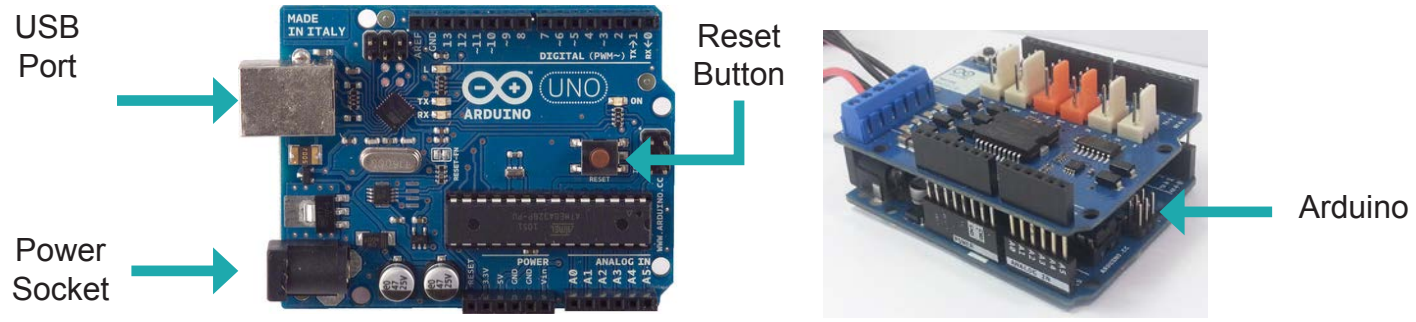
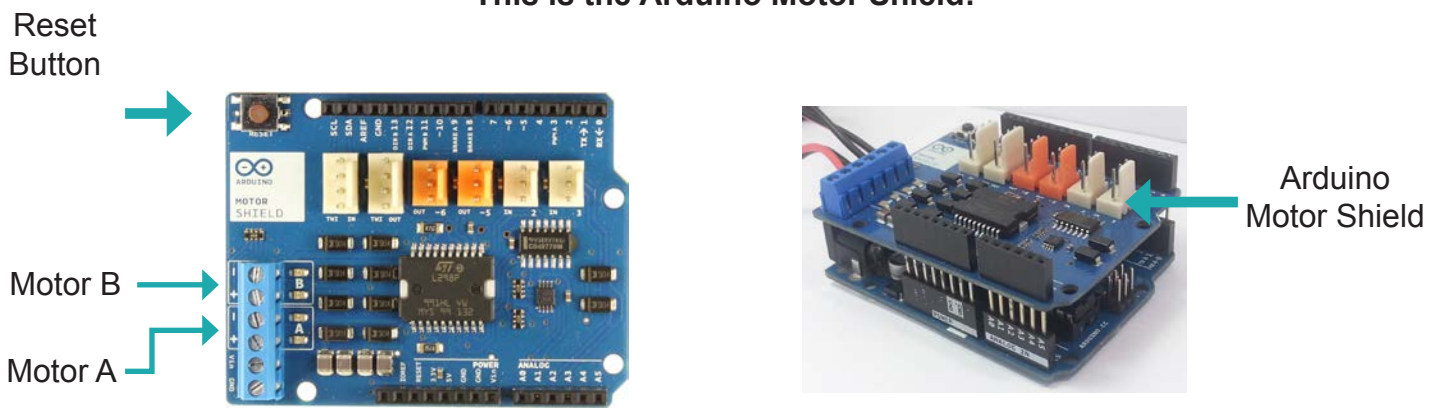
Mechanisms

In this tutorial we will review how to program the Arduino to control individual motors such that you can control one or more mechanisms that are attached to the motors.

This is the Arduino Uno:



This is the Arduino Motor Shield:

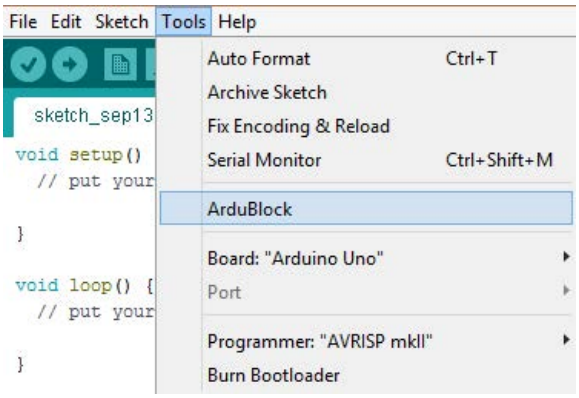


The motor shield is what allows our Arduino to control motors relatively easily without having to worry about too much extra wiring or extra components.

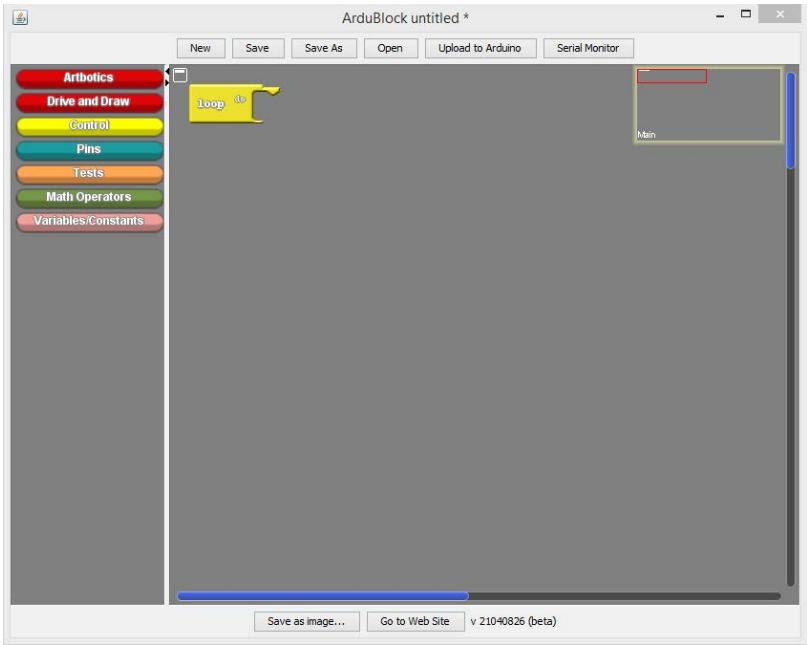
Mechanisms

Open the Arduino IDE that was illustrated earlier. An icon like the one on the right should be on your desktop. If not, navigate to where programs are generally installed or ask your instructor for assistance.

Once open, navigate to **Tools**, and click on **ArduBlock**.



The ArduBlock program should open displaying the block drawer on the left and one loop block in the coding area.



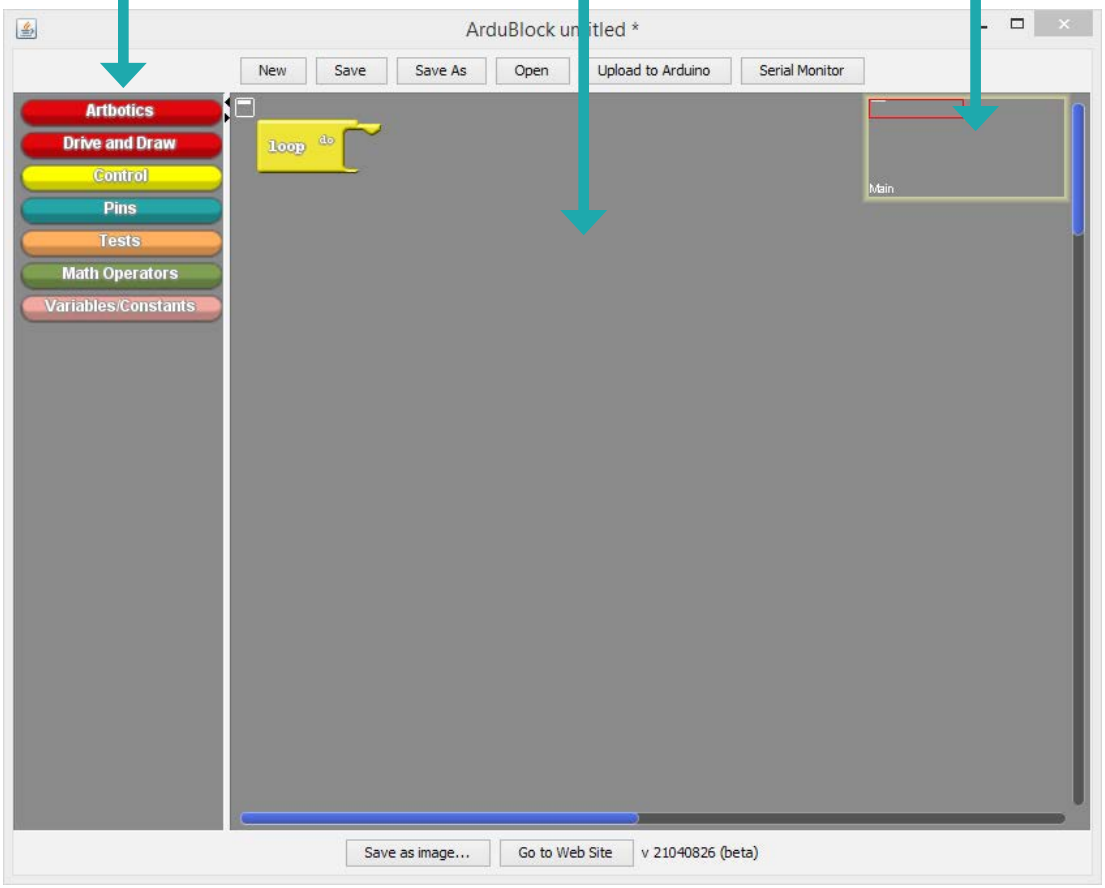
Mechanisms / Where to program

This is the ArduBlock Programming Tool

Block Drawers:
Each contain coding blocks

Coding Area:
Area to drop and arrange coding blocks

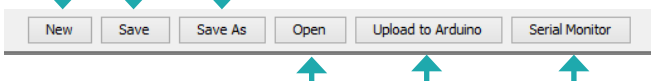
Overview:
Quickly jump to sections of your code



New: Create new file

Save: Save current file

Save As: Save as different name



Open: Open saved file

Upload to Arduino:
Put code on to your Arduino

Serial Monitor:
Read and write data to and from your Arduino

With this programming tool we will be able to design code and upload it to the Arduino Uno. By following these steps you will be able to drive motors for an interactive sculpture.

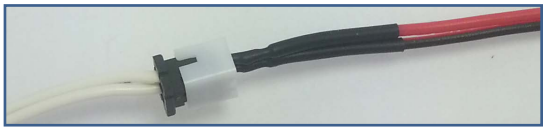
All the while we encourage you to explore and play with the provided coding blocks.

Mechanisms

1

There are two motor controllers, A and B, that are marked on the terminals of the Motor Shield. They will be connected to the motors that you will be using for this exercise.

Follow the red and black wires of terminal **A** to its end and connect the wire to one of the DC motors that you have prepared for the exercise. Do the same for terminal **B**.



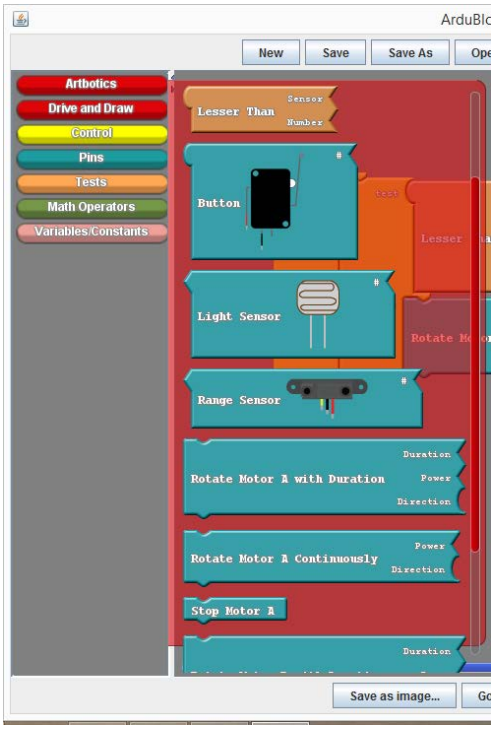
Using these two blocks, and other **Rotate Motor** blocks in the **Artbotics drawer**, you can directly control the motor terminals.

These blocks use 3 arguments:

Duration: Determines how many seconds the motor will spin for.
Power: Determines the speed at which the motor will spin; 1 (slow) to 5 (fast).
Direction: Determines the direction the motors will spin;
HIGH, or LOW(opposite of HIGH)

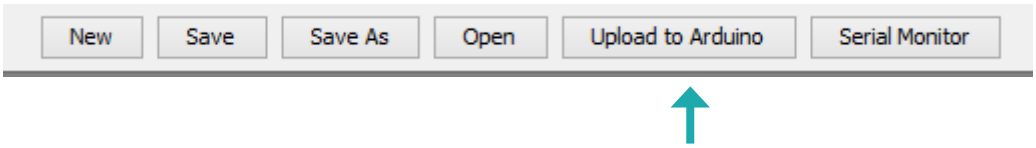
To program the motors go to the **Artbotics Drawer** at the top of the list on the left side of the window.

From the drawer select **Rotate Motor A with Duration** and snap it onto the **Loop** block in the coding area. This block will set the physical motor connected to terminal A to spin with a given duration, power and direction.



2 Go back to the **Artbotics Drawer** and select the **Rotate Motor B with Duration**. Snap it in place under Rotate Motor A. This block will rotate motor B for a given duration, speed and direction.

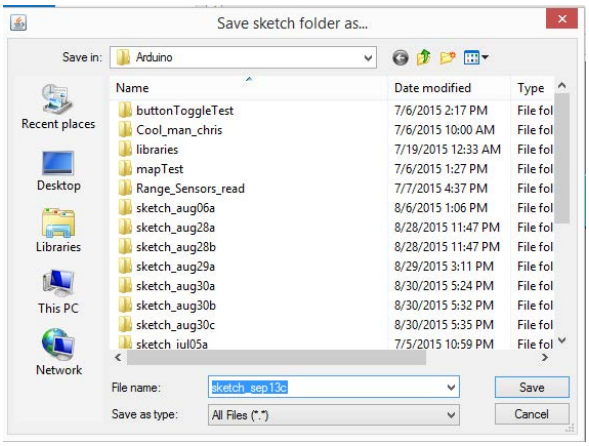
Now that you have your programming blocks setup, **make sure your Arduino is plugged in with its battery pack turned off** and press the **Upload to Arduino** button. This will translate your blocks into code, then compile it which essentially translates the code into a format the Arduino can understand. After compilation, the Arduino IDE will upload the data to the Arduino.




```
#include <Artbotics.h>

Motor motorA(3, 12);
Motor motorB(11, 13);

void setup()
{
}

void loop()
{
  motorA.rotate( 2, 3, HIGH);
  motorB.rotate( 1, 3, HIGH);
}
```

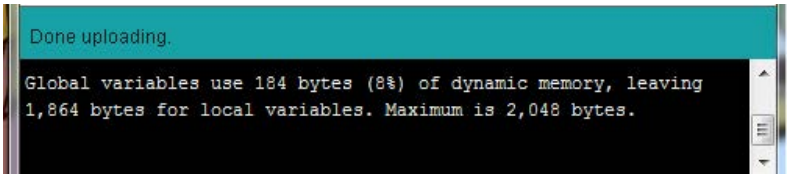
Note: the code that is shown inside the Arduino IDE after you hit **Upload to Arduino** is a direct representation of the blocks.

After you click the Upload to Arduino button, a window will pop up to set the file name of the program. Set this to whatever you like and click Save.



3

When compilation and uploading is complete the black window at the bottom of the Arduino IDE should look something like the image below.

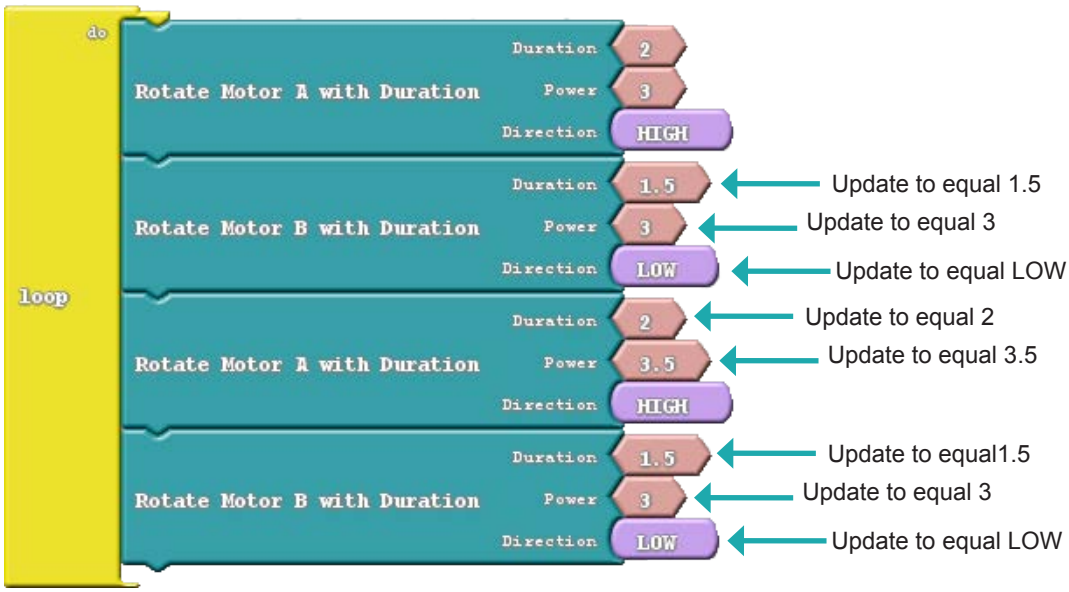


The motors may begin to rotate a little bit. For them to work at full power, turn the battery pack on.

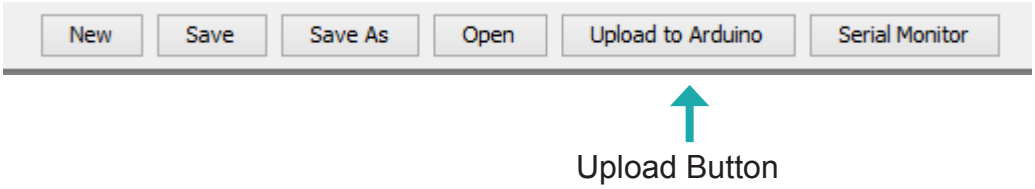
Once on, you should observe motor A rotating for 2 seconds, motor B rotating at the same speed for 1 second, then motor A rotating again for 2 seconds as the program loops over and over...

You can chain any sequence of commands in the loop block and it will run those commands indefinitely until you turn off and unplug the Arduino.

Switch back to ArduBlock and go to the **Artbotics** drawer. Select another **Rotate Motor A with Duration** block, and another **Rotate Motor B with Duration** block, and place them underneath the other blocks inside of the loop. Update their variables as shown below.

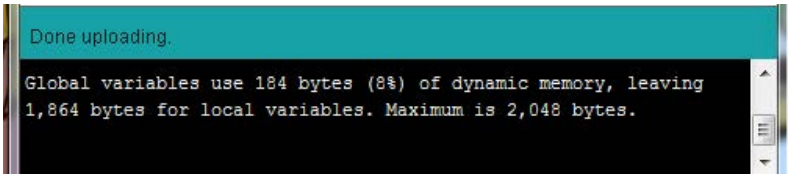


4 Now that you have your programming blocks setup, **make sure your Arduino is plugged in with its battery pack turned off** and press the **Upload to Arduino** button. After compilation, the Arduino IDE will upload the data to the Arduino.



Upload Button

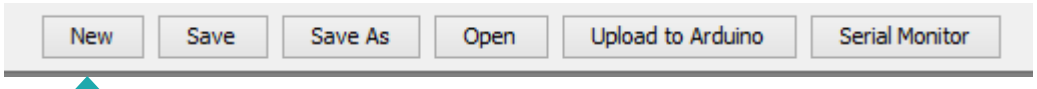
When compilation and uploading is complete the black window at the bottom of the Arduino IDE should look something like the image below.



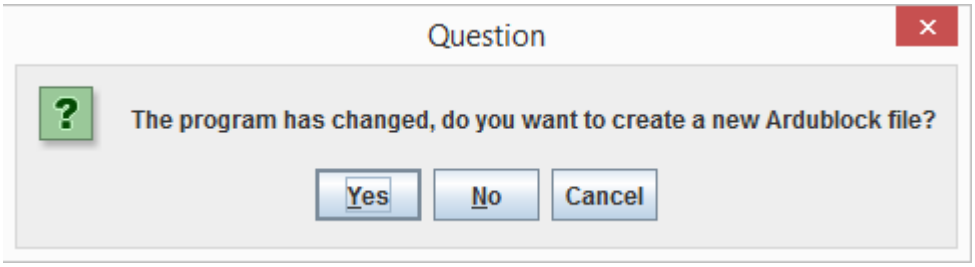
The motors may begin to rotate a little bit. For them to work at full power, turn the battery pack on.

Once on you should observe motor A rotating in the HIGH direction for 2 seconds, then motor B rotating at the same speed in the LOW direction for 1.5 seconds, then motor A rotating in the HIGH direction slightly faster for 2 seconds and finally motor B rotating in the LOW direction for 1.5 seconds.

You can chain any sequence of commands to the loop block and it will run those commands indefinitely until you turn off and unplug the Arduino.



New Button

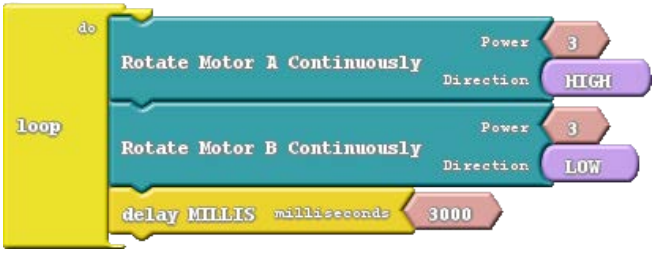


Press the **New** button on the menu bar at the top of the ArduBlock program. A pop up will appear, click **Yes** and the coding area will be cleared leaving only the **Loop Block**

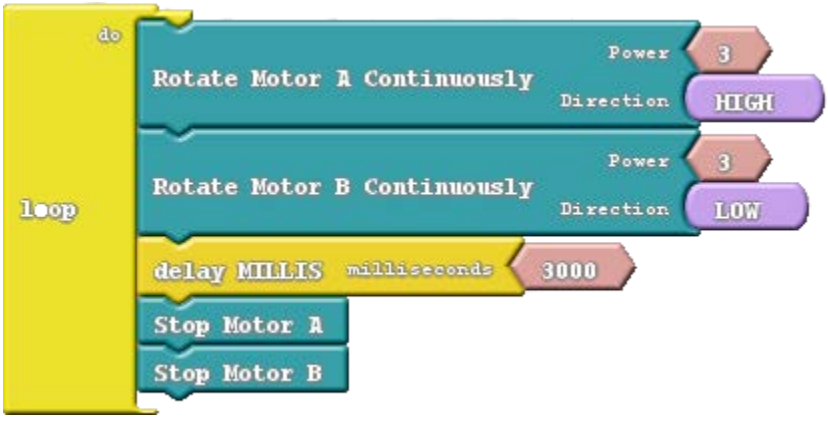
5

Go to the **Artbotics Drawer** and select **Rotate Motor A Continuously** and snap it into the loop block. The rotate continuously block will rotate a motor indefinitely at a given power and direction until it is commanded to stop with a different block.

Go to the **Artbotics Drawer** and select **Rotate Motor B Continuously** and snap it into the loop block.



Go to the **Control Drawer** and select **delay MILLIS** and snap it into the loop block. Then update its milliseconds value to 3000. This block will cause our program to stop executing code blocks for the given duration. In this case 3000 milliseconds, or 3 seconds.



Go to the **Artbotics Drawer** and select **Stop Motor A** and snap it into the loop block. Then go back to the **Artbotics Drawer** and select **Stop Motor B** and snap it into the loop block as well.

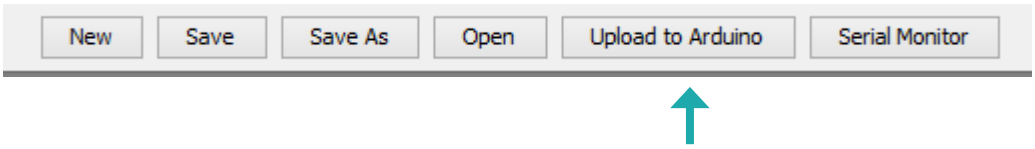
These two code blocks will tell motor A and motor B to shut off when they are executed.

6

Go to the **Control Drawer** and select **delay MILLIS** and snap it into the loop block. Then update its milliseconds value to 3000.

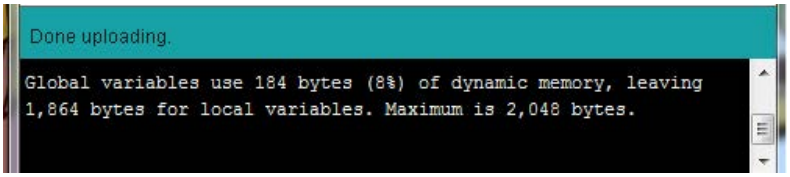
Another delay is being added in order to observe the motors turning off. Without this second delay the motors would turn back on immediately because it is inside of a loop.

Now that you have your programming blocks setup, **make sure your Arduino is plugged in with its battery pack turned off** and press the **Upload to Arduino** button. After compilation, the Arduino IDE will upload the data to the Arduino.



Upload Button

When compilation and uploading is complete the black window at the bottom of the Arduino IDE should look something like the image below.



The motors may begin to rotate a little bit. For them to work at full power, turn the battery pack on.

Once on you should observe Motor A rotating HIGH and motor B rotating LOW for 3 seconds, then the motors will turn off for 3 seconds, finally the program will loop back around and turn the motors on again.

You can chain any sequence of commands to the loop block and it will run those commands indefinitely until you turn off and unplug the Arduino.