

Arduino Robot Lab 01



Simple Movement and Sensor Data Acquisition
Isaac Daffron & Zachary Collins - Spring 2018

The Robot

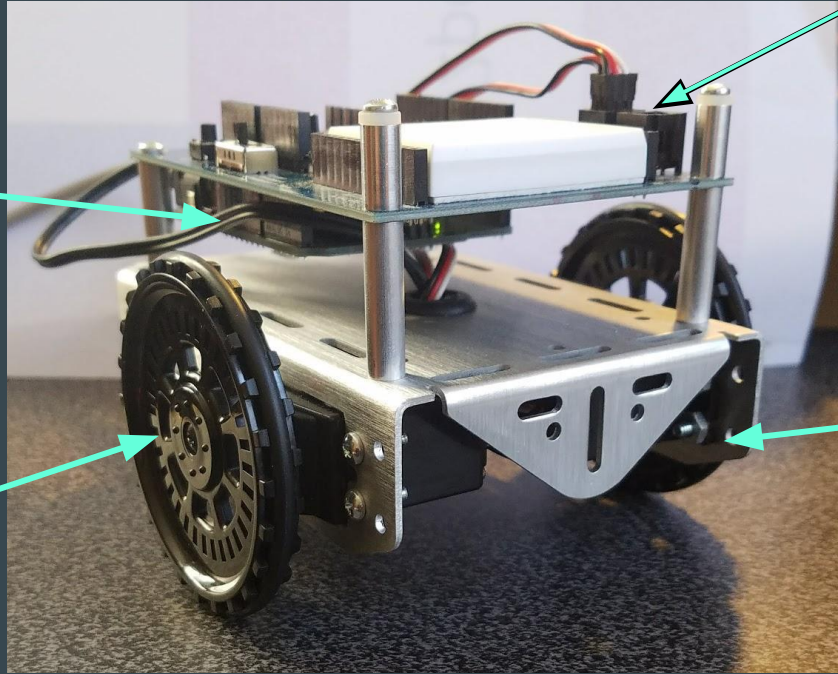
Parallax Shield with Arduino

Arduino Microcontroller

Servo with Wheel Attachment

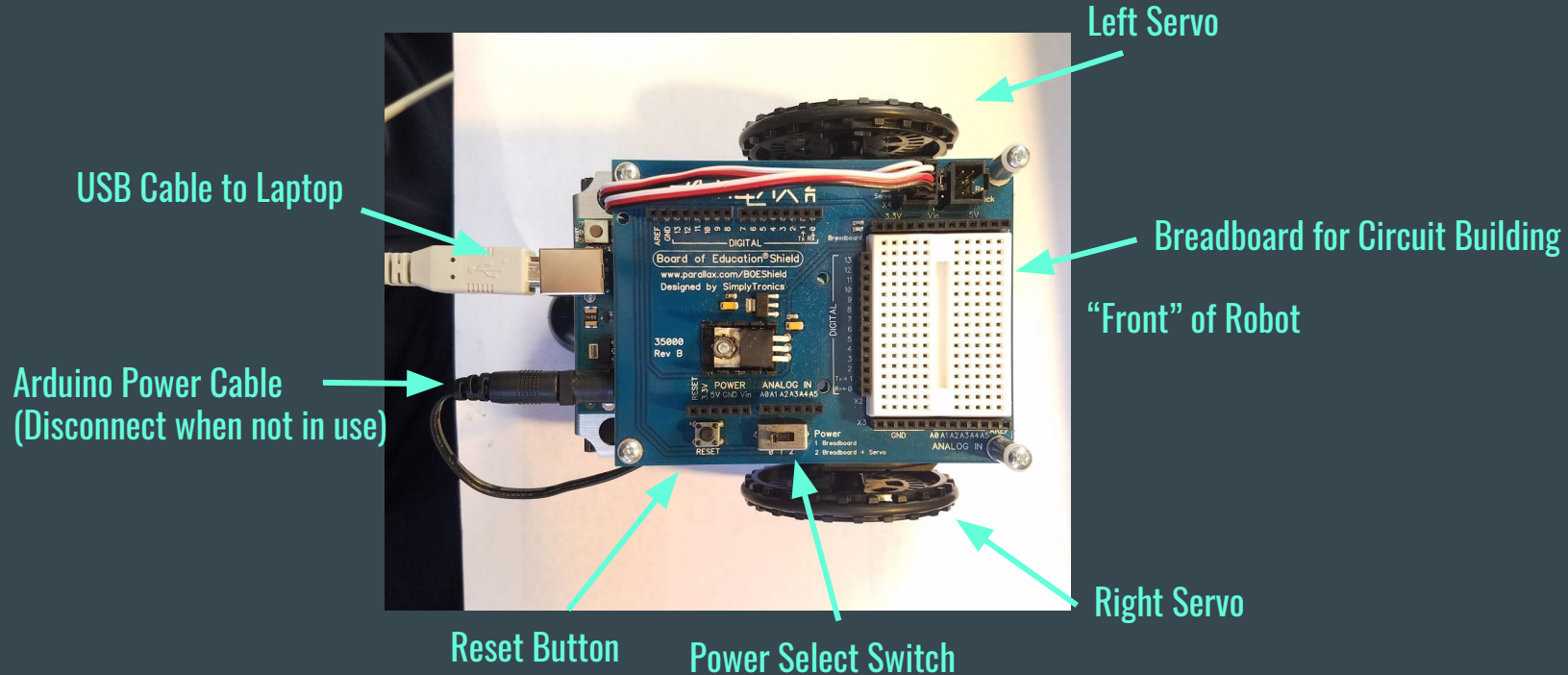
Parallax Board for Circuit Building

"Front" of the Robot



The Robot

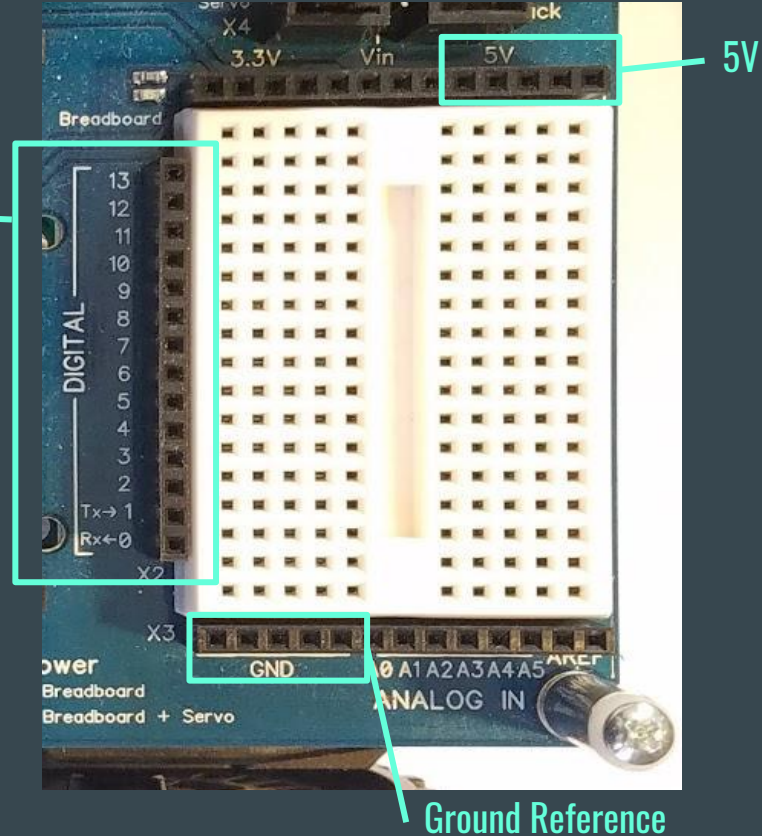
Hopefully you have a USB port on at least one of your computers!



Breadboard

This is where you will build your analog circuits.

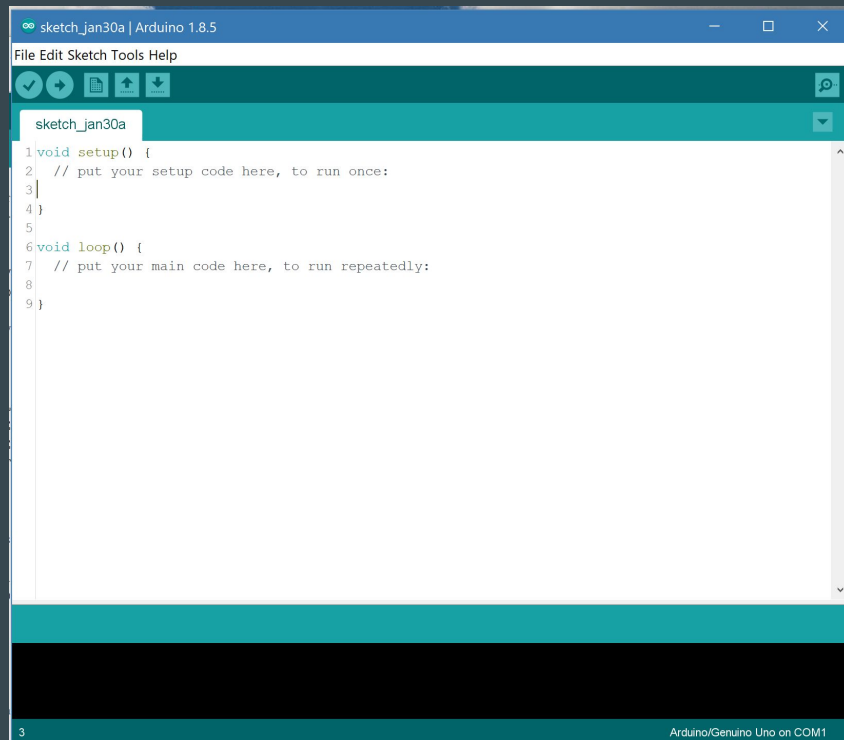
Digital Inputs/Outputs
You can read/output 0 or +5V on these pins, corresponding to a zero or 1.



Arduino IDE

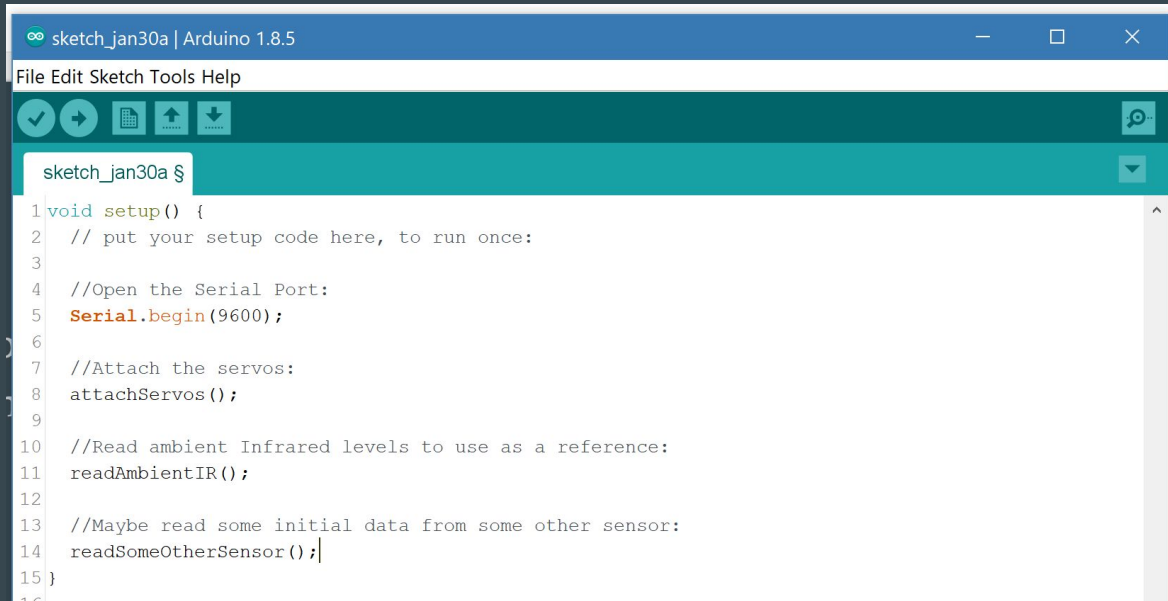
We will be programming using the Arduino's own IDE.

- Download at <https://www.arduino.cc/en/Main/Software> (aka just Google “Arduino IDE Download”)
- Write code in “Sketches”
- These Sketches get uploaded to the Arduino board on the Robot.
- Every Sketch Has a `setup()` function And a `loop()` function. There is no `main()`! (that you can see).



Arduino IDE - setup()

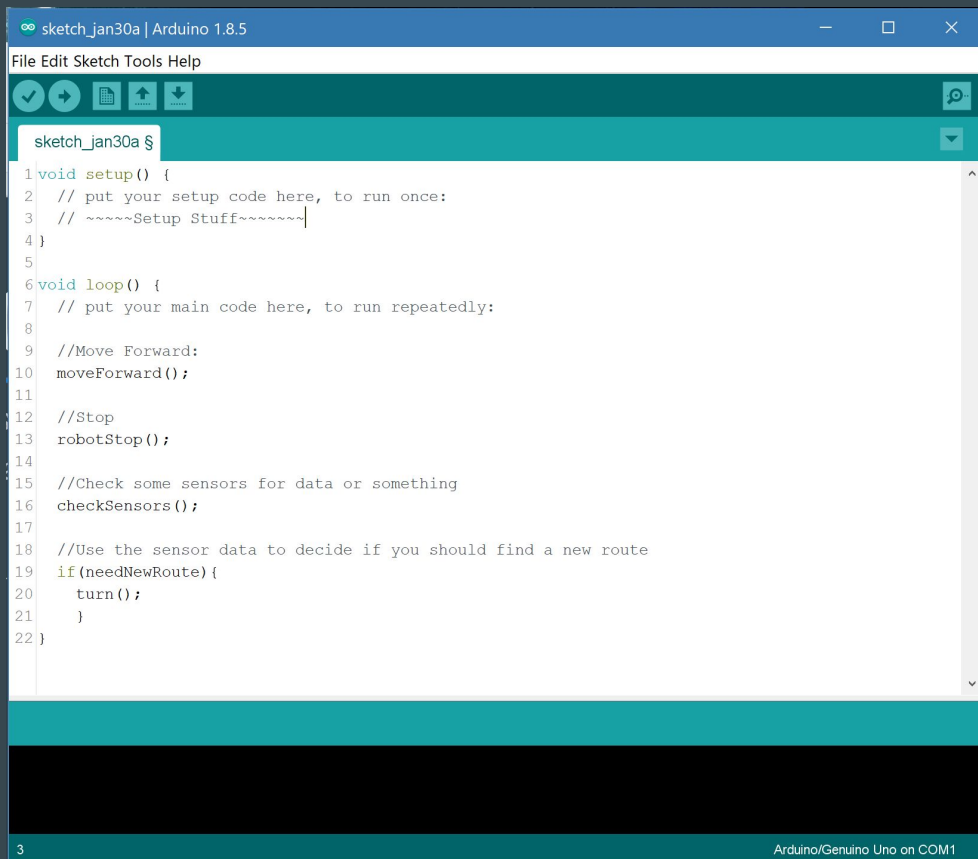
- First function to run when the robot is turned on or reset.
- Initialize everything your robot needs to start running.
- Example:

A screenshot of the Arduino IDE interface. The title bar shows 'sketch_jan30a | Arduino 1.8.5'. The menu bar includes 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. Below the menu bar is a toolbar with icons for opening, saving, and running. The main text area shows the following code:

```
1 void setup() {  
2   // put your setup code here, to run once:  
3  
4   //Open the Serial Port:  
5   Serial.begin(9600);  
6  
7   //Attach the servos:  
8   attachServos();  
9  
10  //Read ambient Infrared levels to use as a reference:  
11  readAmbientIR();  
12  
13  //Maybe read some initial data from some other sensor:  
14  readSomeOtherSensor();  
15 }
```

Arduino IDE - loop()

- loop() runs as soon as setup() is finished.
- It runs continuously, as long as the robot is on.
- Most of your code will be implemented in loop().
- Note that in setup() and loop() we are calling lots of functions! Modularize your code into functions!



```
sketch_jan30a | Arduino 1.8.5
File Edit Sketch Tools Help

sketch_jan30a $
1 void setup() {
2   // put your setup code here, to run once:
3   // ~~~~~Setup Stuff~~~~~
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9   //Move Forward:
10  moveForward();
11
12  //Stop
13  robotStop();
14
15  //Check some sensors for data or something
16  checkSensors();
17
18  //Use the sensor data to decide if you should find a new route
19  if (needNewRoute) {
20    turn();
21  }
22 }
```

3 Arduino/Genuino Uno on COM1

Part 1: Simple Robot Movement - Objective

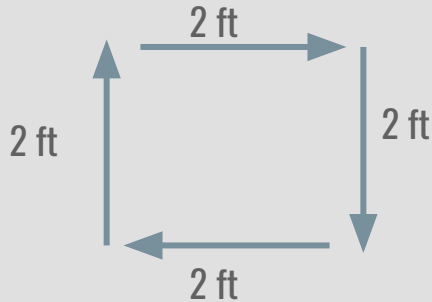
Objective: Demonstrate your knowledge of functions, the Arduino IDE, and servo control by making your robot move through a series of simple shapes.

Shape 1 : Line

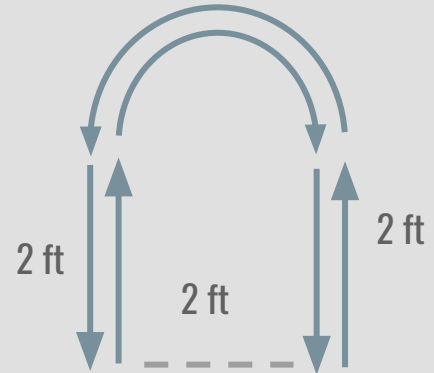


*Note: Robot should 'reverse' back to start

Shape 2 : Square



Shape 3: This Thing



Part 1: Servo Functions

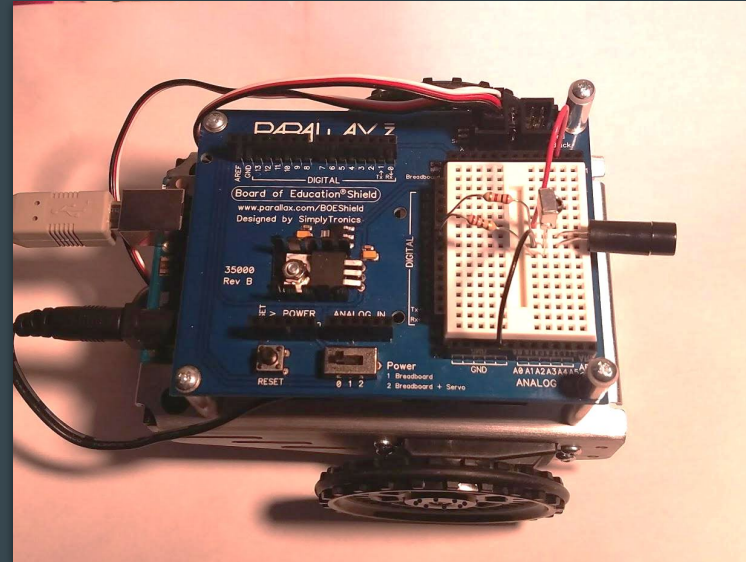
- `servoVarName.WriteMicroseconds(microseconds);`
 - Tells the servo to rotate. The argument *microseconds* indicates the speed of rotation. To go straight, for example, both left and right servo will have to go at the same speed, and turn the same direction.
 - Have to send *microseconds* = 1500, to stop rotation. It is the zero of the servo. The further the input value if from 1500, the faster the servo will turn. 1400 will make it turn faster than 1490.
 - When *microseconds* > 1500, the servo will turn the opposite direction than if *microseconds* < 1500.
 - Right and Left servos are mirrored! That means they will turn opposite directions if given the same input *microseconds*. You will spin in circles and go nowhere!
 - Basically, `WriteMicroseconds(1600)` is full speed one direction, and `WriteMicroseconds (1400)` is full speed the opposite direction.

Part 1: Suggested Plan

- Write functions for moving forward/backwards, stopping and turning
 - For big sweeping turns, the inside wheel moves slower than the outside wheel.
 - Have the wheels move in opposite directions to turn in place.
- Place the functions in the `loop()` in the correct order to go through the movements.

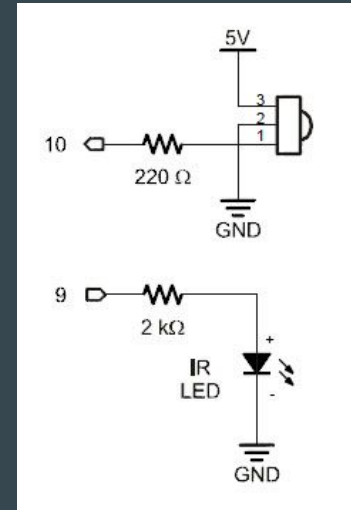
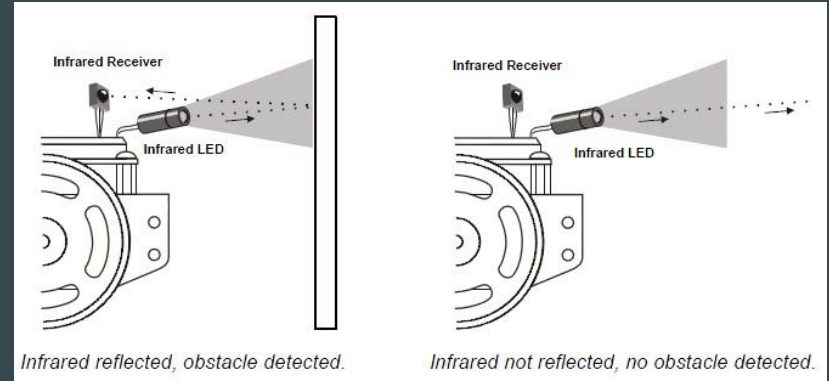
Part 2: Infrared Sensors

- We can improve our robot by building circuits on its breadboard
 - Send output to pins
 - Read information from pins
 - Attach sensors to get information from the outside world



Part 2: Infrared Sensor Circuit

- Circuit components
 - IR Receiver (1)
 - 220 Ohm Resistor (1)
 - 2k Ohm Resistor (1)
 - IR LED (1)
 - Wires (At least 2)
- First step: build the circuits shown here
 - Have an instructor or TA check your circuit before you go on!



Part 2: Infrared Sensor Code

- Reading from a pin
 - Declare pin as INPUT instead of OUTPUT
 - Use `digitalRead(int pin)` to read from a pin
- First, run sample code:
 - Use given `irDetect` function to print the receiver pin's input
 - What happens when you put your hand close to the sensor?

Part 2: Infrared Sensor Project

Objective: Demonstrate your knowledge of basic circuits and programming control structures by making your robot avoid walls.

- Combine IR sensor code with Shape 1
- Make a collision detection system
 - Make the robot drive continuously in a straight line
 - Stop before you crash.
- Too easy? Try making the robot do something (turn, reverse, etc.) after it stops

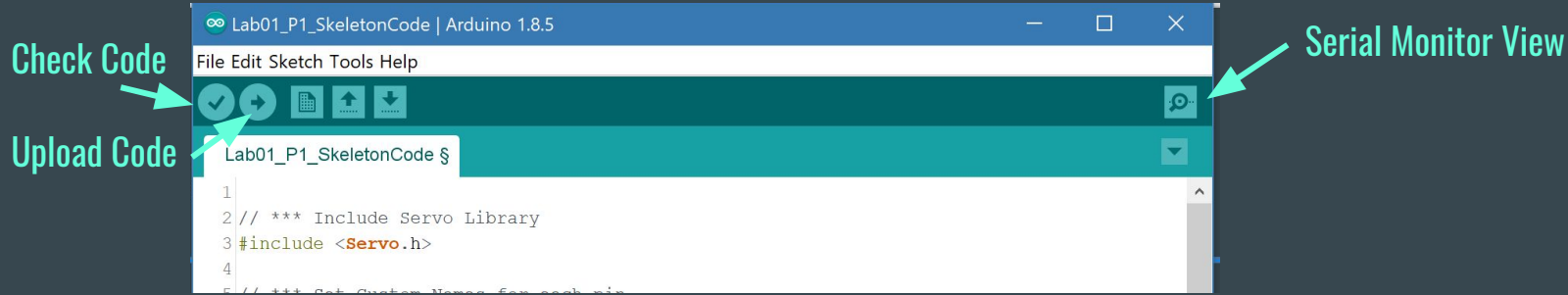
Shape 1 : Line



*Note: Don't reverse this time! Keep going until you find a wall.

Other Useful Tips

Navigating IDE:



Writing to serial Monitor:

- `Serial.println("item_to_print");`
 - Prints the argument on a new line each time.

Skeleton Code and Grading

Code: <https://goo.gl/Bcdb3a>

Make sure you get each section checked off by a TA or instructor!

- Showing up: 10%
- Part 1: 60%
 - Shape 1: 20%
 - Shape 2: 20%
 - Shape 3: 20%
- Part 2: 30%