All About Arduino Libraries
Created by Bill Earl

Last updated on 2019-05-29 11:20:09 PM UTC
Arduino Libraries

Arduino libraries are a convenient way to share code such as device drivers or commonly used utility functions. This guide details how to install libraries on your computer. For an excellent introduction to Arduino libraries and what you can do with them, see the Libraries page from Arduino Tips, Tricks and Techniques (https://adafru.it/aYG).

There are three general types of Arduino Libraries:

Standard Libraries

The Arduino IDE comes with a set of standard libraries for commonly used functionality. These libraries support all the examples included with the IDE. Standard library functionality includes basic communication functions and support for some of the most common types of hardware like servo motors and character LCD displays.

Standard Libraries are pre-installed in the "Libraries" folder of the Arduino install. If you have multiple versions of the IDE installed, each version will have its own set of libraries. For the most part, it is not a good idea to change the Standard Libraries or install your libraries in the same folder.

Library Manager Libraries

The newer Arduino IDE has a Library Manager which facilitates installing third-party libraries submitted to Arduino for use. Adafruit has most of its libraries and drivers in the Library Manager for easy use.

User Installed Libraries

There are many other libraries with useful functionality and device drivers for all sorts of hardware. These drivers are available from places like Arduino Playground, Github and Google Code.

User installed libraries should be installed in your sketchbook libraries folder so they can be used with all versions of the IDE. This way, if a new version of the IDE is released, you don't have to re-install all your favorite libraries!
Library Manager

The Library Manager is a newer feature of the Arduino IDE. Many of the popular libraries, including most of Adafruit's, are in the Library Manager. If you are looking to work with other people's libraries, first check the Library Manager to see if the library is there first rather than downloading a copy into your local libraries folder.

To start, load the Arduino IDE. Then use the menus at the top to select Sketch -> Include Library -> Manage Libraries...

You will then get a list of all the great libraries you can install!

Most often you may use the Search features at the top to find a specific library.

Above, the user typed adafruit into the search bar. All modern Adafruit libraries begin with the word Adafruit.

You can scroll down the list to find the library you are looking for. Say we wanted the Circuit Playground library like shown below. You can click the Install button to have it available or if it is already installed, you can select the version you want (often the latest but sometimes an earlier library for compatibility).
Often (but not always), loading a library will bring the availability of additional examples. In the menu, go to Files -> Examples then select your library name (if present) and browse the additional examples. Loading the Adafruit CircuitPlayground library above also loaded 12 examples and 4 directories with additional examples.

The latest versions of the Arduino IDE will alert you on startup that newer versions of your installed libraries (and boards) are available, if you would like to upgrade. You can safely ignore these in a pinch but you'll want to get the latest libraries when you start a new project.
Installing Libraries Manually

The Library Manager is the simplest and safest way to install libraries. If possible, that is the method you should use.

If the library you want to use is not available via the library manager, you can get the source code and place it into your machine's local library folder. The location of the folder is rather specific.

See the file hierarchy below:

![File Hierarchy Diagram]

It is important to install your libraries in the correct location. Otherwise the compiler will not be able to locate them when you try to compile and upload your sketches.

Incorrectly installed libraries can cause many problems with the IDE.

Locate your Sketchbook Folder and the Libraries Folder Inside

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.

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

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"!

User installed libraries should go in a folder named Libraries, located inside your sketchbook folder. This is where the IDE will look for user installed libraries.

On version 1.0.2 and later of the Arduino IDE, the "Libraries" folder is created automatically. On earlier versions of the IDE, you may need to create it when you install your first library.
Open the Preferences Dialog

Click File -> Preferences in the IDE.

Find the Sketchbook Location

Usually a folder named Arduino in your Documents folder.

Once you know the location, navigate to this folder in Windows Explorer or mac Finder.
Create a Libraries Folder

If there is no Libraries folder there, create one.

Right click in Windows Explorer and select New -> Folder.

Rename to Libraries

Right click on the New Folder and select Rename. Then type in Libraries.
Installing a Library on Windows

Follow these steps to install a library in Windows.

Close the Arduino IDE
First make sure that all instances of the Arduino IDE are closed. The IDE only scans for libraries at startup. It will not see your new library as long as any instance of the IDE is open!

Download the Zip File
Make sure to select a released version of the library. Some library repositories are still in development, and may contain unreleased or untested code. The "Releases" button takes you to the tested and released versions.

1. Click the "Releases" button on the Github repository page to find the released version of the library.
2. Then download the zip file.
Open the Zip File
Open the Zip File and copy the library master folder.

Paste into your Libraries Folder
Open your sketchbook Libraries folder and paste the master folder you copied from the .Zip.

Give it a Legal Name
The IDE will not recognize folders with dashes in the name. So you must rename the Github Master Folder. Underscores are OK!
Re-start the IDE

Restart the Arduino IDE and verify that the library appears in the File->Examples menu.

Load one of the library examples to test.

Verify that it Compiles

Click the check-mark icon in the upper left and verify that the example sketch compiles without errors.
Installing a Library on Mac OSX

Follow these steps to install a library in Mac OSX.

Close the Arduino IDE
First make sure that all instances of the Arduino IDE are closed. The IDE only scans for libraries at startup. It will not see your new library as long as any instance of the IDE is open!

Download the Zip File
Make sure to select a released version of the library. Some library repositories are still in development, and may contain unreleased or untested code. The "Releases" button takes you to the tested and released versions.

1. Click the "Releases" button on the Github repository page to find the released version of the library.
2. Then download the zip file.

Find it in the Downloads Folder
OSX will automatically open the zip file there.
Drag it to your Libraries Folder
Open your sketchbook Libraries folder and drag the master folder from Downloads into it.

Give it a Legal Name
The IDE will not recognize folders with dashes in the name. So you must rename the Github Master Folder. Underscores are OK!

Re-start the IDE
Restart the Arduino IDE and verify that the library appears in the File->Examples menu.

Load one of the library examples to test.
Verify that it Compiles

Click the check-mark icon in the upper left and verify that the example sketch compiles without errors.
Installing a Library on Linux

Follow these steps to install a library in Linux.

Close the Arduino IDE

First make sure that all instances of the Arduino IDE are closed. The IDE only scans for libraries at startup. **It will not see your new library as long as any instance of the IDE is open!**

Download the Zip File

Make sure to select a released version of the library. Some library repositories are still in development, and may contain unreleased or untested code. The "Releases" button takes you to the tested and released versions.

1. Click the "Releases" button on the Github repository page to find the released version of the library.
2. Then download the zip file.
Save the Zip File
Save the zip file to a convenient location.

Open the Zip File
Open the Zip File and copy the library master folder.
Start the Extraction
Select "Extract" from the menu and navigate to your Sketchbook/Libraries folder.

Extract into your Libraries Folder
After navigating to the sketchbook/libraries folder, complete the extraction.

Give it a Legal Name
The IDE will not recognize folders with dashes in the name. So you must rename the Github Master Folder. Underscores are OK!
Re-start the IDE
Restart the Arduino IDE and verify that the library appears in the File->Examples menu.

Load one of the library examples to test.

Verify that it Compiles
Click the check-mark icon in the upper left and verify that the example sketch compiles without errors.
Common Library Problems

'xxxx' does not name a type
This is the most common library related error message and it means that the compiler could not find the library. This can be due to:

- Library is not Installed (see previous pages in this guide)
- Wrong Folder Location
- Wrong Folder Name
- Wrong Library Name
- Library Dependencies
- Forgot to Shutdown the IDE

See below for solutions to these problems.

Wrong Folder Location
The IDE will only find standard libraries and libraries installed in the sketchbook Libraries folder. It will not be able to find libraries installed elsewhere.

The Library folder must be at the top level of the Libraries folder. If you put it in a sub-folder, the IDE will not find it.

Note: Some third-party library repositories have different folder structures. You may need to re-arrange things to make sure that the library files are at the top level of the folder. WaveHC is one example of this. The actual library folder is a folder within the top-level repository folder.

Don't have a "Sketchbook" folder
It is there. But on a Windows or Mac/OSX machine it may not be named "Sketchbook". See the page titled "Where to Install your Libraries".

Incomplete Library
You must download and install the entire library. Do not omit or alter the names of any files inside the library folder.
Wrong Folder Name

The IDE will not load files with certain characters in the name. Unfortunately, it doesn’t like the dashes in the zip files names generated by Github. When you unzip the file, rename the folder so that it does not contain any ‘illegal’ characters. Simply replacing each dash (‘-‘) with and underscore (‘_’) usually works.

Wrong Library Name

The name specified in the #include of your sketch must match exactly (including capitalization!) the class name in the library. If it does not match exactly, the IDE will not be able to find it. The example sketches included with the library will have the correct spelling. Just cut and paste from there to avoid typos.

Multiple Versions

If you have multiple versions of a library, the IDE will try to load all of them. This will result in compiler errors. It is not enough to simply rename the library folder. It must be moved outside of the sketchbook Libraries folder so the IDE won’t try to load it.

Library Dependencies

Some Libraries are dependent on other libraries. For example, most of the Adafruit Graphic Display libraries are dependent on the Adafruit GFX Library. You must have the GFX library installed to use the dependent libraries.

“Core” Libraries

Some libraries cannot be used directly. The GFX library is a good example of this. It provides core graphics functionality for many Adafruit displays, but cannot be used without the specific driver library for that display.

Forget to shutdown the IDE

The IDE only searches for libraries at startup. You must shut down ALL instances of the IDE and restart before it will recognize a newly installed library.