Installing Arduino IDE a learn.sparkfun.com tutorial

Available online at: http://sfe.io/t61

Contents

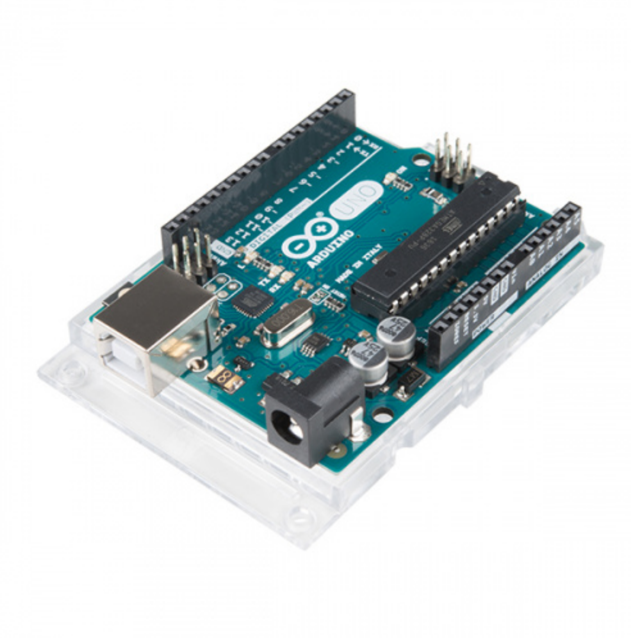
Introduction

This tutorial will walk you through downloading, installing, and testing theArduino software (also known as the Arduino IDE - short for Integrated Development Environment). Before you jump to the page for your operating system, make sure you've got all the right equipment.

Required Materials

To follow along with this tutorial, you will need the following materials. You may not need everything though depending on what you have. Add it to your cart, read through the guide, and adjust the cart as necessary.

- A computer (Windows, Mac, or Linux)
- An Arduino-compatible microcontroller (anything from this guide should work)
- A USB A-to-B cable, or another appropriate way to connect your Arduino-compatible microcontroller to your computer (check out thisUSB buying guide if you're not sure which cable to get).



Arduino Uno - R3

DEV-11021
$27.60
138
Favorited Favorite 174
Wish List

USB Cable A to B - 6 Foot

CAB-00512
$4.50
1
Favorited Favorite 13
Wish List

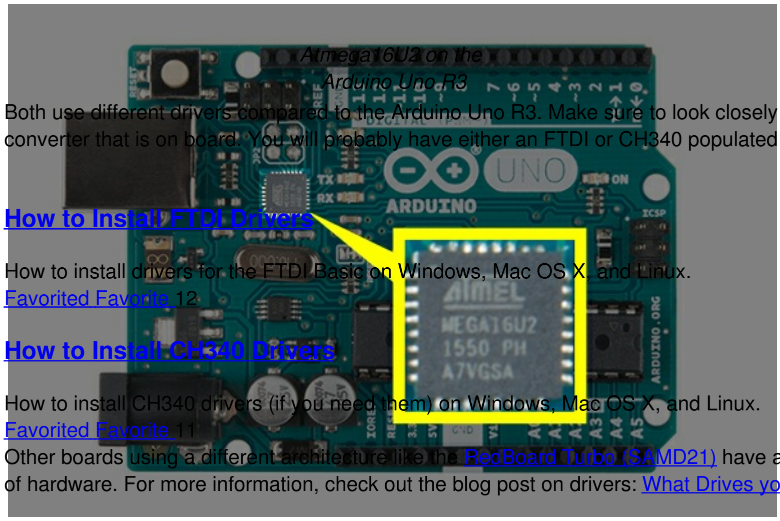
Suggested Reading

If you're new to Arduino in general, you'll want to check out this tutorial to familiarize yourself with everyone's favorite microcontroller platform.

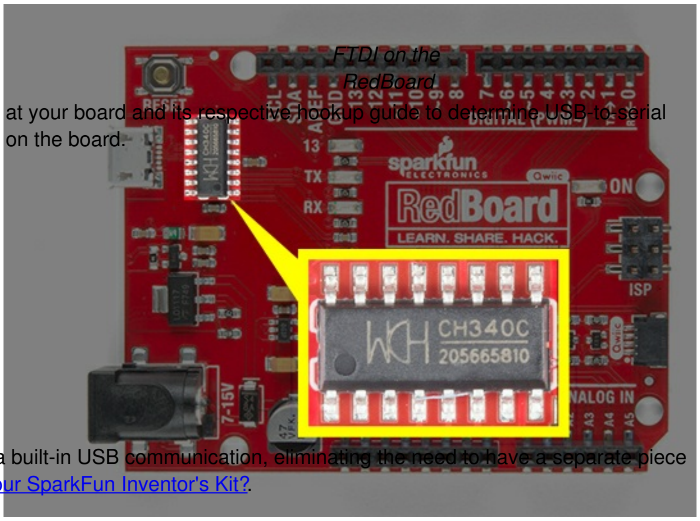
What is an Arduino?

February 26, 2013

What is this 'Arduino' thing anyway? This tutorials dives into what an Arduino is and along with Arduino projects and widgets.
Favorited Favorite 56
Note: There are several variants that use the Arduino Uno R3 footprint. Depending on the design, you may need to install additional drivers for your USB-to-serial converter before you are able to able to upload code to your microcontroller. For example, the RedBoard uses an FTDI while the RedBoard Plus uses the CH340.



Atmega16U2 on the Arduino Uno R3



FTDI on the RedBoard

Both use different drivers compared to the Arduino Uno R3. Make sure to look closely at your board and its respective hookup guide to determine USB-to-serial converter that is on board. You will probably have either an FTDI or CH340 populated on the board.

How to Install FTDI Drivers

How to install drivers for the FTDI Basic on Windows, Mac OS X, and Linux.
Favorited Favorite 12

How to Install CH340 Drivers

How to install CH340 drivers (if you need them) on Windows, Mac OS X, and Linux.
Favorited Favorite 11

Other boards using a different architecture like the RedBoard Turbo (SAMD21) have a built-in USB communication, eliminating the need to have a separate piece of hardware. For more information, check out the blog post on drivers: What Drives your SparkFun Inventor's Kit?.

If you're ready to get started, click on the link in the column on the left that matches up with your operating system, or you can jump to your operating system here.

- Windows
- Mac
- Linux

Downloading the Arduino IDE

You can download the Arduino IDE from their website. They have installation instructions, but we will also go over the installation process as well. Make sure you download the version that matches your operating system.

Click for Arduino IDE Download Page

The installation procedure is fairly straightforward, but it does vary by OS. Here are some tips to help you along.

Troubleshooting Tips

- We recommend using a computer with a full desktop operating system like Windows 7/10 (**avoid** Windows 8 if you can), Mac OSX, and certain flavors Linux (check the Arduino FAQ page for compatibility).
    - If you are not a technical or computer savy individual and you have your choice of computers, I highly recommend using a **Windows 7, 10, or 11** computer. You will usually run into the the least issues, if any, with these operating systems.
- We do **NOT** recommend using a Chromebook, Netbook, tablet, phone, or the Arduino Web IDE in general. You will be responsible for troubleshooting any driver or Arduino Web IDE issues.
- As of the writing of this tutorial (updated 12-14-2018), the most recent and stable release of the Arduino IDE is version 1.8.5. We recommend using that version of the Arduino IDE; you can download the previous releases here.
- Raspberry Pi users with Raspbian installed should use the **Linux ARM** download. We do not recommend using the command line installation. It will install the oldest release of Arduino, which is useless when it comes to installing new boards definitions or libraries.
- For additional troubleshooting tips, here is a troubleshooting guide from Arduino.

Windows

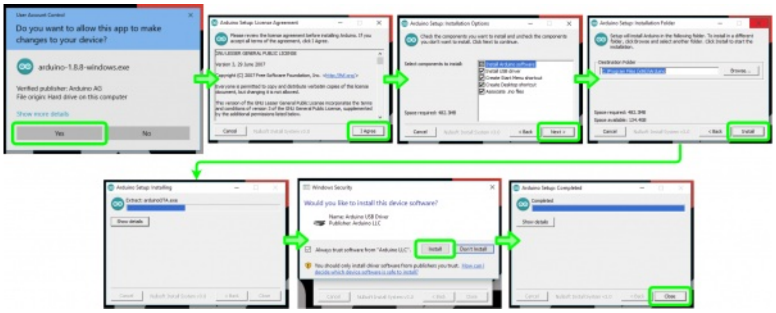
This page will show you how to install and test the Arduino software with a Windows operating system (Windows 10, Windows 7, Vista, and XP).

Go to the Arduino download page and download the latest version of the Arduino software for Windows if you have not already.

Click for Arduino IDE Download Page

Installer

The Windows version of Arduino is offered in two options: an installer or a zip file. The **installer** is the easier of the two options, just download that, and run the executable file to begin the installation.


User Account Control

Do you want to allow this app to make changes to your device?

arduino-1.8.8-windows.exe

Verified publisher: Arduino AG
File origin: Hard drive on this computer

Arduino Setup: License Agreement

Arduino Setup: Installation Options

Arduino Setup: Installation Folder

Arduino Setup: Installing

Windows Security

Would you like to install this device software?

Name: Arduino USB Driver
Publisher: Arduino LLC

Arduino Setup: Completed

Windows install steps. Click the image to get a bigger view.

When you're prompted to install a driver during installation, select "Install". This will install drivers for Arduino specific boards (like the Uno, Nano, etc.) that you may use in the future.

ZIP

If you choose to download the**zip file** version of Arduino, you'll need to extract the files yourself. Don't forget which folder you extract the files into! You will need to run the executable Arduino file in the folder to start the Arduino IDE.

When the download is finished, un-zip it and open up the Arduino folder to confirm that yes, there are indeed some files and sub-folders inside. The file structure is important so don't be moving any files around unless you really know what you're doing.

Note: On Windows 10, there is an option to install Arduino through their app store. we do not recommend installing the Arduino IDE from the app store. You may run into issues because the OS will automatically update to the most recent release of the Arduino IDE, which may have unknown bugs.

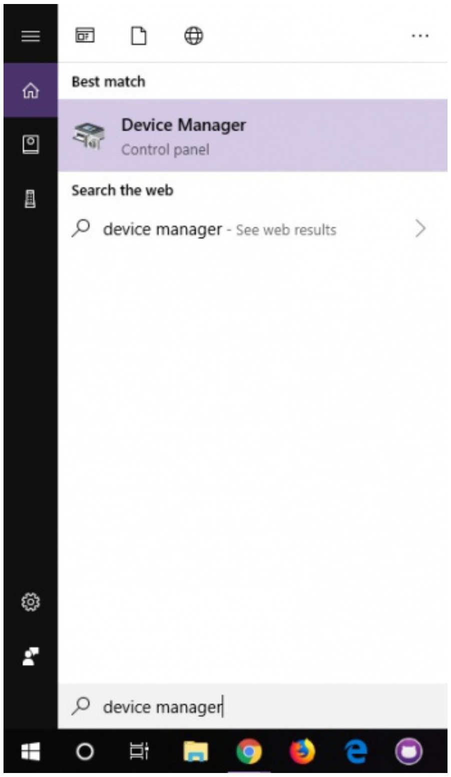
Connecting Your Arduino

Power up your Arduino by connecting your Arduino board to your computer with a USB cable (or FTDI cable if you're using an Arduino Pro). You should see the an LED labeled 'ON' light up. (this diagram shows the placement of the power LED on the UNO).

Drivers for Arduino Uno on Windows

To install the drivers for the Arduino Uno, you will need to plug in your board to your computer's USB port. Once the board is connected, you will need to wait for Windows to begin it's driver installation process. After a few moments, the process will probably fail, despite its best efforts.

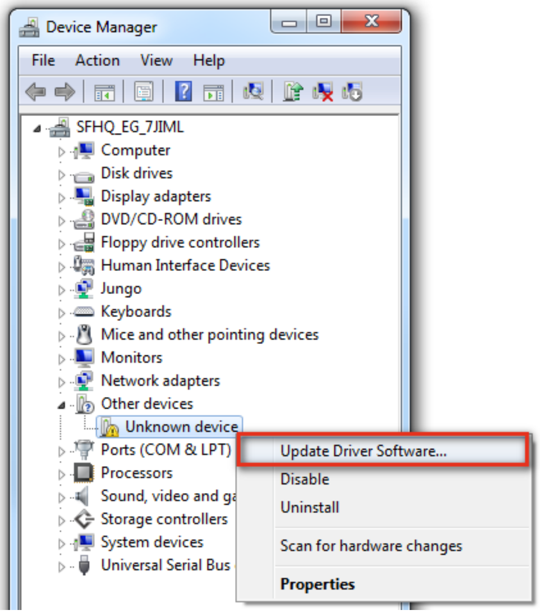
Open up a search, type in**Device Manager**, and hit ENTER.



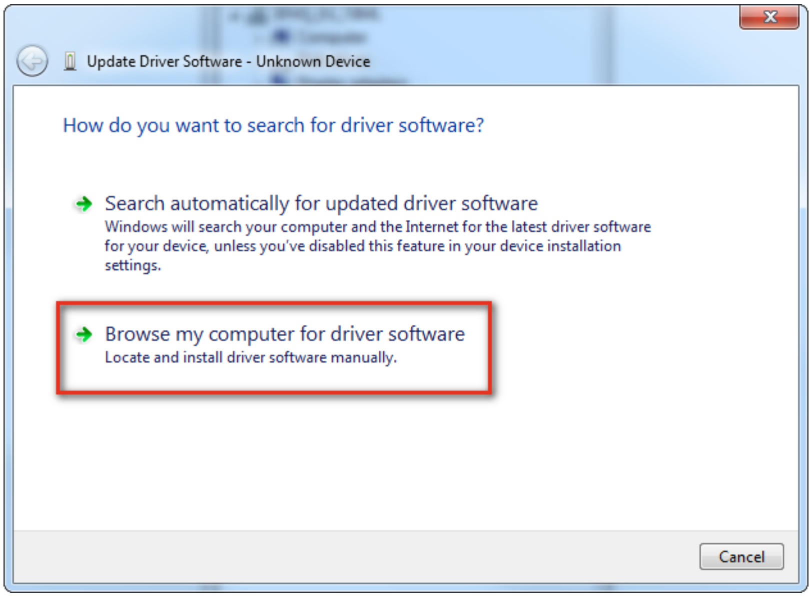
Note: Searching is the easiest method to open the Device Manager. However, there is more than one method of opening the device manager. The longer method is to click on the **Start Menu** > **Windows System** > **System and Security** > **System** > **Device Manager**.

Look under Ports (COM & LPT) tree. 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)**" or "**Unknown device**" port and choose the "**Update Driver Software**" option.



Next, choose the "**Browse my computer for Driver software**" option



How do you want to search for driver software?

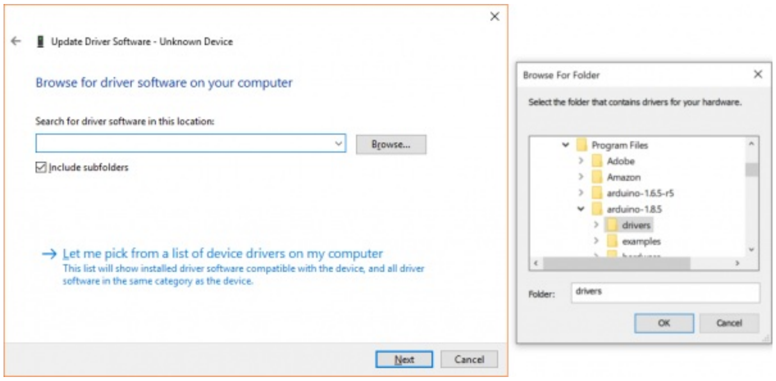
Search automatically for updated driver software

Windows will search your computer and the Internet for the latest driver software for your device, unless you've disabled this feature in your device installation settings.

Browse my computer for driver software

Locate and install driver software manually.

Finally, navigate to the Arduino IDE folder. This should be where you unzipped the Arduino IDE (e.g. it should be similar to the following path with a different version number: *C:\Program Files\arduino-1.8.5\drivers*). Depending on what version of Windows you have, you may be able to select the Uno's driver file, named "**Arduino.inf**", located in the "**Drivers**" folder (not the "FTDI USB Drivers" sub-directory). If you cannot see the***.inf** file, it is probably just hidden. You can select the "**drivers**" folder with the "**include sub-folders**" option selected instead.



Browse for driver software on your computer

Search for driver software in this location:

☑ Include subfolders

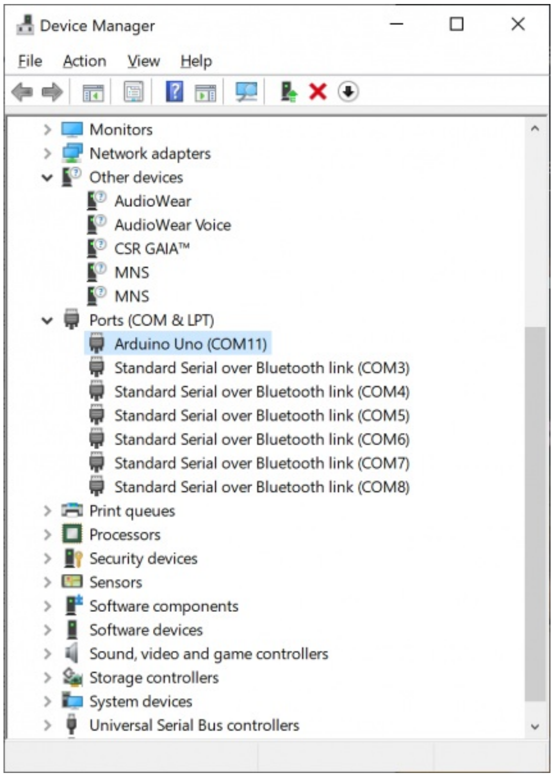
Let me pick from a list of device drivers on my computer

This list will show installed driver software compatible with the device, and all driver software in the same category as the device.

Select the folder that contains drivers for your hardware.

Folder: drivers

Windows will finish up the driver installation from there! Your computer will enumerate with a COM port. You may see a COM port number depending on what is currently saved in your computer. Try to remember what the number is when uploading. If not, you can always navigate back to the device manager and power cycle the Arduino to determine what number your Arduino enumerated on.


Device Manager
File
Action
View
Help
Monitors
Network adapters
Other devices
AudioWear
AudioWear Voice
CSR GAIA™
MNS
MNS
Ports (COM & LPT)
Arduino Uno (COM11)
Standard Serial over Bluetooth link (COM3)
Standard Serial over Bluetooth link (COM4)
Standard Serial over Bluetooth link (COM5)
Standard Serial over Bluetooth link (COM6)
Standard Serial over Bluetooth link (COM7)
Standard Serial over Bluetooth link (COM8)
Print queues
Processors
Security devices
Sensors
Software components
Software devices
Sound, video and game controllers
Storage controllers
System devices
Universal Serial Bus controllers

For earlier versions of the Arduino boards (e.g. Arduino Duemilanove, Nano, or Diecimila) check outthis page for specific directions.

Arduino.cc - Guide: Windows

Drivers for RedBoard on Windows

Depending on your board, there may be different USB-to-serial converters on your board. If you are using a RedBoard you will need to go to theHow to Install FTDI Drivers tutorial. If you are using the or RedBoard Qwiic, you will need to go to theHow to Install CH340 Drivers tutorial. Make sure to look closely at your board and its respective hookup guide to determine USB-to-serial converter that is on board.

How to Install FTDI Drivers

June 4, 2013

How to install drivers for the FTDI Basic on Windows, Mac OS X, and Linux.
Favorited Favorite 12

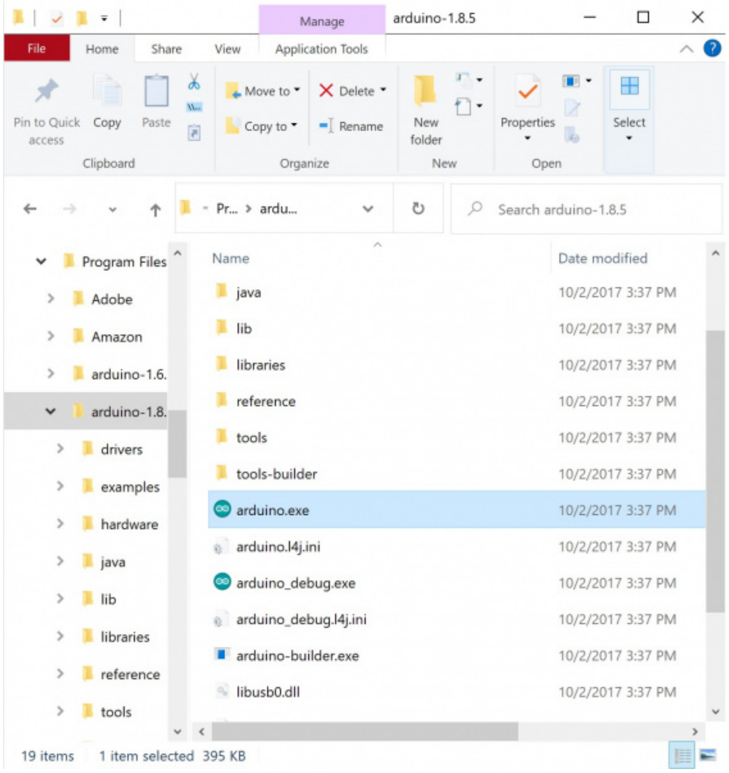
How to Install CH340 Drivers

August 6, 2019

How to install CH340 drivers (if you need them) on Windows, Mac OS X, and Linux.
Favorited Favorite 11

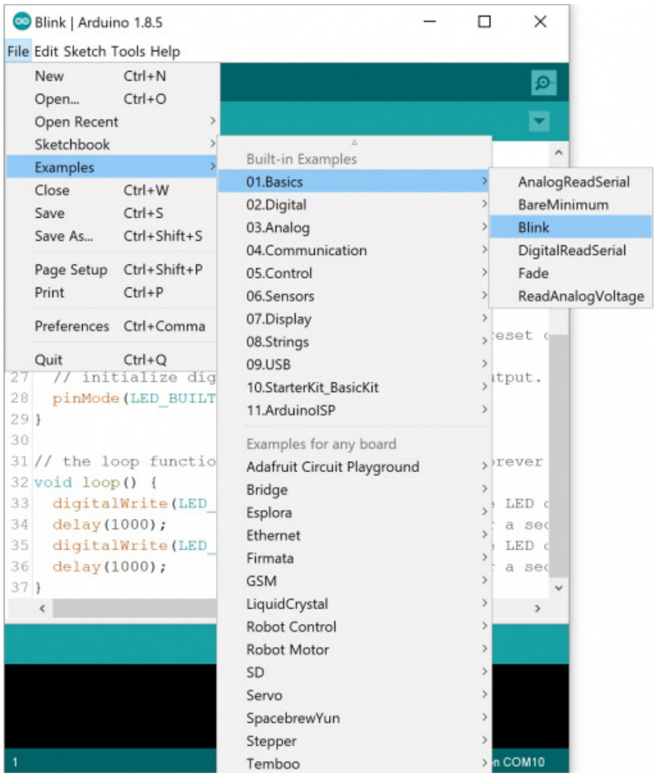
Launch and Blink!

After following the appropriate steps for your software install, we are now ready to test your first program with your Arduino board! Launch the Arduino application. If you disconnected your board, plug it back in.

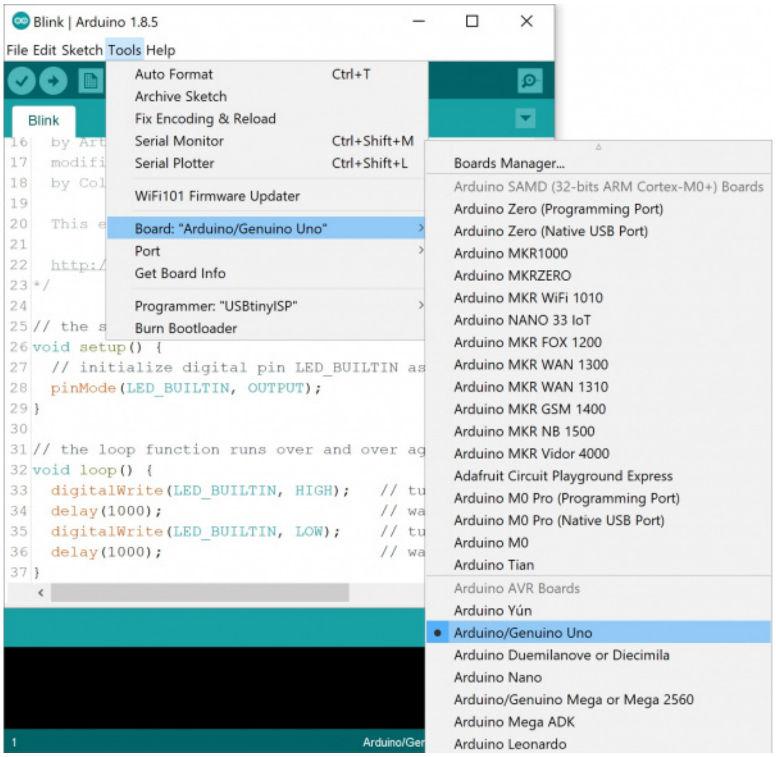


Note: Depending on your method of installing the Arduino IDE, the application may be on your desktop or the program folder.

Open the Blink example sketch by going to:**File** > **Examples** > **01.Basics** > **Blink**.



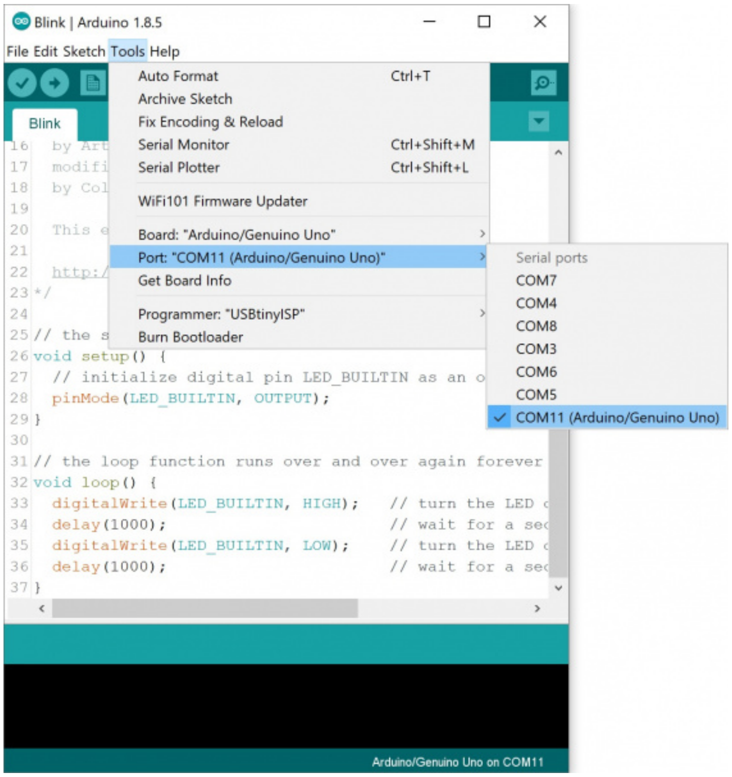
Select the type of Arduino board you're using:**Tools** > **Board** > **Arduino Uno**.



```
  by Art
  modifi
  by Col

  This e

  http:/
*/

// the s
void setup() {
  // initialize digital pin LED_BUILTIN as
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over ag
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // tu
  delay(1000);                       // wa
  digitalWrite(LED_BUILTIN, LOW);    // tu
  delay(1000);                       // wa
}
```

Note: As you move to other architectures, you may need to select a different board definition depending on your development board. For the Arduino Uno R3 and RedBoard development boards with ATmega328P, you can simply select **Arduino Uno**. Certain Arduino IDE versions may have you select **Arduino/ Genuino Uno**.

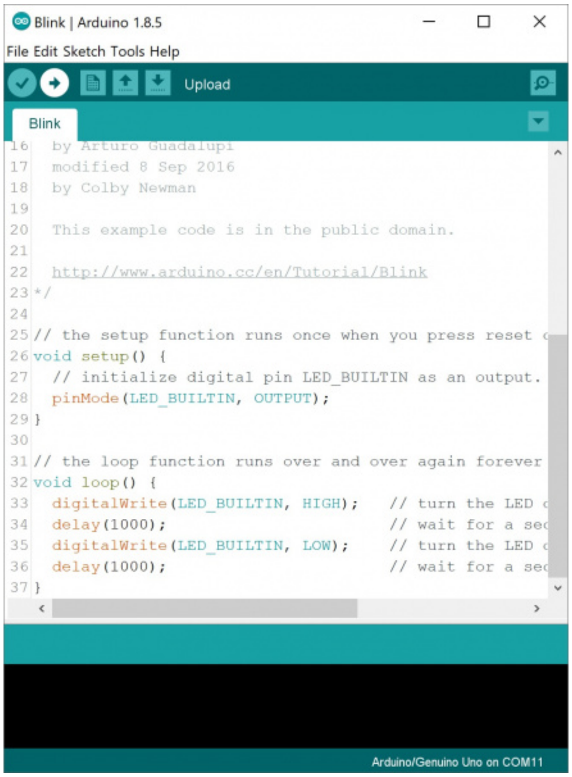
Select the serial/COM port that your Arduino is attached to:**Tools** > **Port** > **COMxx**. In this case it was *COM11*.


Blink | Arduino 1.8.5
File Edit Sketch Tools Help
Blink
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM11 (Arduino/Genuino Uno)"
Get Board Info
Programmer: "USBtinyISP"
Burn Bootloader
Serial ports
COM7
COM4
COM8
COM3
COM6
COM5
✓ COM11 (Arduino/Genuino Uno)
Arduino/Genuino Uno on COM11

```
  by Art
  modifi
  by Col

  This e

  http:/
*/

// the s
void setup() {
  // initialize digital pin LED_BUILTIN as an o
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED c
  delay(1000);                       // wait for a sec
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED c
  delay(1000);                       // wait for a sec
}
```

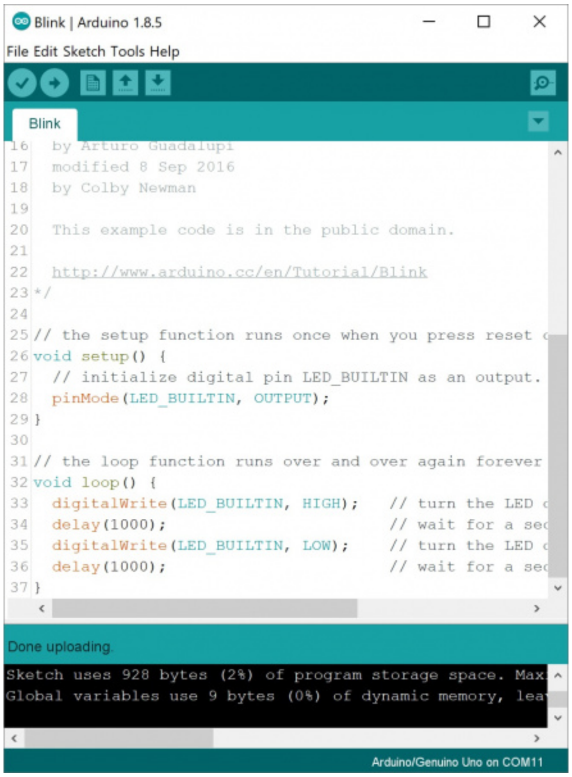
Note: If you're not sure which serial device is your Arduino, take a look at the available ports, then unplug your Arduino and look again. The one that disappeared is your Arduino.

With your Arduino board connected, and the Blink sketch open, press the **Upload**" button.



```
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset o
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED o
  delay(1000);                       // wait for a sec
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED o
  delay(1000);                       // wait for a sec
}
```

After a second, you should see some LEDs flashing on your Arduino, followed by the message **Done Uploading**" in the status bar of the Blink sketch.



```
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset o
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED o
  delay(1000);                       // wait for a sec
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED o
  delay(1000);                       // wait for a sec
}
```

Done uploading.

```
Sketch uses 928 bytes (2%) of program storage space. Max
Global variables use 9 bytes (0%) of dynamic memory, lea
```

If everything worked, the onboard LED on your Arduino should now be blinking! You just programmed your first Arduino!

Note: Depending on the architecture and development board, the built-in LED may be defined on a different pin. You may need to adjust`LED_BUILTIN` or pin `13` to a different value before uploading.

Troubleshooting

[This guide]() from Arduino has some more details and troubleshooting tips if you get stuck.

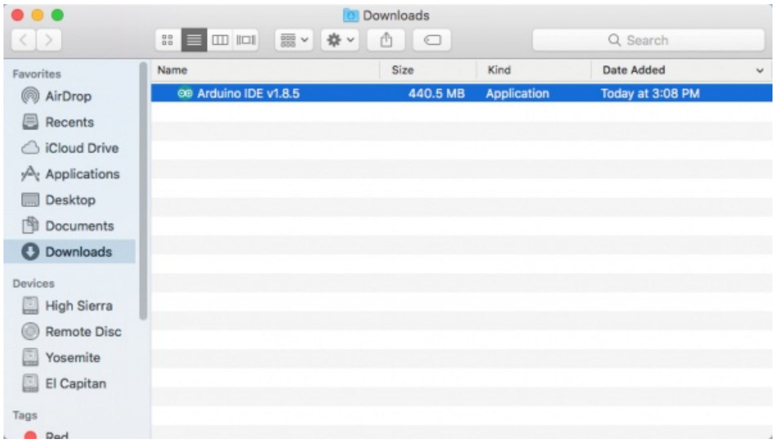
Mac

This page will show you how to install and test the Arduino software on a Mac computer running OSX.

Go to the Arduinodownload page and download the latest version of the Arduino software for Windows if you have not already.

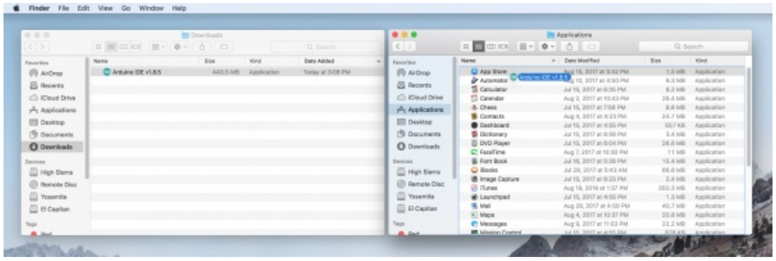
Click for Arduino IDE Download Page

When the download is finished your Mac should automatically un-zip the contents. This will probably in you**Downloads** folder.



Name	Size	Kind	Date Added
Arduino IDE v1.8.5	440.5 MB	Application	Today at 3:08 PM

You can leave the program in the**Downloads** folder or move it into your Applications folder.



Connecting Your Arduino

Power up your Arduino by connecting your Arduino board to your computer with a USB cable (or FTDI cable if you're using an Arduino Pro). You should see the an LED labeled 'ON' light up. (this diagram shows the placement of the power LED on the UNO).

Drivers for Arduino Uno on Mac

For the Arduino Uno, you do not need to install drivers for Mac! You can skip on down and try uploading a blink sketch!

Drivers for RedBoard on Mac

Depending on your board, there may be different USB-to-serial converters on your board. If you are using a RedBoard you will need to go to theHow to Install FTDI Drivers tutorial. If you are using the or RedBoard Qwiic, you will need to go to theHow to Install CH340 Drivers tutorial. Make sure to look closely at your board and its respective hookup guide to determine USB-to-serial converter that is on board.

How to Install FTDI Drivers

June 4, 2013

How to install drivers for the FTDI Basic on Windows, Mac OS X, and Linux.
Favorited Favorite 12

How to Install CH340 Drivers

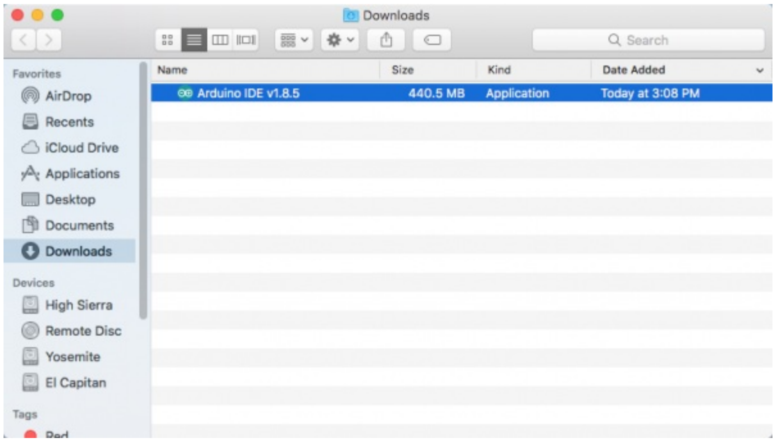
August 6, 2019

How to install CH340 drivers (if you need them) on Windows, Mac OS X, and Linux.
Favorited Favorite 11

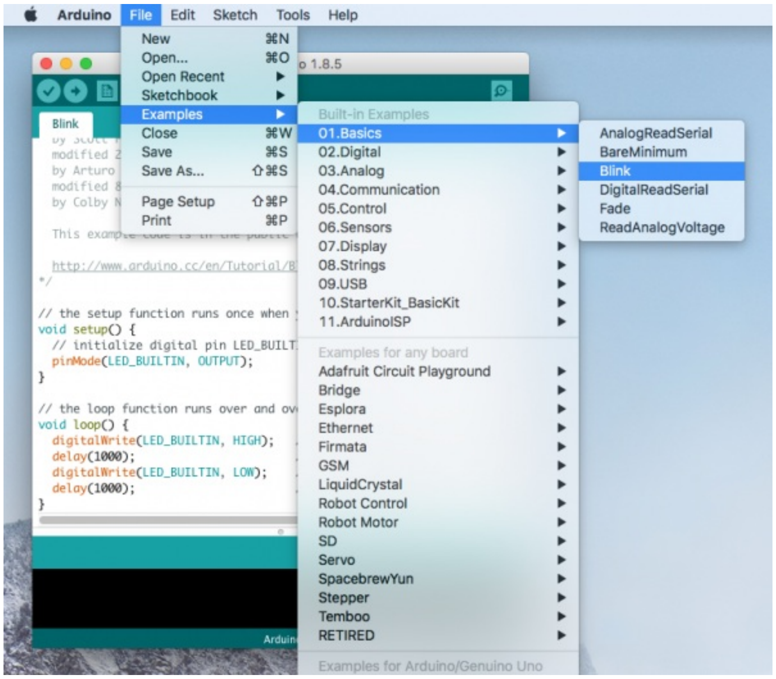
Launch and Blink!

After following the appropriate steps for your software install, we are now ready to test your first program with your Arduino board!

Launch the Arduino application wherever the program is located. If you disconnected your board, plug it back in.

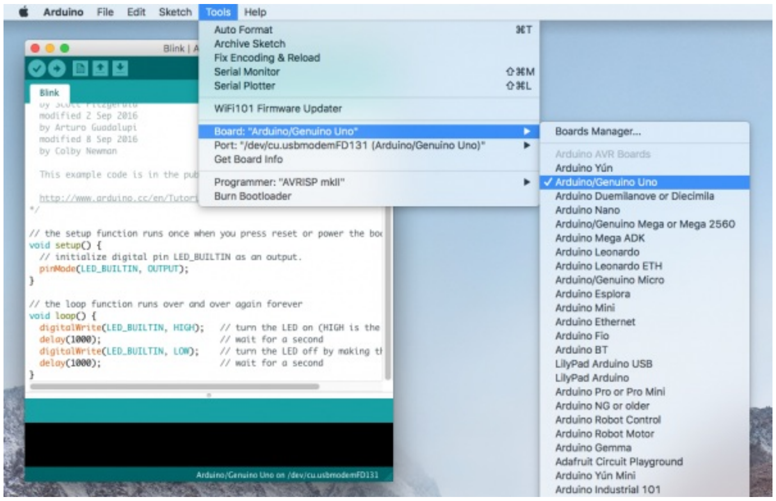


Open the Blink example sketch by going to:**File** > **Examples** > **01.Basics** > **Blink**.



```
  by Scott F
  modified 2
  by Arturo
  modified 8
  by Colby N

  This example code is in the public

  http://www.arduino.cc/en/Tutorial/B
*/

// the setup function runs once when
void setup() {
  // initialize digital pin LED_BUILT
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and ov
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
}
```

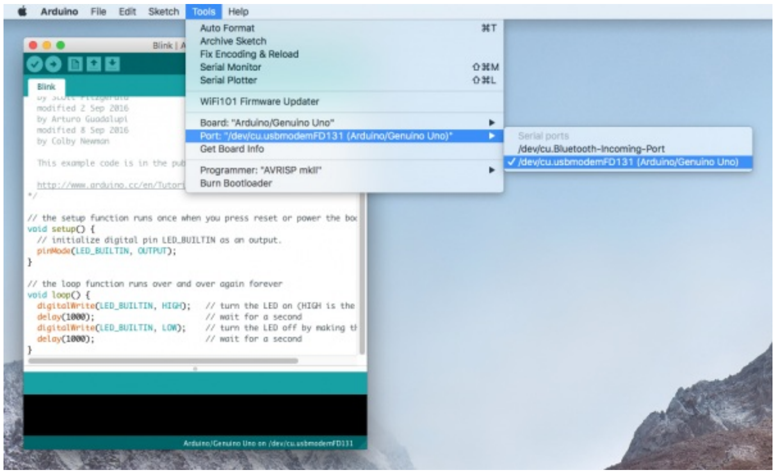
Select the type of Arduino board you're using:**Tools** > **Board** > **Arduino Uno**.



```
  by Scott Fitzgerald
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the pub

  http://www.arduino.cc/en/Tutori
*/

// the setup function runs once when you press reset or power the boa
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making th
  delay(1000);                       // wait for a second
}
```

Note: As you move to other architectures, you may need to select a different board definition depending on your development board. For the Arduino Uno R3 and RedBoard development boards with ATmega328P, you can simply select **Arduino Uno**. Certain Arduino IDE versions may have you select **Arduino/ Genuino Uno**.

Select the serial port that your Arduino is attached to:**Tools** > **Port** > **xxxxxx**. It'll probably look something like "*/dev/tty.usbmodemfd131*" or "*/dev/tty.usbserial-131*" but probably with a different number. In this case, it was*/dev/cu.usbmodemFD131*.



Board: "Arduino/Genuino Uno"

Port: "/dev/cu.usbmodemFD131 (Arduino/Genuino Uno)"

✓ /dev/cu.usbmodemFD131 (Arduino/Genuino Uno)

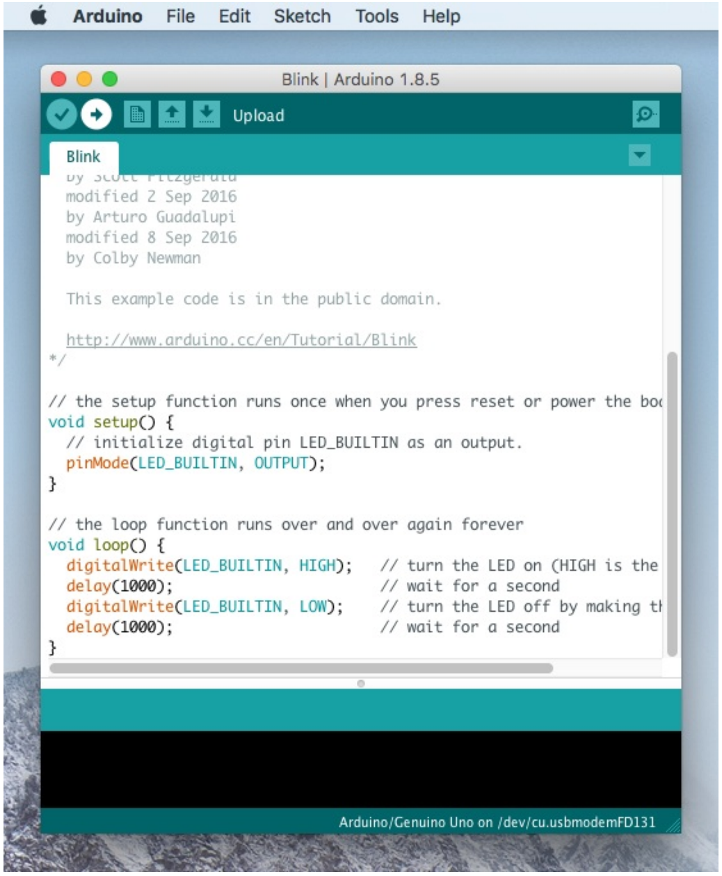
Programmer: "AVRISP mkII"

```
  by Scott Fitzgerald
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the pub

  http://www.arduino.cc/en/Tutori
*/

// the setup function runs once when you press reset or power the boc
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making th
  delay(1000);                       // wait for a second
}
```

Arduino/Genuino Uno on /dev/cu.usbmodemFD131

Note: If you're not sure which serial device is your Arduino, take a look at the available ports, then unplug your Arduino and look again. The one that disappeared is your Arduino.

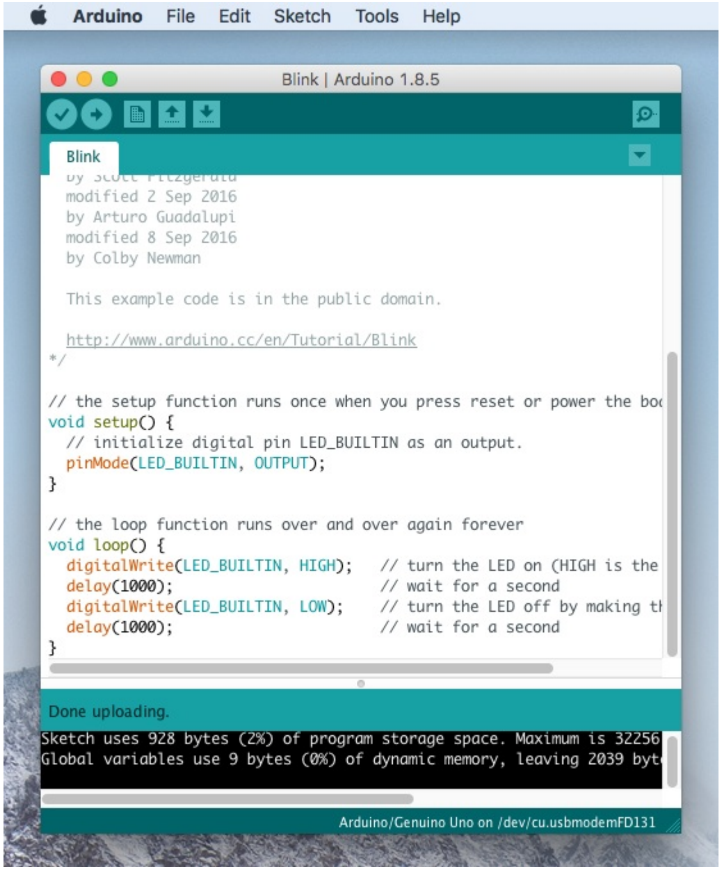
With your Arduino board connected and the Blink sketch open, press the **Upload**" button.



```
  by Scott Fitzgerald
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the boc
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making th
  delay(1000);                       // wait for a second
}
```

Arduino/Genuino Uno on /dev/cu.usbmodemFD131

After a second, you should see some LEDs flashing on your Arduino, followed by the message **Done Uploading**" in the status bar of the Blink sketch.



```
  by Scott Fitzgerald
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the boo
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making th
  delay(1000);                       // wait for a second
}
```

Done uploading.

```
Sketch uses 928 bytes (2%) of program storage space. Maximum is 32256
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 byt
```

Arduino/Genuino Uno on /dev/cu.usbmodemFD131

If everything worked, the onboard LED on your Arduino should now be blinking! You just programmed your first Arduino!

Note: Depending on the architecture and development board, the built-in LED may be defined on a different pin. You may need to adjust LED_BUILTIN or pin 13 to a different value before uploading.

Troubleshooting

If you're having problems, check out this troubleshooting guide from Arduino.

Linux

If you are a Linux user, you probably know that there are many different distribution 'flavors' of Linux out there. Unsurprisingly, installing Arduino is slightly different for many of these distributions. Luckily, the Arduino community has done an excellent job of providing instructions for most of the popular versions. Click on the link below that covers your flavor of Linux:

- ArchLinux
- Debian
- Fedora
- Gentoo
- MEPIS
- Mint
- openSUSE
- Puppy
- Pussy
- Slackware
- Ubuntu
- Xandros (Debian derivative) on Asus Eee PC

If the above directions did not work for you, or you don't see your distribution, try this catch-all guide.

You can go to the download page and download the latest version of Arduino for Linux (there are 32-bit and 64-bit versions available) when your system is properly set up.

Click for Arduino IDE Download Page

Note: Raspberry Pi users with Raspbian installed should use the **Linux ARM** download. Do not use the command line installation process. For more information, please refer to this blog post from Arduino.

Connecting Your Arduino

Power up your Arduino by connecting your Arduino board to your computer with a USB cable (or FTDI cable if you're using an Arduino Pro). You should see the an LED labeled 'ON' light up. (this diagram shows the placement of the power LED on the UNO).

ZIP

Note: You'll need to know if your Linux distribution is running on 32-bit or 64-bit when downloading the Arduino IDE. Type the following command in the serial terminal to determine what version your OS is running on.

```
uname -a
```

If you receive an output with x86_64, this indicates that it is a 64-bit OS.

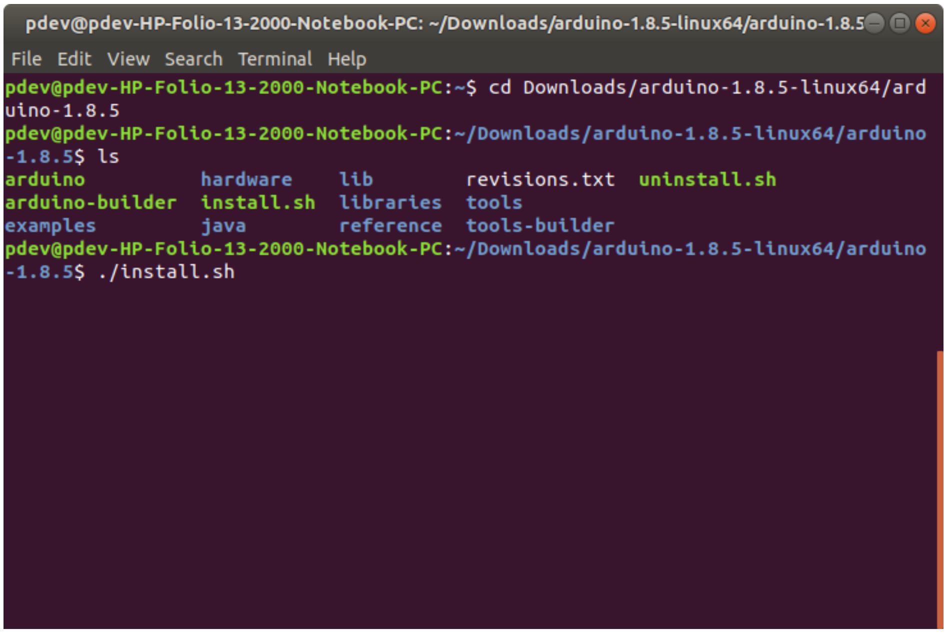
Once you download the zip file version of Arduino, you'll need to extract the files yourself. Don't forget which folder you extract the files into! Once downloaded, open a serial terminal and navigate to the Arduino program folder that was unzipped using the cd command and the folder path. In this case, we downloaded Arduino IDE v1.8.5, 64-bit version for the Ubuntu distribution. The path and folders may be different depending on the version that is downloaded. You may need to use the ls command to navigate.

```
language:bash
cd /Downloads/arduino-1.8.5-linux64/arduino-1.8.5
```

Once you are in the Arduino program folder, you'll need to enter the following command in the terminal window to install.

```
language:bash
./install.sh
```

At this point, your terminal window will probably look like the following.


pdev@pdev-HP-Folio-13-2000-Notebook-PC: ~/Downloads/arduino-1.8.5-linux64/arduino-1.8.5
File Edit View Search Terminal Help

```
pdev@pdev-HP-Folio-13-2000-Notebook-PC:~$ cd Downloads/arduino-1.8.5-linux64/ard
uino-1.8.5
pdev@pdev-HP-Folio-13-2000-Notebook-PC:~/Downloads/arduino-1.8.5-linux64/arduino
-1.8.5$ ls
arduino          hardware    lib        revisions.txt  uninstall.sh
arduino-builder  install.sh  libraries  tools
examples         java        reference  tools-builder
pdev@pdev-HP-Folio-13-2000-Notebook-PC:~/Downloads/arduino-1.8.5-linux64/arduino
-1.8.5$ ./install.sh
```

When finished, the Arduino IDE will be installed on the desktop!

Drivers for RedBoard on Linux

For the Arduino Uno, you do not need to install drivers for Linux! You may need to change the COM port permissions which will be explained further below. You can skip on down and try uploading a blink sketch!

Drivers for RedBoard on Linux

Depending on your board, there may be different USB-to-serial converters on your board. If you are using a RedBoard you will need to go to the How to Install FTDI Drivers tutorial. If you are using the or RedBoard Qwiic, you will need to go to the How to Install CH340 Drivers tutorial. Make sure to look closely at your board and its respective hookup guide to determine USB-to-serial converter that is on board.

How to Install FTDI Drivers

June 4, 2013

How to install drivers for the FTDI Basic on Windows, Mac OS X, and Linux.
Favorited Favorite 12

How to Install CH340 Drivers

August 6, 2019

How to install CH340 drivers (if you need them) on Windows, Mac OS X, and Linux.

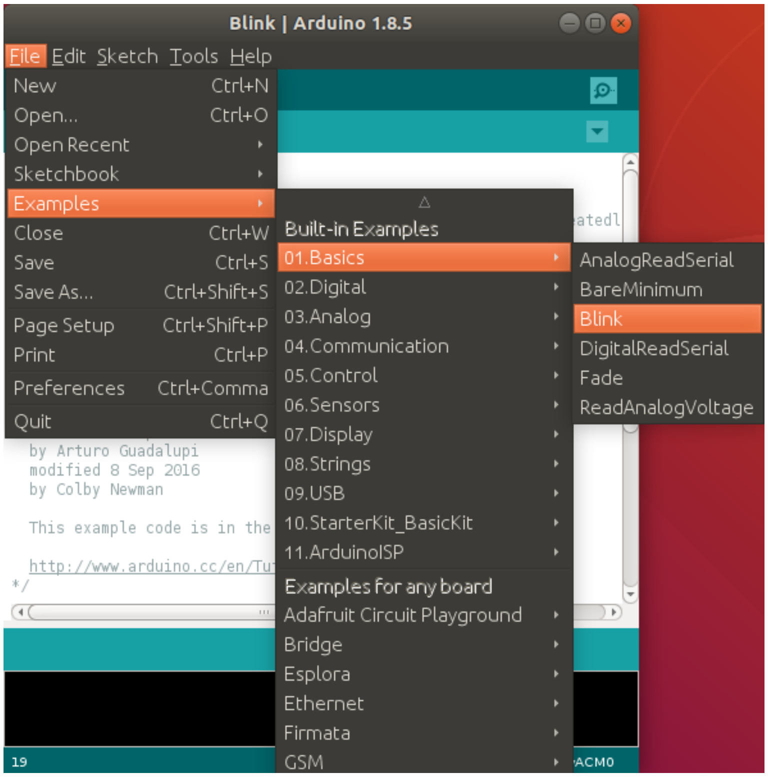
Favorited Favorite 11

Launch and Blink!

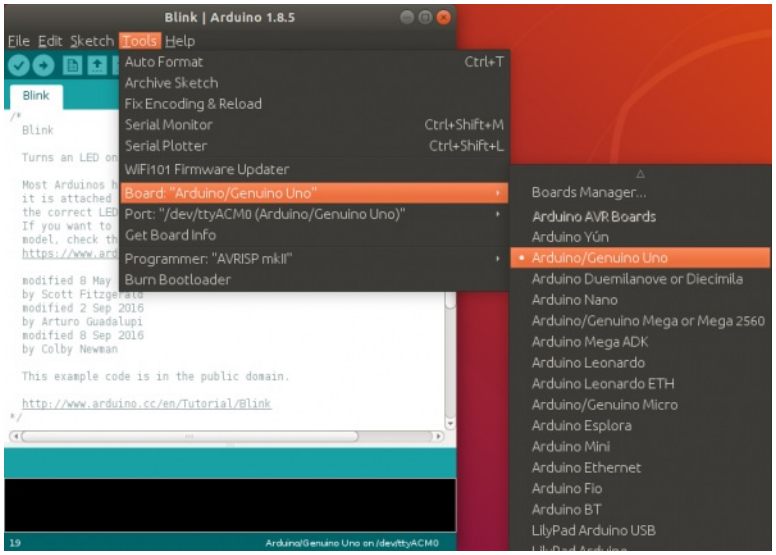
After following the appropriate steps for your software install, we are now ready to test your first program with your Arduino board!

Launch the Arduino application. If you disconnected your board, plug it back in.

Open the Blink example sketch by going to:**File** > **Examples** > **01.Basics** > **Blink**.

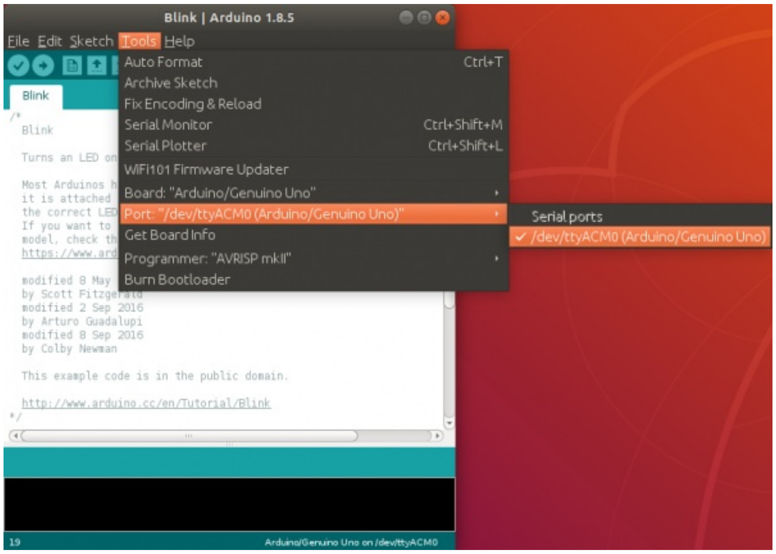


Select the type of Arduino board you're using:**Tools** > **Board** > **Arduino Uno**.



Note: As you move to other architectures, you may need to select a different board definition depending on your development board. For the Arduino Uno R3 and RedBoard development boards with ATmega328P, you can simply select **Arduino Uno**. Certain Arduino IDE versions may have you select **Arduino/ Genuino Uno**.

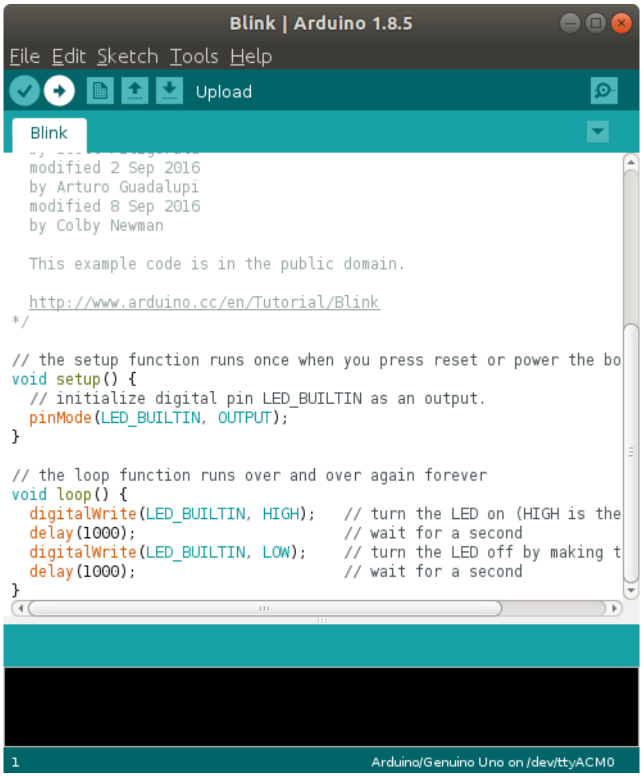
Select the serial port that your Arduino is attached to:**Tools** > **Port** > **xxxxxx** (it'll probably look something like *"/dev/ttyACM0*" but probably with a different number)


Blink | Arduino 1.8.5
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "/dev/ttyACM0 (Arduino/Genuino Uno)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
✓ /dev/ttyACM0 (Arduino/Genuino Uno)
Blink
19
Arduino/Genuino Uno on /dev/ttyACM0

```
/*
  Blink

  Turns an LED on

  Most Arduinos h
  it is attached
  the correct LED
  If you want to
  model, check th
  https://www.ard

  modified 8 May
  by Scott Fitzgerald
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
*/
```

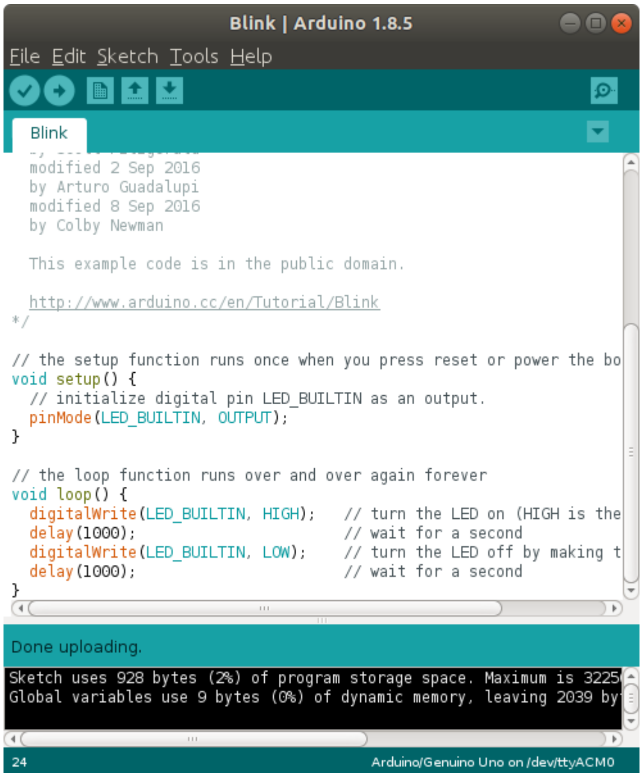
Note: If you're not sure which serial device is your Arduino, take a look at the available ports, then unplug your Arduino and look again. The one that disappeared is your Arduino.

With your Arduino board connected and the Blink sketch open, press the **Upload**" button.


Blink | Arduino 1.8.5
File Edit Sketch Tools Help
Upload
Blink
1
Arduino/Genuino Uno on /dev/ttyACM0

```
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the bo
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making t
  delay(1000);                       // wait for a second
}
```

After a second, you should see some LEDs flashing on your Arduino, followed by the message **Done Uploading**" in the status bar of the Blink sketch.



```
  by Scott Fitzgerald
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the bo
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making t
  delay(1000);                       // wait for a second
}
```

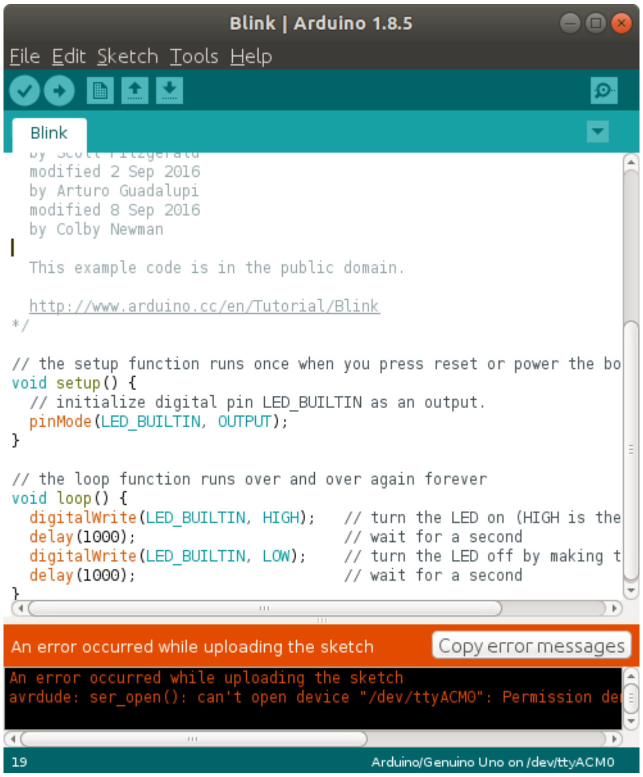
Done uploading.

```
Sketch uses 928 bytes (2%) of program storage space. Maximum is 3225
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 by
```

Troubleshooting Tip: Having trouble uploading? If you receive the following error when uploading, this could bedue to a few reasons. Most likely there is something with the user permissions if you have not used the Arduino Uno on your computer before.

```
avrdude: ser_open(): can't open device "/dev/ttyACM0" Permission denied Problem uploading to board.
```

You'll see the output at the bottom of the Arduino IDE.



```
  by Scott Fitzgerald
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the bo
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making t
  delay(1000);                       // wait for a second
}
```

An error occurred while uploading the sketch

```
An error occurred while uploading the sketch
avrdude: ser_open(): can't open device "/dev/ttyACM0": Permission de
```

Try adjusting the permissions by opening a terminal window.

```
ls -l /dev/ttyACM*
```

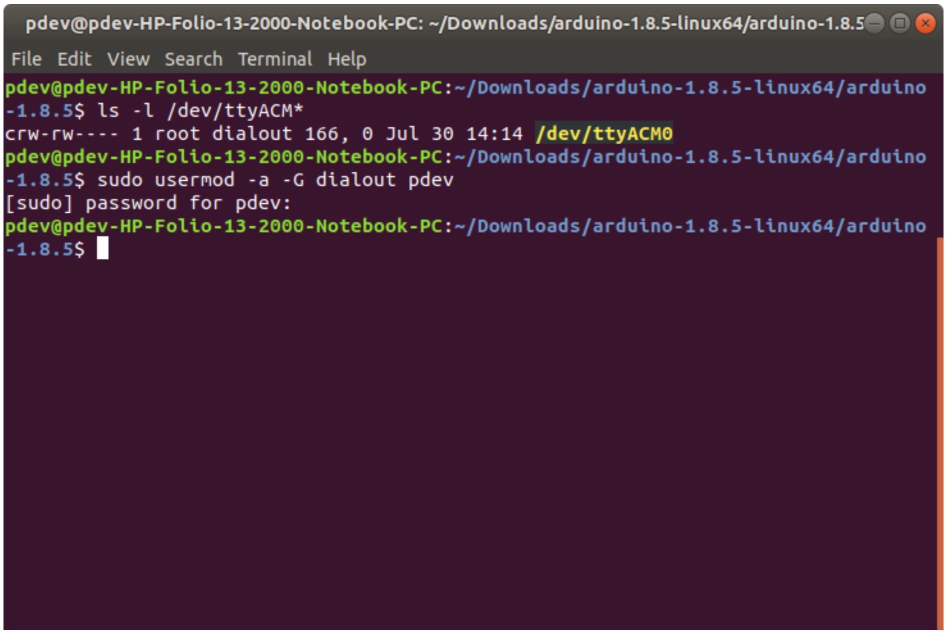
You'll get something similar to the following output.

```
crw-rw---- 1 root dialout 188, 0 Jul 30 14:14 /dev/ttyACM0
```

Then type the following command where the *<username>* is your Linux account user. Make sure to enter your password to change the settings.

```
sudo usermod -a -G dialout <username>
```

You'll probably see something similar to the output below with a different username. In this case, our username was "pdev."


pdev@pdev-HP-Folio-13-2000-Notebook-PC: ~/Downloads/arduino-1.8.5-linux64/arduino-1.8.5
File Edit View Search Terminal Help

```
pdev@pdev-HP-Folio-13-2000-Notebook-PC:~/Downloads/arduino-1.8.5-linux64/arduino
-1.8.5$ ls -l /dev/ttyACM*
crw-rw---- 1 root dialout 166, 0 Jul 30 14:14 /dev/ttyACM0
pdev@pdev-HP-Folio-13-2000-Notebook-PC:~/Downloads/arduino-1.8.5-linux64/arduino
-1.8.5$ sudo usermod -a -G dialout pdev
[sudo] password for pdev:
pdev@pdev-HP-Folio-13-2000-Notebook-PC:~/Downloads/arduino-1.8.5-linux64/arduino
-1.8.5$ 
```

You'll need to restart your Linux for the changes to take effect. Since we are already in the terminal window, type the following command to restart your computer. Of course, you can use the GUI as well to restart if you prefer.

```
sudo reboot
```

If everything worked, the onboard LED on your Arduino should now be blinking! You just programmed your first Arduino!

Note: Depending on the architecture and development board, the built-in LED may be defined on a different pin. You may need to adjust LED_BUILTIN or pin 13 to a different value before uploading.

Troubleshooting

The Arduino Playground Linux section is a great resource for figuring out any problems with your Arduino installation.

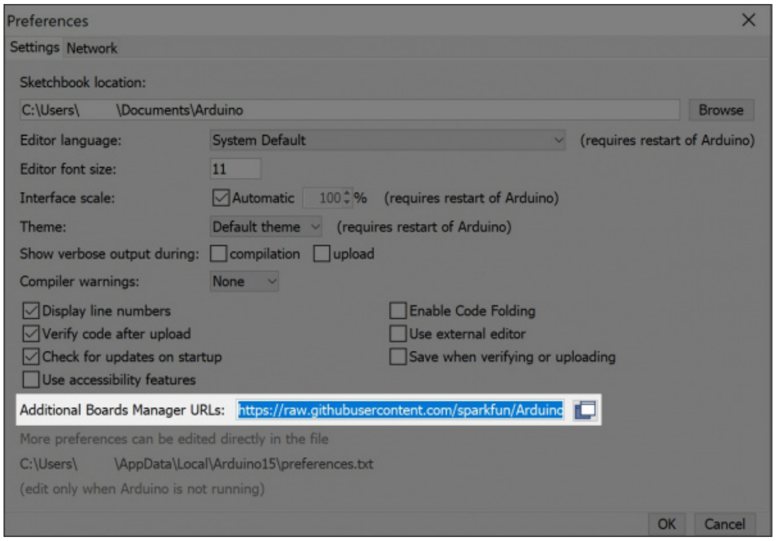
Board Add-Ons with Arduino Board Manager

With Arduino v1.6.4+, a new boards manager feature makes it easy to add third-party boards (like the SparkFun RedBoard, Digital Sandbox, and RedBot) to the Arduino IDE.

GitHub: SparkFun Arduino Boards

To start, highlight and copy (CTRL + C / CMD + C) the text below for the boards manager URL. You'll need this to configure Arduino.

```
language:bash
https://raw.githubusercontent.com/sparkfun/Arduino_Boards/main/IDE_Board_Manager/package_sparkfun_index.json
```



Preferences

Settings | Network

Sketchbook location:

C:\Users\ \Documents\Arduino Browse

Editor language: System Default (requires restart of Arduino)

Editor font size: 11

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Theme: Default theme (requires restart of Arduino)

Show verbose output during: □ compilation □ upload

Compiler warnings: None

☑ Display line numbers □ Enable Code Folding

☑ Verify code after upload □ Use external editor

☑ Check for updates on startup □ Save when verifying or uploading

□ Use accessibility features

Additional Boards Manager URLs: https://raw.githubusercontent.com/sparkfun/Arduino

More preferences can be edited directly in the file

C:\Users\ \AppData\Local\Arduino15\preferences.txt

(edit only when Arduino is not running)

OK Cancel

Heads Up: GitHub has recently moved away from having default branches as "master" and changed the default to "main". SparkFun has followed this move as well. If you're unable to see a particular board inside of Arduino it may be that you are using the older "master" link. As a result, you will only see the previous changes on the old master branch. Please change to the 'main' json url listed above and all should be well.

Open up Arduino:

- *Configure the Boards Manager*
  - For Windows and Linux, head to **File > Preferences > Additional Boards Manager URLs** and paste (CTRL + V / CMD + V) the link
  - For Macs, head to **Arduino > Preferences > Additional Boards Manager URLs** and paste (CTRL + V / CMD + V) the link
- Click on **Tools > Board > Boards Manager...**
- Select the Type as "**Contributed**" from the drop down menu.
- Click on the **SparkFun AVR Boards** and then click **Install**.

That's it! Boards are all installed. This also gives you access to all of our library files as well through the built-in Library Manager tool in Arduino. Looking for more information about adding other custom boards? Check out the the following tutorial to install other Arduino cores

Installing Board Definitions in the Arduino IDE

September 9, 2020

How do I install a custom Arduino board/core? It's easy! This tutorial will go over how to install an Arduino board definition using the Arduino Board Manager. We will also go over manually installing third-party cores, such as the board definitions required for many of the SparkFun development boards.
Favorited Favorite 3

Resources and Going Further

Now that you've got the Arduino software installed on your system and tested it with your Arduino board successfully, you're ready for your next steps into the world of embedded electronics. If you want to learn about some of the **concepts** that will help you build your projects, check out some of the following tutorials:

Installing an Arduino Library

How do I install a custom Arduino library? It's easy! This tutorial will go over how to install an Arduino library using the Arduino Library Manager. For libraries not linked with the Arduino IDE, we will also go over manually installing an Arduino library.
Favorited Favorite 23

How to Power a Project

A tutorial to help figure out the power requirements of your project.
Favorited Favorite 70

Choosing an Arduino for Your Project

Examining the diverse world of Arduino boards and understanding the differences between them before choosing one for a project.
Favorited Favorite 19

Click Here to See More Tutorials Related to Arduino

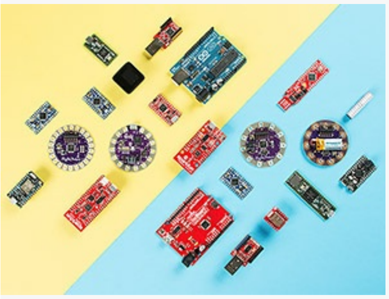
Or check out the some of the Adventures in Science videos focused on Arduino.

SparkFun: Adventures in Science

Looking for the right Arduino?

Check out our **Arduino Comparison Guide**! We've compiled every Arduino development board we carry, so you can quickly compare them to find the perfect one for your needs.

Take me there!



If you'd rather jump right in to **building** something, check out these links to projects here on learn as well as some other places to find Arduino-based projects:

- Arduino Project Hub
- WorkBench: SparkFun
- Instructables: Arduino
- Hackster.io: Arduino
- The ITP Physical Computing Wiki
- Make: Projects

Or any of our tutorials tagged with Arduino.

Experiment Guide for RedBot with Shadow Chassis

This Experiment Guide offers nine experiments to get you started with the SparkFun RedBot. This guide is designed for those who are familiar with our SparkFun Inventor's Kit and want to take their robotics knowledge to the next level.
Favorited Favorite 11

Qwiic Joystick Hookup Guide

Looking for an easy way to implement a joystick to your next Arduino or Raspberry Pi project? This hookup guide will walk you through using the Qwiic Joystick with the Arduino IDE on a RedBoard Qwiic and in Python on a Raspberry Pi.
Favorited Favorite 4

Basic Servo Control for Beginners

An introductory tutorial demonstrating several ways to use and interact with servo motors!
Favorited Favorite 19

SparkFun Air Quality Sensor - SGP30 (Qwiic) Hookup Guide

A hookup guide to get started with the SparkFun Air Quality Sensor - SGP30 (Qwiic).
Favorited Favorite 1