BLUESat Arduino Joystick Activity

This activity will give you an introduction to some of the things we do at BLUESat and give you a chance to write code that solves a problem similar to the ones we solve on the Rover. In this activity, you’ll experiment with a direction controller, which will encode both the direction you want the Rover to move in and the speed at which you want it to move at.

# Materials

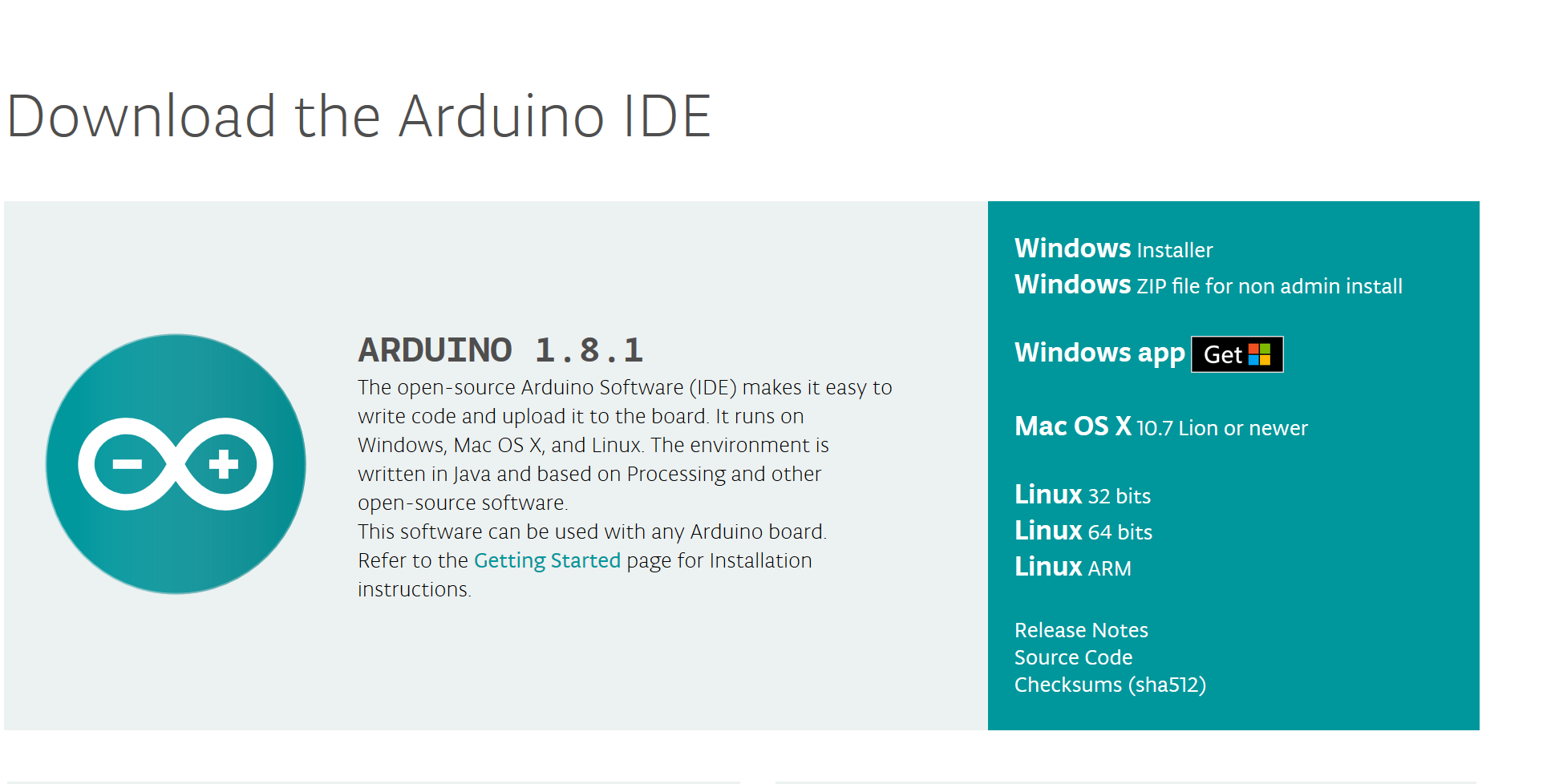
You’ll need the following

* 1 Arduino with associated LEDs set up on a breadboard (this should be provided to you).
* The USB Cable that fits your Arduino
* 1 Joystick

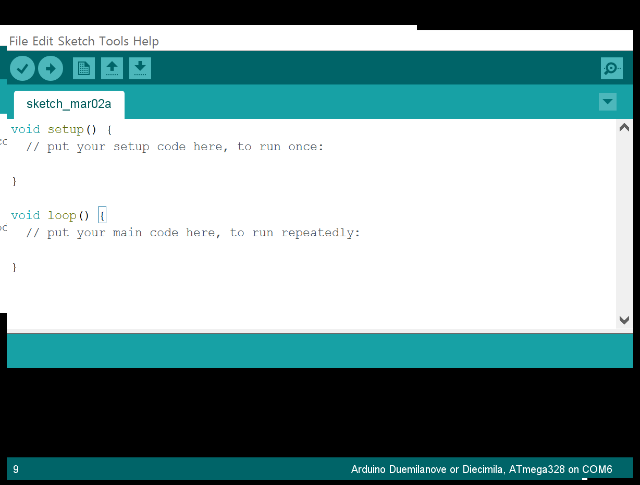
# Install the Arduino IDE and Get the Code

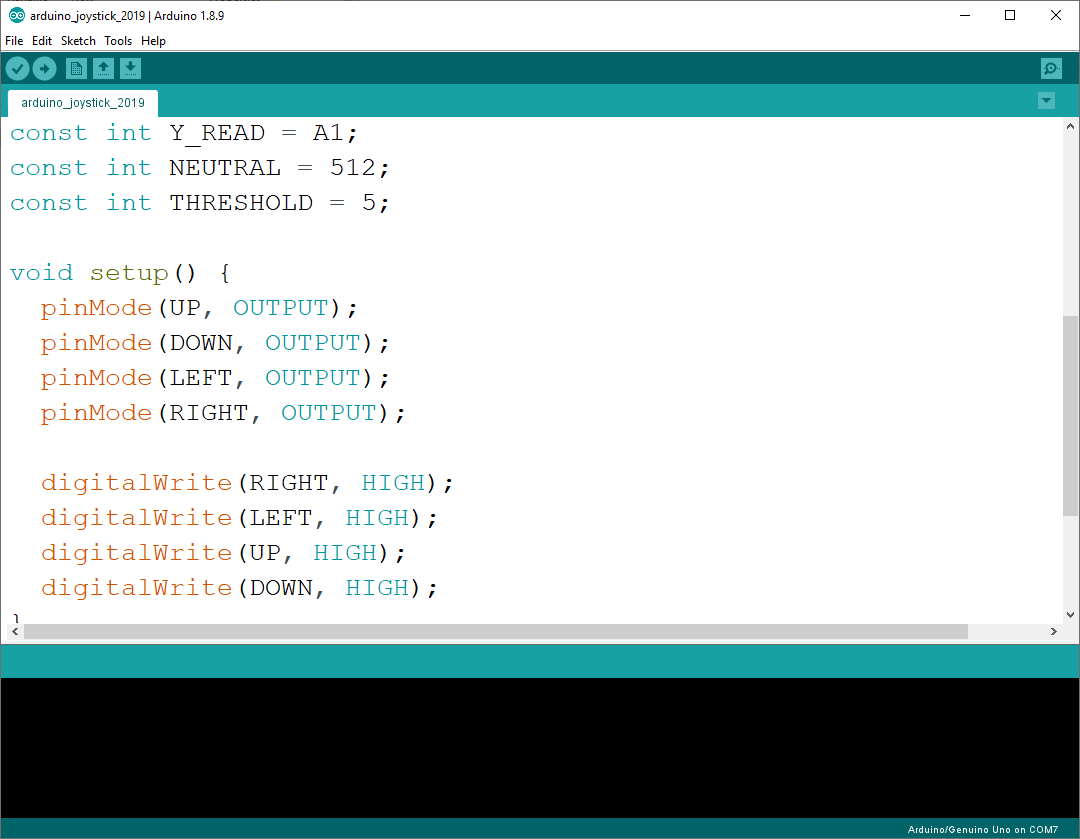
The code you are going to be writing will be in the Arduino environment. You need to set that up, and get the code you’ll be working with.

1. Go to <https://www.arduino.cc/en/Main/Software> and scroll down to the “Download the Arduino IDE” section and click on the download link for your platform. You should choose one of the options highlighted in red.



1. Once the download is complete, run the installer and then open the Arduino IDE. You should see a window like this:

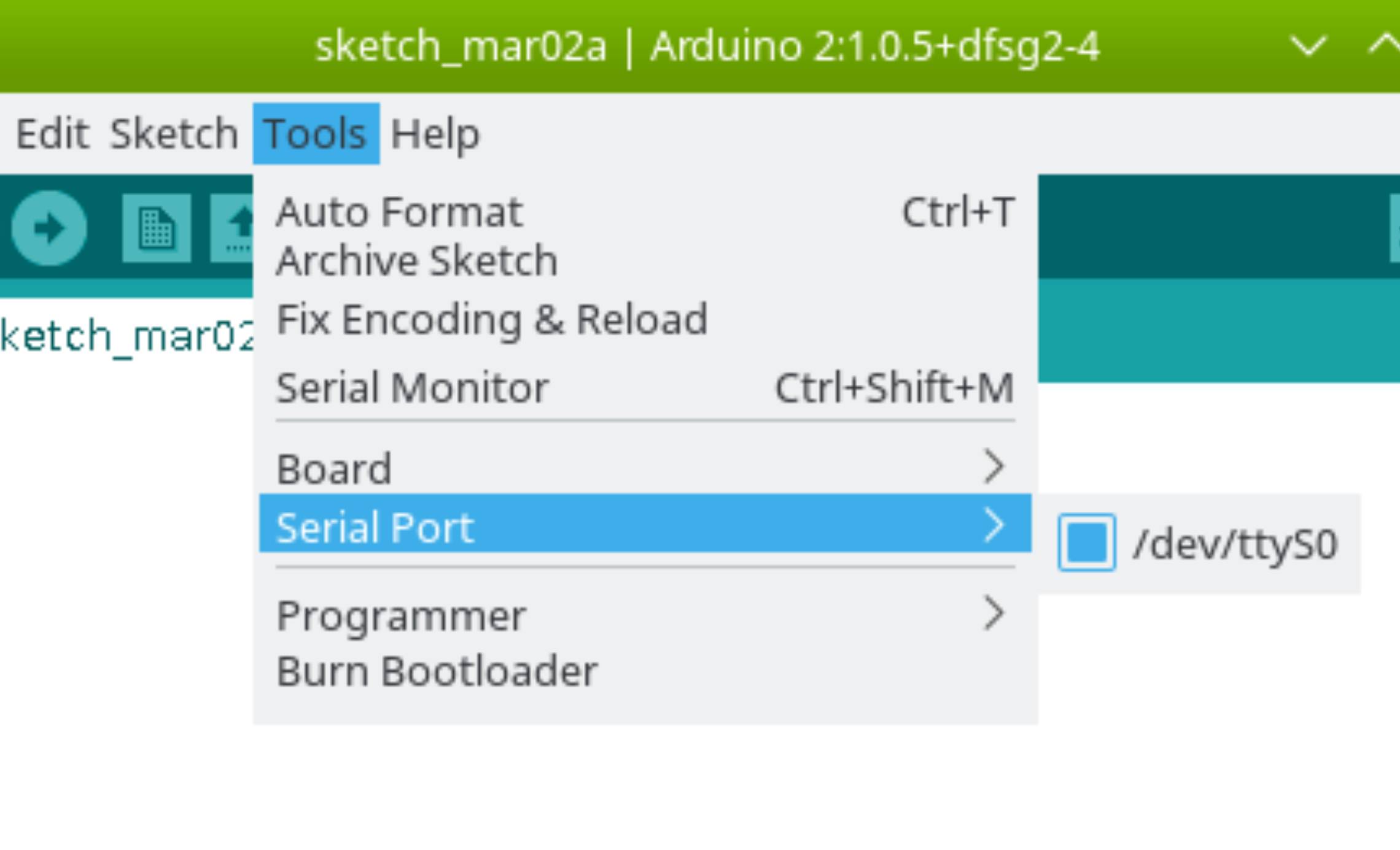
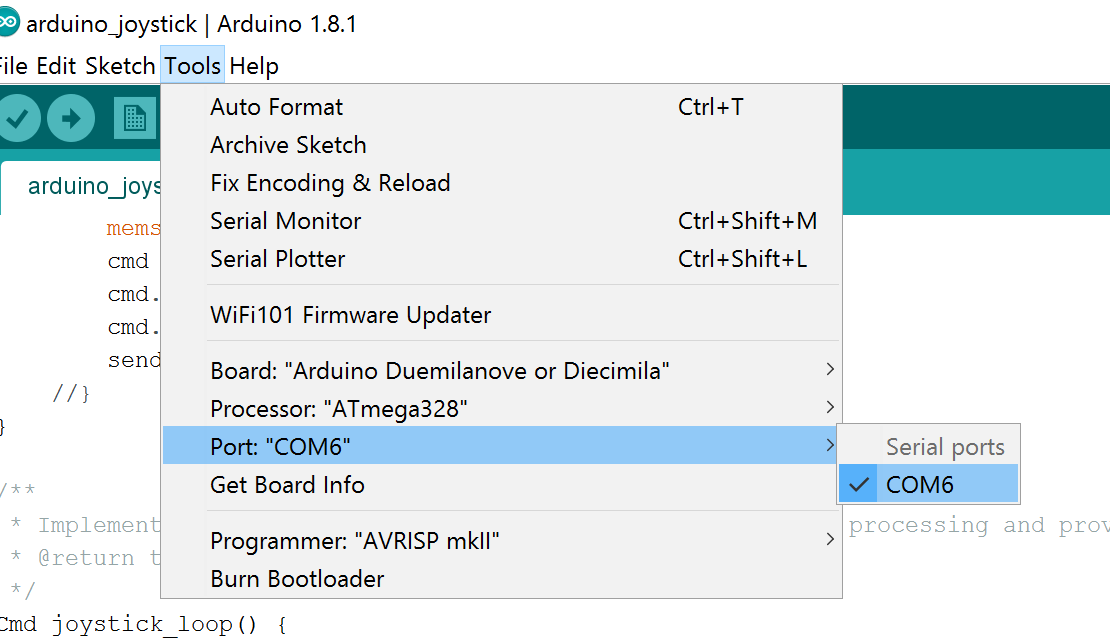


1. This is where you’ll write your code. Setup runs once when the Arduino turns on, and loop runs repeatedly. We will provide you with the code which runs in setup. Loop is the code that runs first after setup has been finished and continues to repeat forever afterwards. This is where you will be placing your code.
2. If you are using a WAVGAT, go to this [page](https://bluesat.atlassian.net/wiki/spaces/GUIL/pages/153845992/Electrical+Prototyping+Tutorial) and find the WAVGAT section.
3. Download the template code from <https://raw.githubusercontent.com/bluesat/arduino_joystick/master/2019/arduino_joystick_2019.ino>.
4. Copy the contents of the template into a new file. Save it somewhere convenient, giving it a name like “arduino\_joystick”. You should be looking at a window which now looks something like this.  
   
5. Don’t touch anything that’s outside of the loop function. The constants defined at the top of the file are there to help you out.

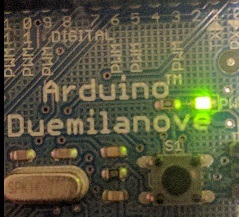
# Running you First Test

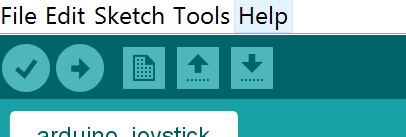
Now you have everything setup, its time to run your first test.

1. Plug the Arduino into your laptop via the USB Cable.
2. Open the “arduino\_joystick.ino” file from the last step in the Arduino IDE, if you don’t have it open already.
3. On Windows menu select “Tools > Port” from the top menu and click on the second item in the menu. It will say “COM6” or similar. On Mac and Linux, select “Tools > Serial Port” and then what looks like the most appropriate value (for Linux this will be similar to “/dev/ttyUSB0 or /dev/ttyACM0” and on Mac it will probably have the word “serial” in it. Ask one of the instructors if you are unsure).



*(Left: Windows, Right: Mac and Linux)*

1. Next select Tools > Board, and choose the appropriate setting for the board you have. Your board will either be “Arduino/Genuine Uno” or “Arduino Duemilanove or Diecimila.” It should say which on the board, if it doesn’t chose “Uno” or ask.
2. Click the Upload button, which is an arrow on a green background to program your Arduino.



If you want to compile (but not upload) the code, click the tick to the left of the arrow. This is useful for checking the syntax of your program.

# Using a Joystick

Here’s a quick description of how your LEDs should behave.

* In the neutral position, the LED should not flash at all.
* When you tilt the joystick in a cardinal direction (up, down, left or right), only one LED should blink. The frequency of blinking should be related to the degree to which you have turned the joystick in that direction. Whether the LED blinks faster or slower the more you move the joystick in that direction is up to you.
* When you tilt the joystick in a diagonal direction, two LEDs should flash up. The frequency of the blinking should be the same behaviour as described above.

# Extension Activity

The joystick has a button press pin on it (SW). You could try to wire up the Arduino to monitor this pin and blink all four LEDs at once when the button is pressed. Ask the demonstrators before attempting this activity.