

Seeeduino v4.2



Seeeduino V4.2 is an Arduino-compatible board, which is based on ATmega328P MCU. There're so many Arduinos and Arduino-compatible boards in the world, and we think that Seeeduino V4.2 is one of the best Arduinos, Seeeduino V4.2 is much more stable, easy-to-use and even good looking.

Seeeduino V4.2 is based the Arduino UNO bootloader, and with an ATMEGA16U2 as an Uart-to-USB converter, which means that the board can basically work like a FTDI chip. You can program the board via a micro-usb cable, if you have an Android phone, you can find a micro-usb cable easily. Also you can power the board via a DC Jack input, 7 to 15V is acceptable. There's a switch to choose the system supply voltage, 3.3V or 5V, which is very useful if you want to set the system to 3.3V to save power.

Finally, the three on-board Grove interface can make your board connect to Groves easily. Want to make something awesome, maybe just a Seeeduino V4.2 and some Groves is enough.

Features

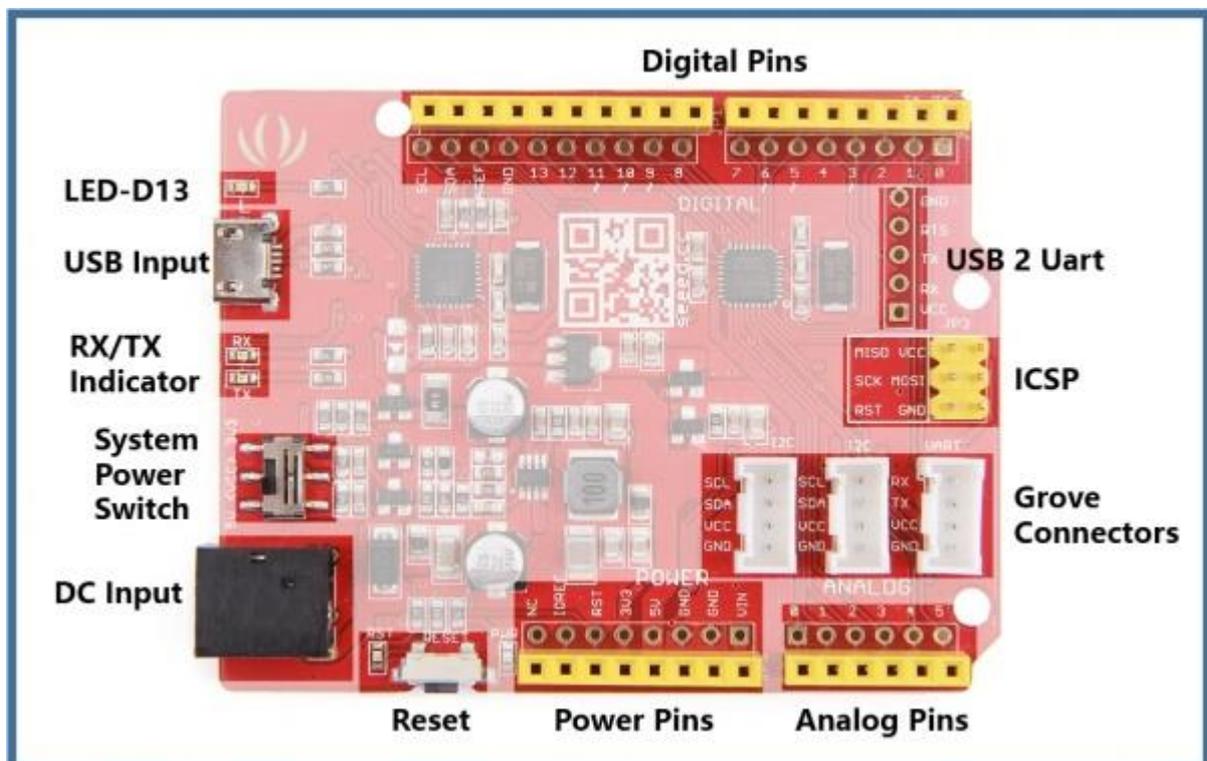
- ATmega328 microcontroller
- Arduino UNO bootloader
- 14 Digital I/O Pins (6 PWM outputs)
- 6 Analog Inputs
- ISP Header
- Arduino UNO-R3 Shield Compatible
- Micro USB programming and power supply
- 3 on-board Grove connectors
- 3.3/5V system operation power switch

Specification

- DC Jack Input: 7v-12v
- DC Out Put Current:
 - 5V PIN: 500mA MAX with Micro USB
 - 5V PIN: 2000mA MAX with DC Jack power supply
 - 3V3 PIN: 500mA MAX
- DC Current per I/O Pin: 40 mA
- Flash Memory: 32 KB
- RAM: 2 KB
- EEPROM: 1 KB
- Clock Speed:16 MHz
- Length: 68.6mm
- Width: 53.4mm
- Weight: 26g

Hardware Overview

Below iamge is the hardware overview of the Seeeduino V4.2.



- LED-D13

A led connect to **D13**, can be used as an indicator

- **USB Input**

Port used to connect the board to your PC for programming. Micro USB is the ubiquitous version of USB, found in most Android phones, and other devices. You probably have dozens of these cables laying around your house.

- **RX/TX Indicator**

The TX and RX LEDs work automatically, they let you know when the board is sending or receiving information respectively.

- **System Power Switch**

Switch used to change the logic level and power output of the board to either **5V** or **3.3V**. Nowadays many new and great sensors are being develop to work with 3.3V, with other duino boards you would need to place a logic level converter between the board and these sensor(s), with the Seeeduino V4.2 board all you have to do is slide the switch!

- **DC Input**

The DC power jack allows your Seeeduino board to be powered from a wall adapter so that you can supply more power to your project if needed, for example when using DC motors or other high power devices. The DC input can be **7V-15V**.

- **Reset**

This button is conveniently placed on the side to allow you to reset the Seeeduino board even when a shield placed on top. This isn't the case in other duino boards where the button is placed on top making it hard to access.

- **Power Pins & Analog Pins**

Just like the Extra Header Digital Pads, these extra connections are something we've personally come to realize people need in their projects, specially the power connections if you want to power more than one sensor/device without the use of a breadboard.

- **Grove Connectors**

[SeeedStudio](#) has a variety of sensors/devices that can make use of this I2C or UART connection. In addition we sell independent Grove connectors to help you make our own sensor connections. The I2C Grove connector is also connected to analog pin A4 and A5 for SDA and SCL respectively if you would like to use those pins instead. The UART Grove connector is connected to digital pins 0 and 1 for RX and TX respectively.

- **U1 ICSP**

This is the ICSP connection for the ATmega328P-MU, it is located in the standard ICSP/SPI position for Arduino Uno, Due, Mega, and Leonardo compatible hardware (e.g. shields) that may use this connector. The SPI pins in this port: MISO, SCK, and MOSI, are also connected to digital pins 12, 13, and 11 respectively just like those of the Arduino Uno.

- **USB 2 Uart**

Pinout of USB-2-Uart

- **Extra Header Pads (Digital)**

Sometimes it is very convenient to connect a sensor/device to your board directly instead of going through a breadboard, or perhaps you want to solder the sensor directly to the board once you've completed your project, or maybe you want to monitor the output of the pins while they're been used by other devices. In any case we have added these extra pads to help you along the way.

Getting Started on Windows

This part is base on the [Getting Started with Arduino on Windows](#), click to view the original page.

Get a Micro-USB cable

You need a Micro-USB cable first, the same as the data cable of an Android Phone.



If you can't find one, you can buy one [here](#).

Download the Arduino Software (IDE)

Get the latest version from the [download page](#). When the download finishes, unzip the downloaded file.

Connect the board

The Seeeduino V4.2 automatically draw power from either the USB connection to the computer or an external power supply.

Connect the Arduino board to your computer using the USB cable. The green power LED (labelled **PWR**) should go on.

Install the drivers

*Installing drivers for the Seeeduino V4.2 with **Windows 7, Vista, or XP***

- Plug in your board and wait for Windows to begin it's driver installation process. After a few moments, the process will fail, despite its best efforts
- Click on the Start Menu, and open up the Control Panel.

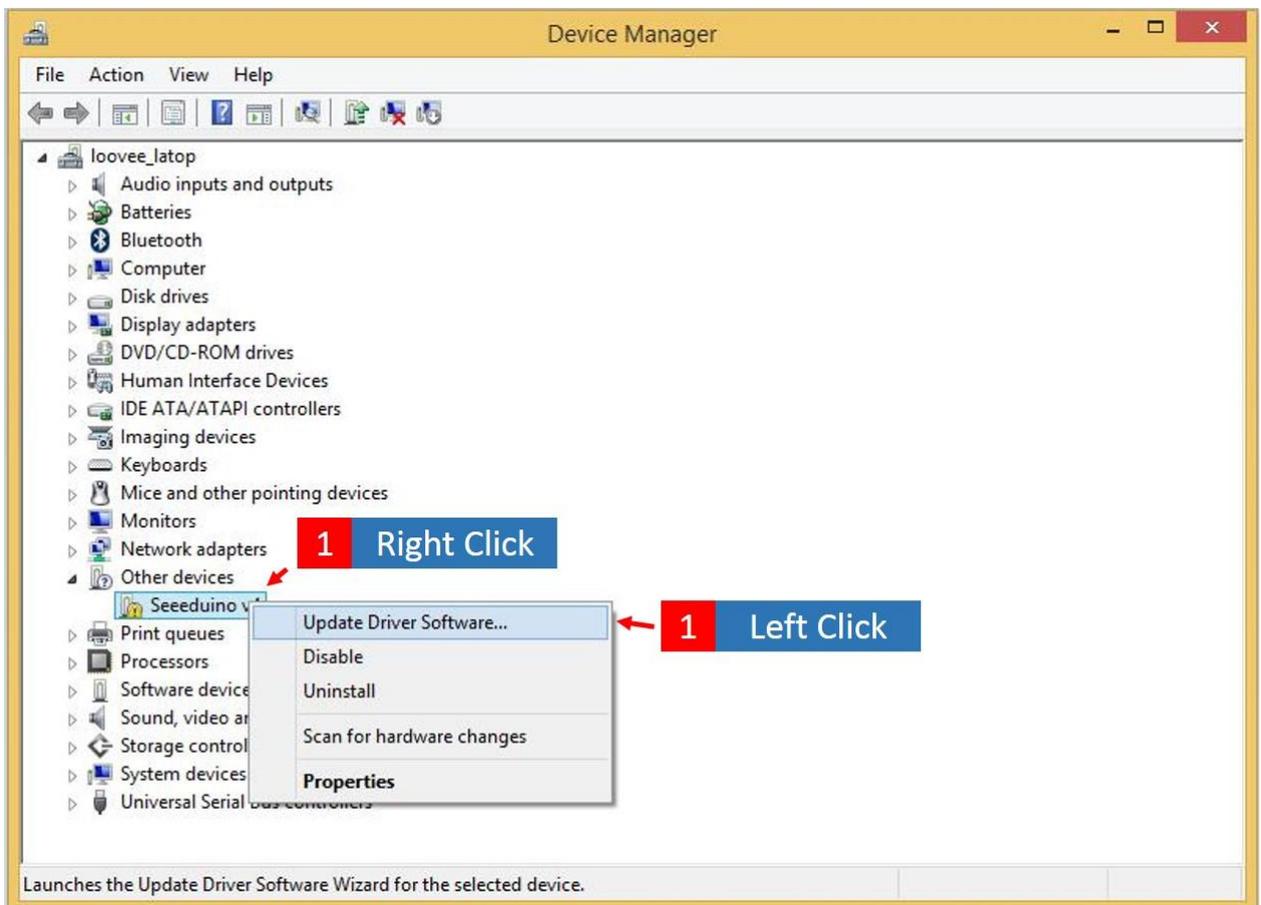
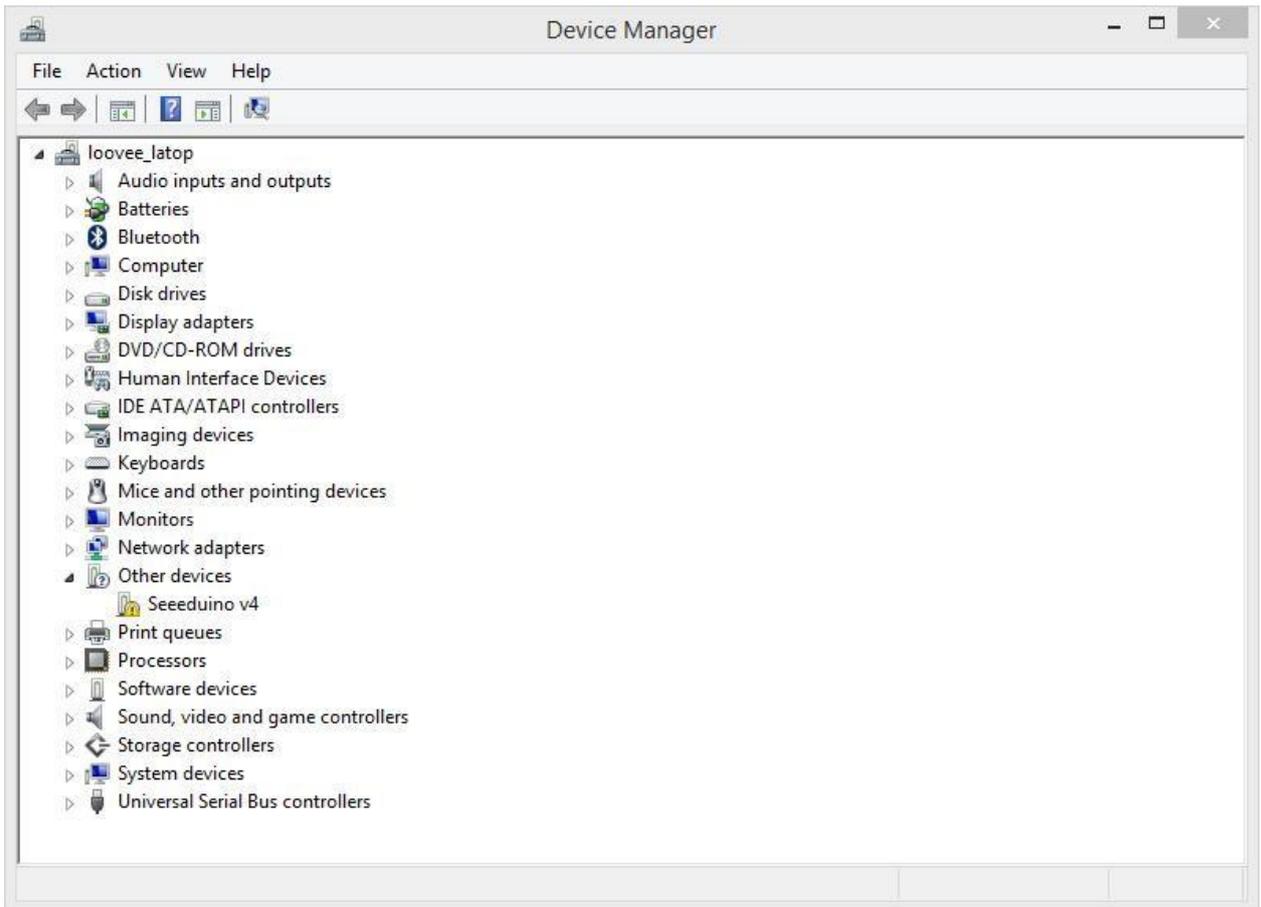
- While in the Control Panel, navigate to System and Security. Next, click on System. Once the System window is up, open the Device Manager.
- Look under Ports (COM & LPT). You should see an open port named "Arduino UNO (COMxx)". If there is no COM & LPT section, look under "Other Devices" for "Unknown Device".
- Right click on the "Arduino UNO (COMxx)" port and choose the "Update Driver Software" option.
- Next, choose the "Browse my computer for Driver software" option.
- Finally, navigate to and select the driver file named "**arduino.inf**", located in the "Drivers" folder of the Arduino Software download (not the "FTDI USB Drivers" sub-directory). If you are using an old version of the IDE (1.0.3 or older), choose the Uno driver file named "**Arduino UNO.inf**".
- Windows will finish up the driver installation from there.

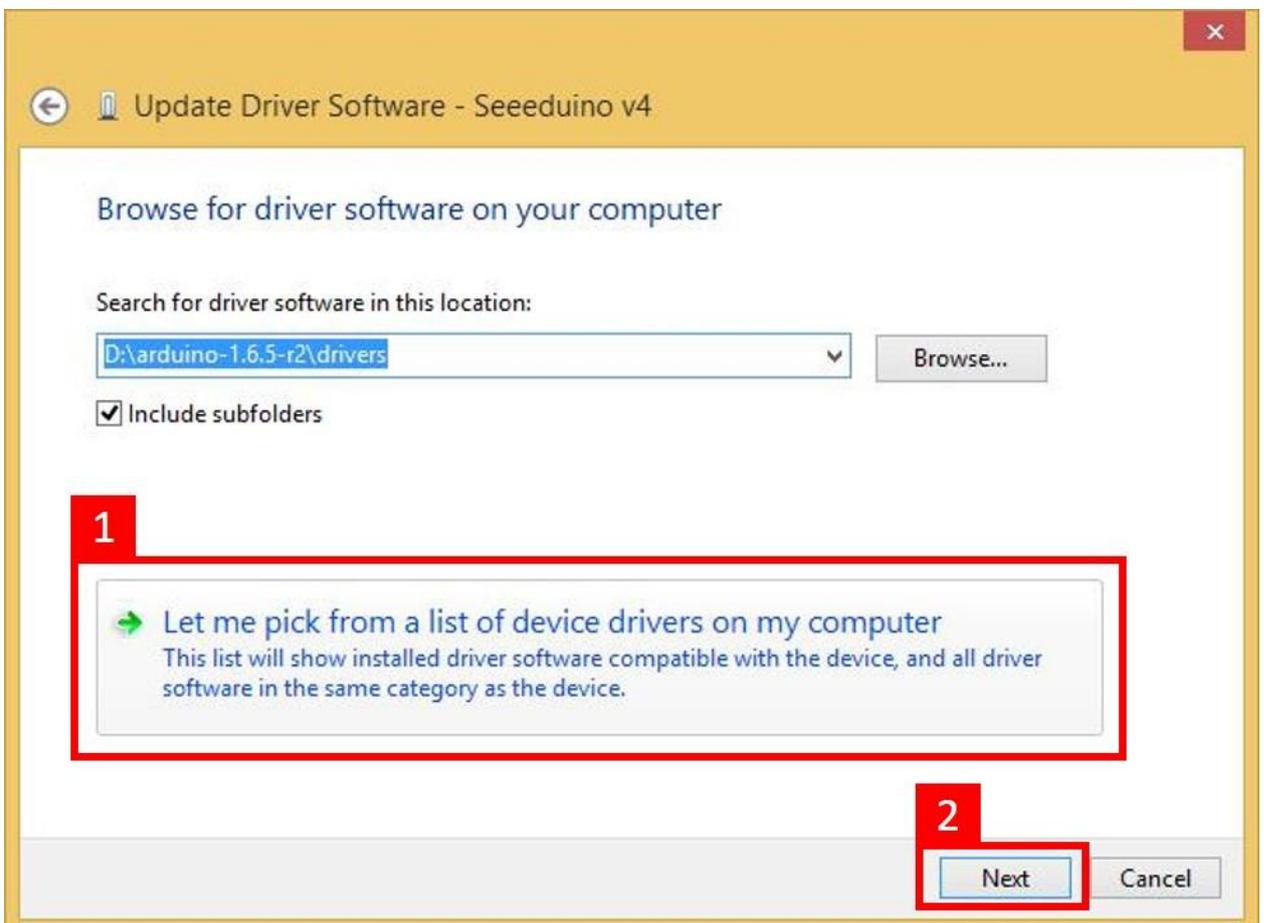
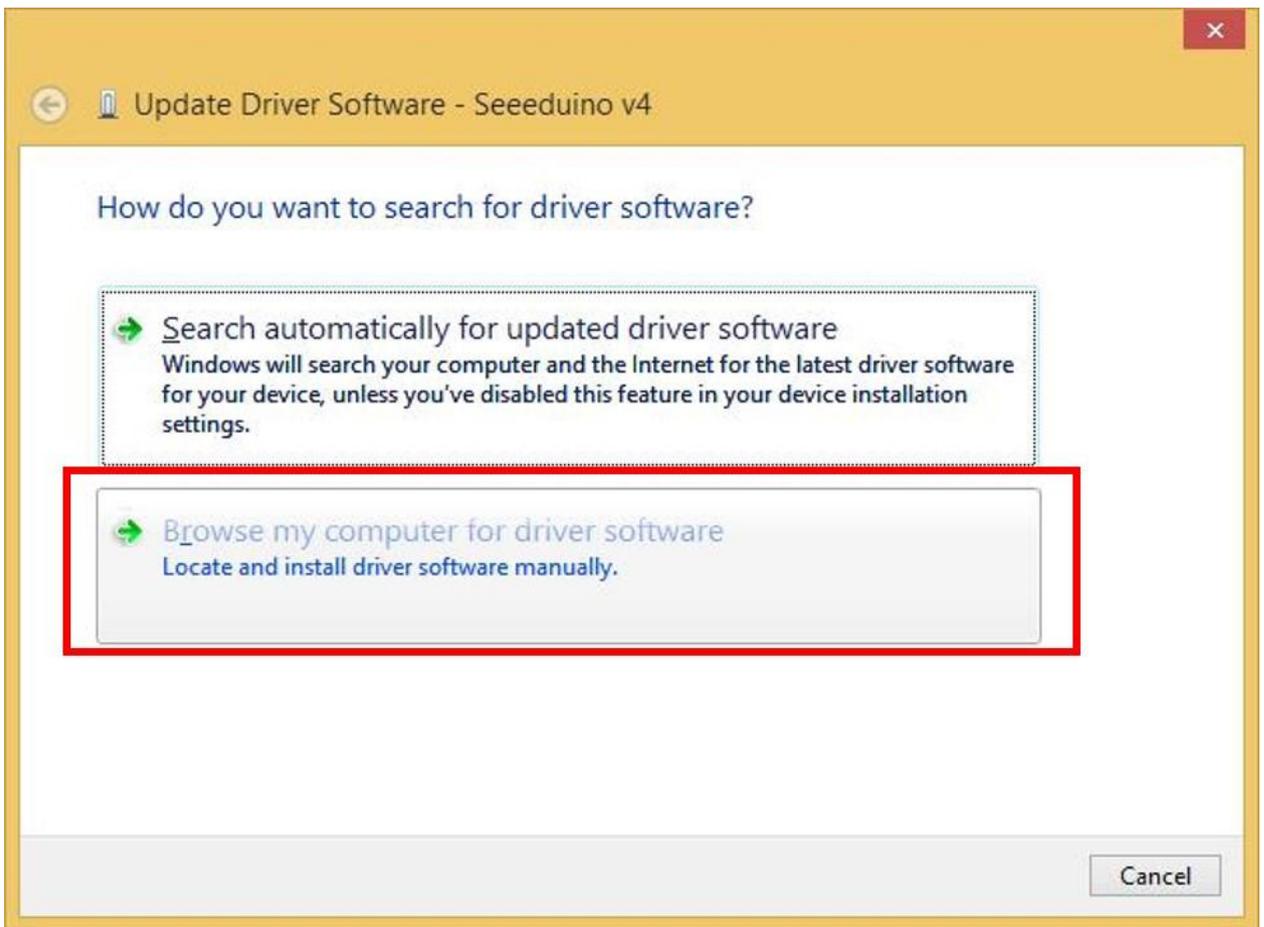
See also: [step-by-step screenshots for installing the Uno under Windows XP](#).

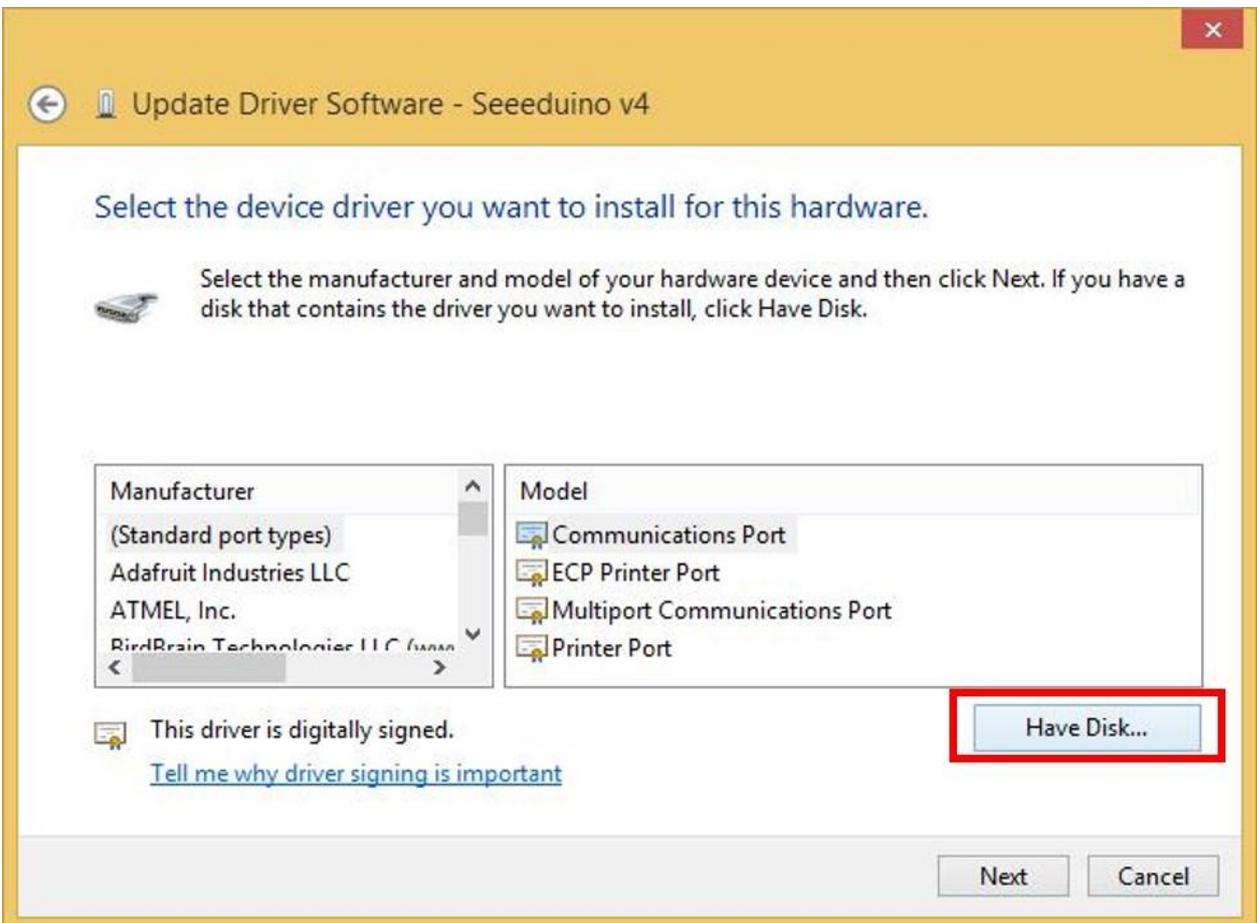
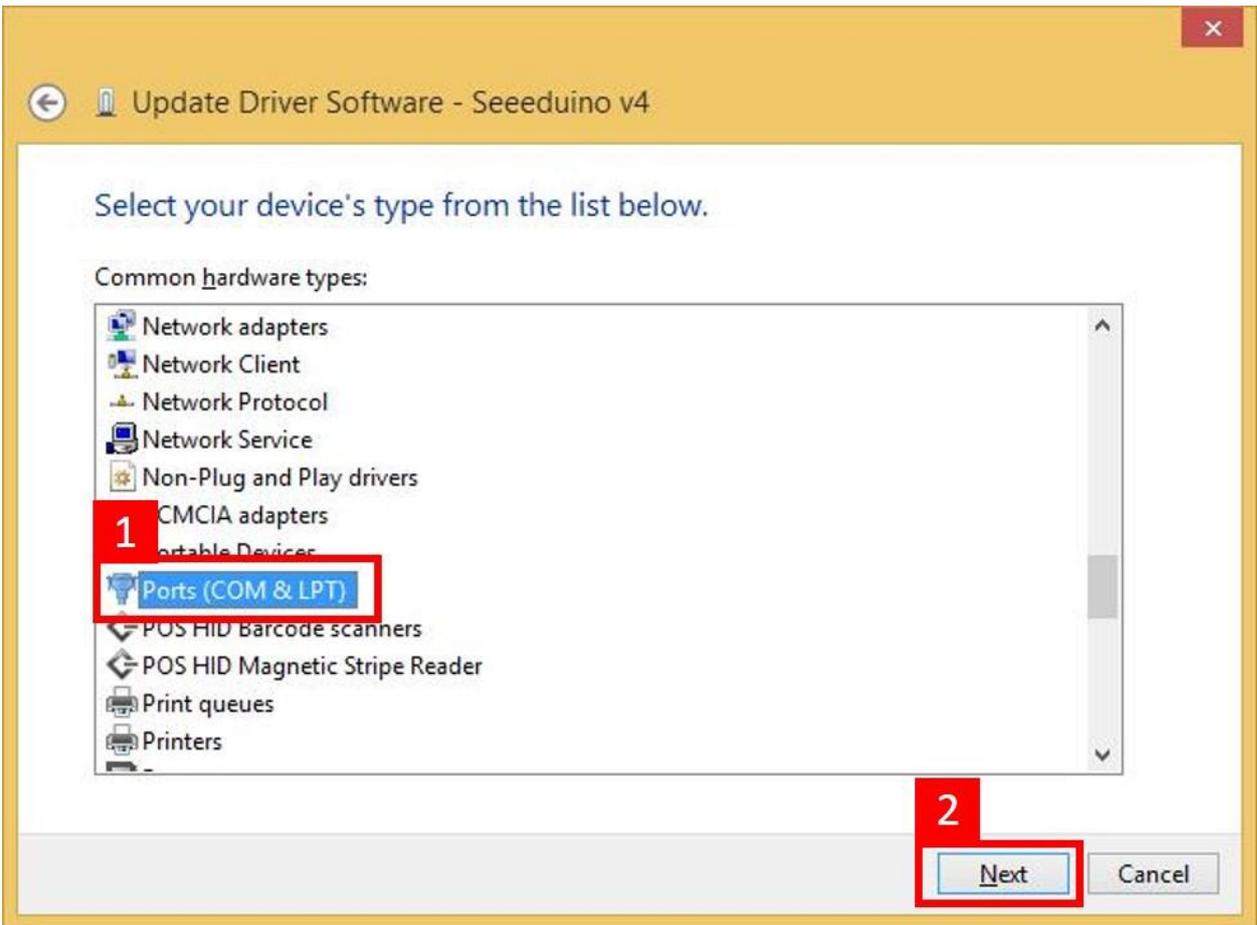
*Installing drivers for the Seeeduino V4.2 with **Windows 8/8.1***

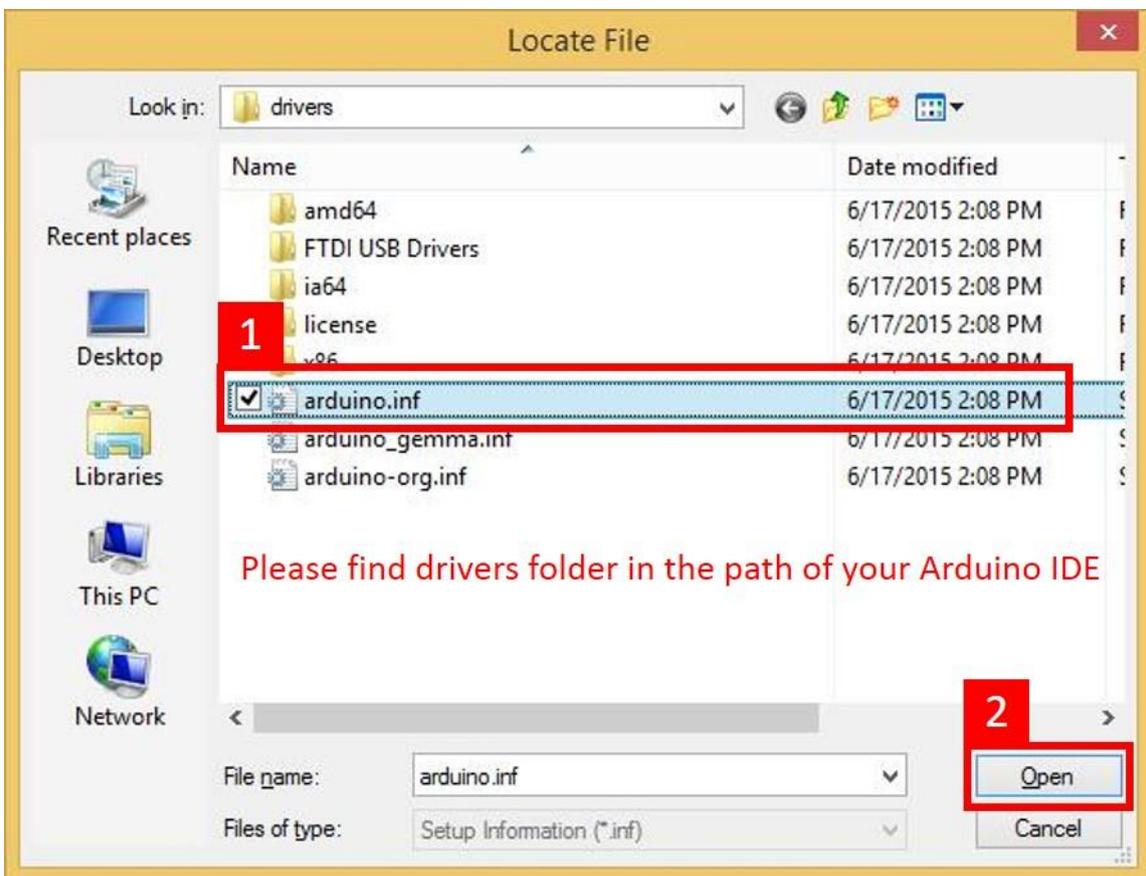
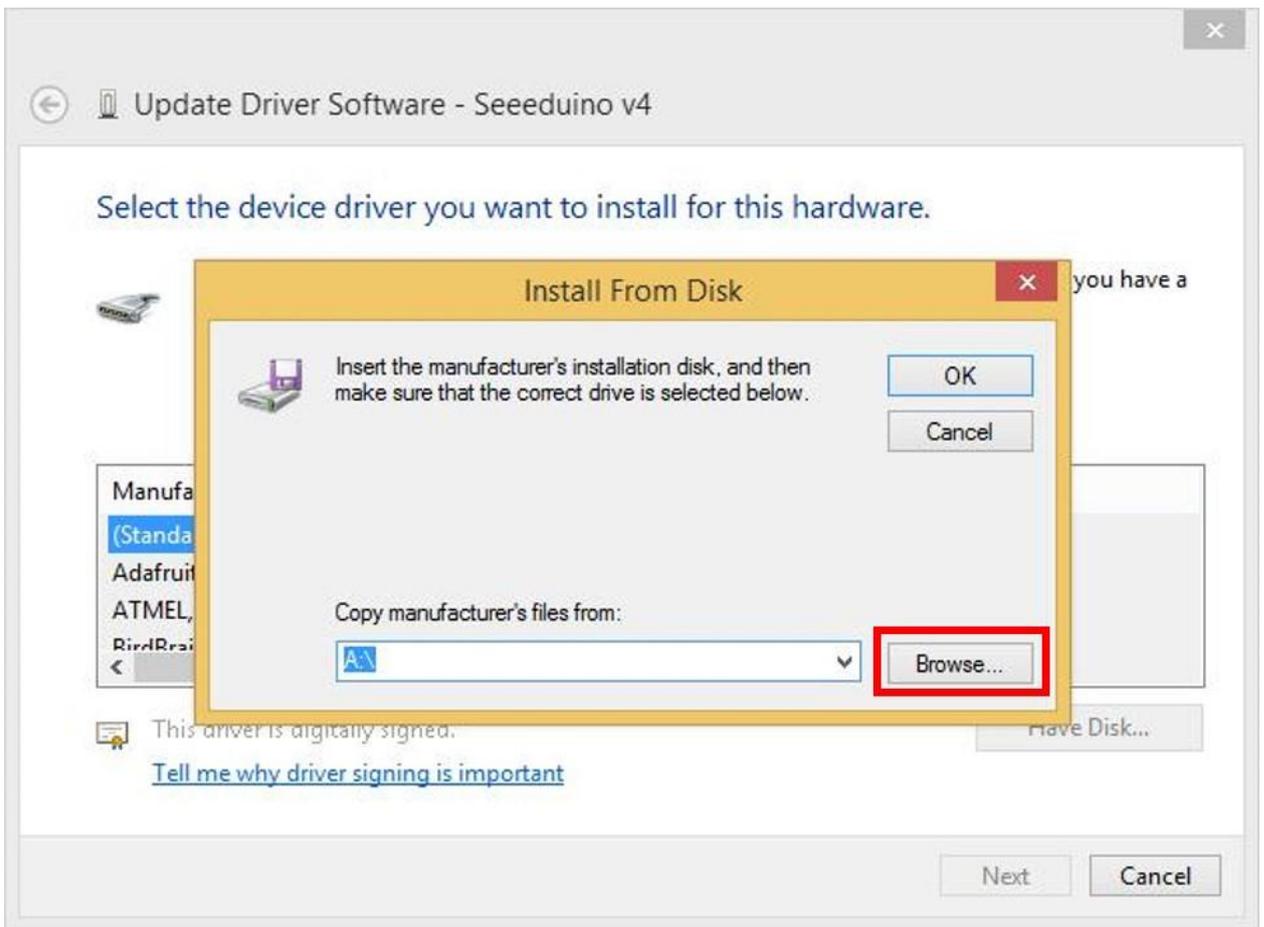
- Plug in your board and wait for Windows to begin its driver installation process. After a few moments, the process will fail, despite its best efforts
- Click on the Start Menu, and open up the Control Panel.
- While in the Control Panel, navigate to System and Security. Next, click on System. Once the System window is up, open the Device Manager.
- Look under Ports (COM & LPT). You should see an open port named "Arduino UNO (COMxx)". If there is no COM & LPT section, look under "Other Devices" for "Unknown Device".
- Right click on the "Arduino UNO (COMxx)" port and choose the "Update Driver Software" option.
- Next, choose the "Browse my computer for Driver software" option.
- Choose Let me pick from a list of device drivers on my computer
- Choose Ports(COM&LPT) and click Next
- Click on Have Disk
- Click on Browse..
- Then, navigate to and select the driver file named "**arduino.inf**", located in the "Drivers" folder of the Arduino Software download (not the "FTDI USB Drivers" sub-directory). If you are using an old version of the IDE (1.0.3 or older), choose the Uno driver file named "**Arduino UNO.inf**".
- Click on OK
- Manufacturer choose Arduino LLC(www.arduino.cc), Model choose Arduino UNO and then click on Next
- Then show up a Update Driver Warning, click on Yes.
- Windows will finish up the driver installation from there.

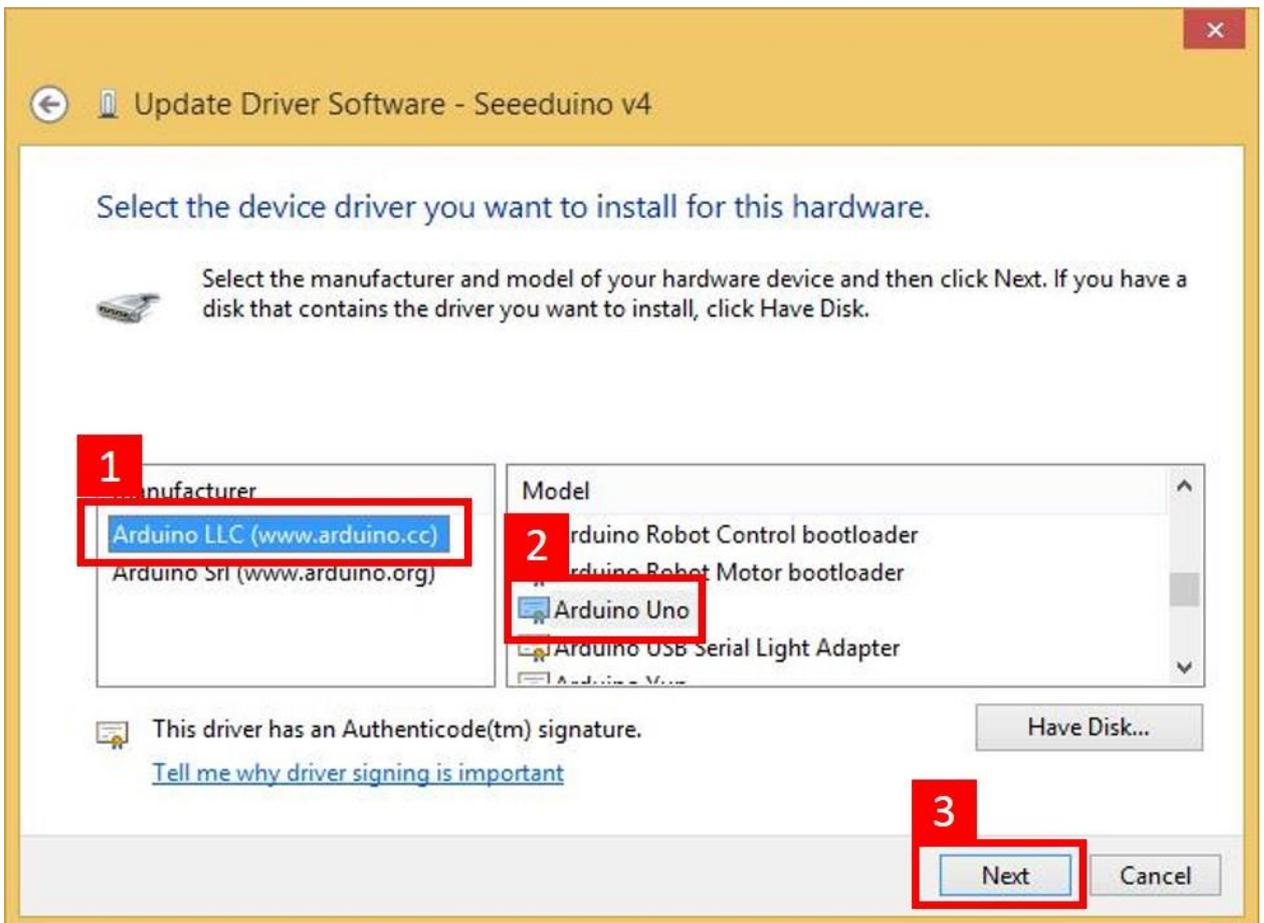
Below is the details, click to view larger image.

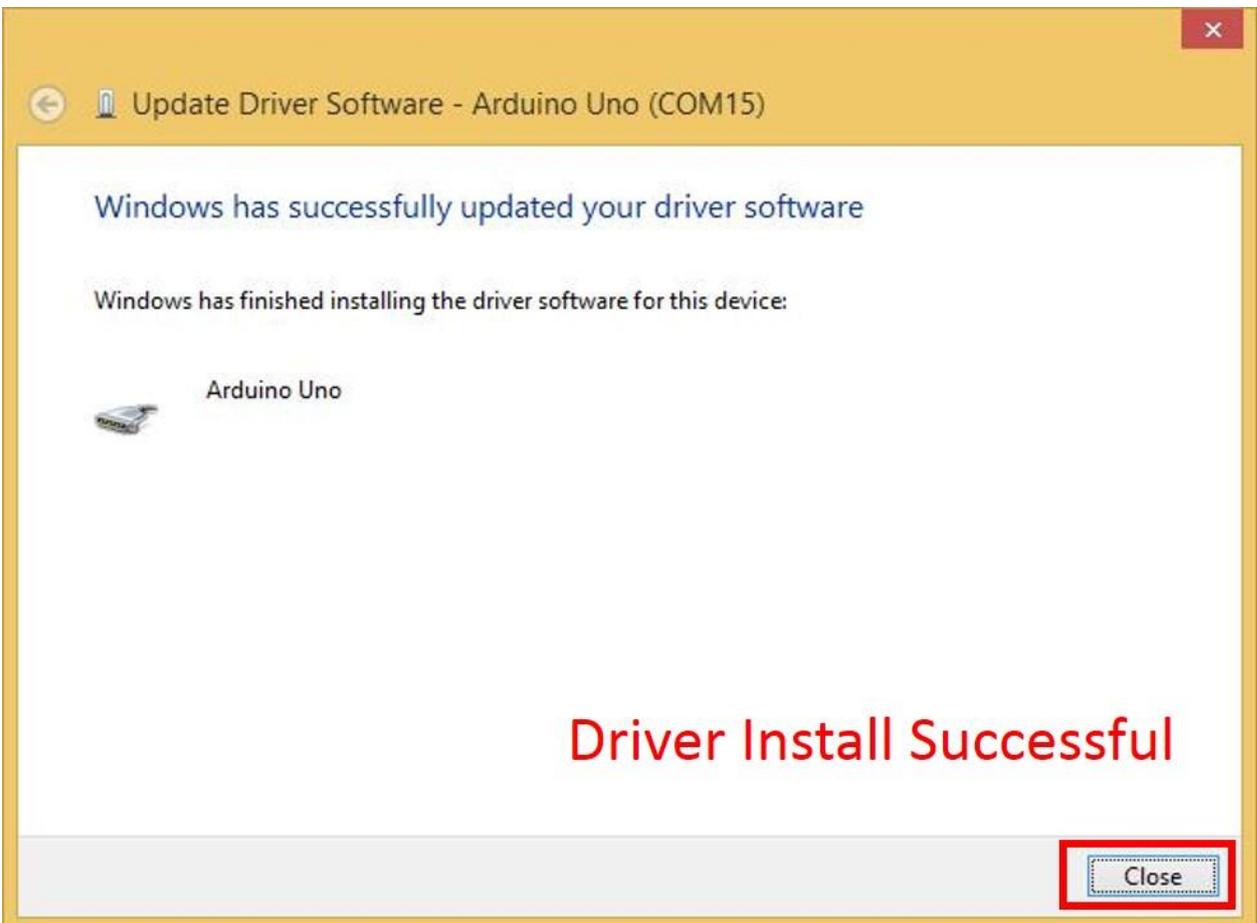
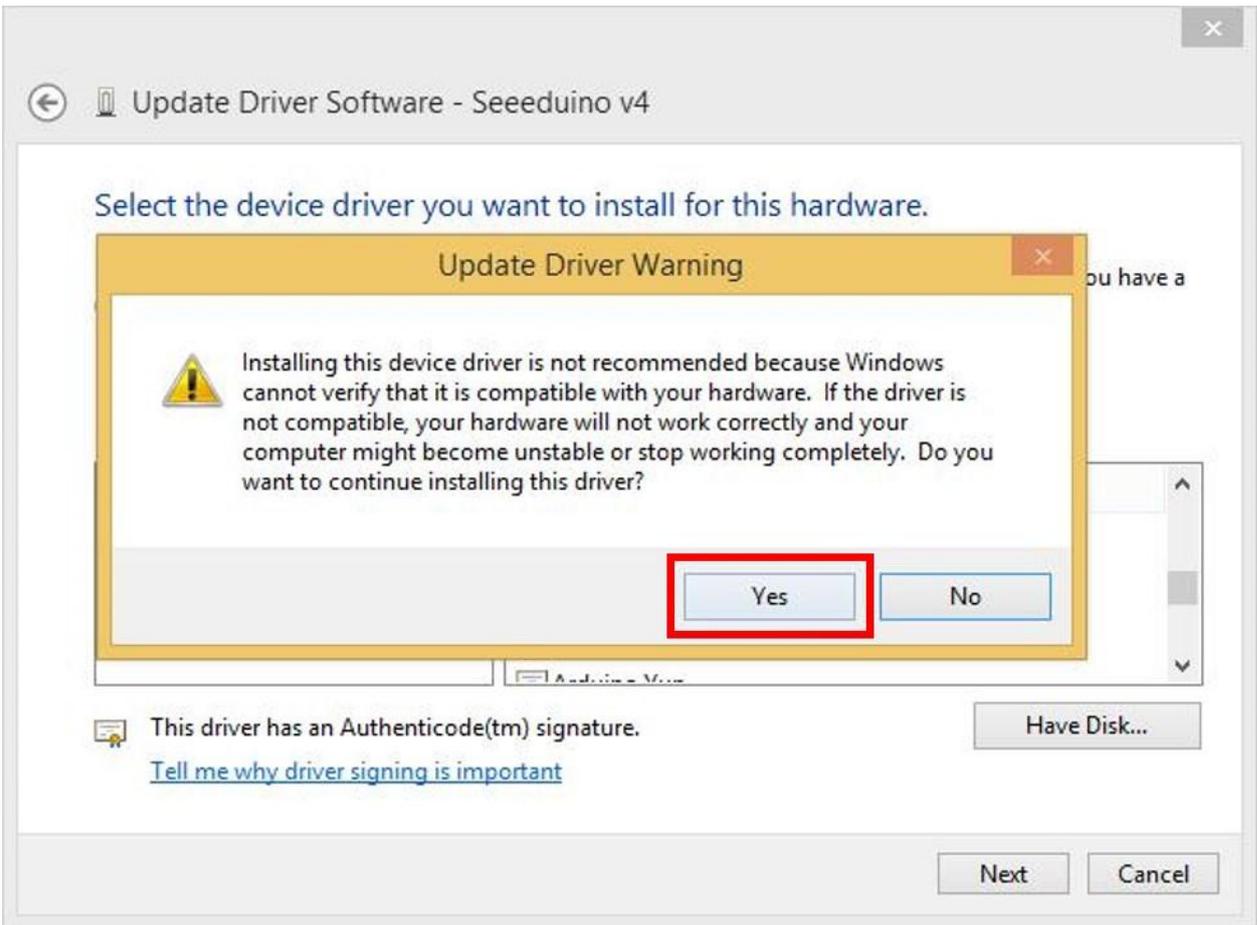










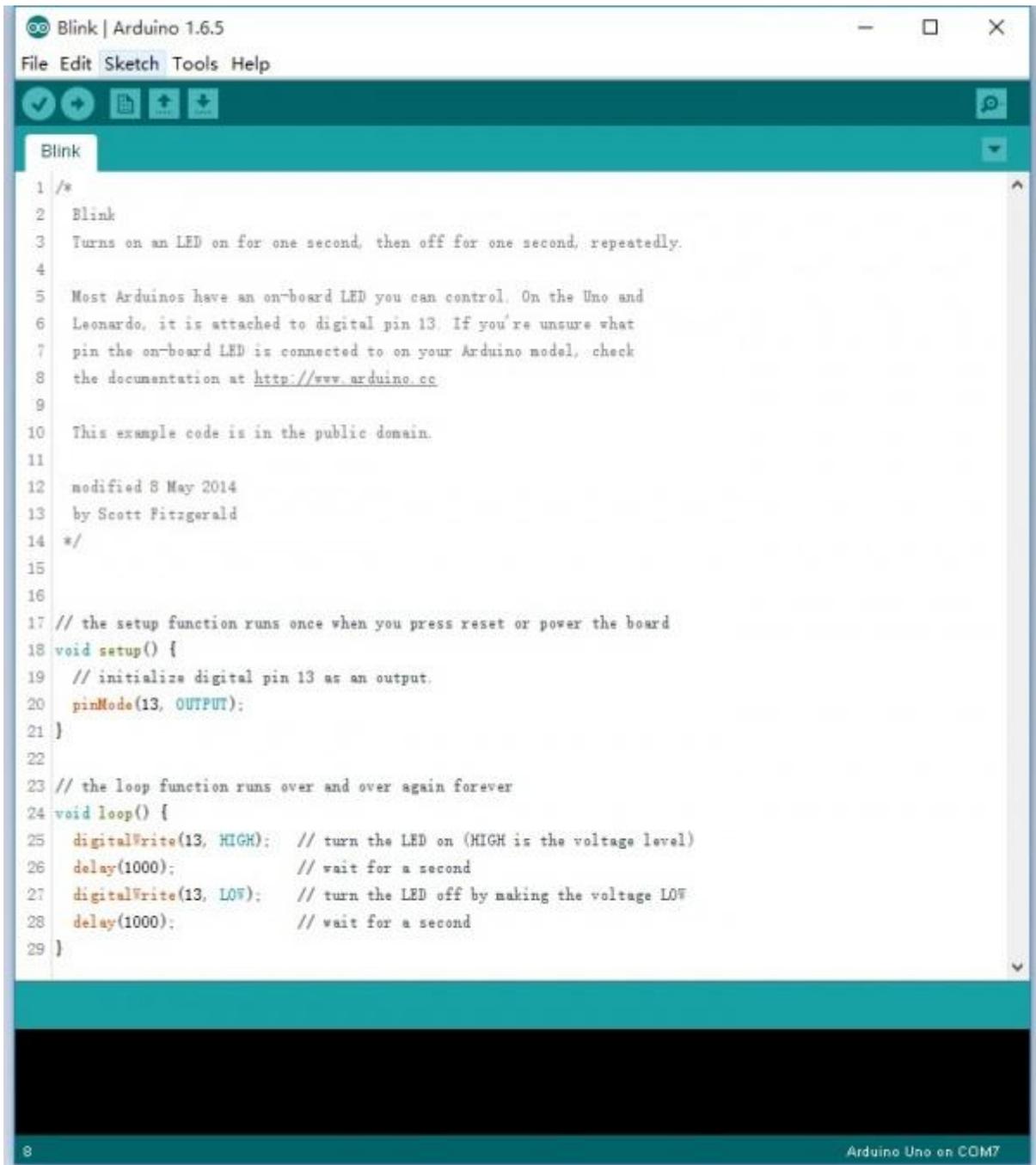


Launch the Arduino application

Double-click the Arduino application (arduino.exe) you have previously downloaded . (Note: if the Arduino Software loads in the wrong language, you can change it in the preferences dialog. See the [Arduino Software \(IDE\) page](#) for details.)

Open the blink example

Open the LED blink example sketch: **File > Examples >01.Basics > Blink.**

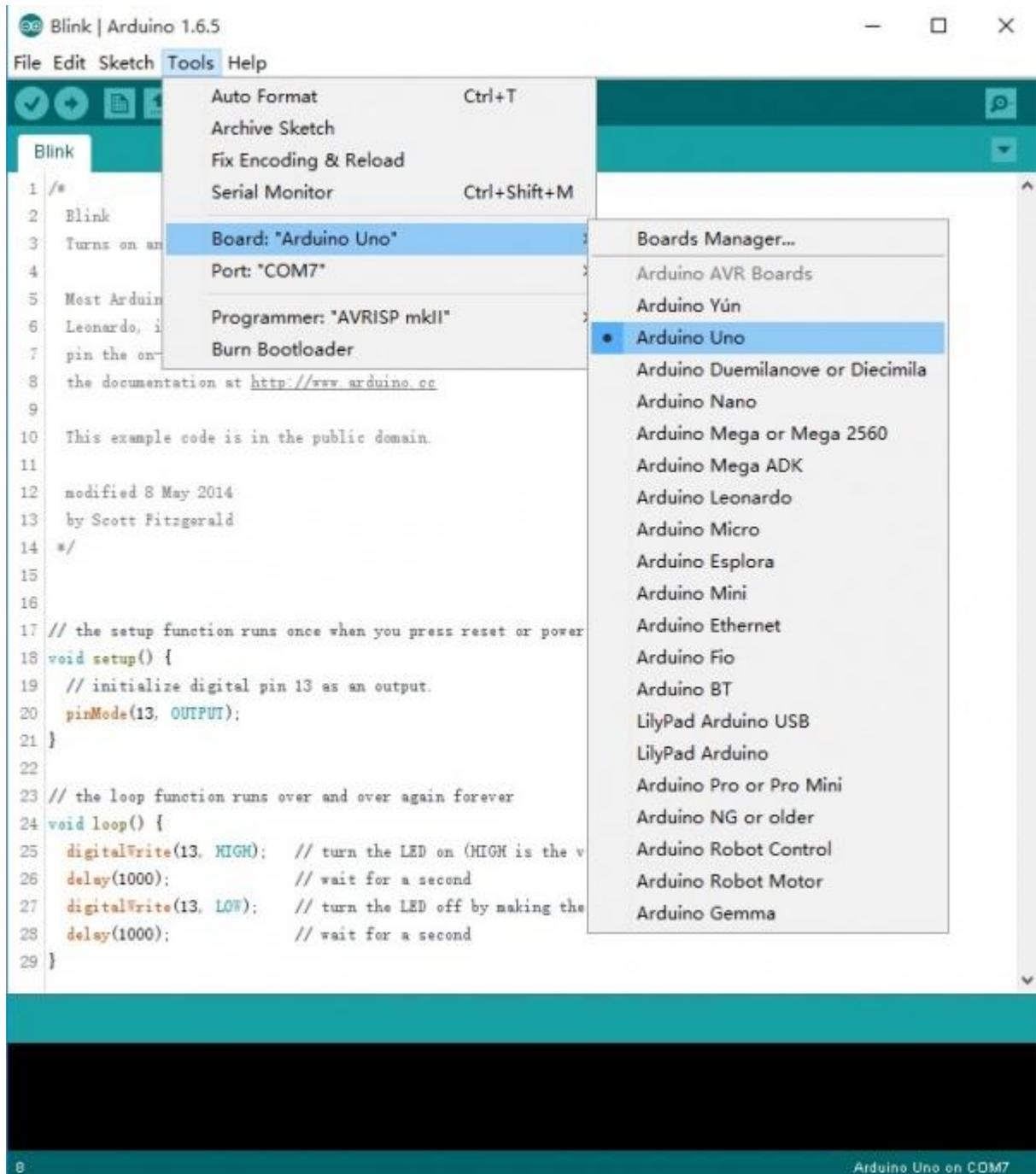
A screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.6.5". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for opening, saving, and running. The main text area shows the code for the "Blink" sketch. The code includes a multi-line comment explaining the sketch's purpose and author information, followed by the C++ code for the setup and loop functions. The setup function initializes pin 13 as an output. The loop function turns the LED on for one second, then off for one second, repeating this cycle. The status bar at the bottom shows "8" on the left and "Arduino Uno on COM7" on the right.

```
1 /*
2  Blink
3  Turns on an LED on for one second, then off for one second, repeatedly.
4
5  Most Arduinos have an on-board LED you can control. On the Uno and
6  Leonardo, it is attached to digital pin 13. If you're unsure what
7  pin the on-board LED is connected to on your Arduino model, check
8  the documentation at http://www.arduino.cc
9
10 This example code is in the public domain.
11
12 modified 8 May 2014
13 by Scott Fitzgerald
14 */
15
16
17 // the setup function runs once when you press reset or power the board
18 void setup() {
19   // initialize digital pin 13 as an output.
20   pinMode(13, OUTPUT);
21 }
22
23 // the loop function runs over and over again forever
24 void loop() {
25   digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
26   delay(1000);           // wait for a second
27   digitalWrite(13, LOW);  // turn the LED off by making the voltage LOW
28   delay(1000);           // wait for a second
29 }
```

Select your board

You'll need to select the entry in the Tools > Board menu that corresponds to your Arduino.

Selecting an Arduino Uno:



Select your serial port

Select the serial device of the Arduino board from the Tools | Serial Port menu. This is likely to be **COM3** or higher (**COM1** and **COM2** are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.

Upload the program

Now, simply click the "Upload" button in the environment. Wait a few seconds - you should see the RX and TX leds on the board flashing. If the upload is successful, the message "Done uploading." will appear in the status bar.



A few seconds after the upload finishes, you should see the pin 13 (L) LED on the board start to blink (in orange). If it does, congratulations! You've gotten Arduino up-and-running. If you have problems, please see the troubleshooting suggestions.

Getting Started on Mac OS X

This part is based on the [Getting Started w/ Arduino on Mac OS X](#), click to view the original page.

Download the Arduino environment

Get the latest version from the [download page](#). When the download is finished, double click the .zip file. This will expand the Arduino application.

Install the Software

Copy the Arduino application into the Applications folder (or elsewhere on your computer).

Connect the board

The Seeeduno V4.2 automatically draw power from either the USB connection to the computer or an external power supply. Connect the Arduino board to your computer using the USB cable. The green power LED (labelled PWR) should go on.

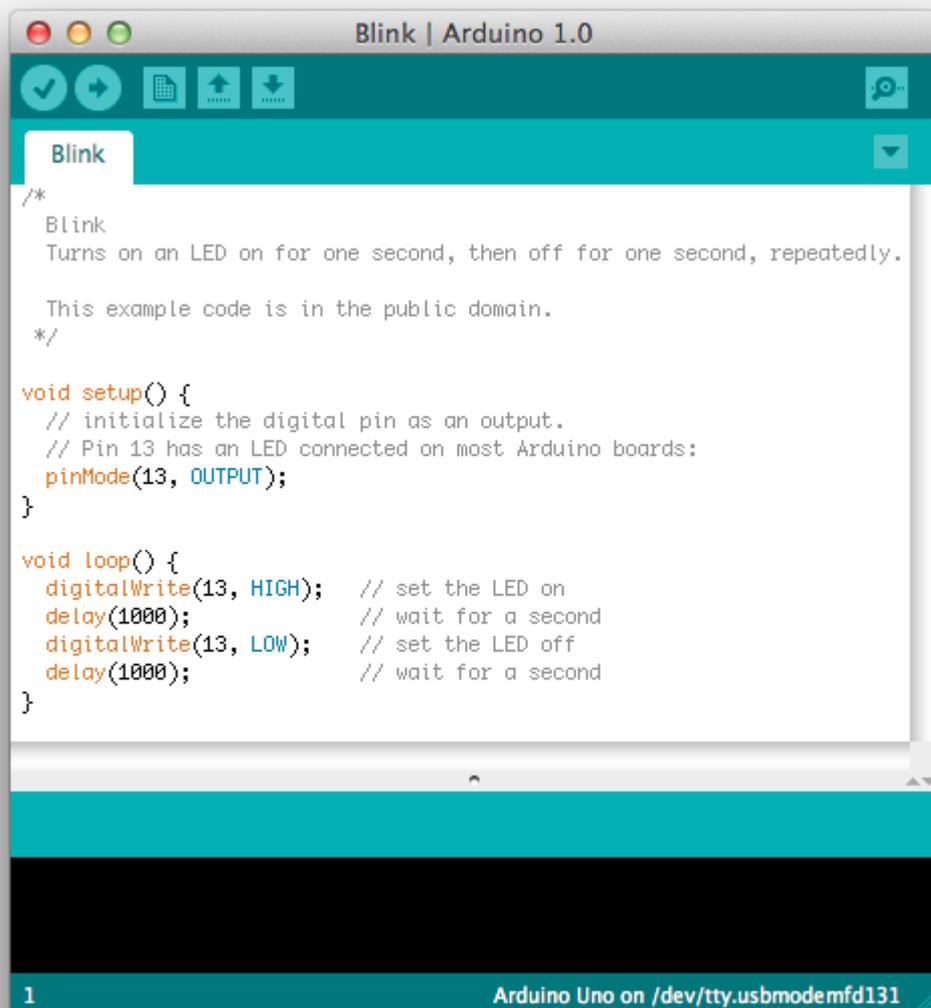
Launch the Arduino application

Double-click the Arduino application.

Note: if the Arduino software loads in the wrong language, you can change it in the preferences dialog. See the [environment page](#) for details.

Open the blink example

Open the LED blink example sketch: **File > Examples > 01.Basics > Blink.**

A screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.0". The top toolbar contains icons for check, run, upload, and download. Below the toolbar is a teal header with the file name "Blink" and a dropdown arrow. The main text area contains the following code:

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

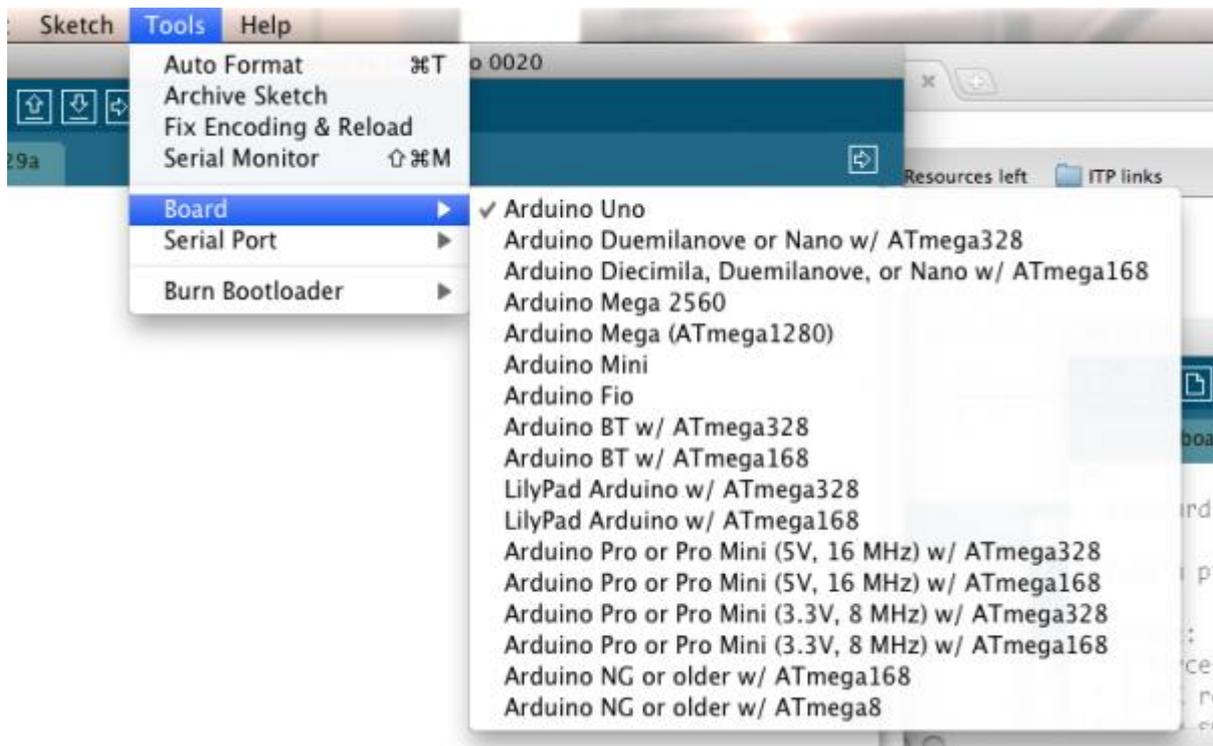
void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000);           // wait for a second
}
```

The bottom status bar shows "1" on the left and "Arduino Uno on /dev/tty.usbmodemfd131" on the right.

Select your board

You'll need to select the entry in the Tools > Board menu that corresponds to your Arduino.

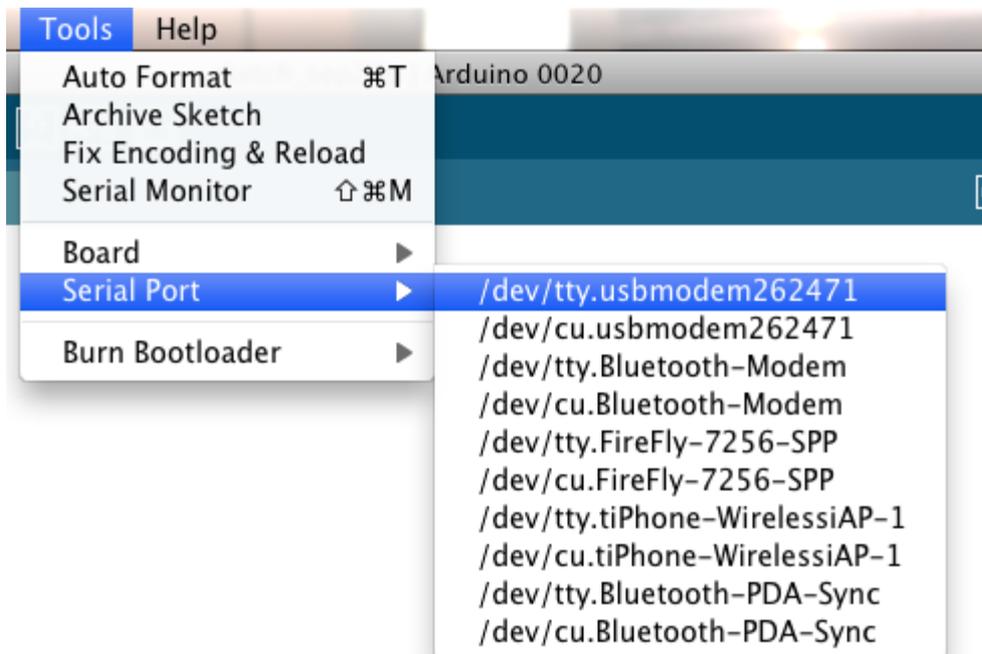


Selecting an Arduino Uno

Details of the board menu entries are available on the [environment page](#).)

Select your Serial Port

Select the serial device of the Arduino board from the Tools > Serial Port menu. On the Mac, this should be something with `/dev/tty.usbmodem` (for the Uno or Mega 2560) or `/dev/tty.usbserial` (for older boards) in it.



Selecting an Uno

Upload the program

Now, simply click the "Upload" button in the environment. Wait a few seconds - you should see the RX and TX leds on the board flashing. If the upload is successful, the message "Done uploading." will appear in the status bar.



A few seconds after the upload finishes, you should see the pin 13 (L) LED on the board start to blink (in orange). If it does, congratulations! You've gotten Arduino up-and-running. If you have problems, please see the troubleshooting suggestions.

Getting Started on Linux

For Linux user, please go to [Installing Arduino on Linux](#)

Resources/References

Resources

- [Schematic of Seeeduino V4.2 in Eagle File](#)
- [Schematic of Seeeduino V4.2 in PDF](#)
- [ATMEGA328P-MU Summary](#)
- [ATMEGA16U2 Summary](#)
- [Other Version of Seeeduino](#)

References

- [Getting Started with Arduino](#)
- [Arduino Language Reference](#)
- [Download the Arduino Software\(IDE\)](#)
- [Arduino FAQ](#)
- [Arduino Intruduction](#)
- [Wikipedia page for Arduino](#)