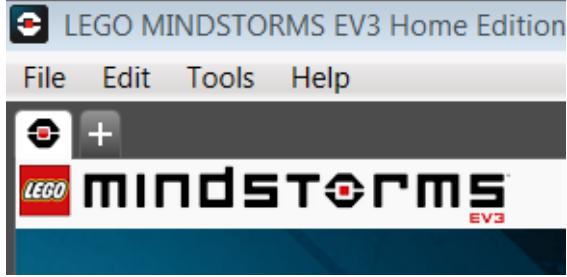


Name**Mindstorm and EMG Lab**

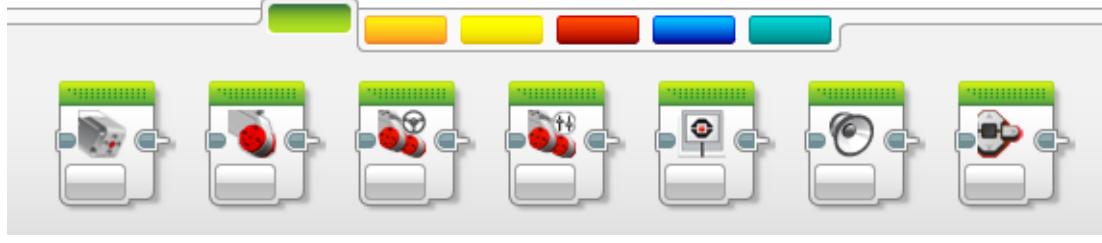
Objective: Learn about basic Mindstorm programing and how a signal from your muscle can be used to control your program.

1. Start the Lego Mindstorm Program with the icon located on the desktop. Open a new project by clicking the plus tab in the upper left corner of the window.

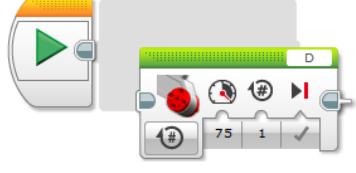


2. Turn on and attach the Mindstorm brick to the computer using the mini usb cable. Plug in 2 large motors and 2 push buttons into the Mindstorm brick.

The Lego Mindstorms program runs off of LabVIEW which is a drag and drop program used to code many things. Mindstorms is set up to have a bottom window with many different colored tabs (shown below). These tabs are organized by these colors; green is action, orange is flow control, yellow is sensor, red is data operation, blue is advanced, and teal is for custom blocks. For this lab you will only be using the first three tabs.



3. Create a function that allows the motor 1 to run.
 - a. Go to the green “action” tab and hover over the second block. This will show that it is the large motor. Drag this block into the main window in front of the play block and release it next to the play block as shown.



- b. Every block has a couple of options allowing some customization. The large motor block lets you choose what motor to control with the letter in the upper right corner of the block (shown as D in the picture above). Clicking it will allow the selection of which port you want to control (motor ports A, B, C, D). Select motor port A to control our first motor.



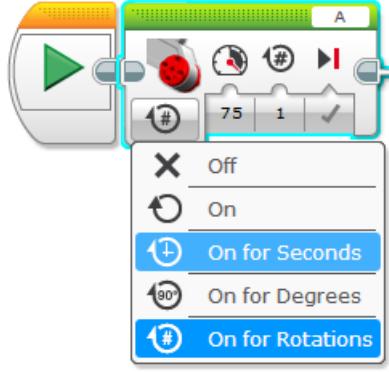
- c. If the EV3 Lego brick is connected to the computer pressing the play block will run the program. Click the play button.



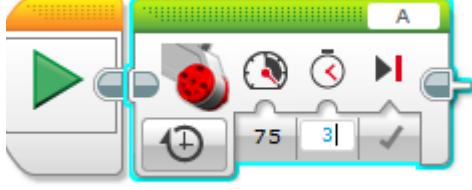
You will notice that when we ran our program to turn on the motor in port A the motor turned on momentarily and then powered off. This is because each block takes a certain amount of time to run before the program continues to the next block. We want to make our motor run for 3 seconds to make it more useful.

4. Change the function so that the motor 1 runs for 3 seconds.

- a. To extend the length of time that the motor runs we can change the settings of the large motor block. Click the circle with the pound symbol to open a drop down menu of options. We want to choose on for seconds.



- b. Now to simply increase the length of time that the motor runs for we can change the seconds parameter from 1 to 3.

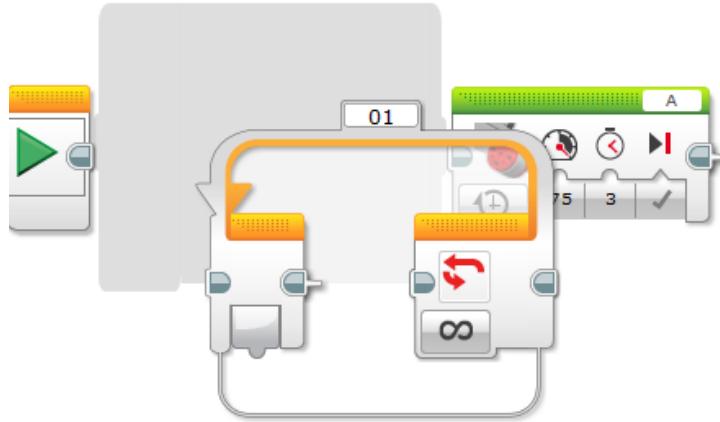


- c. Now run the program again by pressing the play block.

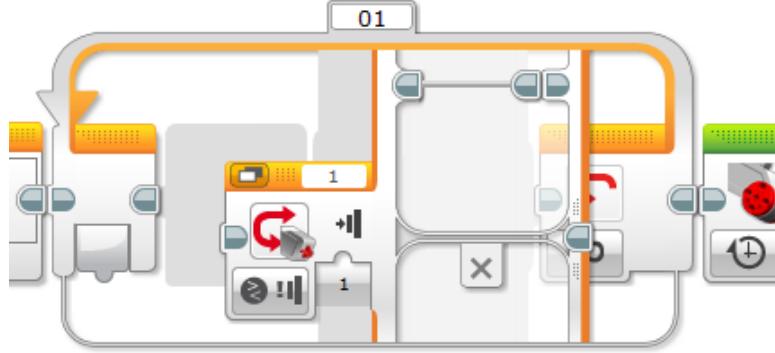
Now that you are getting more comfortable with the program let's add even more functionality. Let's use a button to control the motor.

5. Change the function such that the motor 1 will run when the button is pressed.

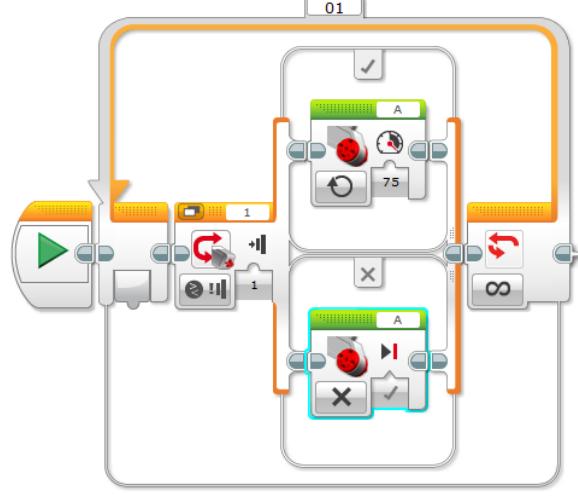
- To make this program we want to utilize loops to our advantage. Go to the orange "flow control" tab at the bottom and drag the third block (loop) into our main window between the play block and the large motor block.



- Now what we want to do is drag the fourth block (switch) from the "flow control" tab into the loop block.

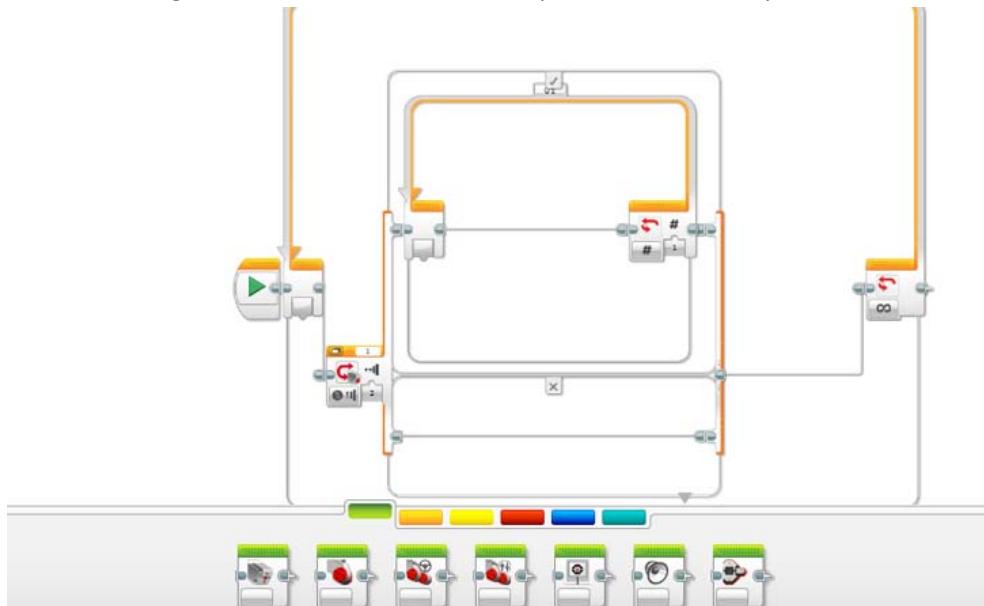


- Now using the features of the switch block we can switch what our program does depending on the state of the parameters which in this case is the button in port 1 being pressed. Change our large motor block to being "on" with our options menu (button under the picture of the motor) and drag it into the top section (true) of the switch case. Also drag a new large motor block into the bottom section (false) of the switch case. Change that motor blocks settings to be "off" and change the port to port A.

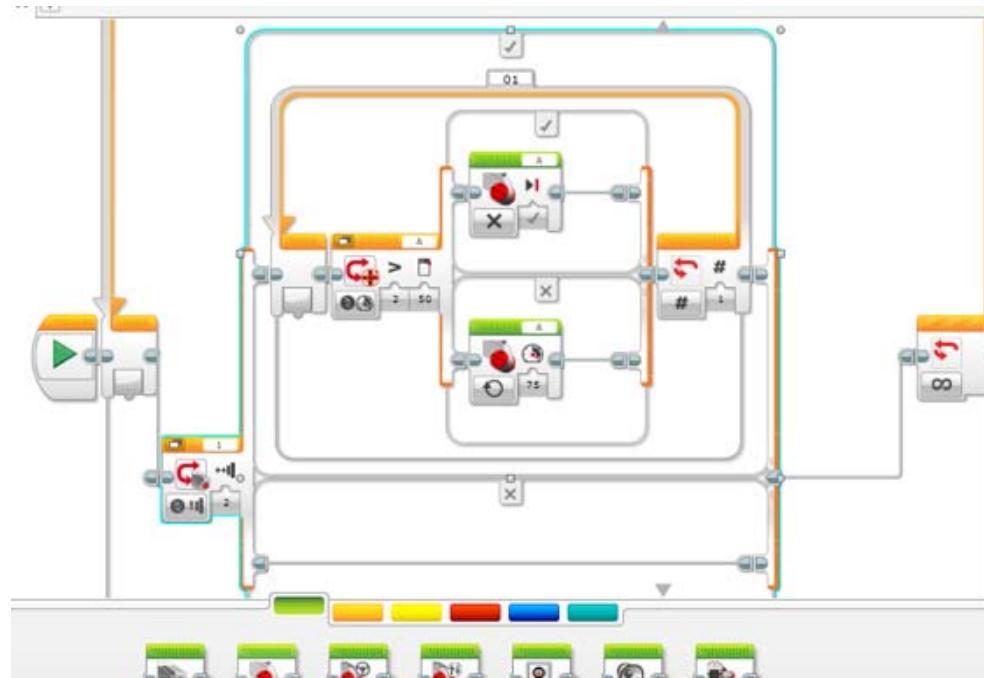


- d. Now the program will enter the loop and if the button value 1 (pressed) it will turn our motor on, and loop back infinitely. It will keep the motor on until the button state is read as 0 (not pressed) which will then cause the motor to stop. Run the program and test how it works. (**This program is in an endless loop and will not end by itself. Stop the program by pressing the stop square at the bottom right of your screen when you are finished testing.**)
6. Change the function such that the motor 1 will run when the button is pressed and stay on until it is pressed again.

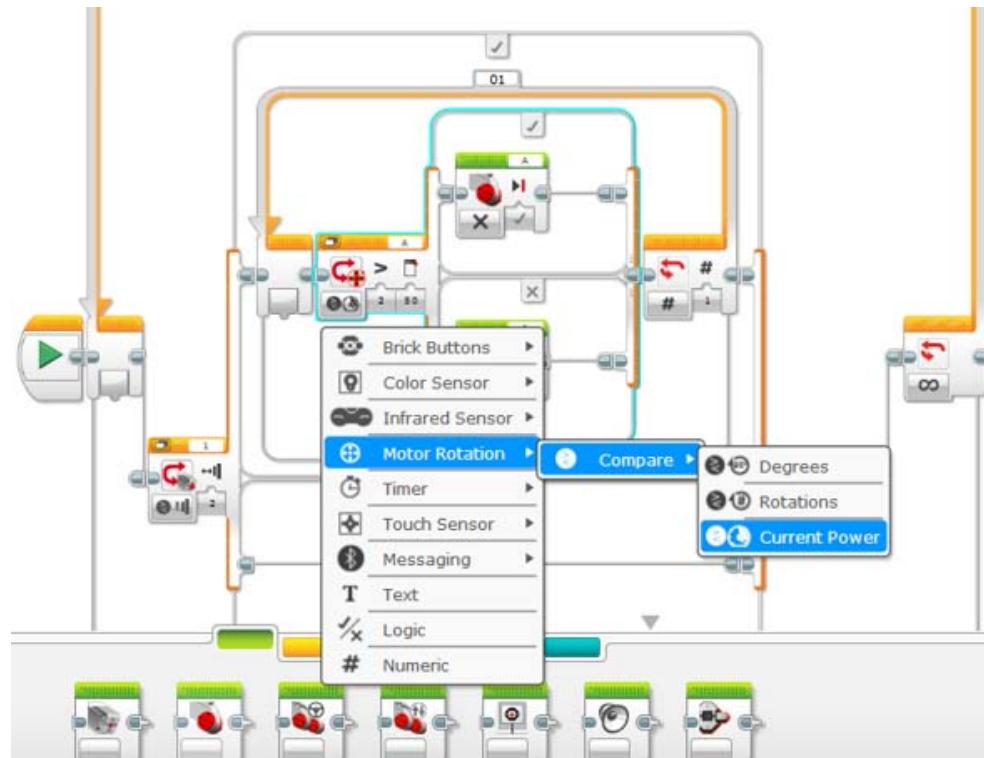
- a. Add another loop (as done in step 5.a), but place it in the true section of the previous switch case. Set this loop to 1 increment (change infinity sign on right side of loop to a 1). In addition change the Initial switch state from pressed (1) to bumped (2).



- b. Add another switch case within the newly created loop. Put the “motor on” block within the false section of the new switch case. Put the “motor off” block within the true section of the new switch case.



- c. Now change the new switch case to act on if the motor is on or off. To do this, change the switch cases parameters to “motor rotation” then “compare” and then “current power”.



- d. Run the program

Now try making your own loops by following steps 7-10. Try to do it without the help of others until there are 30 minutes left in the lab.

7. Create a function such that Motor A turns on for 3 rotations, then Motor B on for 2 rotations, and repeats. You can change from seconds to rotations by clicking under the picture of the motor on the programming block.

8. Change the function so that when the switch is pushed, Motor A runs for 3 seconds and then Motor B runs for 2 seconds then remains still until the switch is pressed. This can be done by putting a loop at the end of the program to loop forever.

9. Change the function so that when the switch is pushed it runs. step 8, and then the next time the switch is pushed it runs Motor A backwards for 3 rotations and then motor B runs backwards for 2 rotations and remains still. You can make the motor go backwards by putting in a negative number of rotations.

10. You can then put in a loop interrupt to make this system repeat multiple times (see next page).