the figure below.

First notice that there are six tabs in the middle of the screen right below the two black boxes. These tabs are labelled: Programs, Images, Sounds, My Blocks, Variables, and Exportable Items. The various files that make up your project are listed here, but as you are just starting, the one you will find most useful is the Programs tab, which lists all the programs that are included in your project.

At the bottom of the screen are five buttons: Copy, Paste, Delete, Import, Export. The Import and Export buttons are a bit more advanced and are less likely to be used in the early going with your teams. It is these buttons that are used to move programs between computers.

Copy, Paste, and Delete are definitely buttons that you will use, and we will cover them here:

- **Delete Button**: Remember how we noted earlier that the x after the program name on one of the program tabs can be used to remove a program from the programming screen. That action, however, does not remove the program from the project. That is done on this program properties screen. If you want to delete a program from a project, select the name of the program in the Programs list, and then click the Delete button. Note that once you have deleted a program from a project, you will not be able to get it back unless you happen to have a copy of it associated with another project.
Copy and Paste Buttons: You would use these buttons if you wanted to move a program from one project to another. Select the program you want to move on the project properties page for the first project and click the Copy button. Use the project tabs to select the second project, click on the wrench icon to the left of the program tabs to open the project properties page, and click the Paste button. The program you copied from the first project will now show up on the program list for the second project.