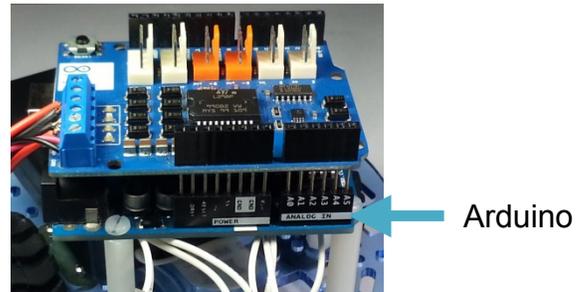
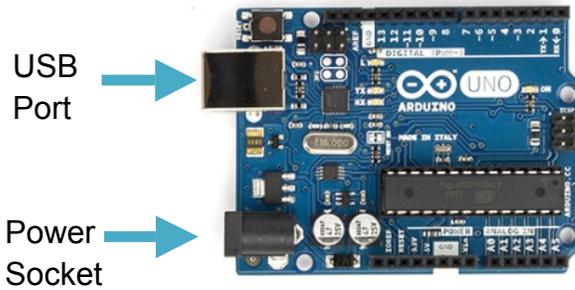
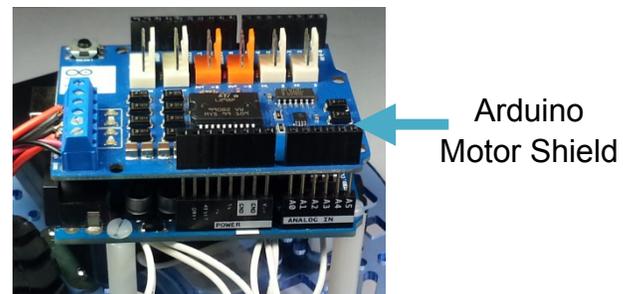
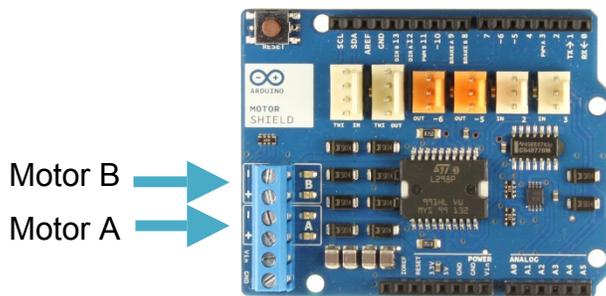


In this tutorial we will review how to program the Arduino to control individual motors such that you can control one or more mechanisms that are attached to the motors.

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 and extra components.

This is the Arduino Software,
called the IDE (Integrated development environment)



Verify:
Check code
for errors

New:
Open new file



Save:
Save current file

Upload: Put code
on to your Arduino

Open:
Open a saved file



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

Editor: Where you
can type your code

Console:
Notifies you of any errors when
compiling or uploading

The line inside the editor
that your cursor is on

Type of Arduino and on what
serial port it is connected to

With this computer program we will be able to write code and upload it to the Arduino Uno.

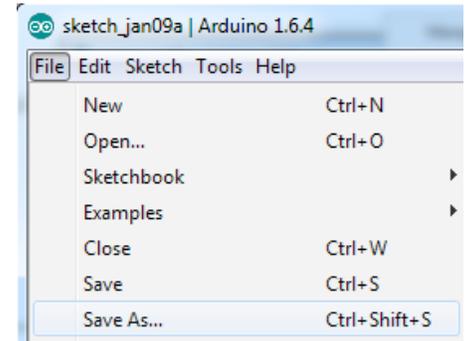
Mechanisms / Starting your program



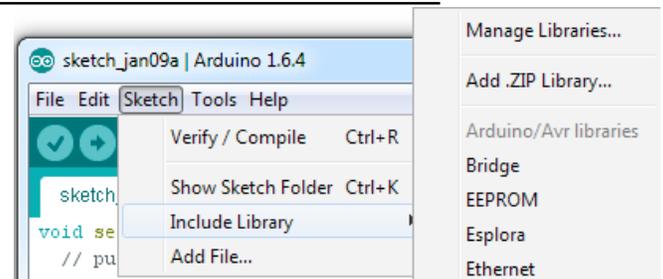
Open the Arduino IDE that was illustrated earlier. An icon like the one on the right here should be on your desktop. If not, navigate to where programs are generally installed or ask your instructor for assistance.



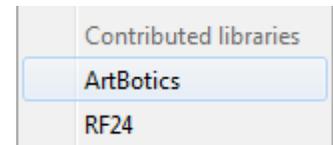
Once open navigate to **File** and **Save As**, giving your program a unique name. It can be *MechBot* or whatever you like.



Before we continue navigate to **Sketch** and then **Include Library**.

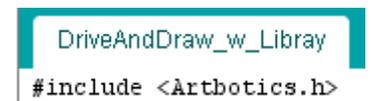


At the very bottom of the next drop down list you should find a library named Artbotics. If not please ask you instructor for assistance.



The top left of the IDE, just above the Editor; you should see a tab with the name of your file. Underneath that at the top of the Editor you should see `#include <Artbotics.h>`.

This line imports the Artbotics library of commands to be used in your program. It gives you access to objects with prewritten code that will make driving the car easier.



Make sure to go back and do the previous step if you don't see this or else you will run into trouble later on.

1

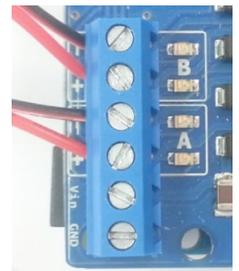
With the Artbotics library setup we can control motors individually using `Motor motorA(3, 12);` and `Motor motorB(11, 13);`. The command `Motor` is a programming object that is defined in the Artbotics library.

Update the code in your IDE to match the code in the box on the right.

```
#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);
}
```

The objects `motorA` and `motorB` will be used to drive the physical motors connected to the Arduino motor shield (shown on the right).



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**.



To make the motor spin we use `motorA.rotate()` and `motorB.rotate()`. In the code above, the motor plugged into port A is controlled using `motorA.rotate(2, 3, HIGH);`

The 2 refers to its **duration** (2 seconds). This can be given decimal values.

The 3 refers to its **power** level (range from 0 – 5), which governs how fast the motor will spin (0 is off and 5 is full power). This can also be given decimal values.

The `HIGH` sets its **direction** (`HIGH` is one direction and `LOW` is the other direction)

Press the verify button  to compile your code. Plug the USB cable into the Arduino, then press the upload button . When the IDE is finished uploading the program, unplug the USB cable from the Arduino and continue.

While the program is running the motor in port A should start spinning for two seconds and then the motor in port B will spin for one second.

2

Now we'll add some more commands, changing the last argument to alter the direction of the motors.

Update the code in your IDE to match the code in the box on the right.

```
#include <Artbotics.h>
Motor motorA(3, 12);
Motor motorB(11, 13);

void setup(){}

void loop(){
  motorA.rotate(2, 3, HIGH);
  motorB.rotate(1.5, 3, LOW);
  motorA.rotate(2, 3.5, HIGH);
  motorB.rotate(1.5, 3, LOW);
}
```

Press the verify button  to compile your code. Plug the USB cable into the Arduino, then press the upload button . When the IDE is finished uploading the program, unplug the USB cable from the Arduino and continue.

While the program is running the motors should alternate running in different directions.

3

You can also control individual motors to run at different power levels and directions at the same time. To do so, we can turn the motors on, then pause for a period of time, then turn them off.

To turn a motor on indefinitely do not pass an argument for duration. For instance, `motorA.rotate(3, HIGH);` `motorB.rotate(3, HIGH);` will turn on both motors indefinitely at the same time.

Then, we command our program to pause using `delay(3000);` which refers to delaying for 3000 milliseconds, or 3 seconds.

To stop both motors after the delay we use `motorA.stop();` and `motorB.stop();` another delay must be added to observe the motors stopping.

Update the code in your IDE to match the code in the box in the upper right.

```
#include <Artbotics.h>
Motor motorA(3, 12);
Motor motorB(11, 13);

void setup(){}

void loop(){
  motorA.rotate(3, HIGH);
  motorB.rotate(3, LOW);
  delay(3000);
  motorA.stop();
  motorB.stop();
  delay(3000);
}
```

Compile your code and then upload it to the Arduino. When you run the program it should turn on both motors at the same time, spinning in opposite directions, and then stop after 3 seconds.