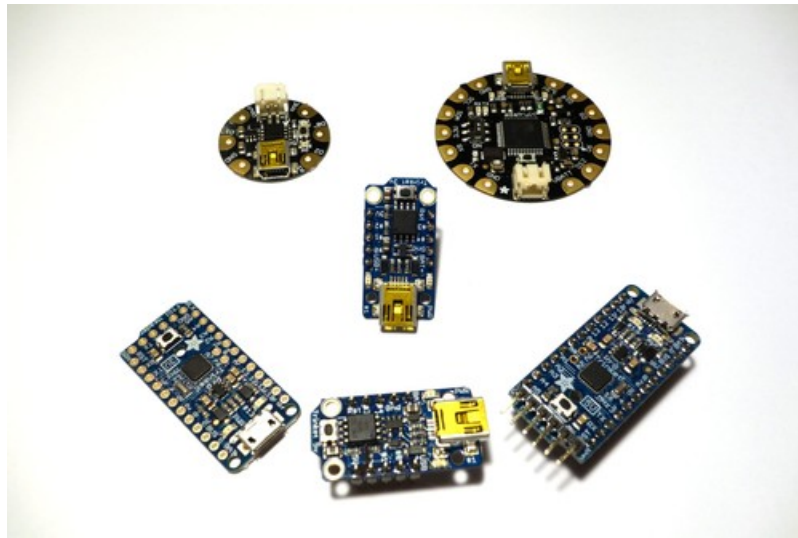




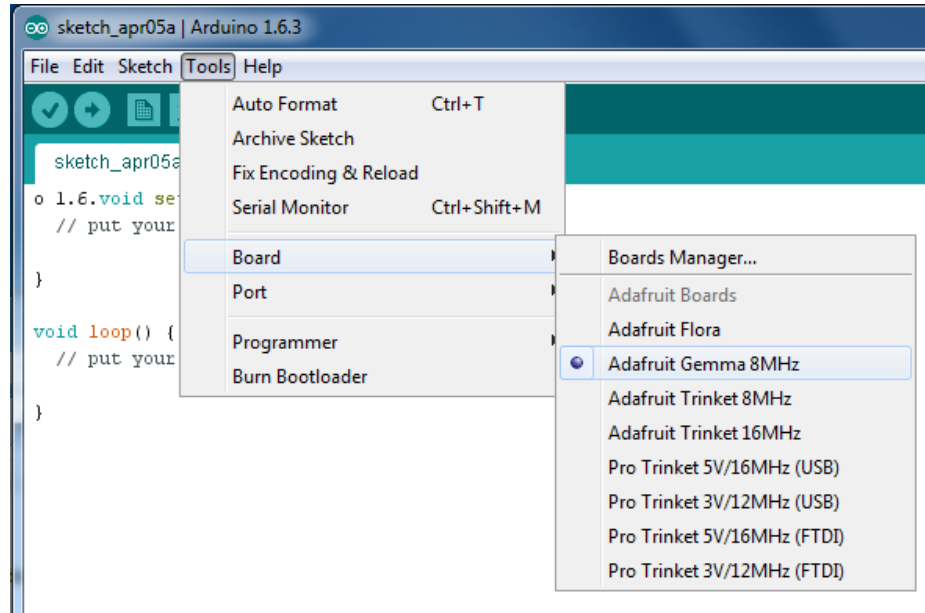
Adafruit Arduino IDE Setup

Created by Tony DiCola



Last updated on 2019-09-05 09:36:27 PM UTC

Overview



This guide will walk you through how to install the Arduino IDE to work with Adafruit's boards like Trinket, Pro Trinket, Gemma, and Flora. It's easy to program these boards by downloading and installing a preconfigured version of the Arduino IDE. You'll be up and running with your Adafruit board in minutes!

For most users follow the [Arduino 1.6.x IDE install page \(https://adafru.it/jDQ\)](https://adafru.it/jDQ) to install a version of the latest Arduino IDE.

For users who need an older version of the Arduino IDE follow the [Arduino 1.0.x IDE install page \(https://adafru.it/jMe\)](https://adafru.it/jMe).

If you aren't sure what Arduino version you need stick with the latest [1.6.x instructions \(https://adafru.it/jDQ\)](https://adafru.it/jDQ)!

And if you're new to using Arduino be sure to check out this [Arduino tips and tricks guide \(https://adafru.it/maj\)](https://adafru.it/maj), and the [official Arduino documentation \(https://adafru.it/aMH\)](https://adafru.it/aMH).

Arduino 1.6.x IDE

The latest version of Arduino is recommended for most users. Follow the steps on this page to learn how to install a version of the Arduino IDE with support for Adafruit's boards.

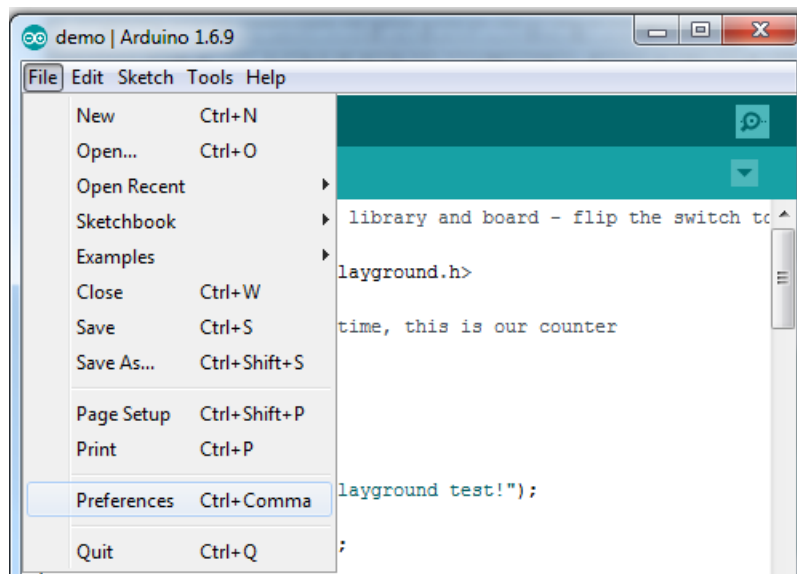
Super Easy Installation (Recommended)

With the latest Arduino IDE (version 1.6.4 and greater) you can now add 3rd party boards directly from the stock IDE! The first thing you will need to do is to download the latest release of the Arduino IDE. You will need to be using **version 1.6.4** or higher:

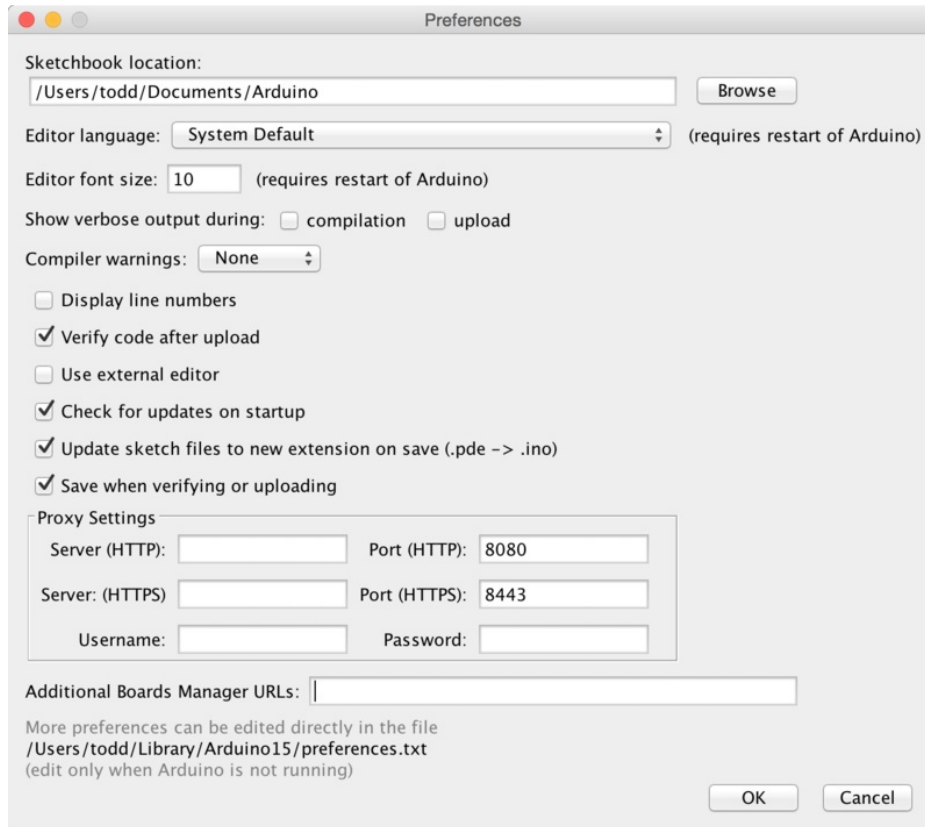
<https://adafru.it/f1P>

<https://adafru.it/f1P>

After you have downloaded and installed the latest version (remember it **must** be 1.6.4 or higher), you will need to start the IDE and navigate to the **Preferences** menu. You can access it from the **File** menu in *Windows* or *Linux*, or the **Arduino** menu on *OS X*.



A dialog will pop up just like the one shown below.



We will be adding a URL to the new **Additional Boards Manager URLs** option. The list of URLs is comma separated, and *you will only have to add each URL once*. New Adafruit boards and updates to existing boards will automatically be picked up by the Board Manager each time it is opened. The URLs point to index files that the Board Manager uses to build the list of available & installed boards.

To find the most up to date list of URLs you can add, you can visit the list of [third party board URLs on the Arduino IDE wiki \(https://adafru.it/f7U\)](https://adafru.it/f7U). We will only need to add one URL to the IDE in this example, but you can add multiple URLs by separating them with commas. Copy and paste the link below into the **Additional Boards Manager URLs** option in the Arduino IDE preferences.





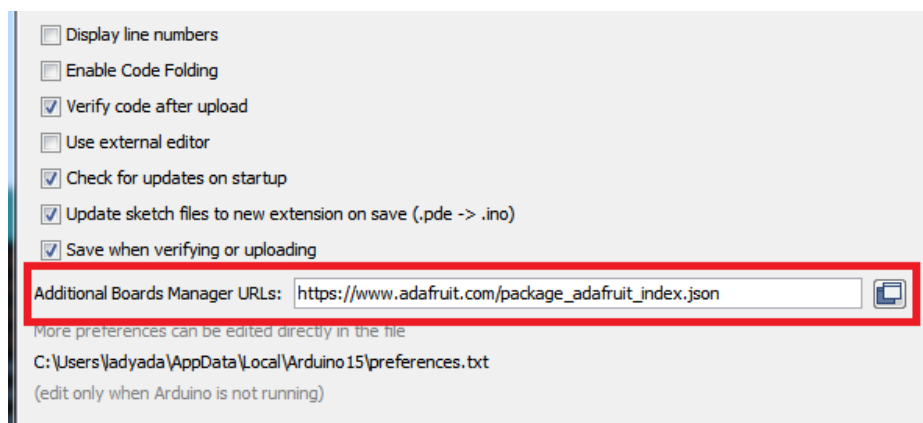
If you don't see the Additional Boards Manager URLs box, make sure you downloaded the Arduino IDE from arduino.cc! Older versions and derivatives of the IDE may not have it

Add the Adafruit Board Support package!

Paste

https://adafruit.github.io/arduino-board-index/package_adafruit_index.json

Into the "Additional Board Managers URLs" box

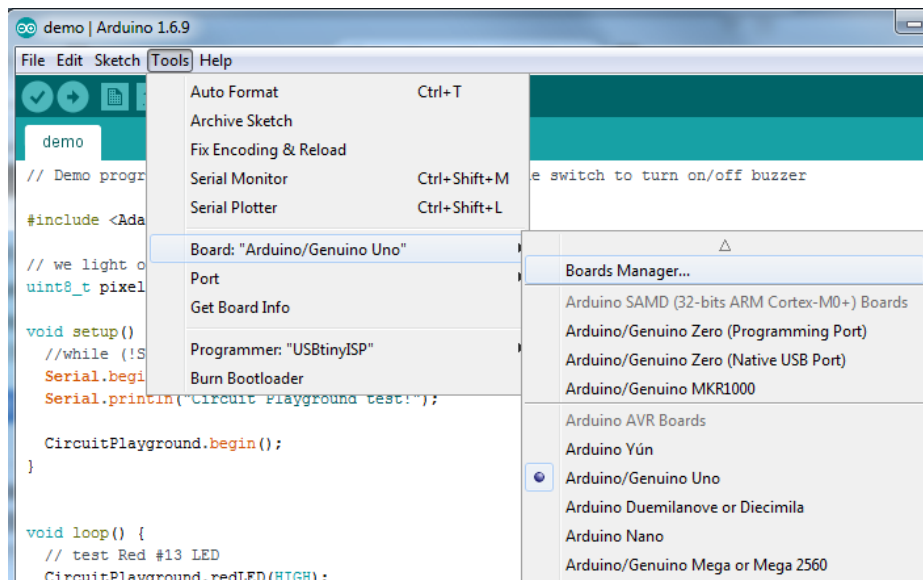


Click **OK** to save the new preference settings. Next we will look at installing boards with the Board Manager.

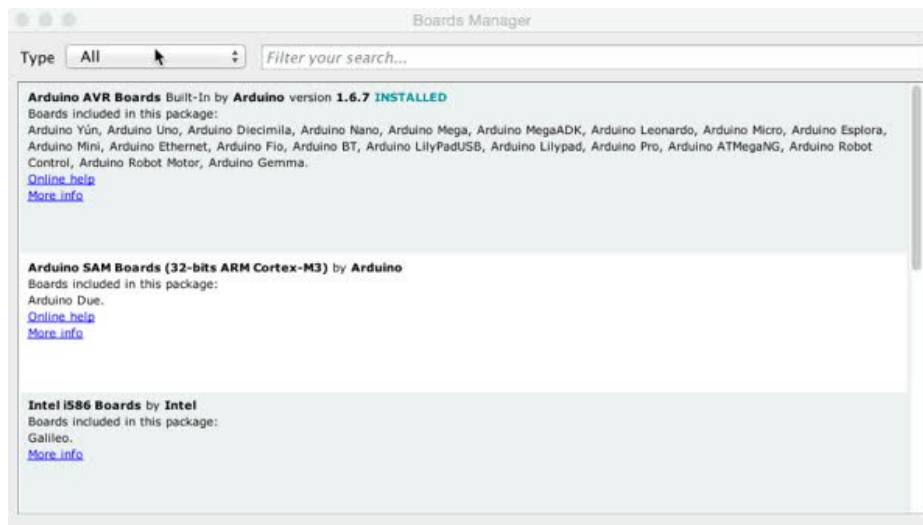
Manage Board Support

Adding the link to the Adafruit board support package does not actually install anything, it only tells the Arduino IDE where to find the software.

Now that you have added the appropriate URLs to the Arduino IDE preferences, you can open the **Boards Manager** by navigating to the **Tools->Board** menu.



Once the Board Manager opens, click on the category drop down menu on the top left hand side of the window and select **Contributed**. You will then be able to select and install the boards supplied by the URLs added to the preferences. In the example below, we are installing support for *Adafruit AVR Boards*, but the same applies to all boards installed with the Board Manager.



Next, **quit and reopen the Arduino IDE** to ensure that all of the boards are properly installed. You should now be able to see the new boards listed in the **Tools->Board** menu.

Finally follow the steps below for your platform to finish the installation - basically installing drivers and permissions management

- [Windows Setup \(https://adafru.it/jAa\)](https://adafru.it/jAa)

- [Mac OSX Setup \(https://adafru.it/mak\)](https://adafru.it/mak)
- [Linux Setup \(https://adafru.it/iOE\)](https://adafru.it/iOE)

Also check out the [troubleshooting section \(https://adafru.it/mal\)](https://adafru.it/mal) for some advice on common errors. Have fun using Adafruit's boards!

Manual Installation (super advanced!)

If you have your own version of the Arduino IDE or would like to install the Adafruit boards yourself then follow the steps below. However it is **highly recommended** that most users stick to the initial methods

With the new version of the Arduino IDE the process of adding support for new boards is much simpler than previous versions. First start by downloading a version of the Arduino 1.6.x IDE and installing it as normal. **Note that these instructions are only tested to work against the Arduino 1.6.4+ IDE, earlier or later versions may or may not work!**

Once you've installed the stock Arduino IDE download and unzip the following file which contains [Adafruit's board definitions \(https://adafru.it/eTY\)](https://adafru.it/eTY):

<https://adafru.it/eTZ>

<https://adafru.it/eTZ>

Unzip the file and navigate inside the **Adafruit_Arduino_Boards** folder to find a **hardware** subfolder with a small hierarchy of files, and a **drivers** folder with Flora drivers specific to Windows.

Next find your installed Arduino IDE's **hardware** subfolder. For Windows and Linux the hardware subfolder should be directly beneath the folder where Arduino's IDE was installed.

For Mac OSX the folder is hidden inside the application bundle. Right click on the Arduino application and click '**Show Package Contents**', then navigate to the **Content -> Java -> Hardware** folder to find the hardware subfolder.

Once you've found Arduino's hardware subfolder carefully merge in the contents of the hardware folder from Adafruit's board definitions, being sure to overwrite any file that conflicts (only avrdude.conf should conflict). When you're done the Arduino hardware folder should have the following hierarchy:

- hardware
 - adafruit
 - avr
 - The contents of the hardware/adafruit/avr folder from Adafruit's board definitions. There will be a boards.txt, platform.txt and variants subfolder.
 - arduino
 - No changes to this hierarchy, it is exactly as provided by the Arduino IDE.
 - tools
 - avr
 - etc
 - **avrdude.conf** - This version of avrdude.conf should be from Adafruit's board definitions and completely overwrites the version provided by the Arduino IDE.

- ... all other files as provided by the Arduino IDE.
- ... all other files as provided by the Arduino IDE.

Finally on Windows only, copy the contents of the drivers folder from the board definitions download into the Arduino IDE's drivers folder (it's a sibling of the hardware folder you found earlier).

That's all you need to do to manually install Adafruit's boards with Arduino! Now follow the steps below for your platform to finish the installation:

- [Windows Setup \(https://adafru.it/jAa\)](https://adafru.it/jAa)
- [Mac OSX Setup \(https://adafru.it/mak\)](https://adafru.it/mak)
- [Linux Setup \(https://adafru.it/iOE\)](https://adafru.it/iOE)

Also check out the [troubleshooting section \(https://adafru.it/mal\)](https://adafru.it/mal) for some advice on common errors. Have fun using Adafruit's boards!

Arduino 1.0.x IDE

The Arduino IDE version 1.0.x is an older version of the Arduino IDE and is provided for users who might require the older IDE for compatibility or other reasons. For most users the newer 1.6.x version is recommended!

Easy Installation

The easiest way to install the Arduino 1.0.x version IDE with support for Adafruit's boards is with the following preconfigured downloads. Just grab the right file for your platform and use it like the normal Arduino IDE. Adafruit's boards like Trinket, Pro Trinket, Gemma, and Flora are configured to show up in the board list automatically!

<https://adafru.it/sD2>

<https://adafru.it/sD2>

<https://adafru.it/sD3>

<https://adafru.it/sD3>

<https://adafru.it/dXa>

<https://adafru.it/dXa>

<https://adafru.it/dXb>

<https://adafru.it/dXb>

Once you've downloaded and installed the IDE follow the steps below for your platform to finish the installation:

- [Windows Setup \(https://adafru.it/jAa\)](https://adafru.it/jAa)
- [Mac OSX Setup \(https://adafru.it/mak\)](https://adafru.it/mak)
- [Linux Setup \(https://adafru.it/iOE\)](https://adafru.it/iOE)

Also check out the [troubleshooting section \(https://adafru.it/mal\)](https://adafru.it/mal) for some advice on common errors. Have fun using Adafruit's boards!

Manual Installation

If you have your own version of the Arduino IDE or would like to install the Adafruit boards yourself then follow the steps below. However it is **highly recommended** that most users stick to the easy install downloads above!

Step 0. Install Arduino IDE

This tutorial will base the IDE off of v1.0.5. You can try later versions but v1.0.5 is at least guaranteed to work

Adding Adafruit board support does not affect any other boards that are installed so you can continue to use the IDE with any Arduino board currently supported.

Linux note: You can find instructions for manually modifying the Arduino IDE to support Trinket/Gemma/Flora in the [README of the Trinket Arduino Linux github repository \(https://adafru.it/dXd\)](https://adafru.it/dXd).

Step 1. Add ATtiny85 Support

Download the following file by pressing the button.

<https://adafru.it/dYP>

<https://adafru.it/dYP>

Unzip it and move the **hardware** folder from the zip file and place it into to your *Arduino sketchbook* folder. Your sketchbook folder is the folder where the Arduino IDE stores your sketches. This folder is automatically created by the IDE when you install it. If this is your first time using the Arduino IDE, it will be empty!

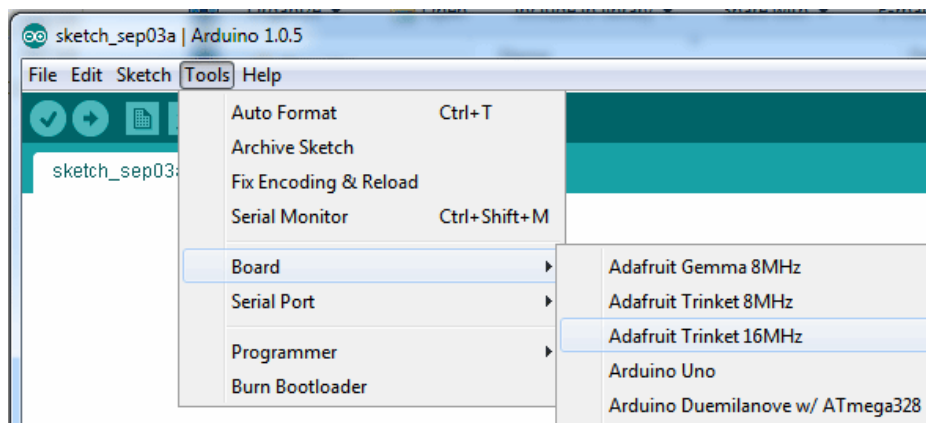
On Linux machines, the folder is named "Sketchbook" and it is typically located in /home/[username]

On Windows and Macintosh machines, the default name of the folder is "Arduino" and is located in your Documents folder.



This is a common source of confusion on Windows and Mac machines, your sketchbook folder is not named "sketchbook" it is named "Arduino"!

Now you can start (or restart the the IDE) and check the **Tools->Board** menu, you should see the three new entries for Trinket and Gemma:



OK you are half done! Next is updating the avrdude configuration file.

Step 2. Updating avrdude.conf

The second step is to update the AVR chip program upload helper to be a little more patient with the ATtiny85 bootloader we have on the Trinket. We will update the description of the chip's erase cycle to be longer, to avoid timeouts and errors.

Windows users can download the new **avrdude.conf** by clicking this button:

<https://adafru.it/eU0>

<https://adafru.it/eU0>

Mac users should download this version and uncompress it

<https://adafru.it/eU1>

<https://adafru.it/eU1>

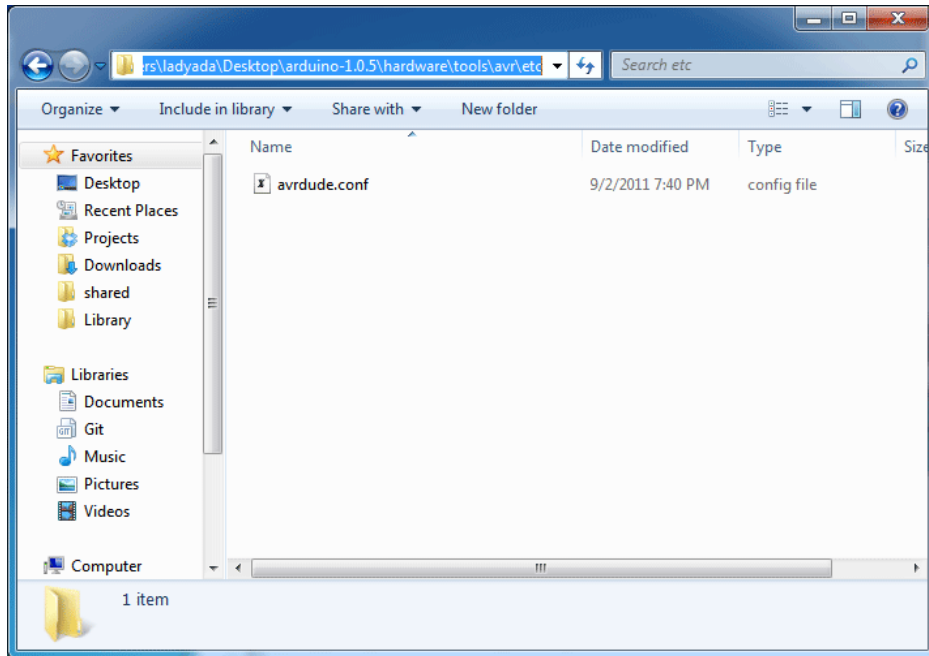
On Linux download this version of the file and rename it to just avrdude.conf:

<https://adafru.it/eU2>

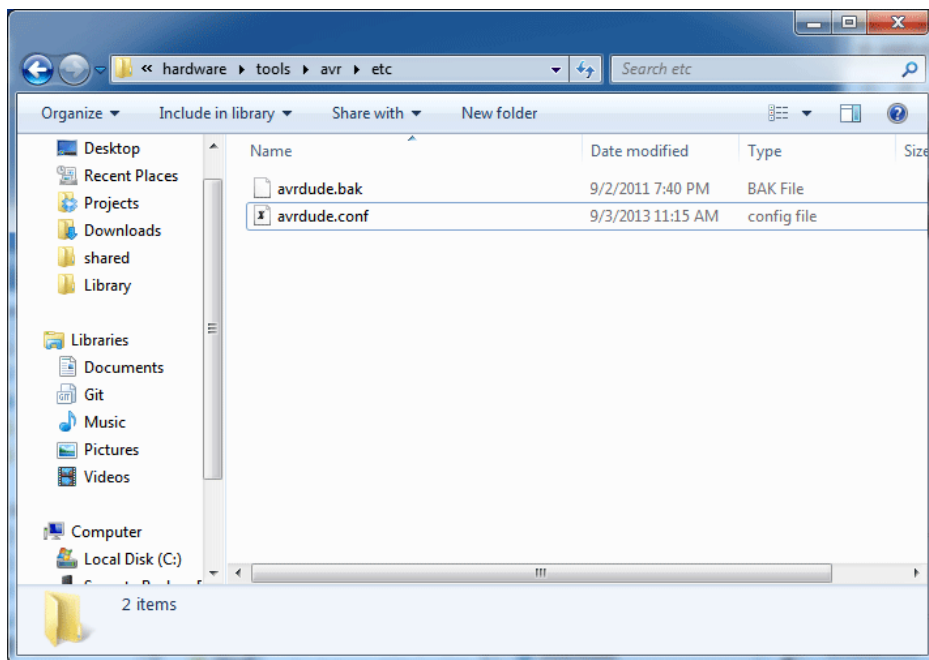
<https://adafru.it/eU2>

Now we will hunt for the original **avrdude.conf** file. If you are using a Mac, right-click on the Arduino application icon and select “Show Package Contents” then navigate through the Contents & Java folders. On Windows you will have to find the installation directory, which may be a folder on the Desktop or possibly in **C:\Program Files** if you used the installer. Likewise in Linux it is where-ever you uncompressed the folder.

Now find the **hardware\tools\avr\etc** folder, and inside you should see the **avrdude.conf** file. You can also use your operating system's **find** tool to locate it.



Rename the old `avrdude.conf` to `avrdude.bak` and copy over the *new* `avrdude.conf` to the same folder



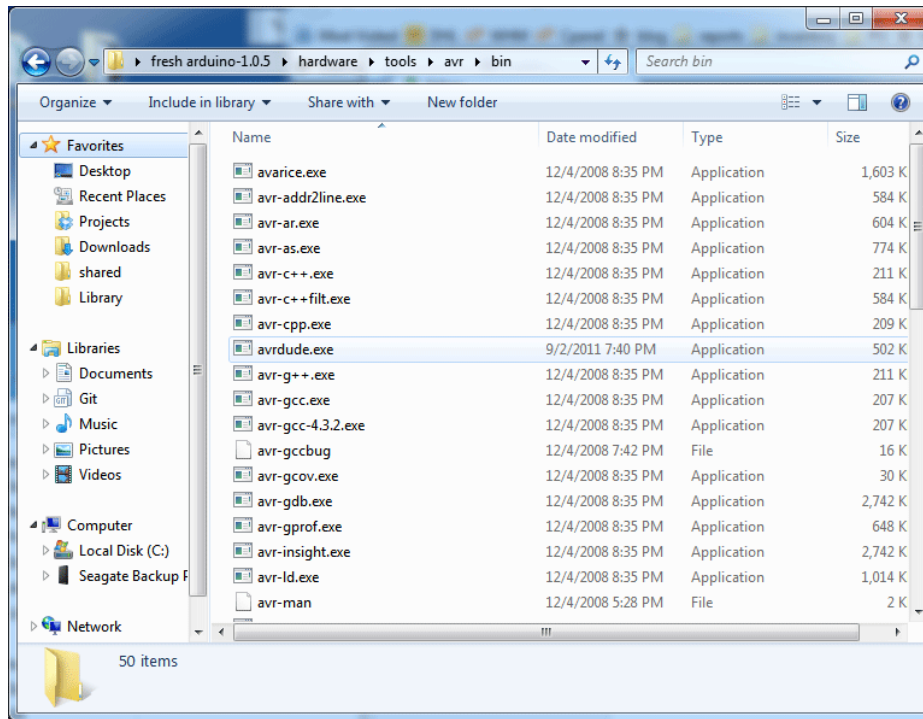
Be sure you rename the old file before replacing it with the new one so you are sure to make the swap!

Step 3. Update 'ld' linker

There's a bug in the 'linker' used by Arduino on Mac & Windows, where you can't make sketches that are larger than 4K on the Attiny85. Since its really likely you'll make sketches this large, we suggest replacing it. It's a lot like replacing the `avrdude.conf`

On Windows: explore the Arduino folder and get to the `hardware\tools\avr\bin` subfolder, you'll see a lot of files starting with `avr-xxx`.

On Mac: Explore the App and find `Adafruit Arduino.app/Contents/Resources/Java/hardware/tools/avr/avr/bin`



Download the `ld` for your operating system and replace the one you have now (thanks to TCWorld for this fix! (<https://adafru.it/sD4>))

<https://adafru.it/eU3>

<https://adafru.it/eU3>

<https://adafru.it/eU4>

<https://adafru.it/eU4>

Now restart the Arduino IDE. If using Trinket or Gemma you are done with setup! Now follow the steps below for your platform to finish the installation:

- [Windows Setup \(https://adafru.it/jAa\)](https://adafru.it/jAa)
- [Mac OSX Setup \(https://adafru.it/mak\)](https://adafru.it/mak)
- [Linux Setup \(https://adafru.it/iOE\)](https://adafru.it/iOE)

However if you're using Flora, continue on to setup the IDE for Flora.

Flora Setup for Arduino 1.0.x

Find your `boards.txt` file its in the `Arduino/hardware/arduino` folder on Windows for version 1.0 IDE

On a Mac right click the application and select "Show Package Contents," then navigate

to Contents/Resources/Java/hardware/Arduino) and add to the bottom:

```
flora8.name=Adafruit Flora
flora8.upload.protocol=avr109
flora8.upload.maximum_size=28672
flora8.upload.speed=57600
flora8.upload.disable_flushing=true
flora8.bootloader.low_fuses=0xff
flora8.bootloader.high_fuses=0xd8
flora8.bootloader.extended_fuses=0xcb
flora8.bootloader.path=caterina
flora8.bootloader.file=Caterina-Flora8.hex
flora8.bootloader.unlock_bits=0x3F
flora8.bootloader.lock_bits=0x2F
flora8.build.mcu=atmega32u4
flora8.build.f_cpu=8000000L
flora8.build.vid=0x239A
flora8.build.pid=0x8004
flora8.build.core=arduino
flora8.build.variant=flora
```

- Place in `arduino-1.0.5\drivers` (Windows only):

<https://adafru.it/eU5>

<https://adafru.it/eU5>

- Add to `hardware/arduino/variants/flora` (windows) or `hardware/arduino/avr/variants/flora` (mac) (create folder):

<https://adafru.it/eU6>

<https://adafru.it/eU6>

- Add to `hardware/arduino/bootloaders/caterina` (windows) or `hardware/arduino/avr/bootloaders/caterina` (mac)

<https://adafru.it/eU7>

<https://adafru.it/eU7>

Edit `hardware/arduino/cores/arduino/USBCore.cpp` (windows) or `hardware/arduino/avr/cores/arduino/USBCore.cpp` (mac)

find:

```
#elif USB_PID == 0x9208
  'L','i','l','y','P','a','d','U','S','B',' ',' ',' ',' ',' ',' ',' '
```

Add right below it

```
#elif USB_PID == 0x8004
  'A','d','a','f','r','u','i','t',' ',' ','F','l','o','r','a',' ',' ',' '
```


Windows Driver Installation



Mac and Linux do not require drivers, only Windows folks need to do this step

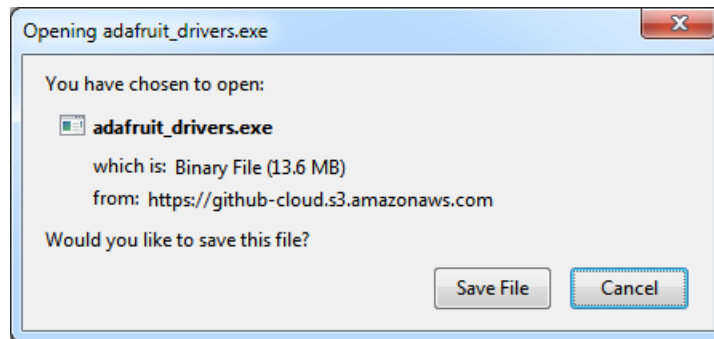
Before you plug in your board, you'll need to possibly install a driver!

Click below to download our Driver Installer.

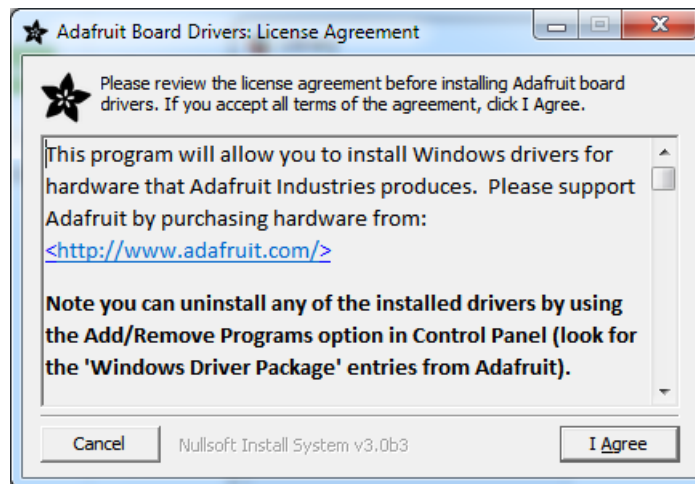
<https://adafruit.it/AB0>

<https://adafruit.it/AB0>

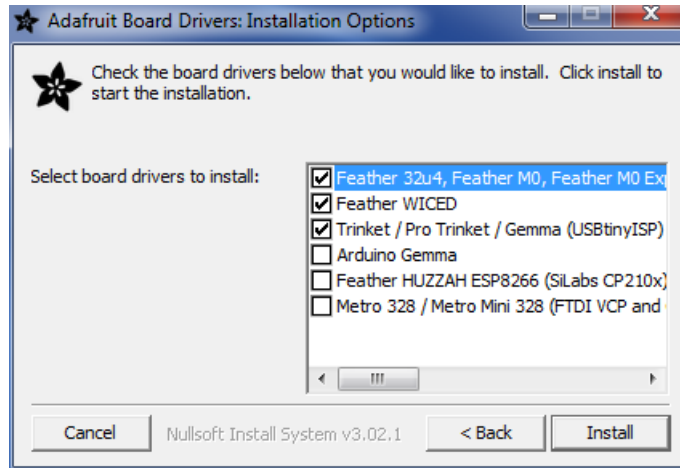
Download and run the installer.



Run the installer! Since we bundle the SiLabs and FTDI drivers as well, you'll need to click through the license



Select which drivers you want to install, we suggest selecting all of them so you don't have to do this again!



On Windows 7, by default, we install a single driver for most of Adafruit's boards, including the **Feather 32u4**, the **Feather M0**, **Feather M0 Express**, **Circuit Playground**, **Circuit Playground Express**, **Gemma M0**, **Trinket M0**, **Metro M0 Express**. On Windows 10 that driver is not necessary (it's built in to Windows) and it will not be listed.

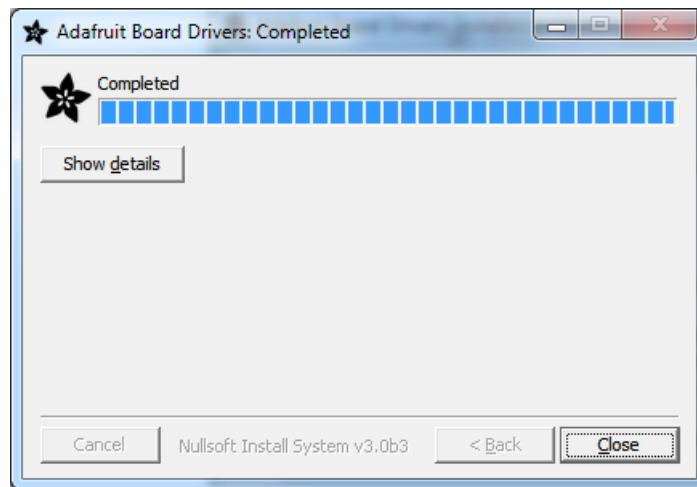
The **Trinket / Pro Trinket / Gemma / USBtinyISP** drivers are also installed by default.

You can also, optionally, install the **Arduino Gemma** (different than the Adafruit Gemma!), **Huzzah** and **Metro 328** drivers.

Click **Install** to do the installin'.



Note that on Windows 10, support for many boards is built in. If you end up not checking any boxes, you don't need to run the installer at all!



Manual Driver Installation

If windows needs the driver files (inf/cat) for some reason you can get all the drivers by downloading the source code zip file from this link:

<https://adafru.it/AB0>

<https://adafru.it/AB0>

And point windows to the **Drivers** folder when it asks for the driver location

Mac OSX Setup

If you're using Mac OS Mavericks, Yosemite or later you may need to update the setting to permit running Arduino IDE

1. Open the, **System Preferences** from the Apple Menu.
2. Open the, **Security and Privacy** control panel.
3. Click on the **General** tab.
4. Click the **Lock Icon** and log in
5. Change **Allow Apps Downloaded From** to **Anywhere**
6. Open the downloaded IDE.
7. Go back to the Security preferences and change the selection back to **Mac App Store and identified developers**
8. You only need to go through this procedure once. Mavericks will remember that it's OK to run the app.
9. That's it, you're ready to start using the Arduino IDE with Adafruit's board on OSX!

Linux Setup

If you missed it on a previous page, the following link will let you download the latest 1.6.x version of the Arduino IDE for Linux:

<https://adafru.it/f85>

<https://adafru.it/f85>

<https://adafru.it/f86>

<https://adafru.it/f86>



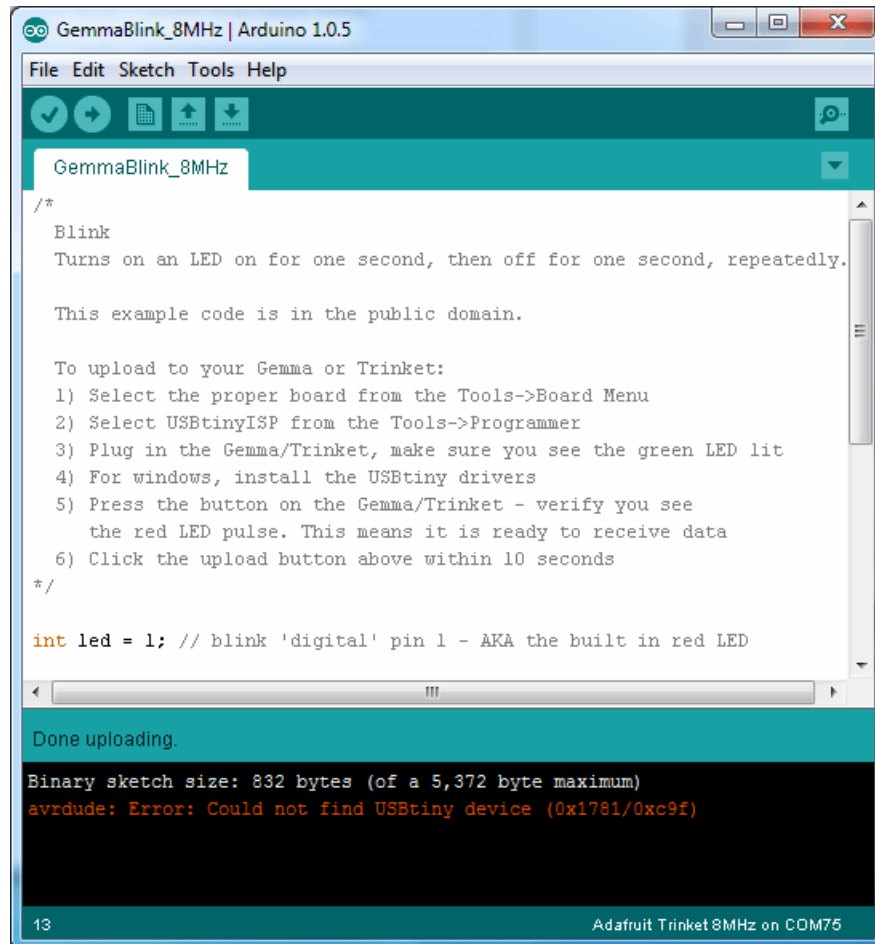
If you're on Linux, and are seeing multi-second delays connecting to the serial console, or are seeing "AT" and other gibberish when you connect, then the modemmanager service might be interfering. Just remove it; it doesn't have much use unless you're still using dial-up modems. To remove, type this command at a shell:

```
sudo apt purge modemmanager
```

Troubleshooting

If you get the error message `avrdude: Error: Could not find USBtiny device (0x1781/0xc9f)`

That means the bootloader wasn't active. Make sure to press the button on the Trinket to activate the bootloader *before* clicking the Upload button.



The screenshot shows the Arduino IDE interface. The title bar reads "GemmaBlink_8MHz | Arduino 1.0.5". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains icons for opening files, saving, and uploading. The sketch editor shows a blink sketch with the following content:

```
/*  
Blink  
Turns on an LED on for one second, then off for one second, repeatedly.  
  
This example code is in the public domain.  
  
To upload to your Gemma or Trinket:  
1) Select the proper board from the Tools->Board Menu  
2) Select USBtinyISP from the Tools->Programmer  
3) Plug in the Gemma/Trinket, make sure you see the green LED lit  
4) For windows, install the USBtiny drivers  
5) Press the button on the Gemma/Trinket - verify you see  
   the red LED pulse. This means it is ready to receive data  
6) Click the upload button above within 10 seconds  
*/  
  
int led = 1; // blink 'digital' pin 1 - AKA the built in red LED
```

Below the sketch editor, a status bar indicates "Done uploading." and "Binary sketch size: 832 bytes (of a 5,372 byte maximum)". The console window shows the error message: `avrdude: Error: Could not find USBtiny device (0x1781/0xc9f)`. The bottom status bar shows "13" and "Adafruit Trinket 8MHz on COM75".

If you get a lot of red text, errors and also a warning about Verification Failed

Check that you updated the `avrdude.conf` file above - if you don't update the description of the Attiny85 in the configure file by replacing it, the IDE won't know to be patient with the Trinket/Gemma bootloader and will have many upload errors

The screenshot shows the Arduino IDE interface. The top window displays the sketch code for 'GemmaBlink_8MHz'. The code includes instructions for setting up a red LED on pin 1 and a loop that blinks it every 1000ms. Below the code editor, the 'Serial Monitor' window shows the output of the upload process. The output indicates a successful upload but includes several error messages from avrdude related to USB communication and verification.

```
GemmaBlink_8MHz
3) Plug in the Gemma/Trinket, make sure you see the green LED lit
4) For windows, install the USBtiny drivers
5) Press the button on the Gemma/Trinket - verify you see
   the red LED pulse. This meas it is ready to receive data
6) Click the upload button above within 3 seconds
*/

int led = 1; // blink 'digital' pin 1 - AKA the built in red LED

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

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
}
```

```
Done uploading.
avrdude: error: usbtiny_receive: usb_control_msg: sending control
message failed, win error: A device attached to the system is not
functioning.

(expected 4, got -5)

avrdude: error: usbtiny_receive: usb_control_msg: sending control
message failed, win error: A device attached to the system is not
functioning.

(expected 4, got -5)
avrdude: verification error, first mismatch at byte 0x0000
0x20 != 0x00
avrdude: verification error; content mismatch
```

20 Adafuit Trinket 8MHz on COM75

On Linux if you get the error message "usbtiny_receive: error sending control message: Protocol error (expected 4, got -71)"

These can generally be ignored and should not interfere with the program upload. Unfortunately Linux's USB core is a little flakey communicating with the ATtiny85 processor on the Trinket/Gemma and can cause these errors. If an upload does fail, try it again as it is likely an intermittent issue.

