Arduino Resources

Overview

The goal of this Arduino Resource is to point new makers to useful resources to help them build up their skills using Arduino. We direct people to external resources and best practices that can help them learn how to use, write, and edit Arduino code.

While we have included links to resources, we have found useful in the past, we cannot guarantee that external links will remain active.

If you have any suggestions to improve this resource, you can add an Issue to the GitHub page for our resources.

Makers Making Change
A Neil Squire Program

Contents

Getting Started with Arduino

Arduino has thousands of tutorials created to get you familiar with using their software. If you are starting out with Arduino, we recommend looking through them and finding tutorials related to what you'd like to learn. If you do not have access to physical hardware but would like to learn and practice with Arduino, you can use TinkerCAD Circuits. The circuits tools allow you to use virtual hardware to make devices and test Arduino code on various boards.

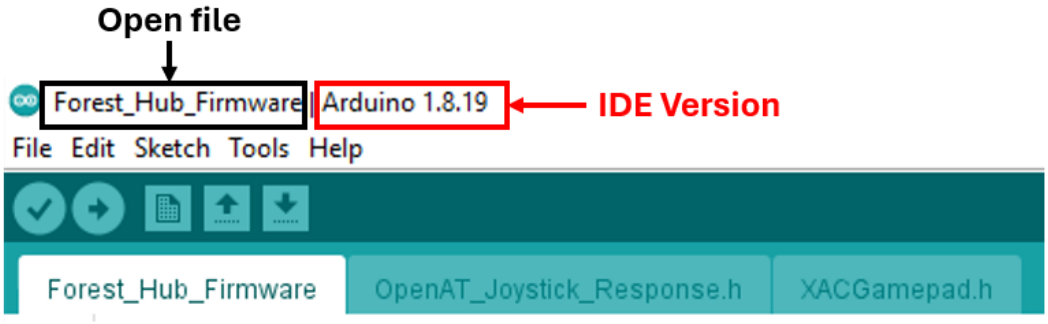
It can be overwhelming, so we have pulled out some of the key things you'll need to know to get started with building MMC devices, starting from installing Arduino to flashing the firmware. Note that all the steps required for a specific device are included in that device's Maker Guide.

Installing Arduino

To install Arduino, go to the Arduino website and download the software for your operating system. Once you've downloaded the installer, click on the file and let it run. You will need administrative privileges on the computer you are installing onto. Unless you know you will need different settings, go with whatever defaults the installer suggests.

Arduino IDE Versions

If you newly install Arduino, you'll likely be using the IDE 2 version, but if you installed Arduino previously, you may be using one of the IDE 1 versions. Tutorial and instruction information will differ between these versions as large changes have been made to the IDE between them. If you are unsure which version you are using, you can open Arduino and look at the top. Arduino will display the file you've opened and the Arduino IDE version, as seen in the figure below.



Installing Libraries

Arduino libraries extend functionality of Arduino by adding code that makes it easy to connect to a sensor, display, etc. The Arduino tutorial for installing libraries is helpful for those just learning.

Flashing Firmware

Flashing firmware is when you upload code to a microprocessor, or board. To successfully flash firmware, the correct board and relevant settings must be selected, the code must compile, the board must be connected, and the correct port must be selected.

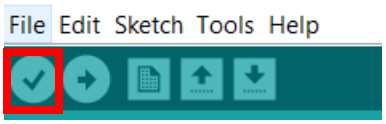
Boards Manager

Board packages are installed through the Boards Manager. These packages are pre-written software that make it easier to interface with different boards. To properly flash firmware you need to have the correct board installed and selected. Some boards will also have multiple options, which will appear when the board is selected. Arduino has tutorials on installing different board packages, selecting boards, and selecting settings on different boards.

Compiling Code

Compiling code is basically translating the code someone has written in to the code language the computer understands. For code to be executable, it must be compiled, but just because the code compiles does not mean it will work as intended. When compiling code, the computer will check that the code will run and give errors or warnings if any issues appear.

To compile to code, click on the checkmark in the top-left corner of Arduino.



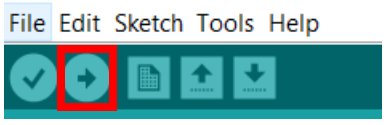
If you run into issues while compiling code, check the Arduino tutorial on compiling errors.

Selecting Port

The port is where the device you're uploading code to is attached (and recognized) by your computer. For our devices, we generally only have one microcontroller so there will only ever be one port option when it's plugged in, but more complex devices and projects may have multiple microcontrollers working together. To select the board, go to "Tools" and look down for the "Port" option, then select the board that is plugged in.

Uploading Code

Once you have the code ready and the port selected, you can upload the code to the board by clicking on the arrow. This will compile the code and upload it to the selected board.



If you are running into issues with uploading code, refer to the Arduino uploading errors for troubleshooting options.

Writing Code

Becoming a good coder takes time and practice. The best way to get better is to start working on problems. We recommend working through tutorials that interest you, then working on problems that keep you engaged.

There are a lot of different styles of coding and best practice suggestions that exist, and many come down to personal preference. If you find a style and best practice that works for you, be consistent and avoid mixing styles. Staying consistent will make your life as a coder easier and make it easier for others to use and understand your code.

A good reference to start from is the Arduino writing style guide.

Syntax

Coding syntax are essentially the rules that the program follows to interpret statements or expressions in the code. As you become more experienced with coding, learning new syntax of different coding languages becomes less daunting, but it can be a major hurdle for new programmers. Learning the Arduino syntax basics will help new programmers get coding.

Commenting

Commenting code is meant to explain what is happening and what variables are meant to do. Good commenting will help with understanding new code, debugging, and coming back to code after not using it for a long time.

We recommend adding comments at the beginning of the code which include all licensing information, as well as a general explanation of what the code will do. For example, the comment at the beginning of the code for the LipSync is copied below:

```
/*
* File: LipSync_Firmware.ino
* Firmware: LipSync
* Developed by: MakersMakingChange
* Version: v4.0.1 (29 April 2024)
  License: GPL v3.0 or later

  Copyright (C) 2024 Neil Squire Society
  This program is free software: you can redistribute it and/or modify it under the terms of
  the GNU General Public License as published by the Free Software Foundation,
  either version 3 of the License, or (at your option) any later version.
  This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY;
  without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
  See the GNU General Public License for more details.
  You should have received a copy of the GNU General Public License along with this program.
  If not, see <http://www.gnu.org/licenses/>
*/
```

Comments in Arduino show as a light grey font, as can be seen in the image below.

```
/*
* File: LipSync_Firmware.ino
* Firmware: LipSync
* Developed by: MakersMakingChange
* Version: v4.0.1 (29 April 2024)
  License: GPL v3.0 or later

  Copyright (C) 2024 Neil Squire Society
  This program is free software: you can redistribute it and/or modify it under the terms of
  the GNU General Public License as published by the Free Software Foundation,
  either version 3 of the License, or (at your option) any later version.
  This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY;
  without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
  See the GNU General Public License for more details.
  You should have received a copy of the GNU General Public License along with this program.
  If not, see <http://www.gnu.org/licenses/>
*/
```

Naming Variables

Make sure variables have useful names so someone can understand what the variable is meant to do or represent. Variable names in Arduino can only contain letters, underscores, and digits, and cannot start with a digit.

Two common conventions for naming variables are using underscores (variable_name) or camel case (variableName) to make it easier for people read the code. We prefer underscores at MMC, but there are pros and cons to each convention.

Finding Help

The online Arduino community is enormous and has answered many different coding questions of other users. The Arduino Forum can be a great place to look for help if you get stuck, whether for software or hardware questions. Stack Overflow also hosts many coding questions and answers, and has tagged sections for Arduino. While these resources can be useful, we cannot guarantee the accuracy of the information provided there, and have no involvement in moderating these forums.

If you are having trouble with building a Makers Making Change device (adding libraries, flashing firmware, etc.), you can use the Makers Making Change Community Forum and the Assembly Help category to post your questions/issues to the MMC community.