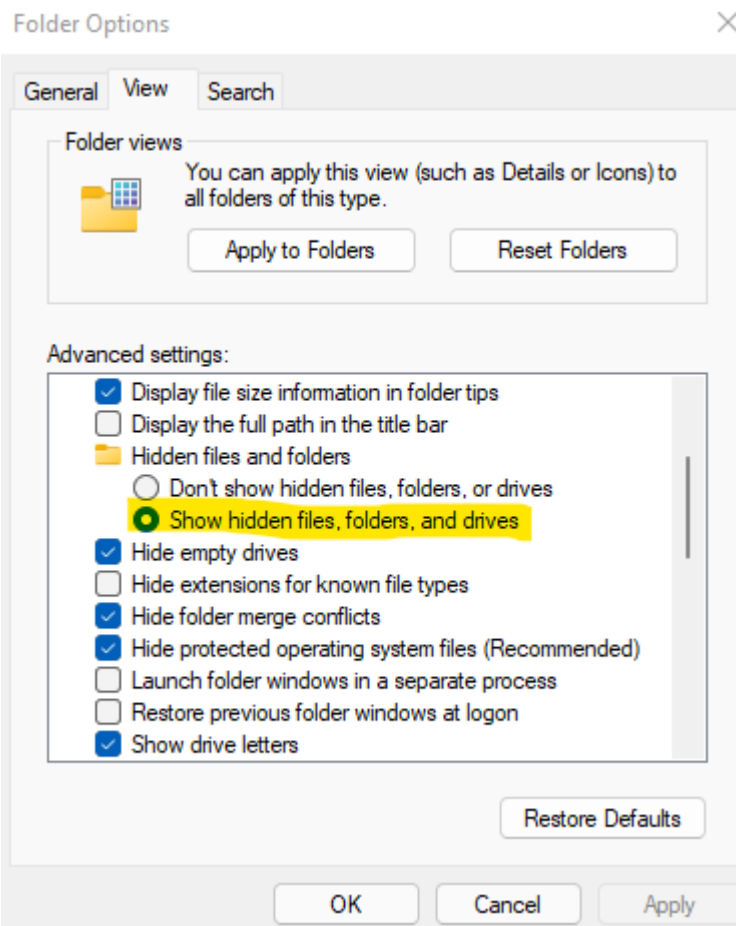


## Section xx: How to "Clean" the IDE

Starting with an Arduino IDE that does not contain any other packages is essential for testing purposes. An IDE with no other packages installed will be considered clean. The IDE appears to have the ability to pull missing variables and tools from other packages that have been installed. This means that testing a new package should be done on a "clean" version of the IDE.

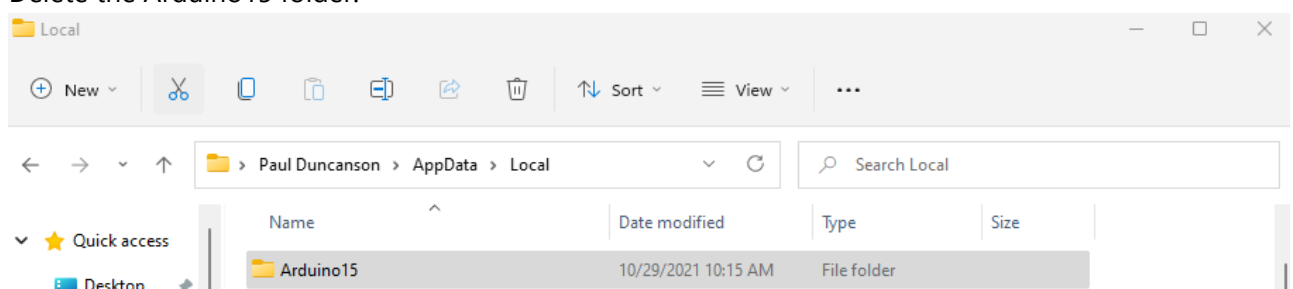
### Remove extra packages

1. Make sure the Arduino IDE and any associated files are closed.
2. Locate the Arduino15 folder.
  - Path: "C:\Users\USERNAME\AppData\Local\Arduino15"
  - On most Windows machines AppData will be a hidden folder, to show the folder select folder options and click "Show hidden files, folders, and drives" under the "Hidden files and folders"



section.

3. Delete the Arduino15 folder.



4. Restart the Arduino IDE.
- The IDE will generate a new empty Arduino15 folder.

Before:

« AppData » Local » Arduino15					Search Arduino15	
Name	Date modified	Type	Size			
cache	10/28/2021 3:08 PM	File folder				
logs	10/28/2021 3:08 PM	File folder				
packages	10/28/2021 3:08 PM	File folder				
staging	10/28/2021 3:08 PM	File folder				
library_index.json	10/29/2021 10:15 AM	JSON File	21,253 KB			
library_index.json.sig	10/29/2021 10:15 AM	SIG File	1 KB			
package_index.json	10/29/2021 10:15 AM	JSON File	416 KB			
package_index.json.sig	10/29/2021 10:15 AM	SIG File	1 KB			
package_Team25Capstone_avrdude_inde...	10/29/2021 10:15 AM	JSON File	1 KB			
preferences.txt	10/29/2021 3:21 PM	Text Document	4 KB			

After:

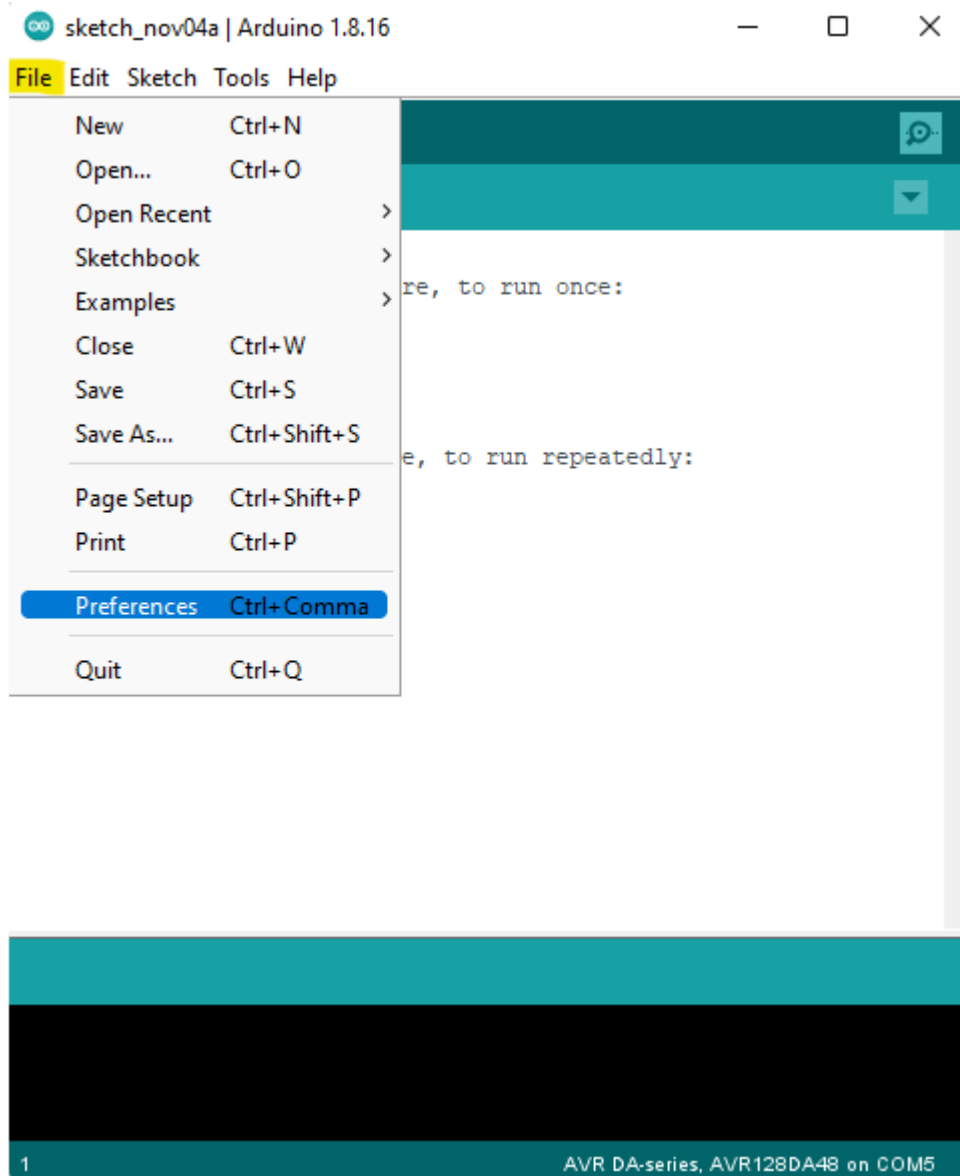
« AppData » Local » Arduino15					Search Arduino15	
Name	Date modified	Type	Size			
logs	10/31/2021 2:38 PM	File folder				
preferences.txt	10/31/2021 2:38 PM	Text Document	3 KB			

# Section xx: How to add a new Package

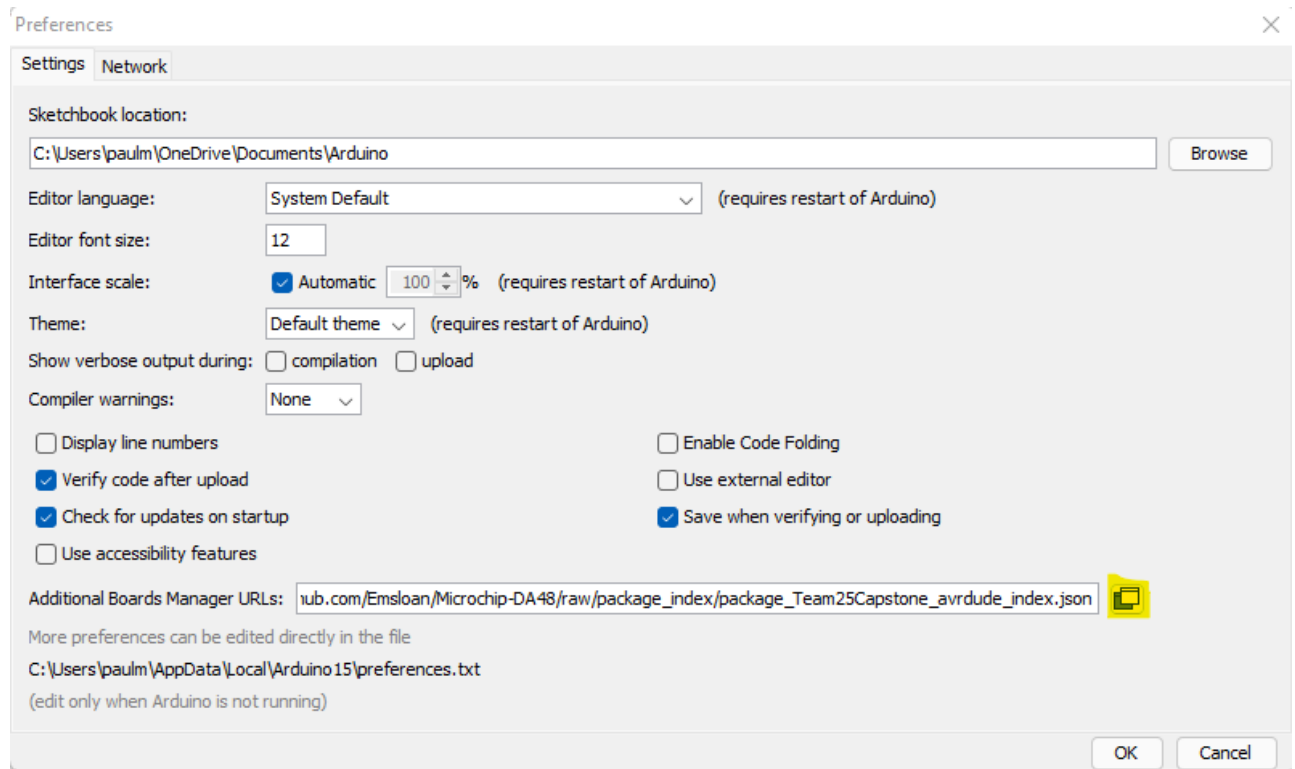
The IDE requires adding URLs pointing to JSON formatted files to install new core packages. The structure of these JSONs is specified by the IDE and will point to necessary tools and information to allow for new boards to interact with the IDE. Please see section xx in this document for more information about creating a JSON file.

Setting up a JSON file in the IDE

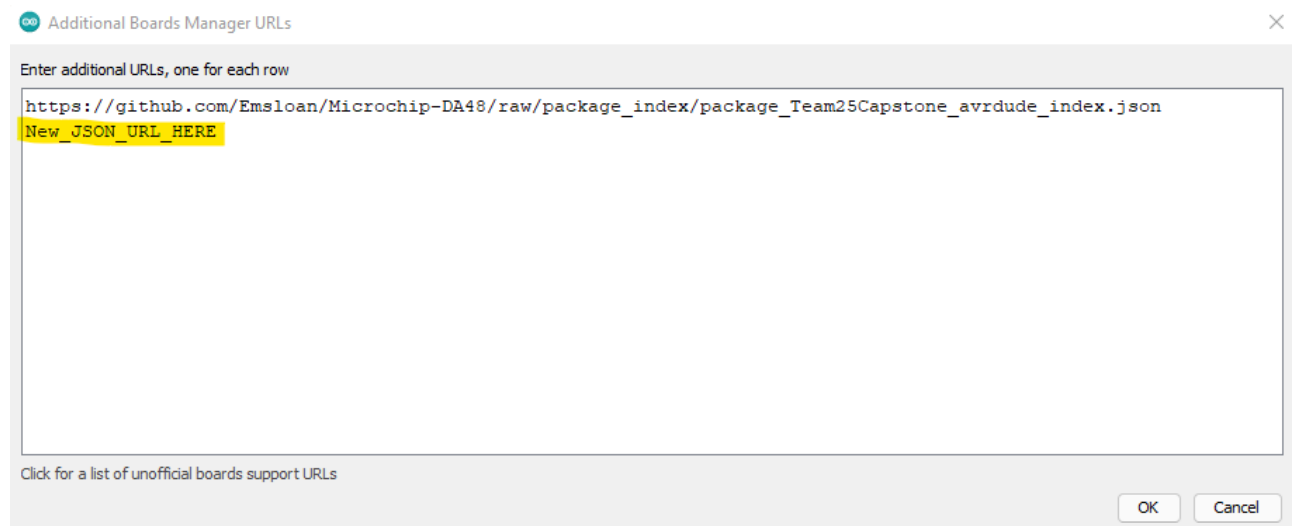
1. Click "File" menu and select the "Preferences" option.



2. In the Preferences menu, click the button to the far right of the "Additional Boards Manager URL" option.

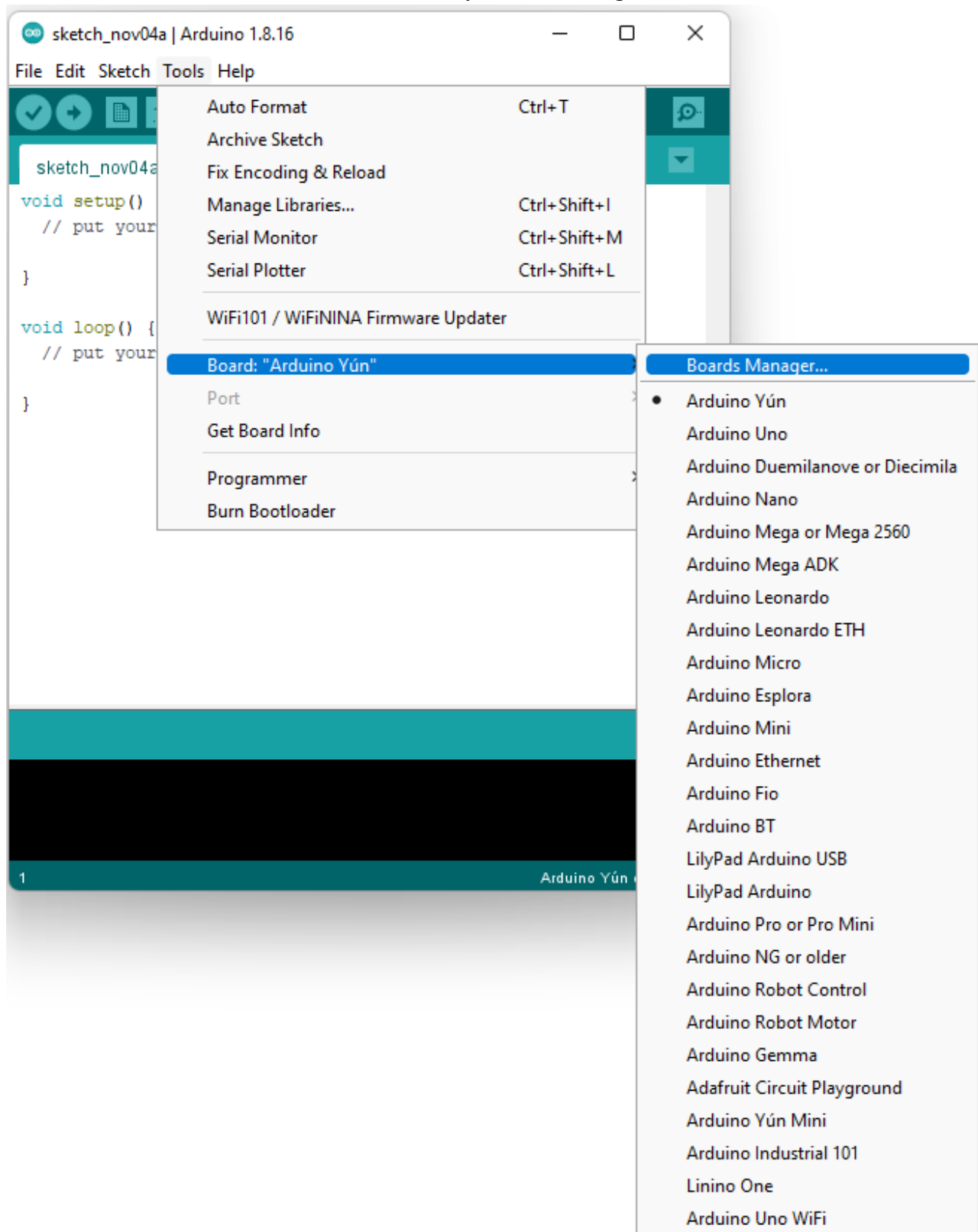


3. Setting up a new JSON requires adding the URL to the next line in the "Additional Boards Manager".



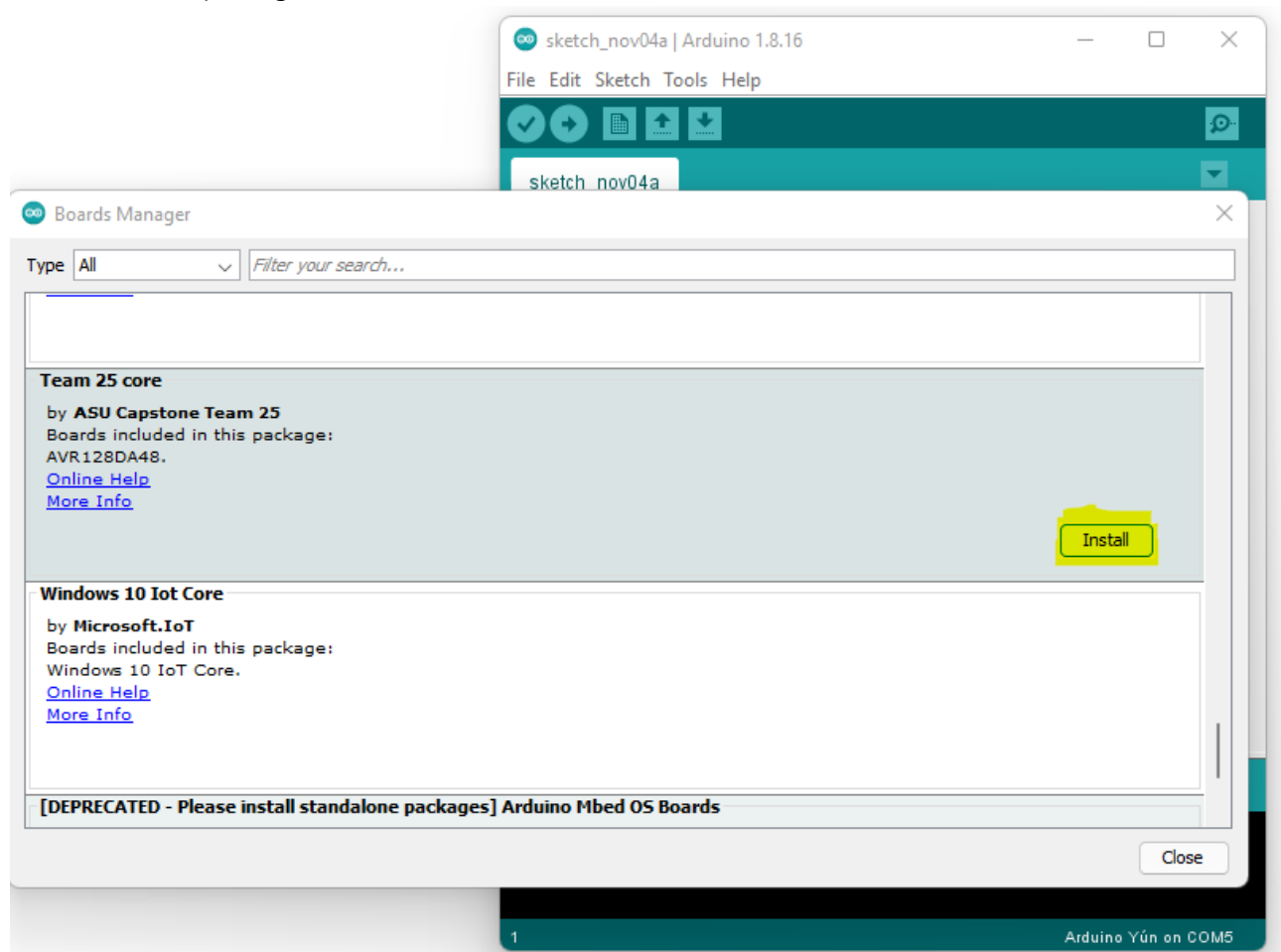
## Installing the Package

4. Click "Tools" menu and select "Board" followed by "Boards Manager".

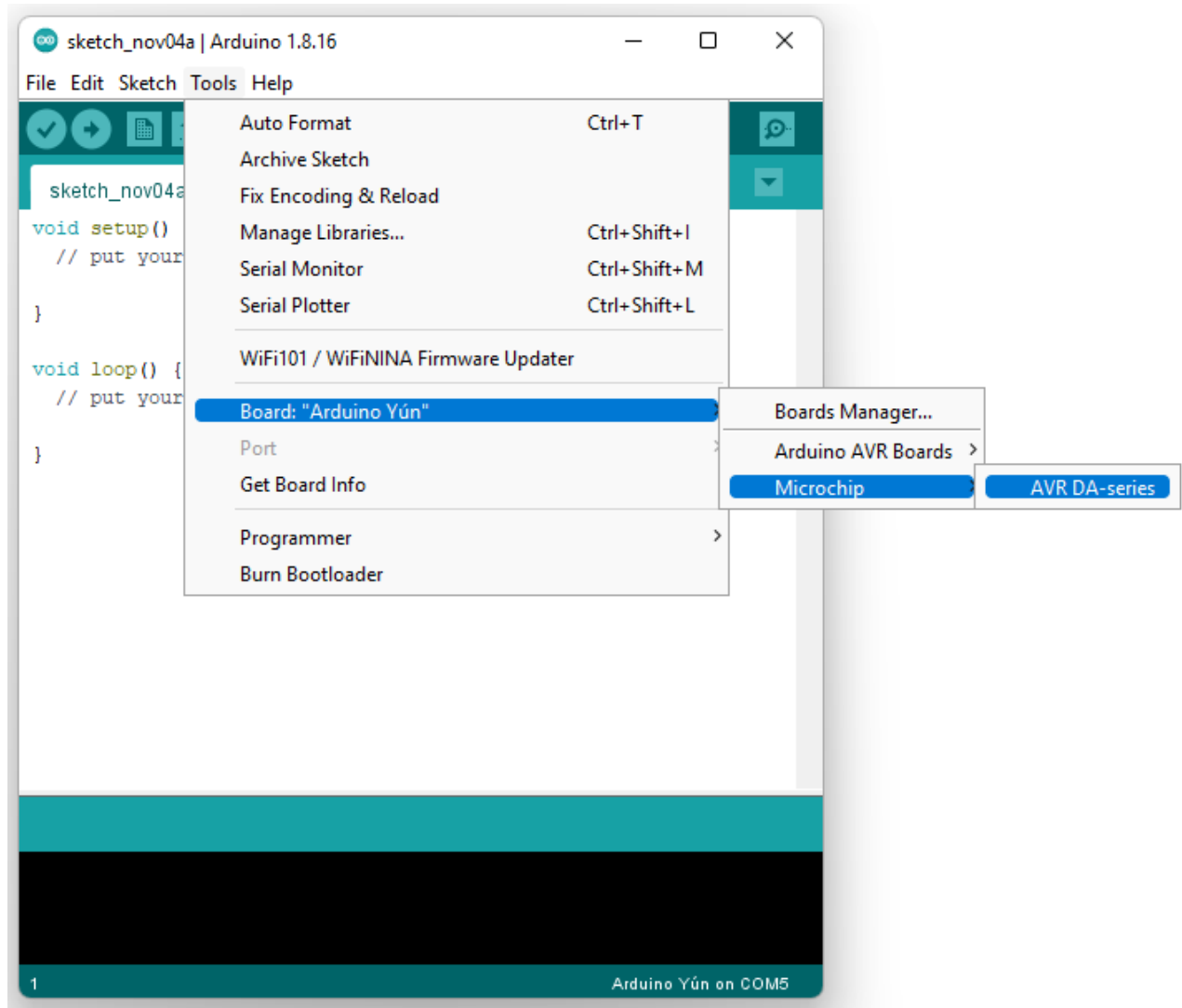


5. The boards manager will update the package options with the new package.

6. Find the added package and select "Install".



7. Installed package will be available under the "Tools" - "Boards" menu.

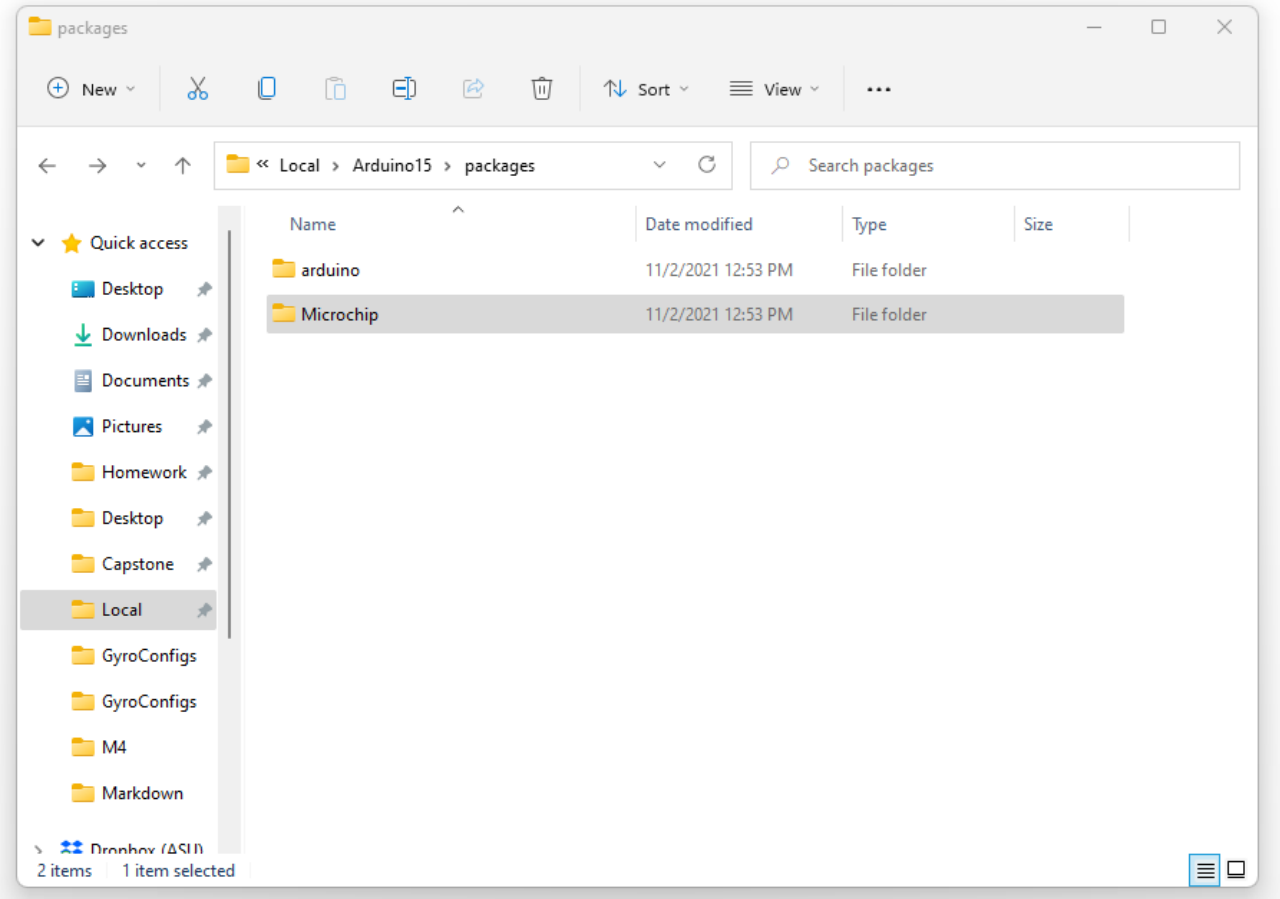


## Section xx: How to test a core package within the IDE

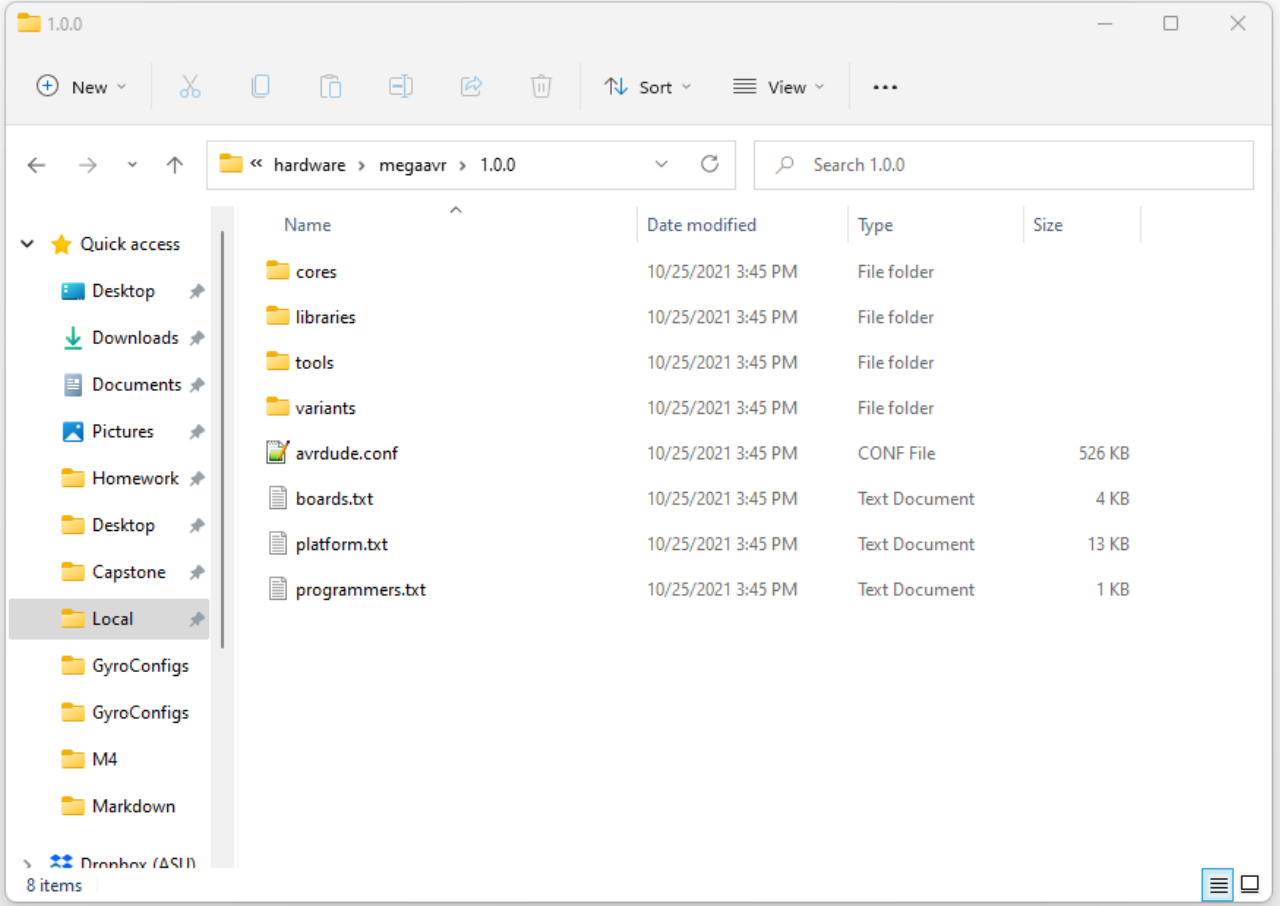
Testing out a new core package can be difficult when every change requires updating the JSON file. These changes require the SHA-256 key to be updated everytime. The IDE will not pull the package's zip file if the SHA-256 code does not match the zip file's source. This means that testing changes made in a package is easier to conduct in an already installed package.

Modifying and testing a core package

1. The "Arduino15" folder contains the folders and files that make a core package.



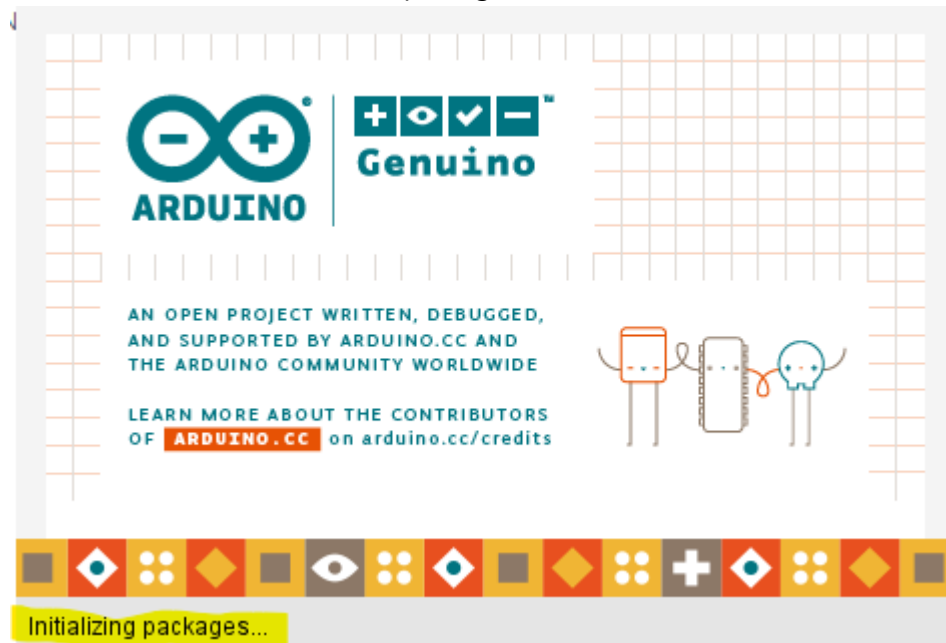
2. Select the desired package from the "Packages" folder.



- 3. Make updates to desired files in the package and save.
- 4. Restart the IDE.



5. The IDE will reinitialize installed packages.



6. Updated package will now be in use.

## Section xx: How to setup the SHA-256 key

---

A Secure Hash Algorithm (SHA) is a hashing function that can be used for validation purposes. The Arduino IDE uses the SHA-256 standard as a method of verifying Arduino core packages. This verification process is necessary to install new packages. Package owners generate a unique SHA-256 key from their core repository and must add this key to the JSON file.

1. The "checksum" section of the JSON contains the unique SHA-256 key.

```

1  {
2    "packages": [
3      {
4        "name": "Microchip",
5        "maintainer": "ASU Capstone Team 25",
6        "websiteURL": "https://github.com/Emsloan/Microchip-DA48",
7        "email": "emsloan@asu.edu",
8        "help": {
9          "online": "https://github.com/Emsloan/Microchip-DA48"
10       },
11       "platforms": [
12         {
13           "name": "Team 25 core",
14           "architecture": "megaavr",
15           "version": "1.0.0",
16           "category": "Contributed",
17           "url": "https://github.com/Emsloan/Microchip-DA48/archive/refs/heads/master.zip",
18           "archiveFileName": "Microchip-DA48-master.zip",
19           "checksum": "SHA-256:87ddb9416625115d108e3318925cdd07b9fda34f096456b34c0830afbd38b27d",
20           "size": "133000",
21           "boards": [
22             {
23               "name": "AVR128DA48"
24             }
25           ],
26           "toolsDependencies": [
27             {
28               "packager": "arduino",
29               "name": "avrdude",
30               "version": "6.3.0-arduino18"
31             }
32           ]
33         },
34       ],
35       "tools": []
36     }
37   ]
38 }

```

2. The SHA-256 key can be generated online here:
  - <https://hash.online-convert.com/sha256-generator>

3. Add a link to the repo's URL and click "Convert file".

## Calculate a SHA hash with 256 bits

Advertisement

### Create your hashes online

Generate a SHA-256 hash with this free online encryption tool. To create a SHA-256 checksum of your file, use the upload feature. To further enhance the security of you encrypted hash you can use a shared key.


Upload and generate a SHA256 checksum of a file:


No file chosen

Or enter the text you want to convert to a SHA-256 hash:

Or enter the URL of the file you want to create a SHA256 hash from:

Or select a file from your cloud storage for a SHA256 conversion:

 Choose from Dropbox

 Choose from Google Drive

#### Optional settings

Shared secret key used for the HMAC variant (optional):

#### Save settings

Save settings as:



[\(Log in to activate\)](#)

(by clicking you confirm that you have understand and agree to our [terms](#))

To get further information of the SHA-256 algorithm, you can visit [FIPS 180-2: Secure Hash Standard \(SHS\)](#)

The input string encoding is expected to be in UTF-8. Different encoding will result in different hash values. Unicode is considered best practices.

#### 4. Copy the "hex:" value into the "checksum" section of the JSON

### Hash converter

#### Conversion Completed

Your hash has been successfully generated.

hex: 38d9ffec51d7274640f580a60d31dfb5afeaa38c950349aa45a8a313626d98e

HEX: 38D9FFEEC51D7274640F580A60D31DFB5AFEAA38C950349AA45A8A313626D98E

h:ex: 38:d9:ff:ee:c5:1d:72:74:64:0f:58:0a:60:d3:1d:fb:5a:fe:aa:38:c9:50:34:9a:a4:5a:8a:31:36:26:d9:8e

base64: ONn/7sUdcnRkD1gKYNMd+1r+qjjjUDSapFqKMTYm2Y4=

#### Want to go premium?

- ✔ convert large files up to 8GB
- ✔ up to 10 concurrent conversions
- ✔ 200 files per conversion
- ✔ higher priority conversion speed
- ✔ Ad-free pages

Upgrade now

#### You now have the following options:

1. Convert another file to **SHA256**
2. Consider a **donation** to support us

#### Are you a developer?

Did you know that [we also have an API](#) to use our service? This allows you to easily integrate our high-quality conversion features into your application.

We also offer a [PHP SDK](#) to quickly get you started using our service.

#### Feedback

Are you satisfied with the conversion? Send us your comment:

Your rating: ☐ Great ☐ Good ☐ Medium ☐ Bad ☐ Worse

If you want you can also send us a comment about our service.

Submit feedback

#### 5. The SHA-256 key must be updated everytime the package repository is changed.

6. If the checksum key doesn't match the zipped package signature, the IDE will throw an error.

