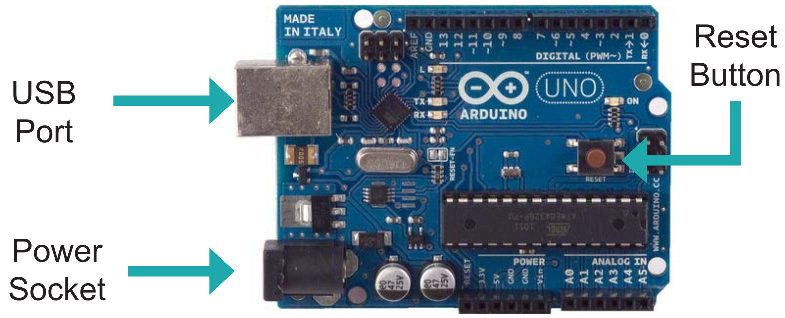
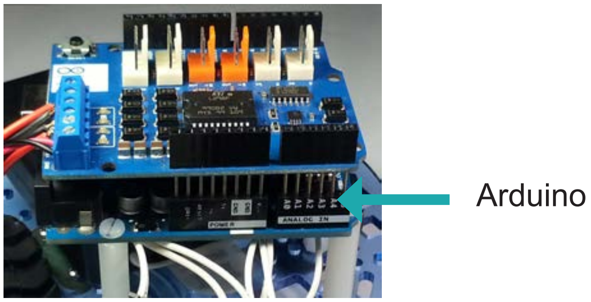
Driving and Drawing

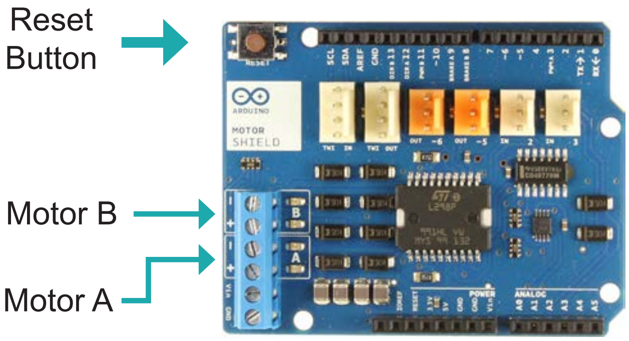
In this tutorial we will be going over how to program the Arduino to drive a car. In this workshop we will introduce some programming fundamentals in a fun exercise, which by the end will have your Arduino drawing neat spirograph patterns and shapes.

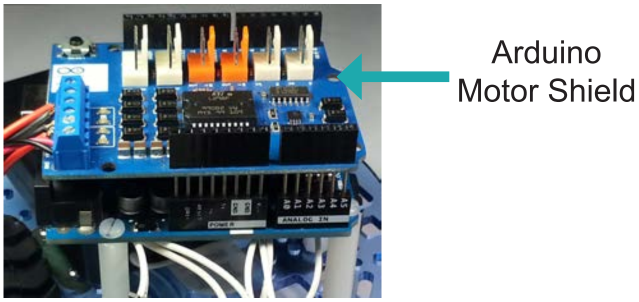
This is the Arduino Uno





This is the Arduino Motor Shield



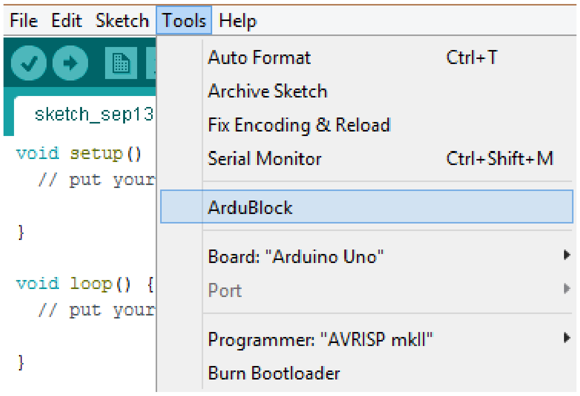


The motor shield is what allows our Arduino to control motors relatively easily without having to worry about too much extra wiring and extra components.

Driving and Drawing

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**.



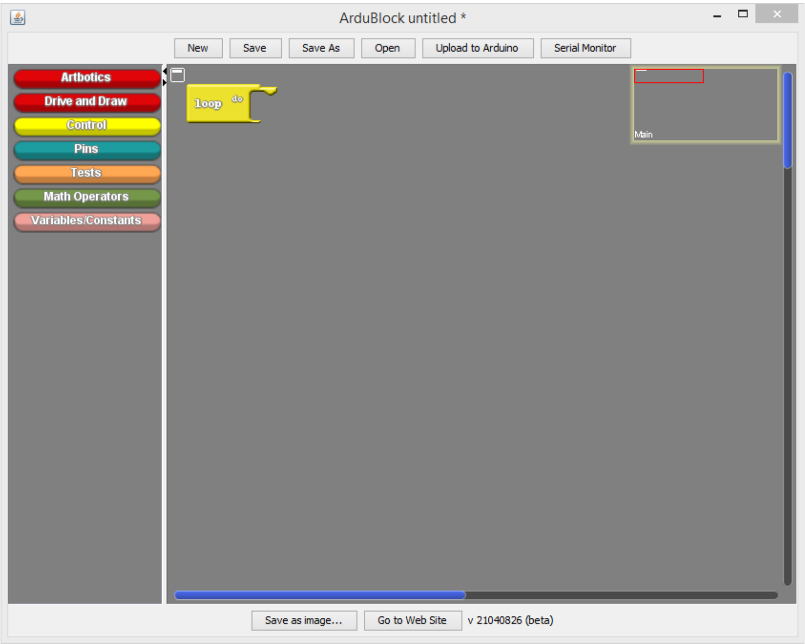
```
void setup()
  // put your

}

void loop() {
  // put your

}
```

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



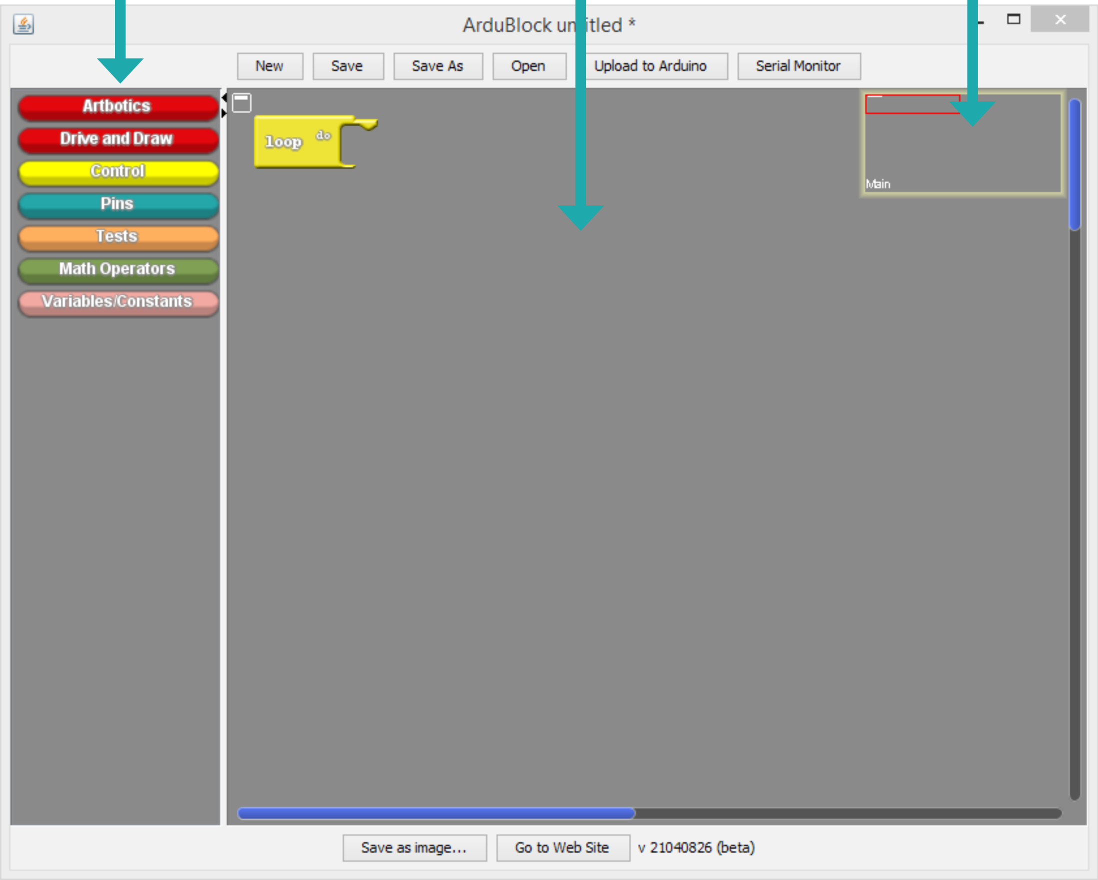
Driving and Drawing / Where to program

This is the ArduBlock Programming Tool

Block Drawers:
Each contain coding blocks

Coding Area:
Area to drop and arrange coding blocks

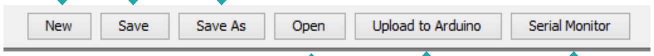
Overview:
Quickly jump to sections of your code



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

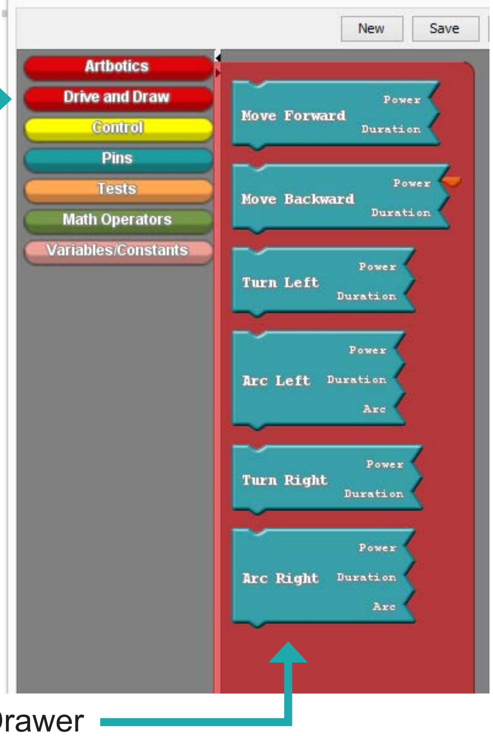
With this programming tool we will be able to design code and upload it to the Arduino Uno. By following these steps you will be able to drive motors for an interactive sculpture.

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

Driving and Drawing

1

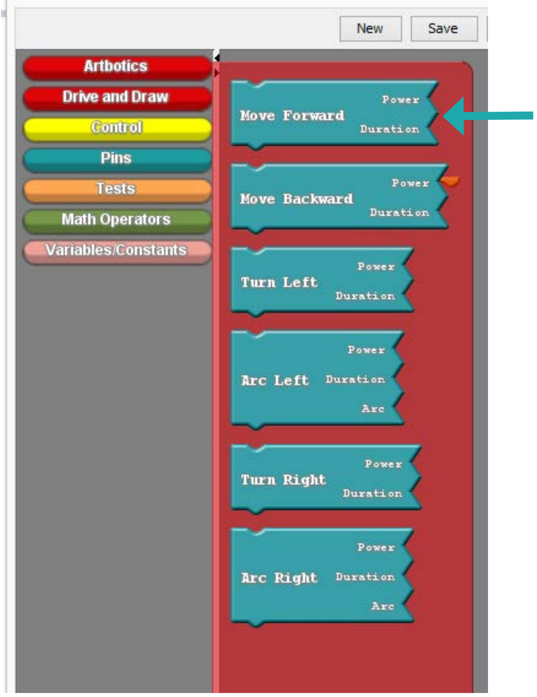
Go to the column on the left and open the **Drive and Draw** block drawer. Here you will see a set of coding blocks designed to drive the Arduino Car.



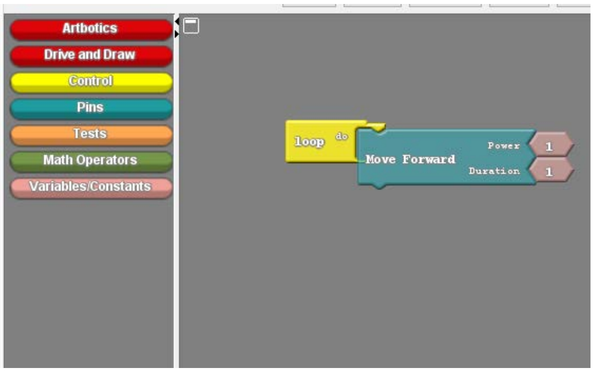
Drive and Draw Drawer Button

Drive and Draw Drawer with Coding Blocks

Select the **Move Forward** block at the top and drag it over into the programming area, lining the socket on the top of the block with the notch on the yellow **Loop** block that is already in the coding area.



Move Forward Block

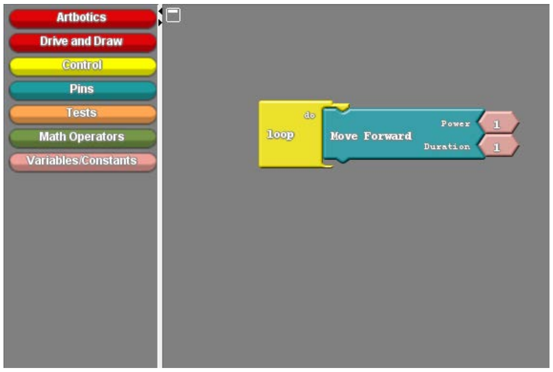


Driving and Drawing

2

You will notice the **loop** block expand and the **Move Forward** block snap in place.

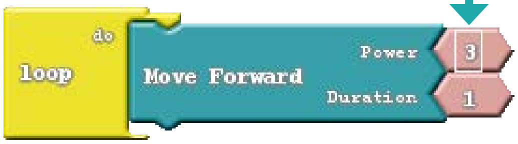
The **loop** block is a control block that is the foundation of any program on Arduino. Any other programming block must be connected within the loop block in order for them to be executed.



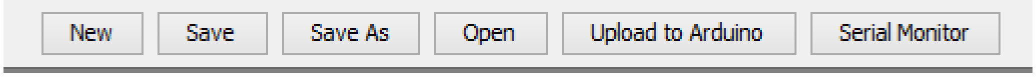
Next update the **number block** (Horizontal diamond) that is slotted in the **Power** socket on **Move Forward**.

Set the move forward command’s Power to 3. Power can be anything between 0 which is off and 5 which is full power.

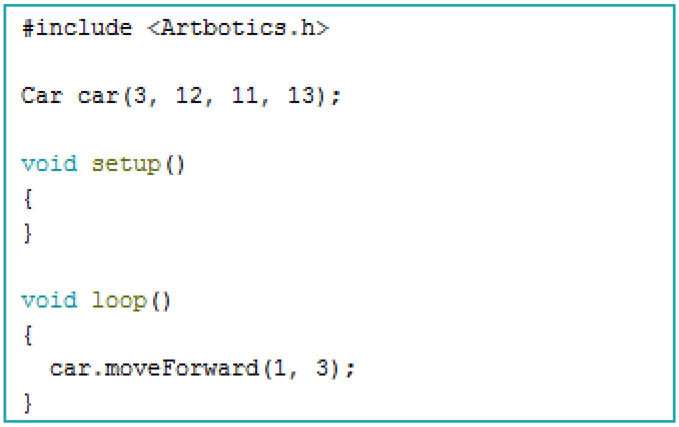
Power: number block



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 Key



```
#include <Artbotics.h>

Car car(3, 12, 11, 13);

void setup()
{
}

void loop()
{
  car.moveForward(1, 3);
}
```

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

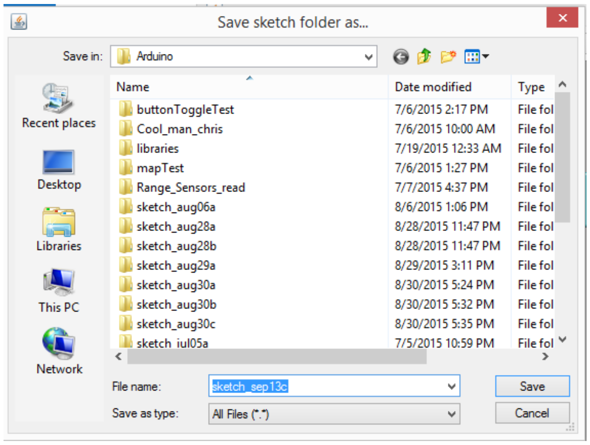
You could alternatively edit and modify this code to get the same results in the following exercises.

Driving and Drawing

3

Note: After uploading your code the program will automatically start running. Be ready to unplug the Arduino or catch it so it doesn't drive away.

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.



Save sketch folder as...

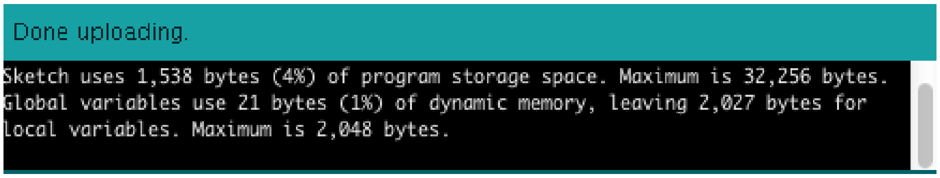
Save in: Arduino

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 the compilation and uploading is complete you should see Done Uploading and text at the bottom of the Arduino Programming IDE simmilar to the image below.

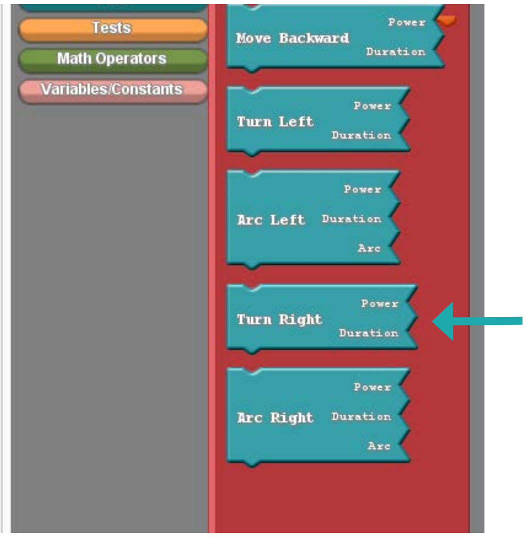


```
Done uploading.
Sketch uses 1,538 bytes (4%) of program storage space. Maximum is 32,256 bytes.
Global variables use 21 bytes (1%) of dynamic memory, leaving 2,027 bytes for
local variables. Maximum is 2,048 bytes.
```

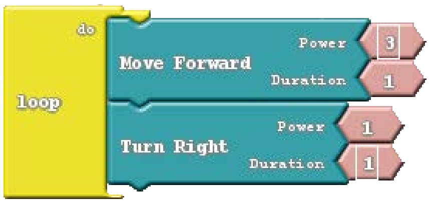
Place the car on a flat surface and switch the battery pack on. The program should start after a moment. If everything works then your car should start driving forward continuously. **Be careful not to let it drive off a table or desk.** When finished, **switch off the battery pack** and continue.

Go to the **Drive and Draw** block drawer and select the **Turn Right** block.

Drag it in place under the **Move Forward** block. It should snap into position, having 1 as both of its default arguments.

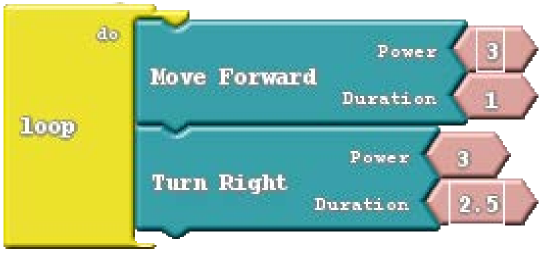


Turn Right Block



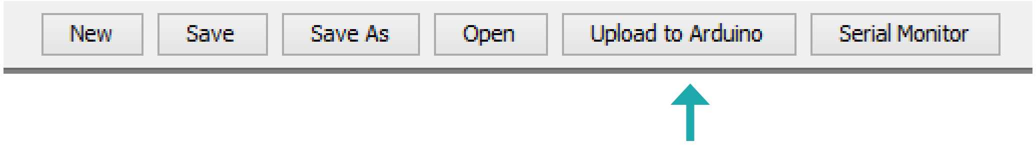
4

Update the default arguments on the turn right block so that **power** is **3** and **duration** is **2.5**.



```
do
loop
  Move Forward   Power 3
                 Duration 1
  Turn Right     Power 3
                 Duration 2.5
```

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

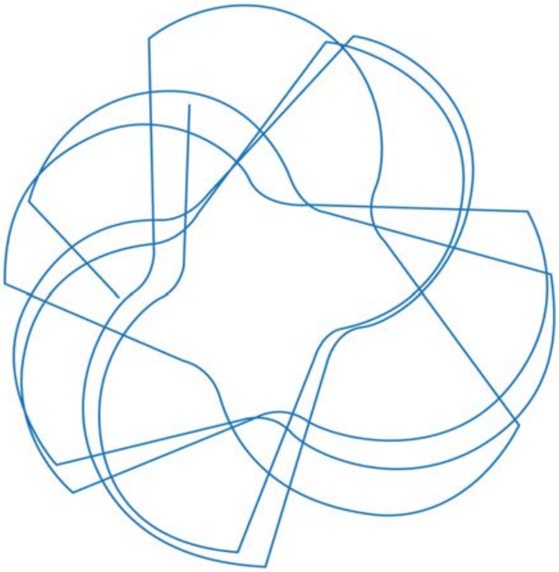


Upload Button

Note: After uploading your code **the program will automatically start running**. Be ready to unplug the Arduino or catch it so it doesn't drive away.

Name your code If the program asks you, then hit save and continue. The Arduino IDE should promptly begin compiling and uploading the interpreted code. When finished unplug the Arduino and continue.

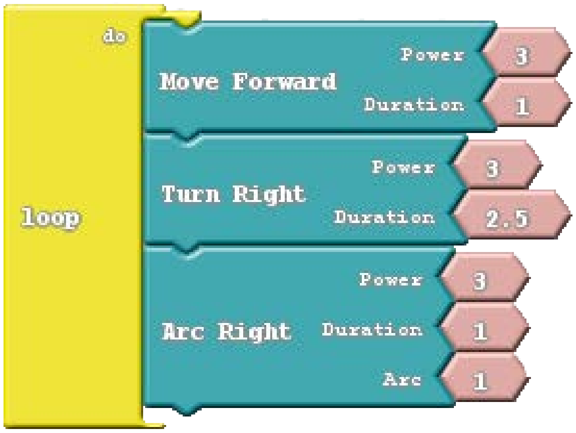
Place the Arduino car into the drawing arena with a marker in the pen holder and switch the battery pack on. Your program should start running, and after a minute or so you should begin to see a pattern like the one below.



When finished, switch off the battery pack on the Arduino car and continue.

5

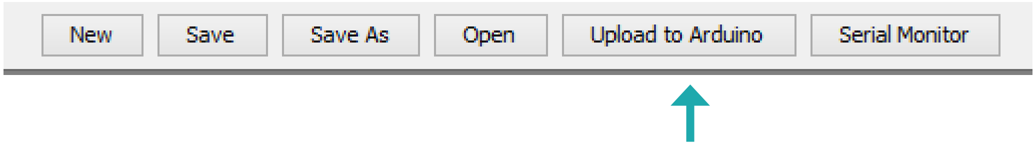
Go back to the **Drive and Draw** block drawer and select an **Arc Right** block.



Arc blocks take a third argument, Arc which is scaled from 0 to 5. Passing 0 will have the car drive straight and 5 will be an in-place turn, anything between 0 and 5 will give a long arc like turn.

Before continuing, update **Power** on **Arc Right** to equal **3**.

Now that you have your programming block 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.



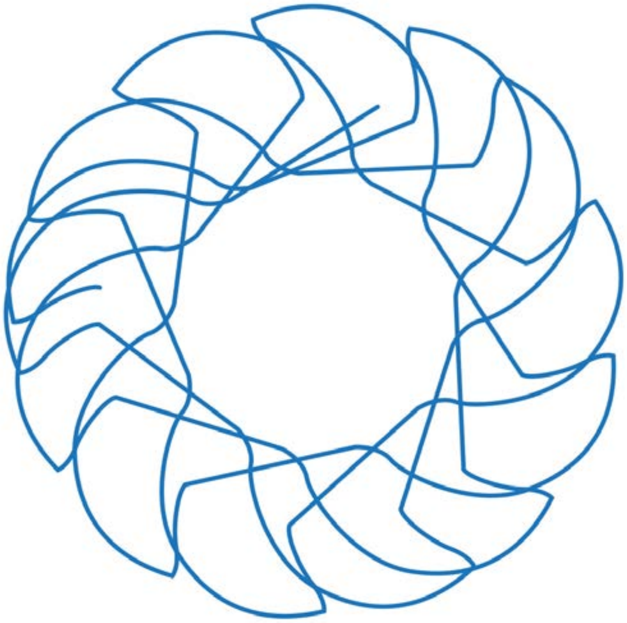
Upload Button

Note: After uploading your code **the program will automatically start running**. Be ready to unplug the Arduino or catch it so it doesn't drive away.

Name your code If the program asks you, then hit save and continue. The Arduino IDE should promptly begin compiling and uploading the interpreted code. When finished unplug the Arduino and continue.

6

Place the Arduino car into the drawing arena with a marker in the pen holder and switch the battery pack on. Your program should start running, and after a minute or so you should begin to see a pattern like the one below.

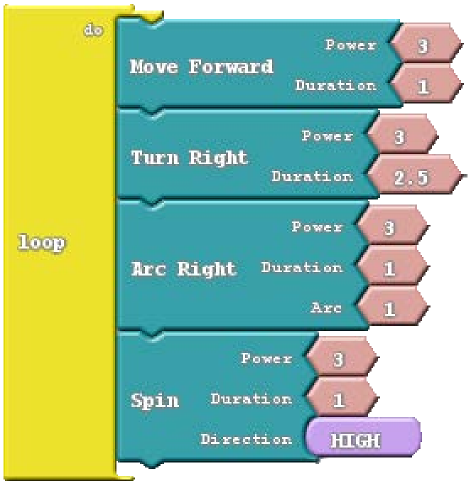


When finished, switch off the battery pack on the Arduino car and continue.

Now go back to the **Drive and Draw** block drawer and select the **Spin** block.

Spin blocks take a third argument, **Direction** which can either be High or Low. Passing High should have the car spin in one direction, Low would have the car spin in the opposite direction.

Before continuing, update **Power** on **Spin** to equal **3**.

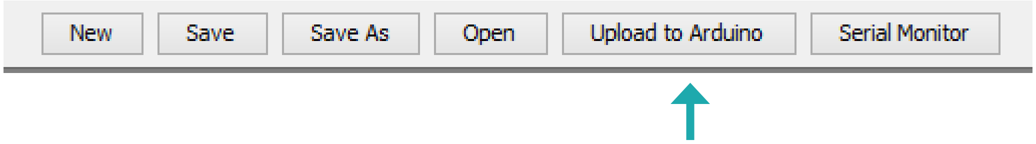


```
loop do
  Move Forward  Power 3  Duration 1
  Turn Right    Power 3  Duration 2.5
  Arc Right     Power 3  Duration 1  Arc 1
  Spin          Power 3  Duration 1  Direction HIGH
```

Driving and Drawing

7

Now that you have your programming block 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.

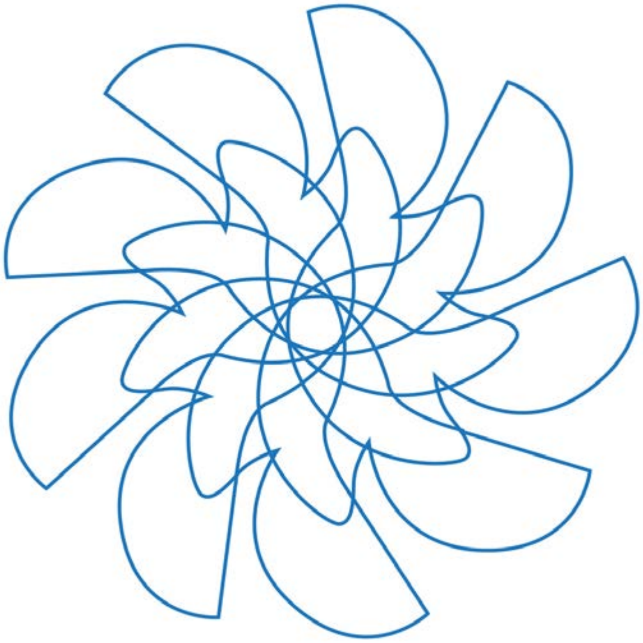


Upload Button

Note: After uploading your code **the program will automatically start running**. Be ready to unplug the Arduino or catch it so it doesn't drive away.

Name your code If the program asks you, then hit save and continue. The Arduino IDE should promptly begin compiling and uploading the interpreted code. When finished unplug the Arduino and continue.

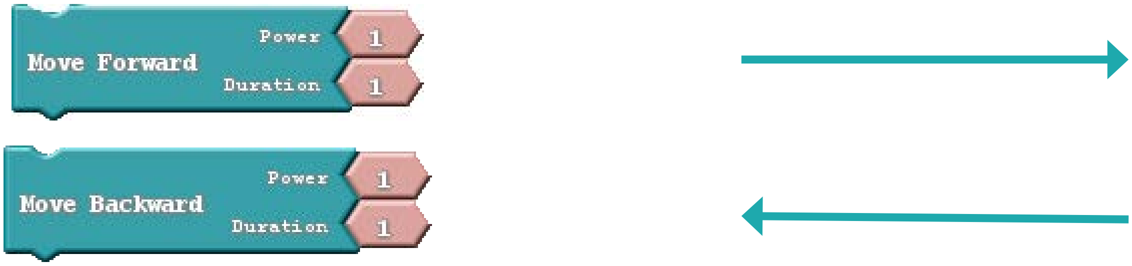
Place the Arduino car into the drawing arena with a marker in the pen holder and switch the battery pack on. Your program should start running, and after a minute or so you should begin to see a pattern like the one below.



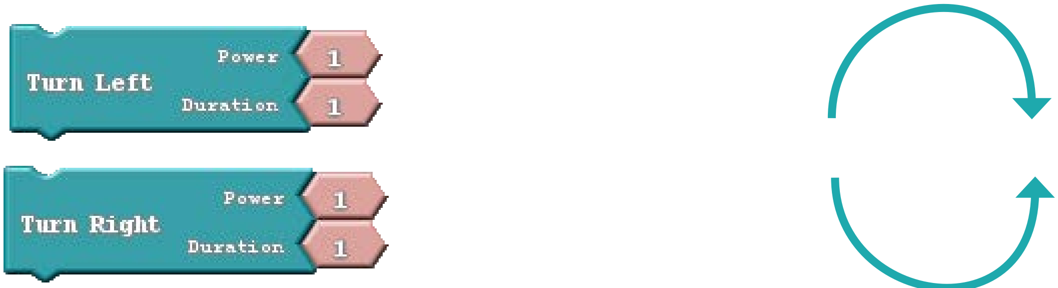
When finished, switch off the battery pack on the Arduino car and begin experimenting with different designs.

Driving and Drawing / Quick Reference

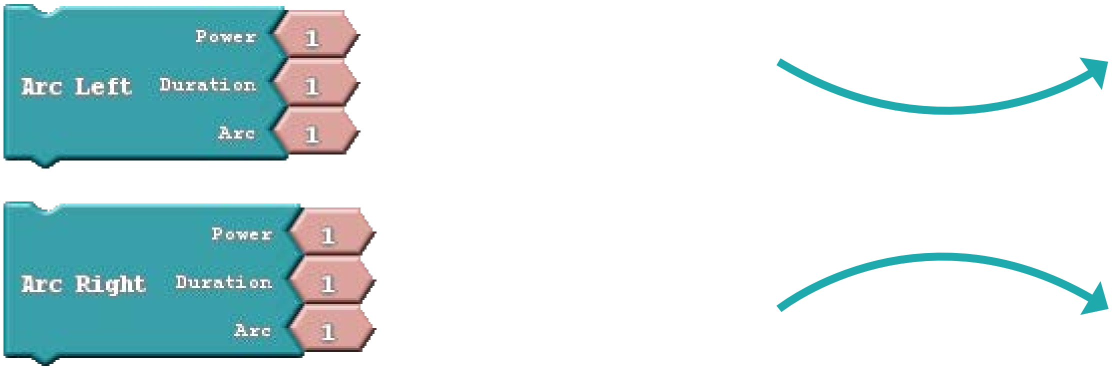
Blocks:



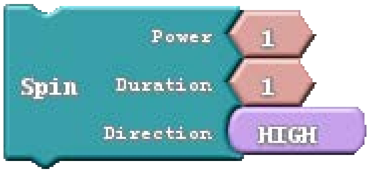
Duration and power both affect the length of the line drawn.

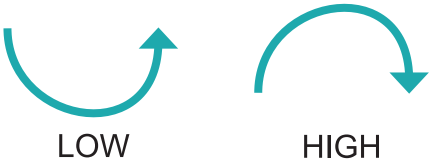


Duration and power both affect the length of ring drawn. For your cars generally 2 seconds duration and 3 power will give you a semi circle (wider then a spin) 4 seconds should give you a full circle.



Arcs are basically straight lines that are curved a little. If you have a small drawing area use long time with medium power and high arc.





Spin is like an in place turn but has a smaller radius. About 1.5 seconds with a power of 3 will give you a half circle. Direction (HIGH or LOW) determines which direction the car will turn.

In our configuration, HIGH should be clockwise and LOW counter-clockwise though this may vary.