

IUEE Project 1, part A

Setup your UNO board.

1 | Get an Arduino board and USB cable

In this tutorial, we assume you're using an Arduino Uno. If you have another board, read the corresponding page in this getting started guide.

You also need a standard USB cable (A plug to B plug): the kind you would connect to a USB printer, for example.



2 | Download the Arduino environment

Get the latest version from the [download page](#).

When the download finishes, unzip the downloaded file. Make sure to preserve the folder structure. Double-click the folder to open it. There should be a few files and sub-folders inside.

3 | Connect the board

The Arduino Uno, automatically draws 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.

4 | Install the drivers

Installing drivers for the Arduino Uno with Windows7, 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)"

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 Uno's driver file, named "**ArduinoUNO.inf**", located in the "Drivers" folder of the Arduino Software download (not the "FTDI USB Drivers" sub-directory).

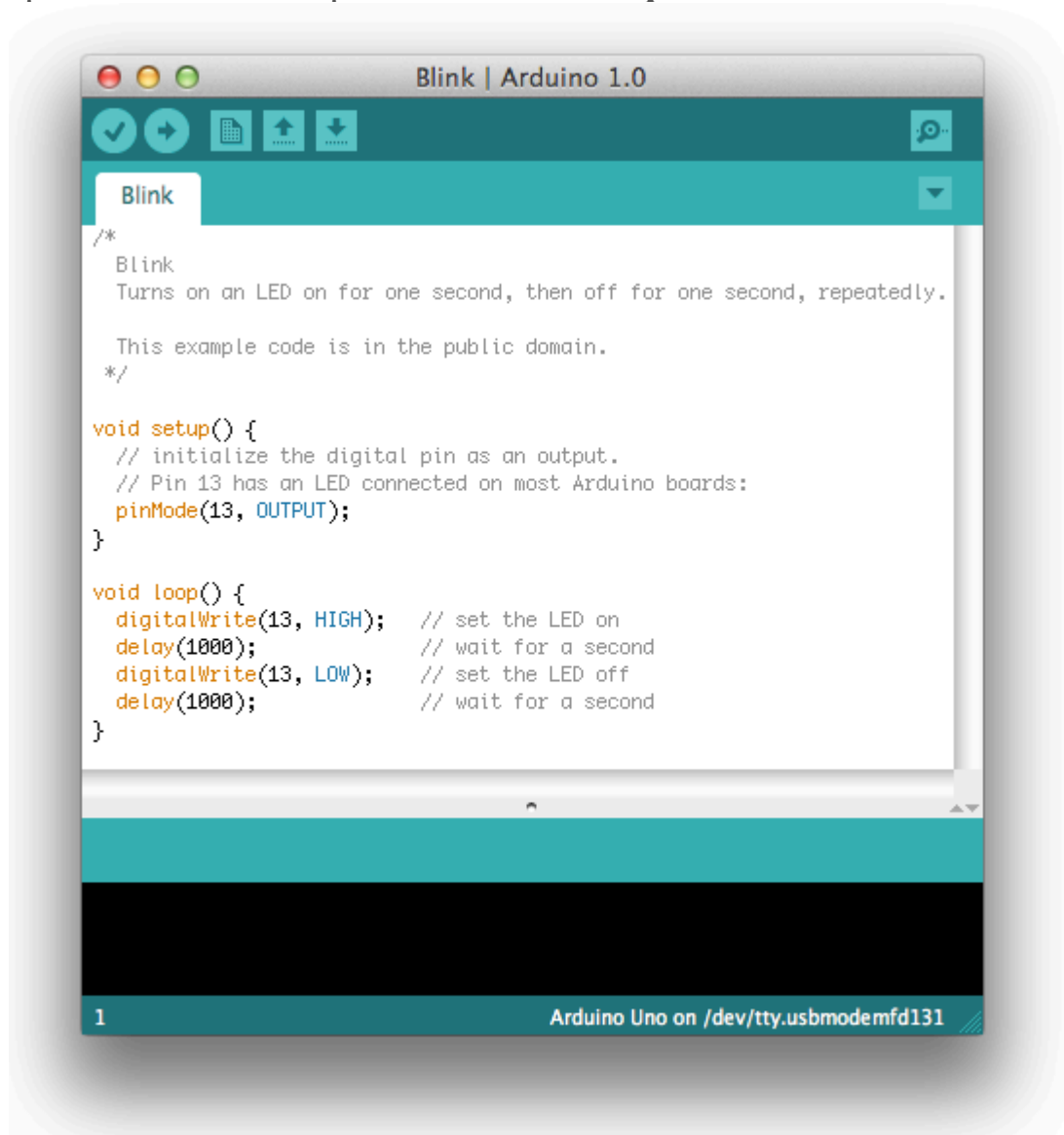
Windows will finish up the driver installation from there. See also: [step-by-step screenshots for installing the Uno under Windows XP](#).

5 | 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.)

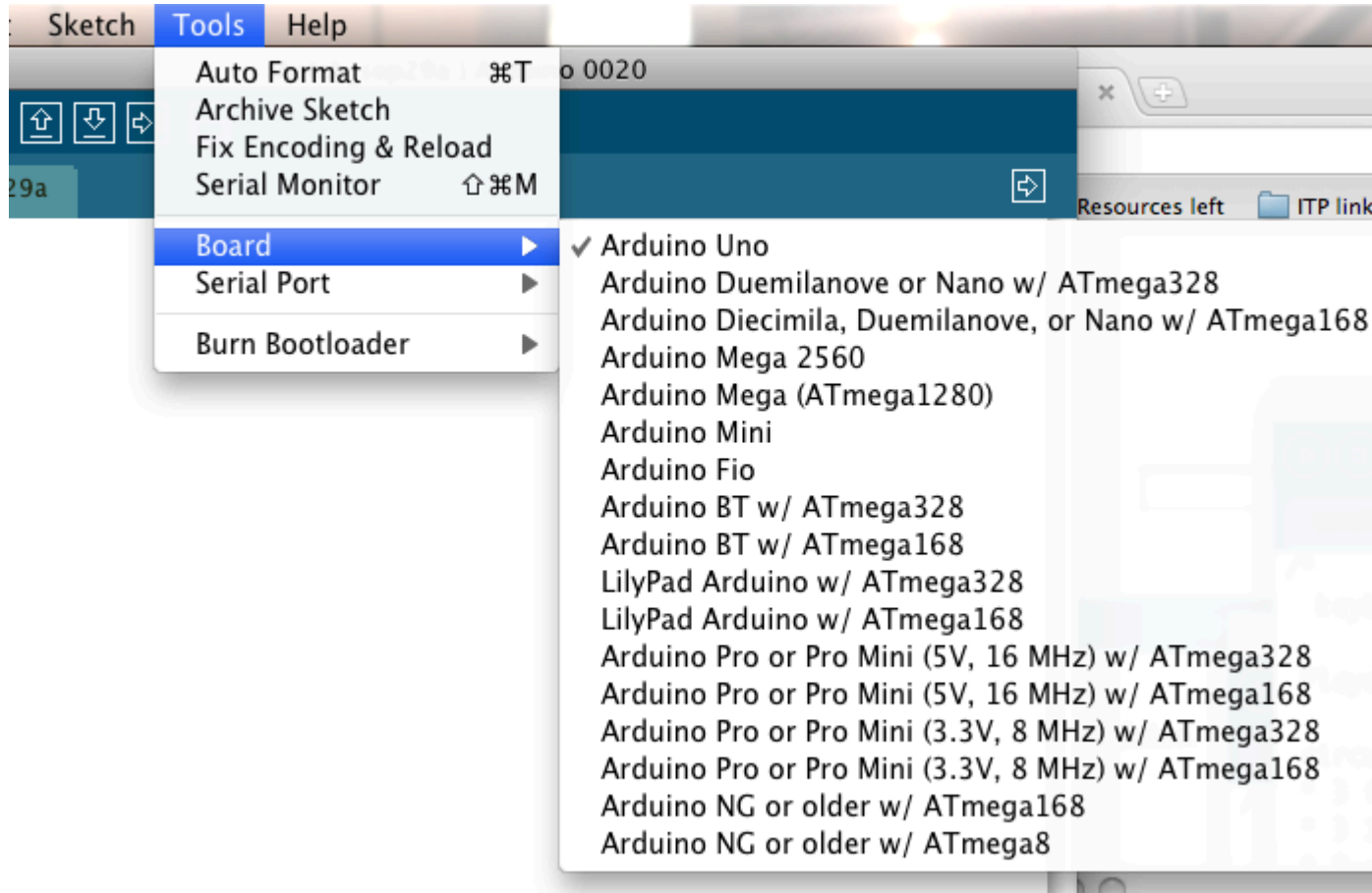
6 | Open the blink example

Open the LED blink example sketch: **File > Examples > 1.Basics > Blink**.



7 | Select your board

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



Selecting an Arduino Uno

8 | 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.

9 | 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.

Intel Ultimate Engineering Experience (IUEE)

If you have problems, please see the [troubleshooting suggestions](#).

You might also want to look at:

the [examples](#) for using various sensors and actuators
the [reference](#) for the Arduino language

The text of the Arduino getting started guide is licensed under a [Creative Commons Attribution-ShareAlike 3.0 License](#). Code samples in the guide are released into the public domain.

IUEE Project 1, part B

Blink

This example shows the simplest thing you can do with an Arduino to see physical output: it blinks an LED.

Hardware Required

Arduino Board
LED
220 (or 270) ohm resistor
breadboard
hook-up wire

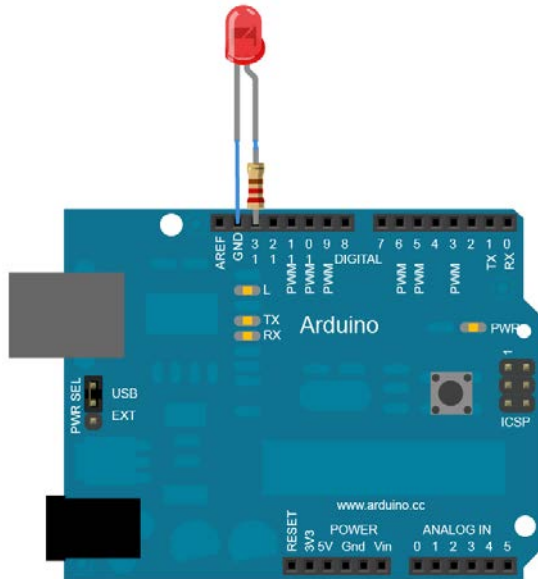
Circuit

To build the circuit, attach a 220-ohm resistor to pin 12. Then attach the long leg of an LED (the positive leg, called the anode) to the resistor. Attach the short leg (the negative leg, called the cathode) to ground. Then plug your Arduino board into your computer, start the Arduino program, and enter the code below.

Most Arduino boards already have an LED attached to pin 13 on the board itself. If you run this example with no hardware attached and without modifying the Blink program, you should see that LED blink.

Note: You will want to use your breadboard to hook up the LED and resistor. This is much easier to connect up the components and we will be using this for the next projects.

click the image to enlarge



Schematic

[click the image to enlarge](#)

Code

Using the Blink.ino program in the examples/01.Basics/Blink directory. Modify the line

```
int led = 13;
```

To

```
int led = 12;
```

Save the file

Save the file under a different name such as BlinkPin12. To do this, use the File/Save As operation on the toolbar. Navigate to the examples/01.Basics/ directory. Create a new folder there named BlinkPin12 and then save your modified code (or sketch) as BlinkPin12.

Repeat Step 9 from Part A

Use the procedure described previously to upload the BlinkPin12 files onto your Arduino UNO. When you upload the file to the Arduino UNO board, the IDE will automatically compile it for you.

You should now see the external LED blinking!

If you have extra time

Try modifying the “blink rate” by looking at the BlinkPin12 code and modifying the lines that control the delays.