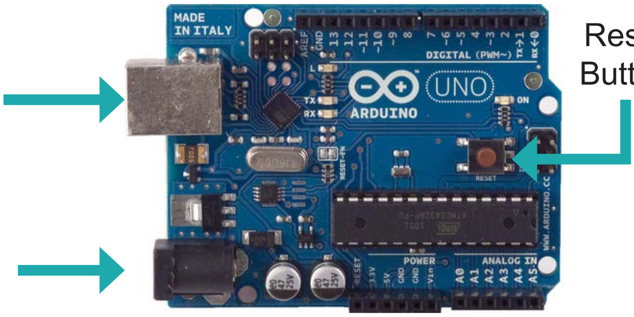
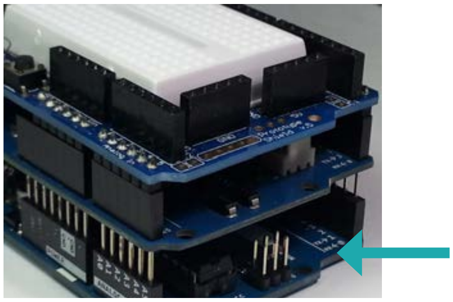
Reacting to Sensors

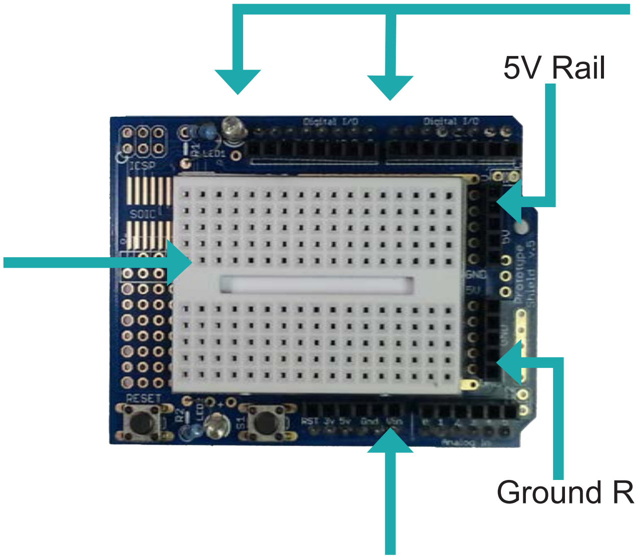
In this tutorial we will be going over how to program the Arduino to react to sensors. By the end of this workshop you will have an understanding of how to use sensors with the Arduino and create an interactive sculpture.

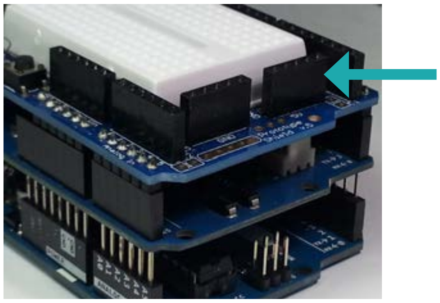
This is the Arduino Uno and motor shield:





This is the Protoshield



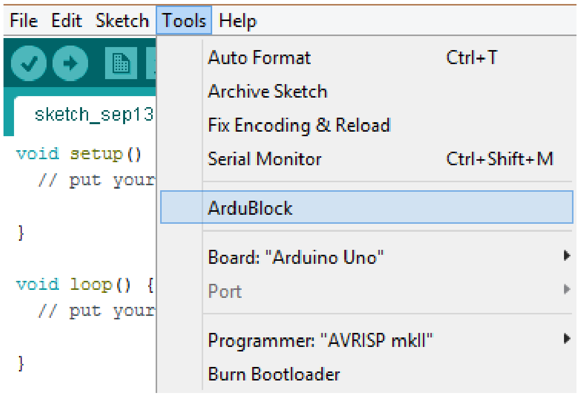


The Proto shield is a bare board that gives us access to a bread board. It also has a 5V and GND rail. We can use these to easily wire up sensors without the need of soldering.

Reacting to Sensors / Starting your program

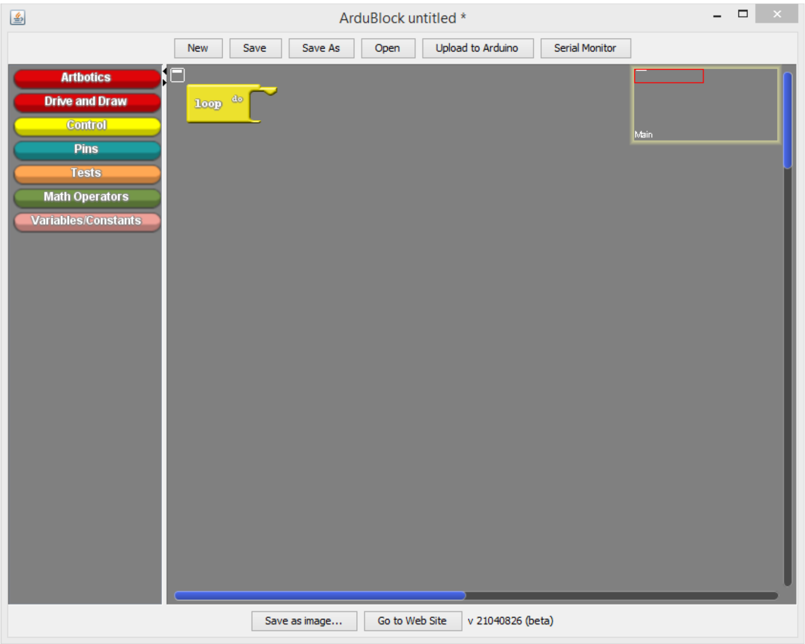
Open the Arduino IDE that was illustrated earlier. An icon like the one on the right here should be on your desktop. If not, navigate to where programs are generally installed or ask your instructor for assistance.

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



```
void setup()
  // put your

}

void loop() {
  // put your

}
```

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



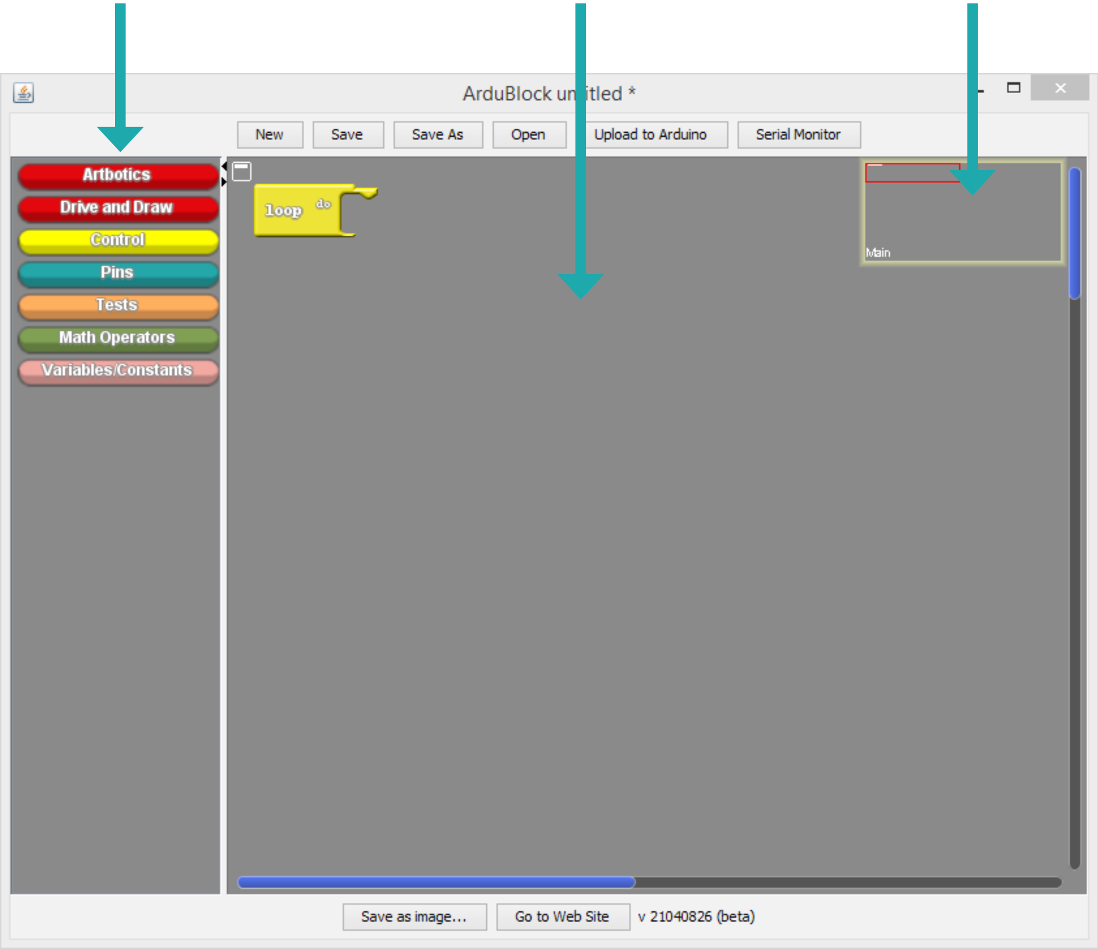
Reacting to Sensors / Where to program

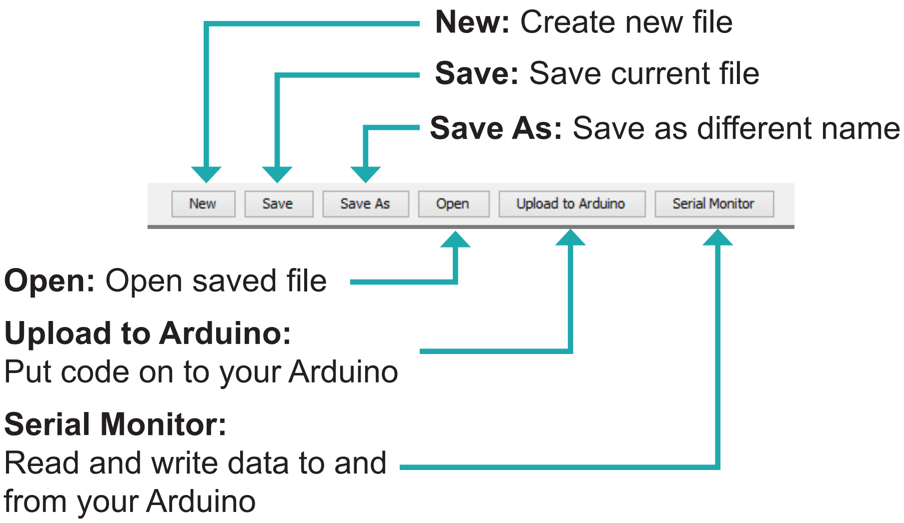
This is the ArduBlock Programming Tool

Block Drawer: 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

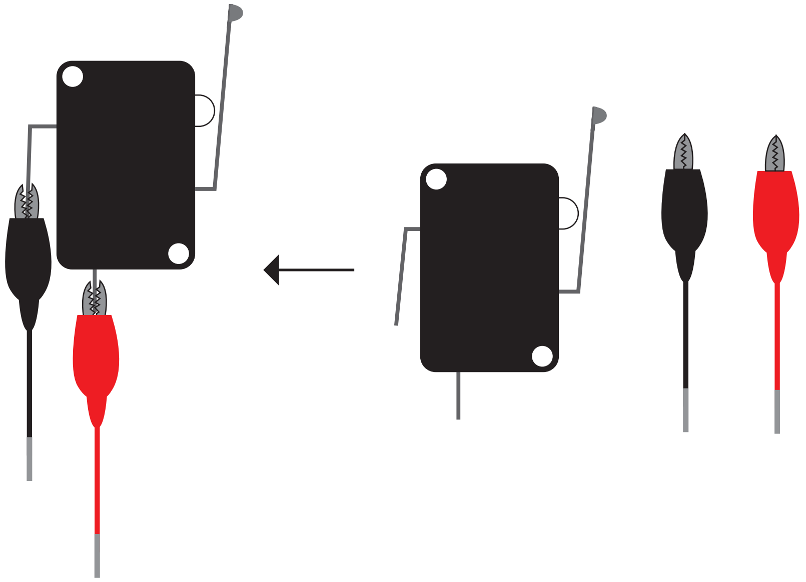
With this programming tool we will be able to design code and upload it to the Arduino Uno. By following these steps you will have your Arduino powered car up and running in no time.

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

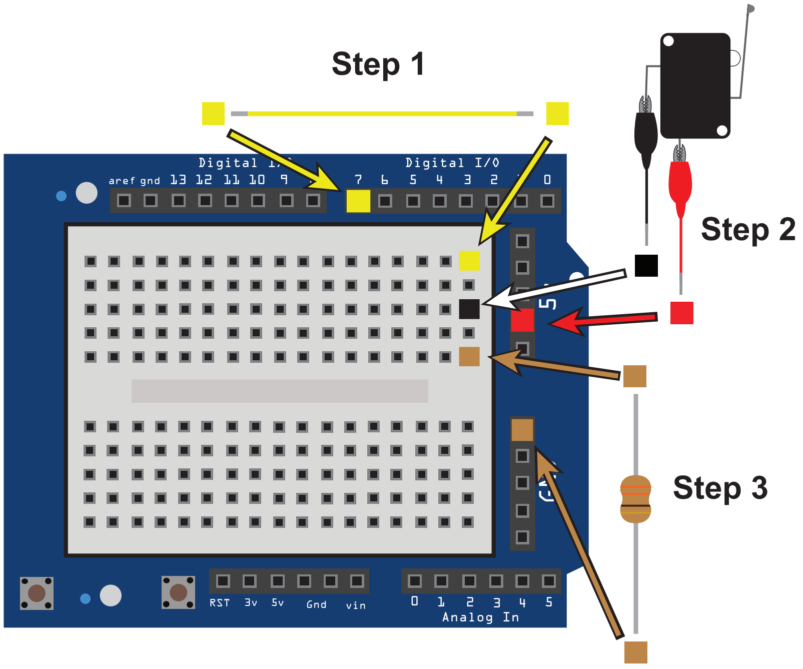
Reacting to Sensor / Button

1

Follow these steps to wire your button to your Arduino



Connect black and red alligator clips to the button’s terminals.



Step1:
Take the yellow wire and plug it into the same column on the breadboard as your switch and the other end into digital pin 7

Step2:
Take the red wire from the button and plug it into 5v
Take the black wire from the button and plug it into the breadboard.

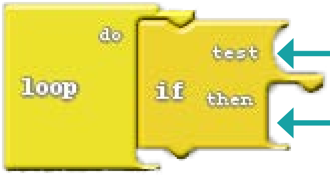
Step3:
Take the resistor and plug it into the same column on the breadboard as your switch and plug the other end into GND

Reacting to Sensors / Button

2 Go to the **Control drawer** and open it. Pick out the **If** block from the drawer and snap it onto the loop block currently present in your coding area.

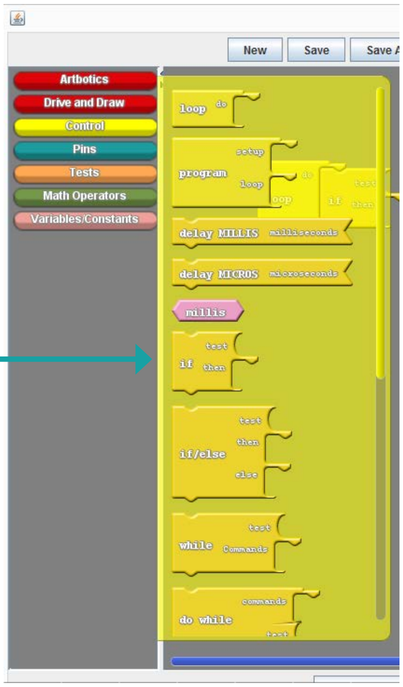
The **If** block is a special control block used to execute commands when certain conditions are met.

The condition is slotted in **test**; code to be executed will be slotted in **then**.



Condition to check

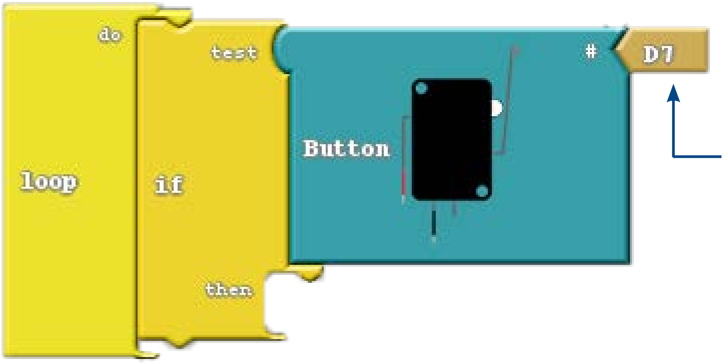
Code to execute



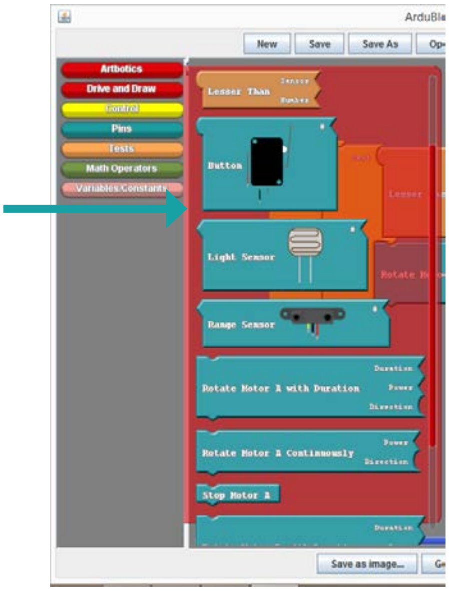
If block in Control drawer

Go to the **Artbotics drawer** and select the **Button** block; notice how it has a rounded plug on its top left corner. Connect this block with the matching rounded **test slot** on the **If** block.

The **test slot** takes any argument that can have a boolean result. Boolean values can either be true or false. In this case, true is when the button is pressed, false is when it is released.



Update to equal D7, which corresponds to digital pin 7 where it is plugged in on the Proto Shield.



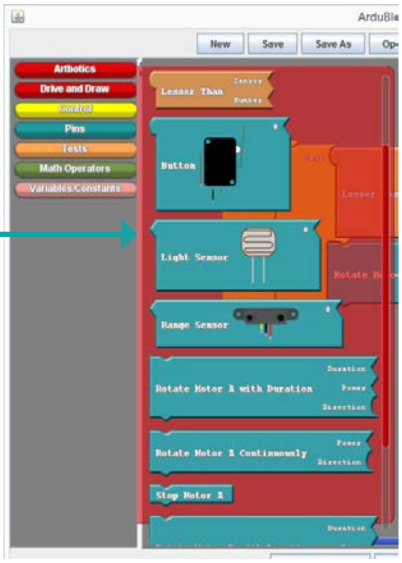
Button Block

Reacting to Sensors / Button

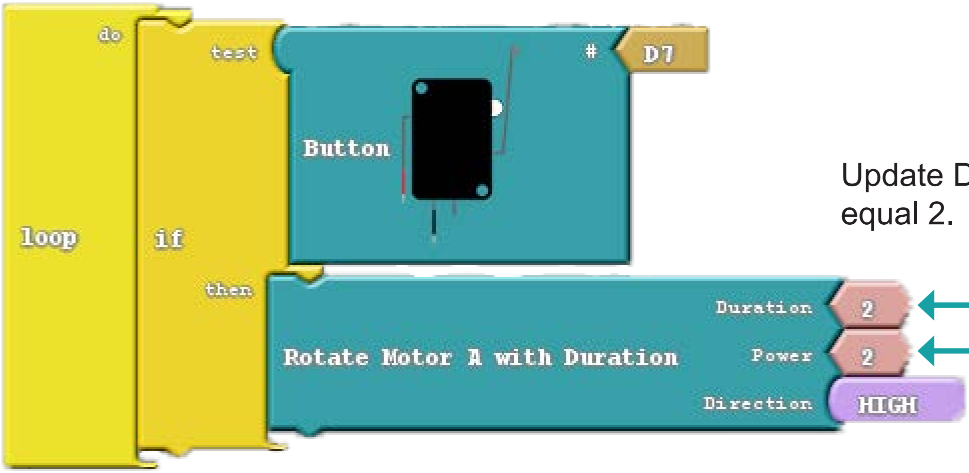
3

Go to the **Artbotics drawer** and select the **Rotate Motor A with Duration** block, this block is a **Command** block for Motor A**.**

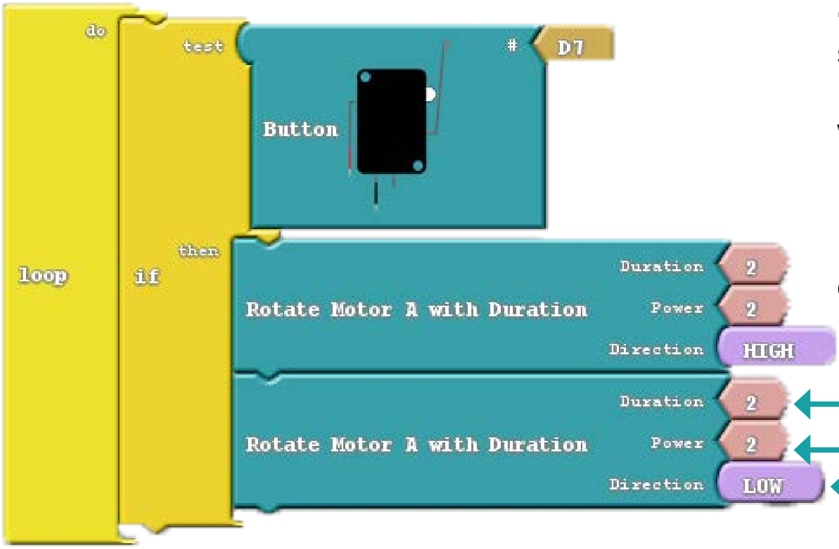
Connect the **Rotate Motor A with Duration** block into the **then slot** of the **If** block**.**



Rotate Motor A with Duration



Update Duration and Power blocks to equal 2.



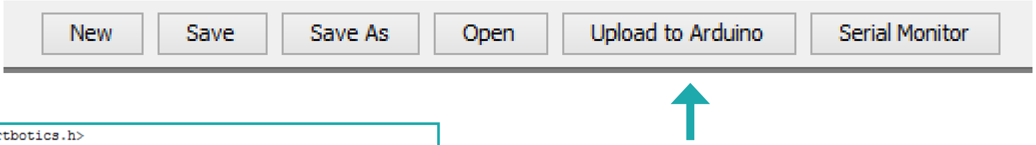
Go to the **Artbotics drawer** again and select another **Rotate Motor A with Duration** block. Connect this block with the previous **Rotate Motor A with Duration** block**.**

Update Duration and Power blocks to equal 2 and Direction to equal LOW.

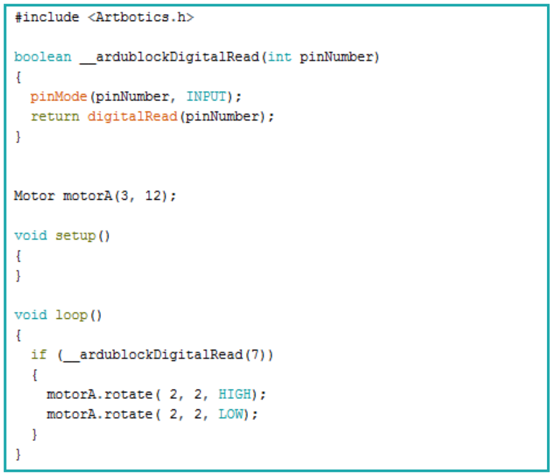
Reacting to Sensors / Range Sensor

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. 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.



Upload Button

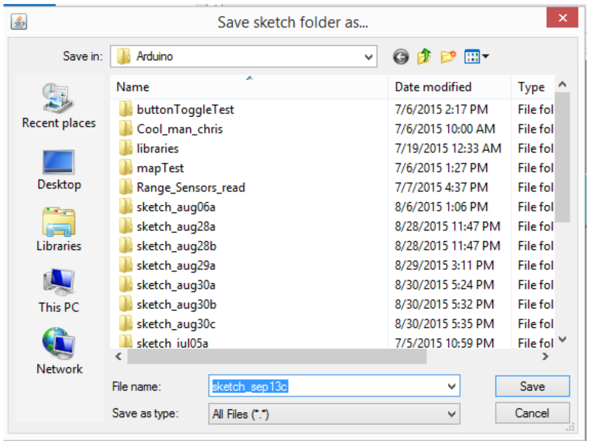


```
#include <Artbotics.h>

boolean __ardublockDigitalRead(int pinNumber)
{
  pinMode(pinNumber, INPUT);
  return digitalRead(pinNumber);
}


Motor motorA(3, 12);

void setup()
{
}

void loop()
{
  if (__ardublockDigitalRead(7))
  {
    motorA.rotate( 2, 2, HIGH);
    motorA.rotate( 2, 2, LOW);
  }
}
```

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

You could alternatively edit and modify this code to get the same results in the following exercises.

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.

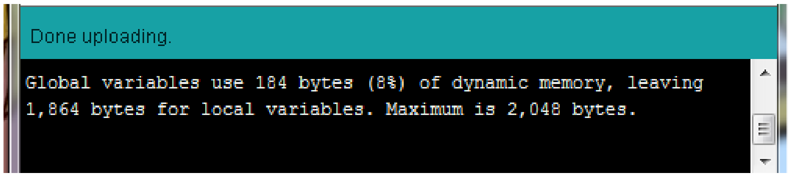


Name	Date modified	Type
buttonToggleTest	7/6/2015 2:17 PM	File fol
Cool_man_chris	7/6/2015 10:00 AM	File fol
libraries	7/19/2015 12:33 AM	File fol
mapTest	7/6/2015 1:27 PM	File fol
Range_Sensors_read	7/7/2015 4:37 PM	File fol
sketch_aug06a	8/6/2015 1:06 PM	File fol
sketch_aug28a	8/28/2015 11:47 PM	File fol
sketch_aug28b	8/28/2015 11:47 PM	File fol
sketch_aug29a	8/29/2015 3:11 PM	File fol
sketch_aug30a	8/30/2015 5:24 PM	File fol
sketch_aug30b	8/30/2015 5:32 PM	File fol
sketch_aug30c	8/30/2015 5:35 PM	File fol
sketch_jul05a	7/5/2015 10:59 PM	File fol

File name: sketch_sep13c

Save as type: All Files (*.*)

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



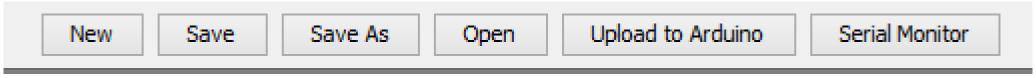
```
Done uploading.

Global variables use 184 bytes (8%) of dynamic memory, leaving
1,864 bytes for local variables. Maximum is 2,048 bytes.
```

5

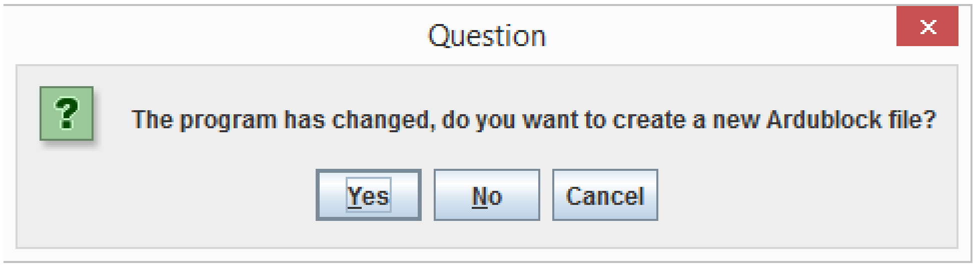
After pressing the button with the battery pack on, you should observe Motor A rotating HIGH for 2 seconds, then motor A rotating LOW for 2 seconds.

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.

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



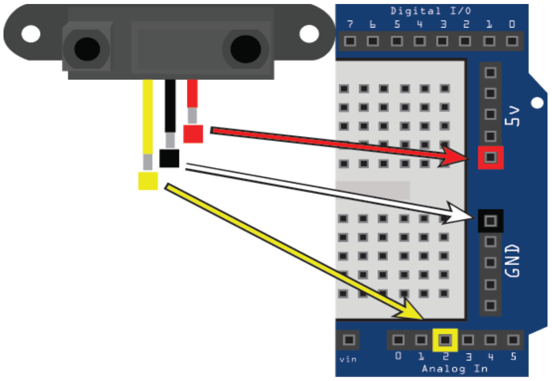
New Button



The program has changed, do you want to create a new Ardublock file?

6

Follow these steps to wire your range sensor to the Arduino

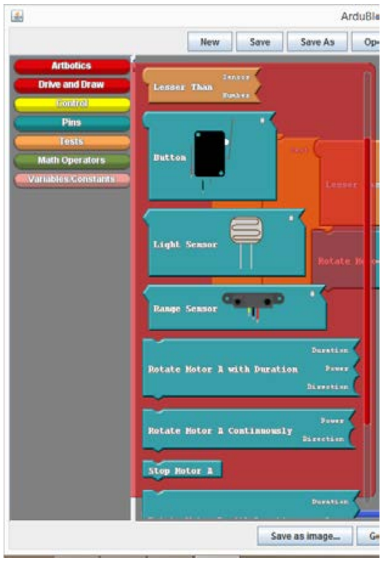


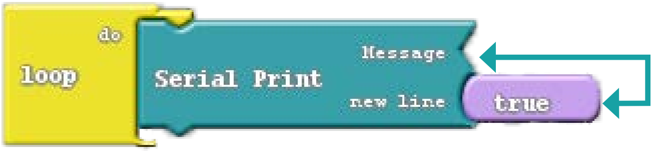
Plug the red wire from the range sensor into 5V, black into GND, and yellow into analog pin 2 (A2)

Reacting to Sensors / Serial Communication

7

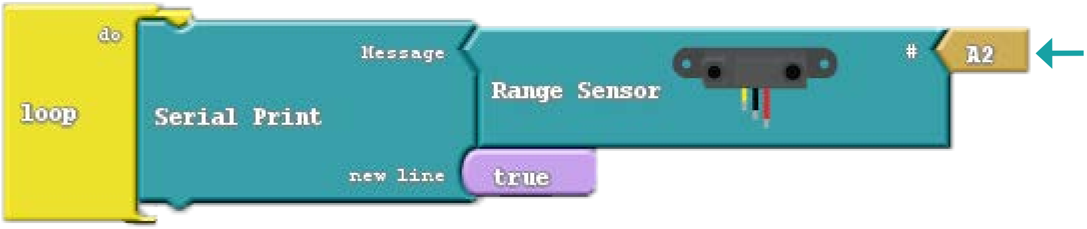
Go to the **Artbotics Drawer** and select **Serial Print** block, This command block takes two blocks as arguments. **Message** which is a number that the command will print out, and **New Line** which determines if values will be printed all on one line or sequentially one after another.





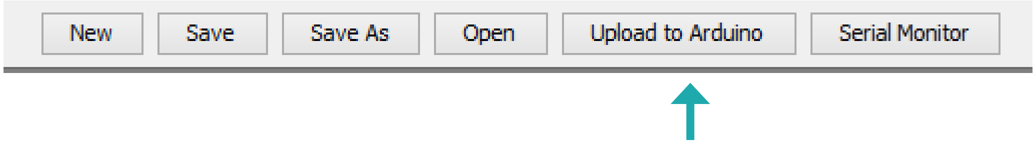
Message slot empty
new line preset to true

Go back to the **Artbotics Drawer** and select the **Range Sensor Block**, this block will return a number equivalent to the distance from its emitter.
Connect the **Range Sensor Block** to the **Serial Print** block by the **Message Slot**.



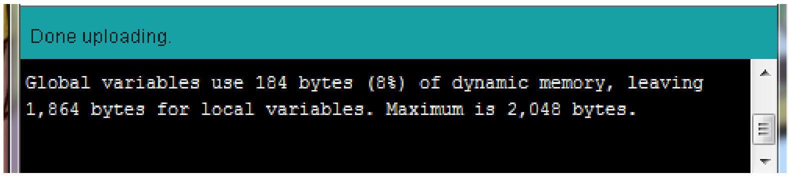
Update the Analog pin from A0 to A2.
A2 (analog pin 2) is the pin used when wiring your range sensor to the proto shield.

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



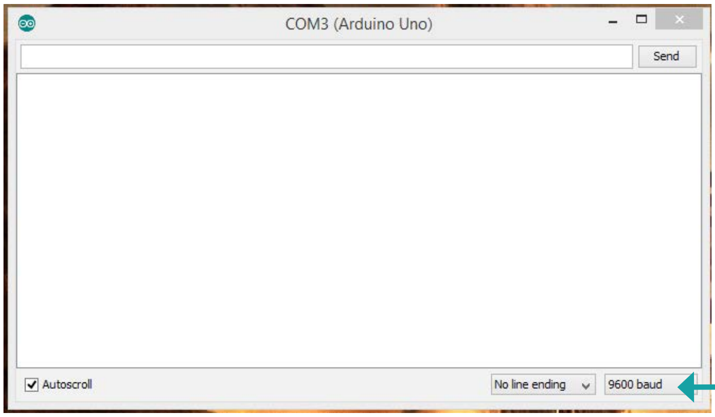
Upload Button

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



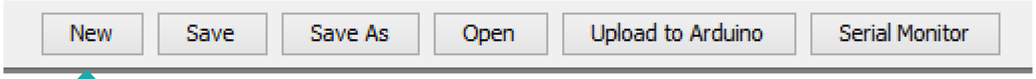
```
Done uploading.

Global variables use 184 bytes (8%) of dynamic memory, leaving
1,864 bytes for local variables. Maximum is 2,048 bytes.
```

8 With your code uploaded and the USB cable still plugged into the Arduino hit the Serial Monitor button 🔍. This will open a console that will start listing numbers every half second. The numbers correspond to the values of the range sensor.

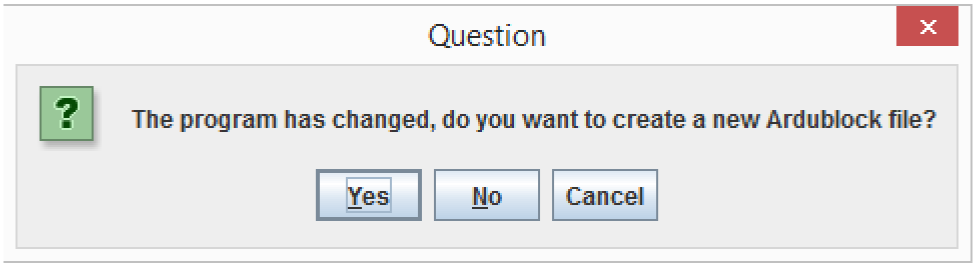


Note: If this value is not 9600, make sure to change it or else you will not get a propper reading.

Place your hand in front of the sensor until it is about 1 foot away. Write down a rough estimate of the value that you see. It should be approximately 300. This number may vary depending on battery life. The number should increase as you move your hand closer to a maxium of roughly 350, and decrease as you move your hand further away to a minimum of about 30.



New Button



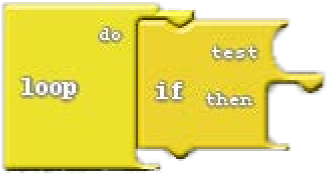
The program has changed, do you want to create a new Ardublock file?

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

Reacting to Sensors / Range Sensor

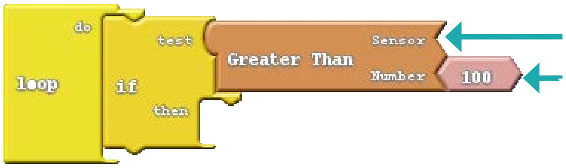
9

Open the **Control Drawer** and select the **If** block, connect it to the **Loop** block.



Open the **Artbotics Drawer** and select the **Greater Than** block, connect it to the **Test Slot** on the **If** block.

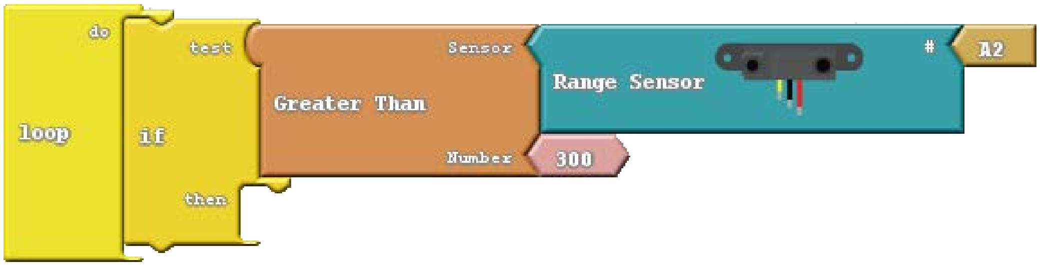
Greater/Less Than blocks return a boolean value. A **Greater Than** block will return true if its sensor reading is greater than the provided constant.



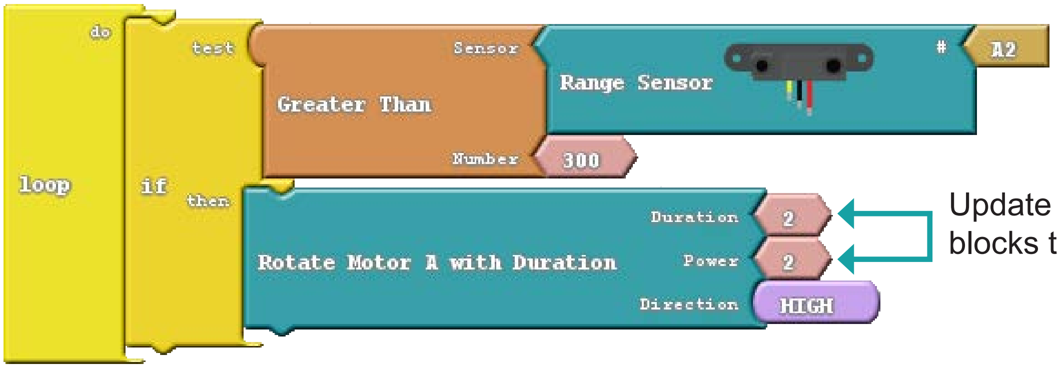
Sensor to compare

Constant to compare with

Open the **Artbotics Drawer** and select the **Range Sensor**, connect it to the sensor slot on the **Greater Than** block.
Then update **Number** to equal the value you got in the previous step; It will be 300 in this example.



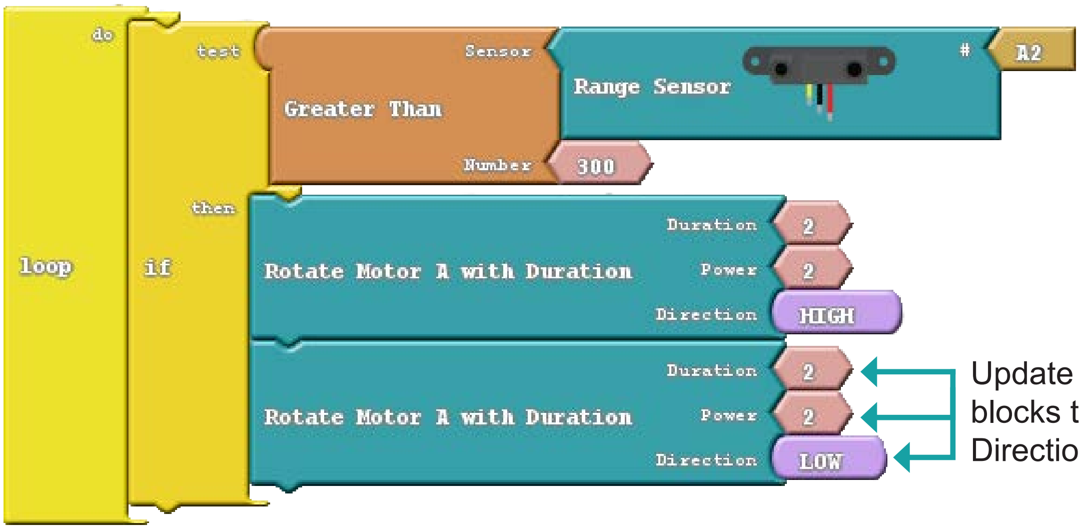
Open the **Artbotics Drawer** and select **Rotate Motor A with Duration,** place this command block in the **then** slot of the **If** block.



Update Duration and Power blocks to equal 2.

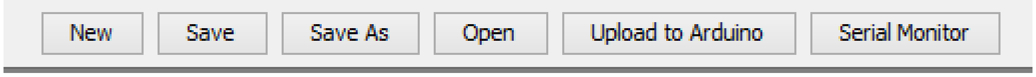
10

Open the **Artbotics Drawer** and select another **Rotate Motor A with Duration,** place this command block in the **then** slot of the **If** block under the first Rotate block.



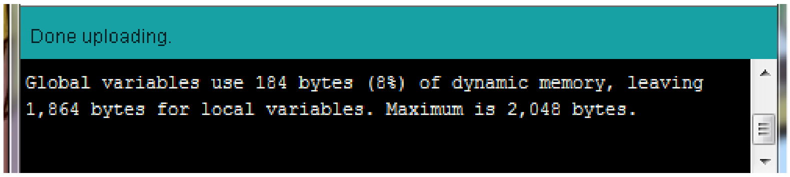
Update Duration and Power blocks to equal 2 and Direction to LOW

Now that you have your programming block setup, **make sure your Arduino is plugged in and turn off the battery pack** 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.



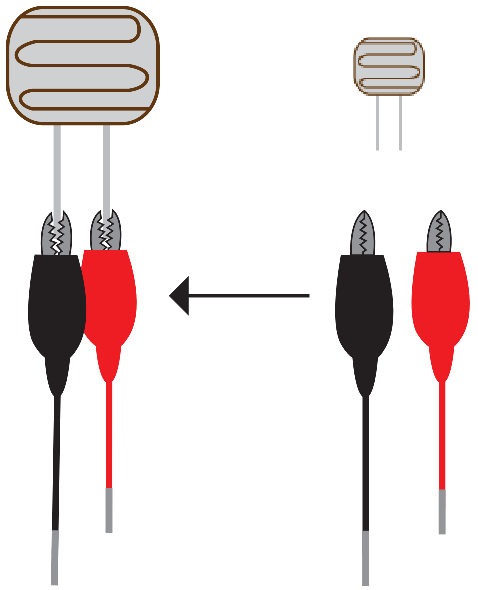
```
Done uploading.
Global variables use 184 bytes (8%) of dynamic memory, leaving
1,864 bytes for local variables. Maximum is 2,048 bytes.
```

After passing your hand over the range sensor with the battery pack on, you should observe Motor A rotating HIGH for 2 seconds, then motor A rotating LOW for 2 seconds.

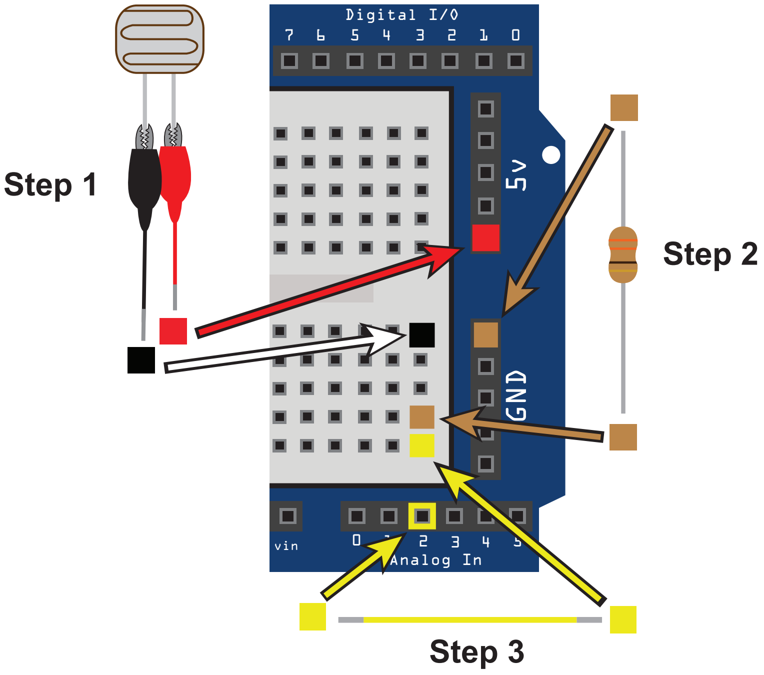
You can chain any sequence of commands to the loop block and it will run those commands indefinitely until you turn off and unplug the Arduino.

11

Follow these steps to wire your light sensor to the Arduino
note: image not to scale



Connect black and red alligator clips to the light sensor's terminals.



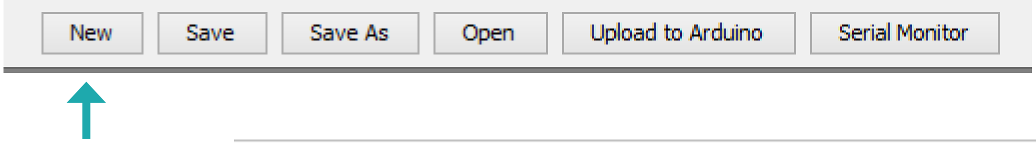
Step1:
Take the red wire from the light sensor and plug it into 5v
Take the black wire from the light sensor and plug it into the breadboard.

Step2:
Take the resistor and plug it into the same column on the breadboard as your light sensor and plug the other end into GND

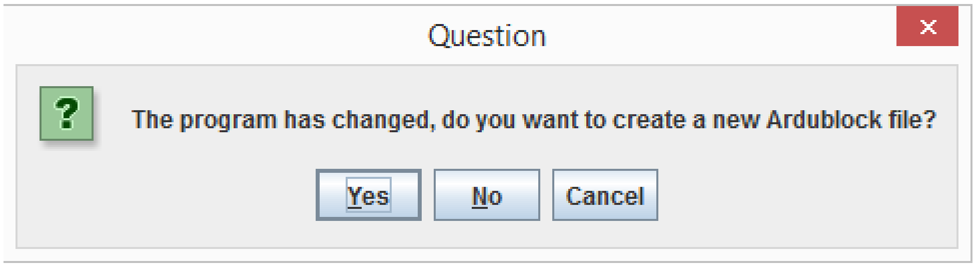
Step3:
Take the yellow wire and plug it into the same column on the breadboard as your light sensor and the other end into Analog pin 2

Reacting to Sensors / Light Sensor

12



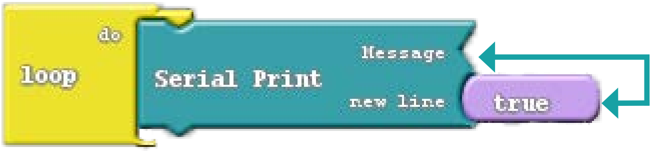
New Button


Question

The program has changed, do you want to create a new Ardublock file?

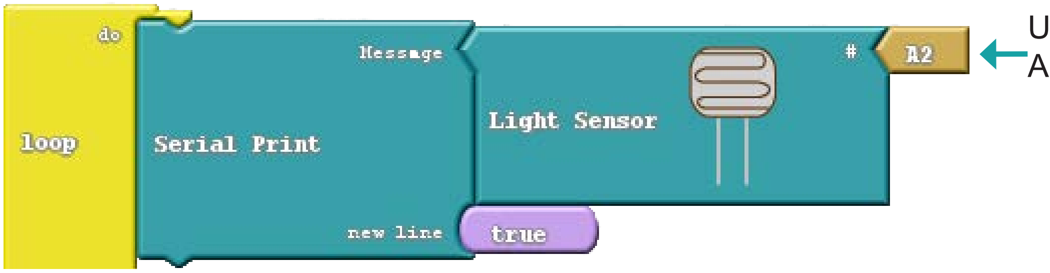
Press the **New** button on the menu bar at the top of the ArduBlock program. A pop up will appear, click **Yes** and the coding area will be cleared leaving only the **Loop** block

Go to the **Artbotics Drawer** and select **Serial Print** block, This command block takes two blocks as arguments. **Message** which is a number that the command will print out, and **New Line** which determines if values will be printed all on one line or sequentially one after another.



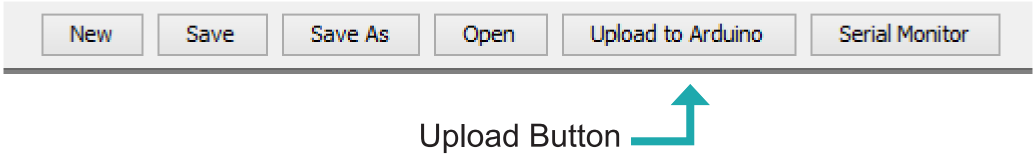
Message slot empty
new line preset to true

Go back to the **Artbotics Drawer** and select the **Light Sensor** block, this block will return a number equivalent to the lighting in the room.
Connect the **Light Sensor** block to the **Serial Print** block by the **Message Slot**.



Update from
A0 to A2

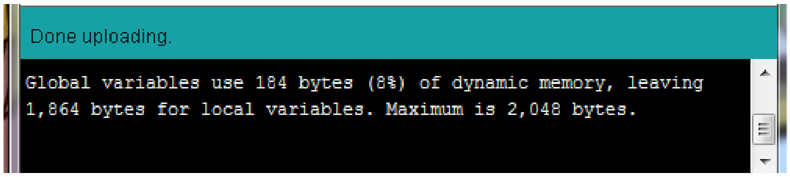
Now that you have your programming blocks setup, **make sure your Arduino is plugged in with its battery pack turned off** and press the **Upload to Arduino** button. After compilation, the Arduino IDE will upload the data to the Arduino.



Upload Button

Reacting to Sensors / Light Sensor

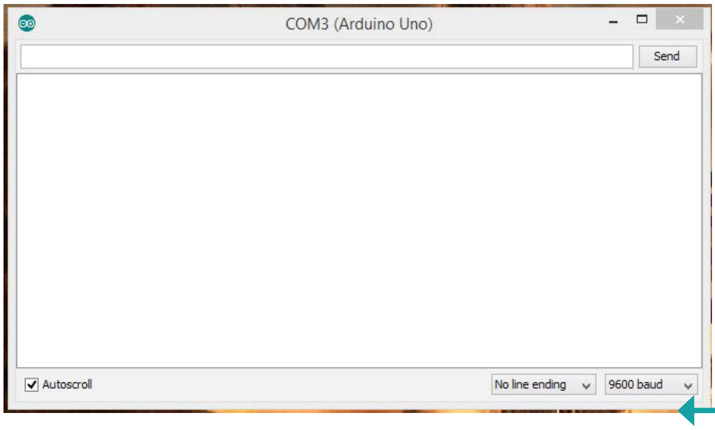
13 When compilation and uploading is complete the black window at the bottom of the Arduino IDE should look something like the image below.



Done uploading.

```
Global variables use 184 bytes (8%) of dynamic memory, leaving
1,864 bytes for local variables. Maximum is 2,048 bytes.
```

With the USB cable still plugged into the Arduino hit the Serial Monitor button . This will open a console that will start listing numbers every half second. The numbers correspond to the values of the light sensor.



COM3 (Arduino Uno)

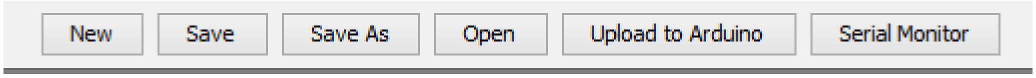
☑ Autoscroll

No line ending

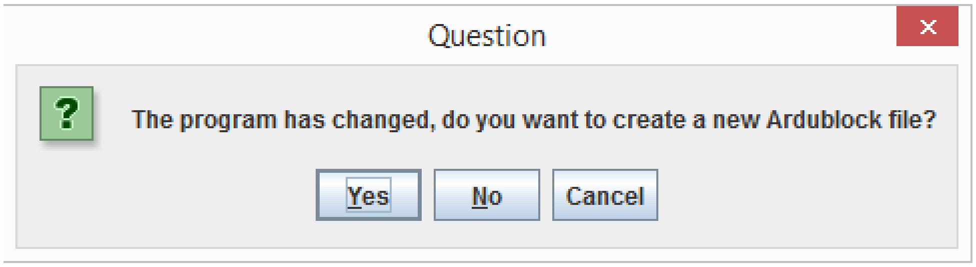
9600 baud

Note: If this value is not 9600, make sure to change it or else you will not get a propper reading.

Now cover the sensor with your hand such that very little light is on it. Take a note of the approximate value you are reading and continue. The value should increase as it gets lighter and decrease as it gets darker. The approximate value will vary with proximity to a light source and ambient lighting in the room.



New Button



Question

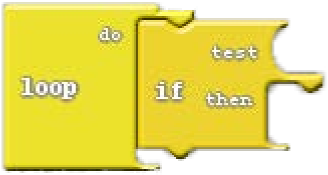
The program has changed, do you want to create a new Ardublock file?

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**

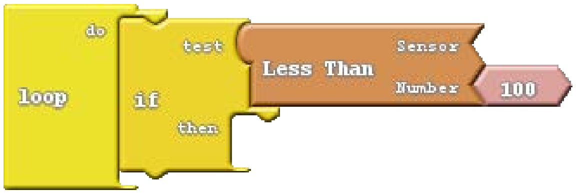
Reacting to Sensors / Light sensor

14

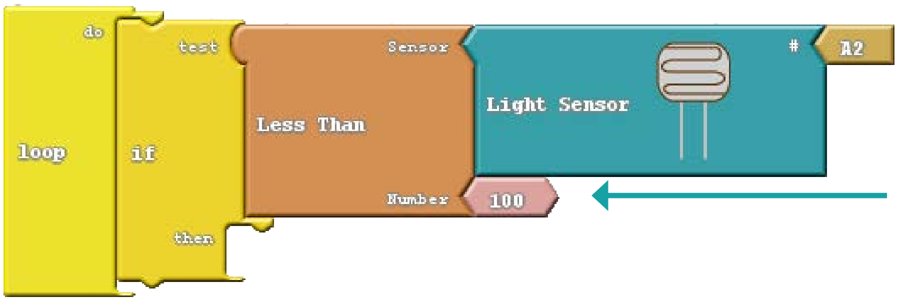
Open the **Control Drawer** and select the **If** block, connect it to the **Loop** block.



Open the **Artbotics Drawer** and select the **Less Than** block, connect it to the test socket on the **If** block.

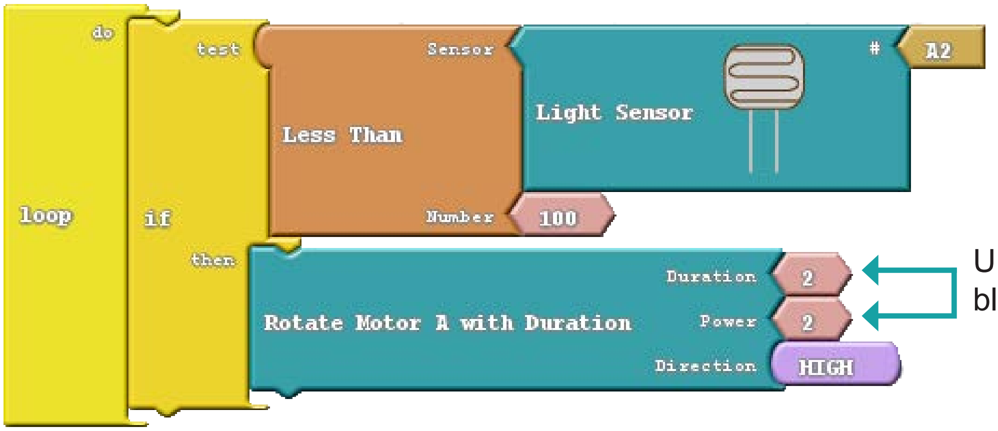


Open the **Artbotics Drawer** and select the **Light Sensor** block, connect it to the Sensor socket of the **Less Than** block.



Note: Update your Number value to reflect your sensor reading in the previous exercise.

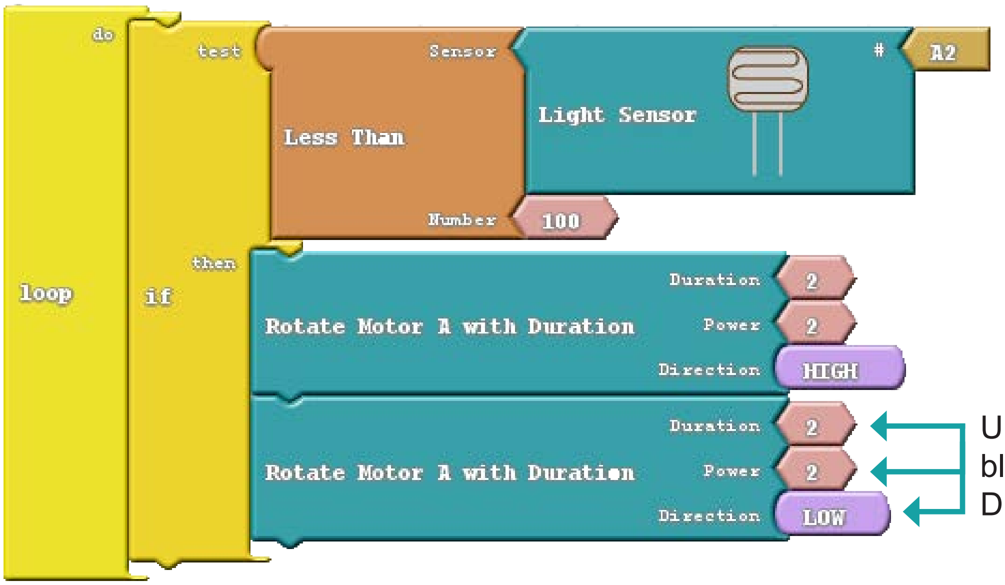
Open the **Artbotics Drawer** and select **Rotate Motor A with Duration**. Then update Duration and Power to equal 2.



Update Duration and Power blocks to equal 2.

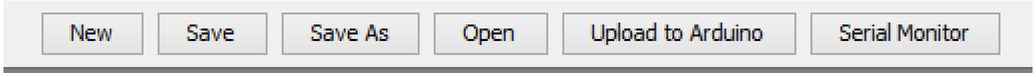
Reacting to Sensors / Light sensor

15 Open the **Artbotics Drawer** and select another **Rotate Motor A with Duration** block, place this command block in the then slot of the **If** block under the first rotate motor block. Update Duration and Power to 2. Update Direction to LOW.



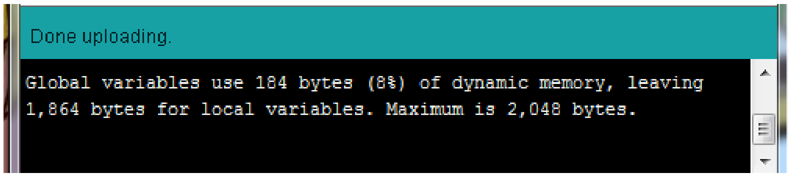
Update Duration and Power blocks to equal 2 and Direction to LOW

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.



```
Done uploading.

Global variables use 184 bytes (8%) of dynamic memory, leaving
1,864 bytes for local variables. Maximum is 2,048 bytes.
```

After covering the light sensor with your hand with the battery pack on, you should observe Motor A rotating HIGH for 2 seconds, then motor A rotating LOW for 2 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.