Setting up Arduino IDE

Installing Libraries

A significant part of the Arduino programming experience is the use of libraries, which are pre-written code packages designed to simplify development. Libraries provide easy-to-use functions for interacting with various hardware components like sensors, motors, and displays, reducing the need to write complex code from scratch. By abstracting away low-level hardware details, libraries make it possible to focus on the core logic of a project, whether you’re a beginner or an experienced developer.

Arduino’s vast collection of libraries can be found and installed directly through the Arduino IDE's Library Manager. This feature allows developers to search, install, and manage libraries effortlessly. Libraries cover a wide range of functionalities, from basic tasks like controlling LEDs to advanced applications involving communication protocols, robotics, or artificial intelligence.

The Arduino community actively contributes to maintaining and expanding this library collection, ensuring that users have access to high-quality, tested resources. Open-source libraries also foster collaboration, enabling developers to improve or customize them. By leveraging libraries, Arduino users can create sophisticated projects efficiently, using well-documented, ready-made solutions. Ultimately, libraries play a crucial role in enhancing productivity and accessibility, making Arduino a powerful platform for both learning and professional development.

Why Use Libraries?

Using libraries in Arduino programming offers several benefits that significantly enhance the development process, making it easier, faster, and more efficient. Here’s why libraries are so essential:

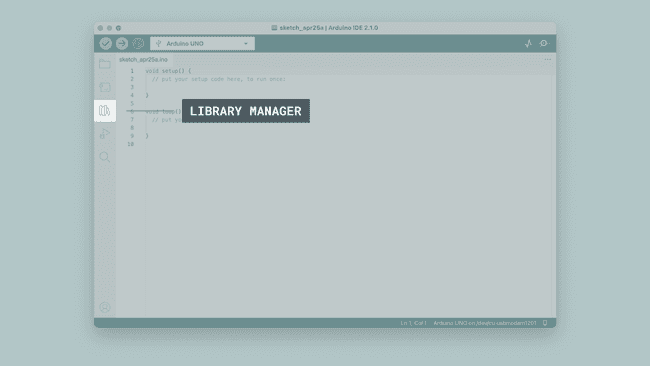
1. **Simplifies Code**: Libraries provide pre-written functions that simplify complex tasks, such as controlling sensors, displays, or motors. Instead of writing intricate code from scratch, developers can call on a library’s functions to achieve the desired results quickly.
2. **Saves Time**: With libraries, developers don’t need to reinvent the wheel. Many common tasks, like reading sensor data or managing communication protocols (e.g., I2C or SPI), are already implemented. This saves valuable time, especially for beginners who may not be familiar with low-level coding details.
3. **Improves Readability and Maintenance**: Libraries make code more readable and organized. Since a library encapsulates specific functionality, the main program remains cleaner and easier to follow. This also simplifies future modifications or debugging.
4. **Reduces Errors**: Libraries are often well-tested by the community, which reduces the risk of errors. By relying on established libraries, developers can avoid common pitfalls associated with coding complex hardware interactions.
5. **Access to Advanced Features**: Libraries enable developers to implement advanced features without needing in-depth knowledge of complex algorithms or hardware intricacies, making sophisticated projects more accessible.
6. **Community Support**: Many libraries are open-source and supported by the vast Arduino community. This means that developers can access documentation, examples, and receive help from other users when they encounter issues.

Installing a Library

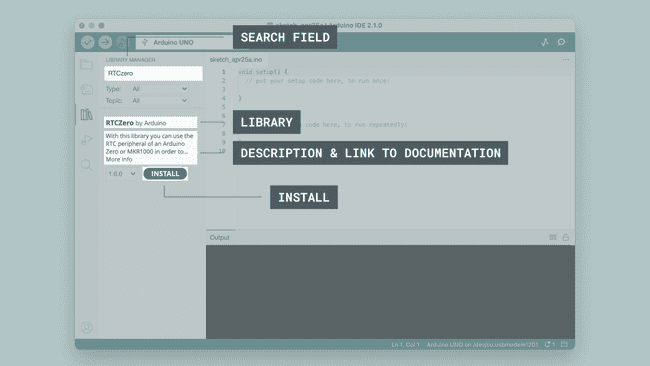
Installing a library is quick and easy, but let's take a look at what we need to do.

**1.** Open the Arduino IDE 2.

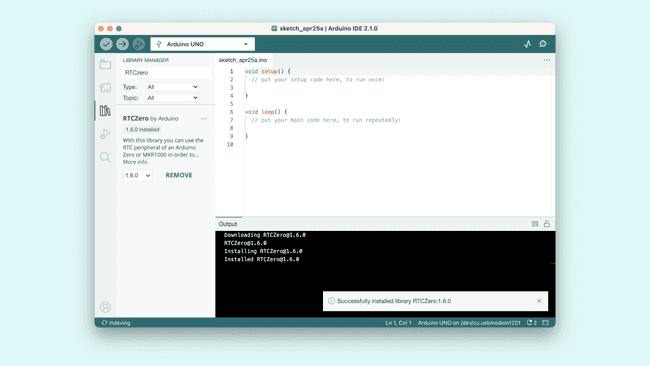
**2.** With the editor open, let's take a look at the left column. Here, we can see a couple of icons. Let's click the on the **"library"** icon.

The library manager.

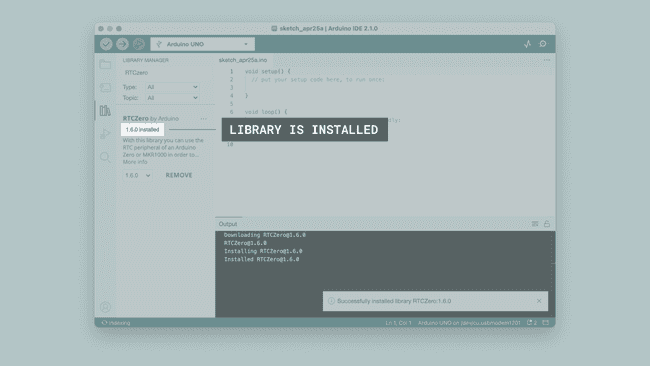
**3.** A list will now appear of all available libraries, where we can also search for the library we want to use. In this example, we are going to install the **RTCZero** library. Click on the **"INSTALL"** button to install the library.

Navigating the library manager.

**4.** This process should not take too long, but allow up to a minute to install it.

Installing a library.

**5.** When it is finished, we can take a look at the library in the library manager column, where it should say **"INSTALLED"**.

Library installed successfully.

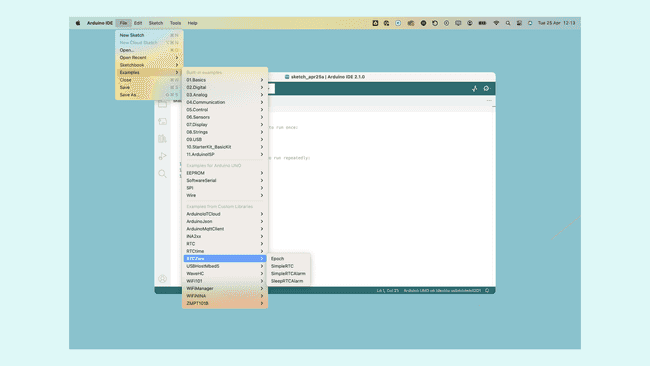
Congratulations! You have now successfully downloaded and installed a library on your machine.

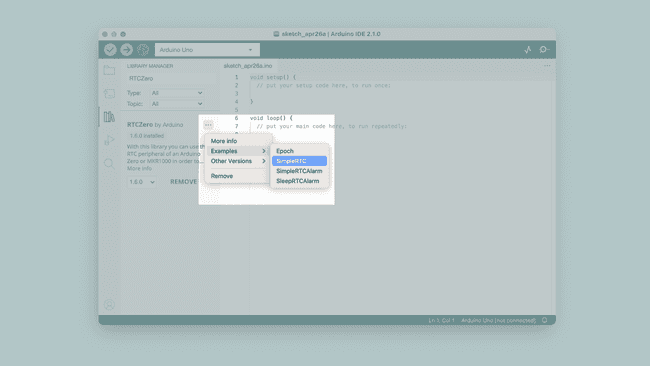
Including a Library

To use a library, you first need to include the library at the top of the sketch.

Including a library in a sketch.

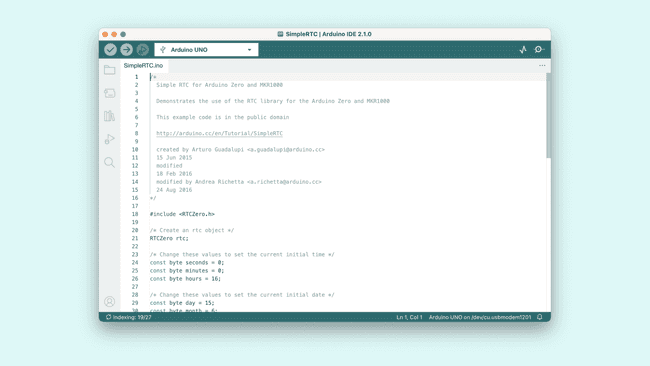
Almost all libraries come with already made examples that you can use. These are accessible through **File > Examples > {Library} > {Example}**. In this example, we are choosing the **RTCZero > SimpleRTC**.

Choosing an example sketch from a library.

You can also access library examples directly in the library manager, by clicking the three dots next to the installed library, like this:

Library examples

The chosen example will now open up in a new window, and you can start using it however you want to.

A library example.

Installing Boards in Arduino ide

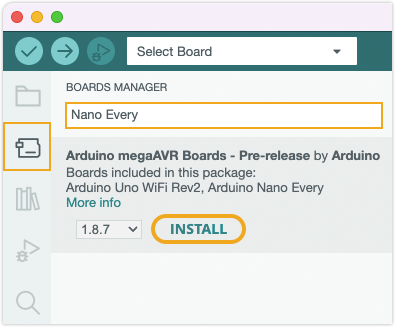
Add Board with board Manager

1. In the menu bar, select Tools > Board > Boards Manager… ,or click on the Boards Manager icon button in the sidebar.
2. Either search for the package name (e.g. “megaAVR”), or the board (e.g. “Uno”, “MKR1000”, or “Portenta”), by typing in the search field.
3. Find the package that includes your board.
4. Click **Install** (the latest version is selected by default).

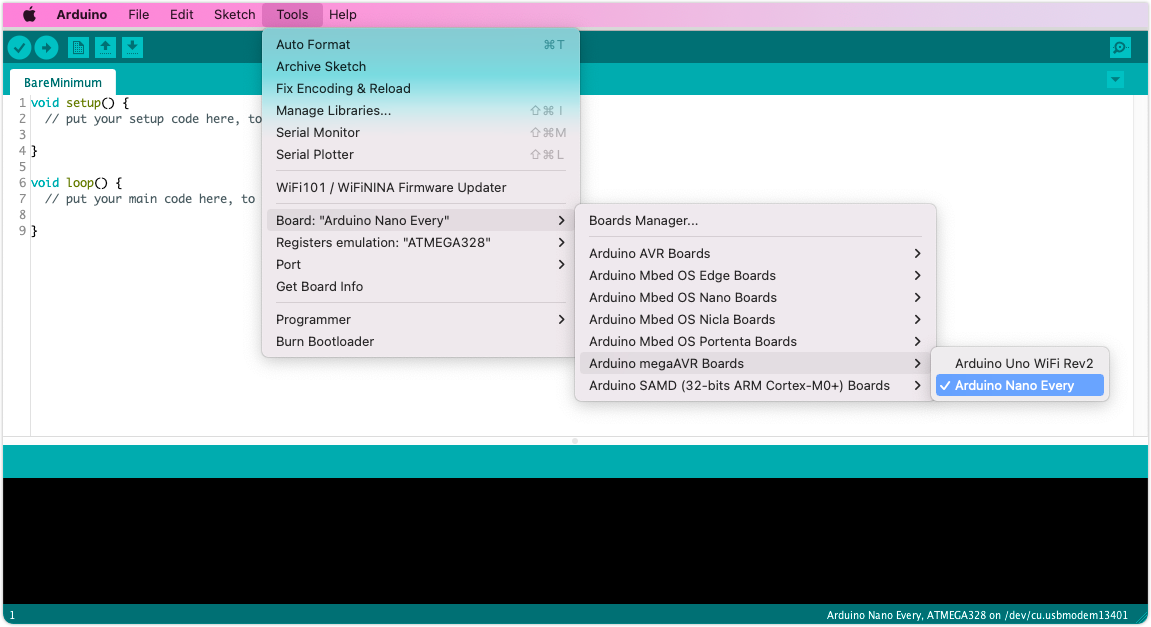
Note

If the package is already installed, you will instead have the option of installing a different version.

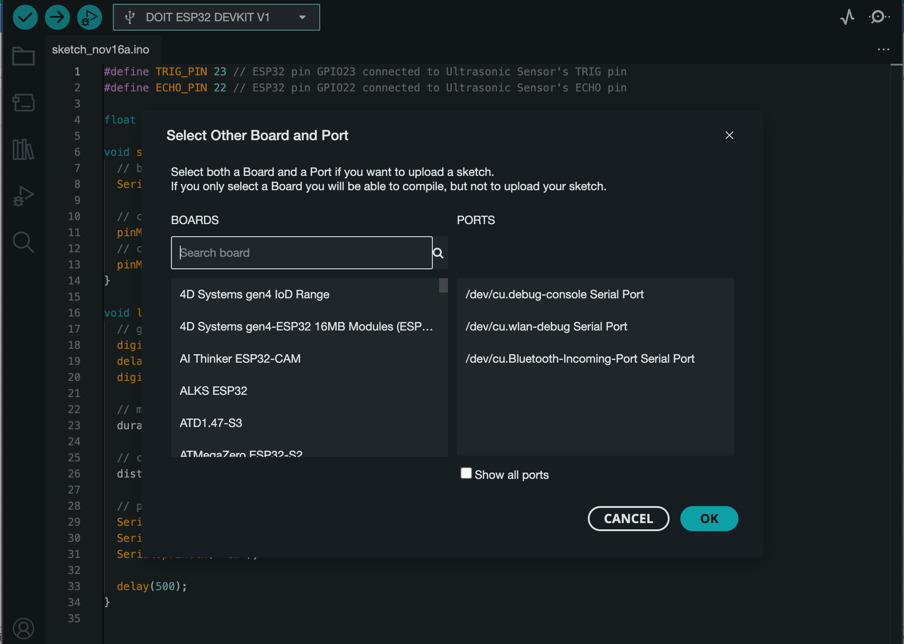
1. Wait for the installation to complete.



You can now select boards from the installed packages in the Tools > Board menu:



Or can direct search for the board by clicking on USB icon near the debugger symbol or uploading symbol.



Add third-party platforms to the Boards Manager in Arduino IDE

By default, Arduino IDE installs and updates the board platforms from the official Arduino package list[[1]](https://support.arduino.cc/hc/en-us/articles/360016466340-Add-third-party-platforms-to-the-Boards-Manager-in-Arduino-IDE#fn1). This lets you install and update Arduino platforms, for use with official Arduino boards, as well as third-party boards compatible with the official Arduino platform.

The Board Manager can also be used for conveniently installing and updating platforms that are not in the official package list, if the creator provides a **platform index URL** (ending with \_index.json).

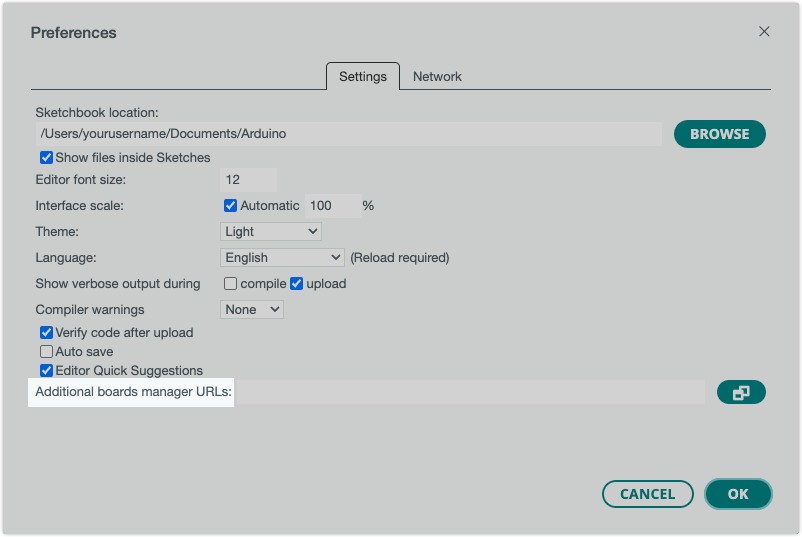
Add or remove additional Board Manager URLs

Follow these steps:

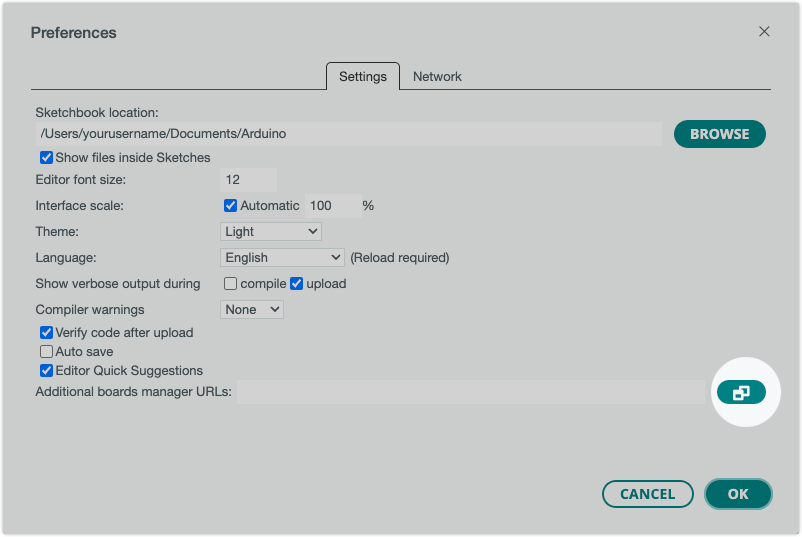
1. Open Preferences using the menu bar:
   * Windows/Linux: **File > Preferences**
   * macOS: **Arduino IDE > Settings**
2. Find the **Additional boards manager URLs** setting toward the bottom.

Note

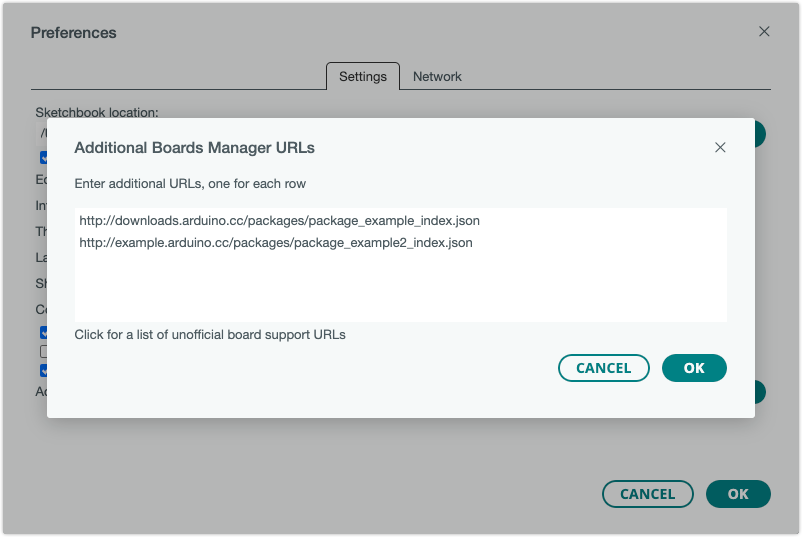
If you can’t find the setting, put your cursor in the middle of the Preferences window and scroll down.



1. Click the button to the right of the text field to open the **Additional Boards Manager URLs** window.



1. Paste or remove URLs in the window. There should be one URL for each line.



1. Click **OK** to close the Additional Boards Manager URLs window.
2. Click **OK** again to close Preferences and save the changes.