Software Setup and Upload Instructions

The following instructions detail the setup process for the eDNA UI and server which are essential for the operation of the sampler.

UI side information and set up instructions

Overview of the software used:

- **React**: JavaScript library for building user interface components.
- **Redux Toolkit**: State management tool, simplifying the use of Redux in the project.
- **Node.js**: JavaScript runtime environment for executing JavaScript code server-side.
- **npm (Node Package Manager)**: Package manager for managing JavaScript dependencies.
- **Tailwind CSS**: Utility-first CSS framework for rapidly building custom designs.
- **Webpack**: Module bundler for bundling JavaScript files and other assets.
- **TypeScript**: Superset of JavaScript, adding static types to the language.
- **Linting Tools**: Code quality tools to identify and fix formatting and syntax issues.
- **GitHub Actions:** CI/CD tool for automating build and linting processes.
- **React Router**: Library for handling routing in React applications.
- **VS Code (Visual Studio Code):** Code editor for development, supporting various programming languages.
- **Windows Subsystem for Linux (WSL)**: WSL is a special feature of windows that lets you use Linux alongside your Windows system.

Typographic conventions

Convention	**Description**
bold	Used for section titles and specific software names.
Red font	Represents Command line interface commands.
Light gray background	Represents pieces of code**.**
Normal text	Regular information and instructions.

System Requirements:

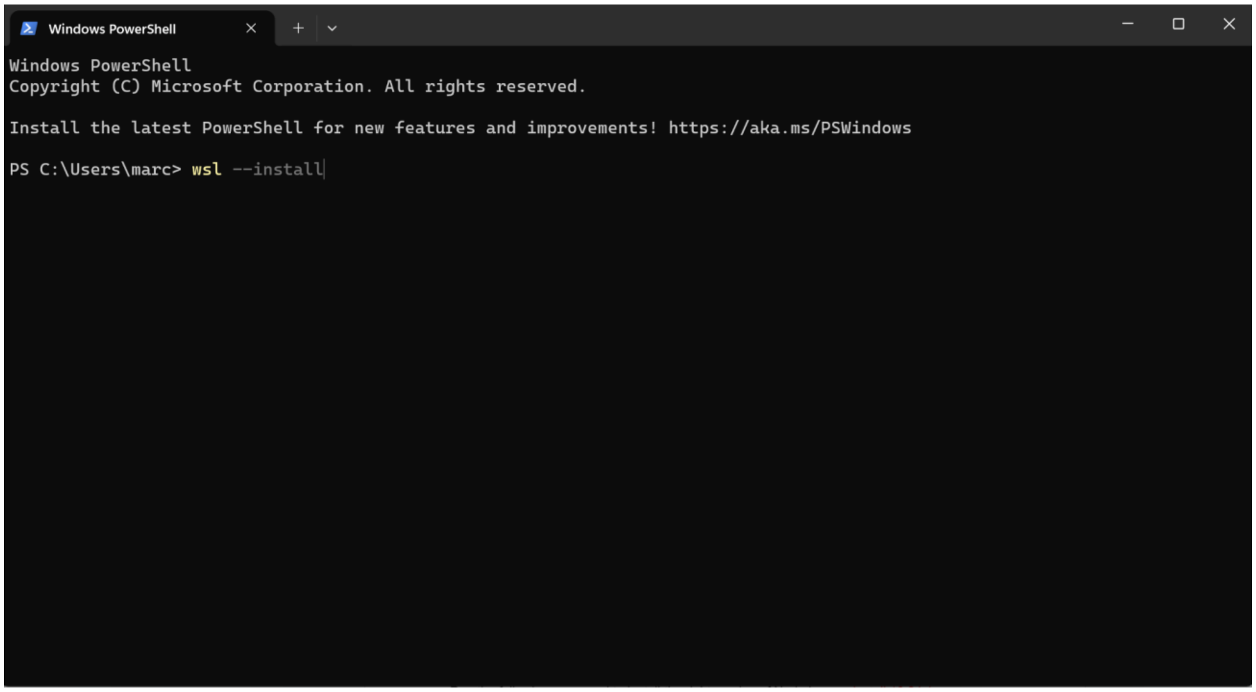
Operating System: Windows, Linux, Mac OS

Processor: A modern multi-core processor, such as Intel Core i3 or above, or an equivalent AMD processor.
RAM: A minimum of 4 GB RAM but 8GB is recommended for optimal performance.
Additional Requirements:

- For Windows users, Windows 10/11 with the ability to run Windows Subsystem for Linux (WSL) .
- Node.js and NPM for JavaScript runtime and package management.
- Node Version Manager (NVM) for managing multiple Node.js versions.

Setup

Windows: Open powershell (you can search for it in the start menu) and type the following command and press enter : wsl --install



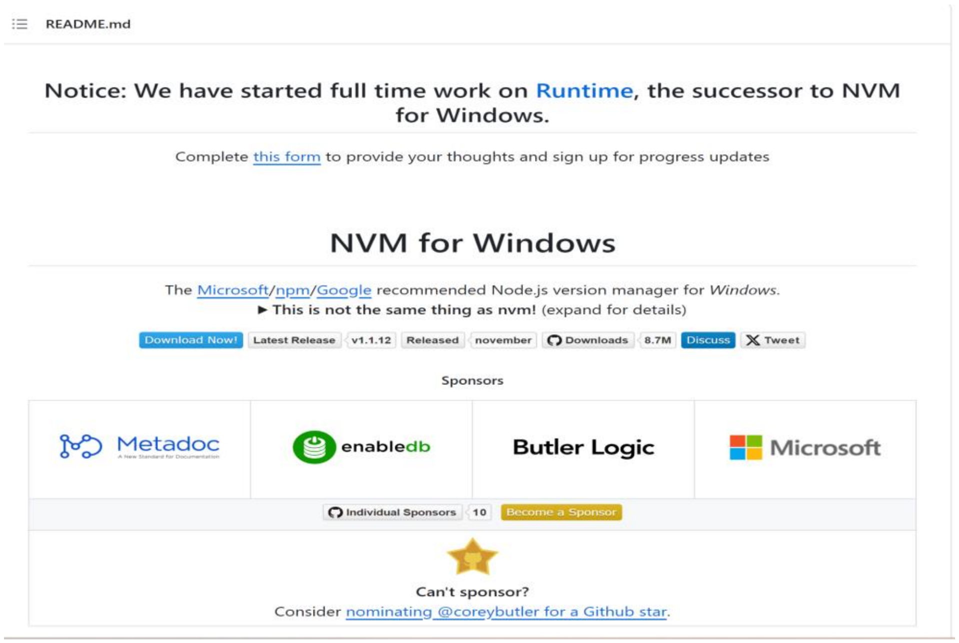
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\marc> wsl --install
```

Figure 1: Windows Powershell
This will set up everything needed to run Linux with Windows, and it will also instal a version of Linux called Ubuntu

Mac/Linux: Standard setup procedures apply.

Dependency Installation

Windows:
To install NVM, navigate to the following Github repository
https://github.com/coreybutler/nvm-windows
The download link should be located in the README.md file.
Click on the blue box on the left seen in figure 2.



README.md

Notice: We have started full time work on Runtime, the successor to NVM for Windows.

Complete this form to provide your thoughts and sign up for progress updates

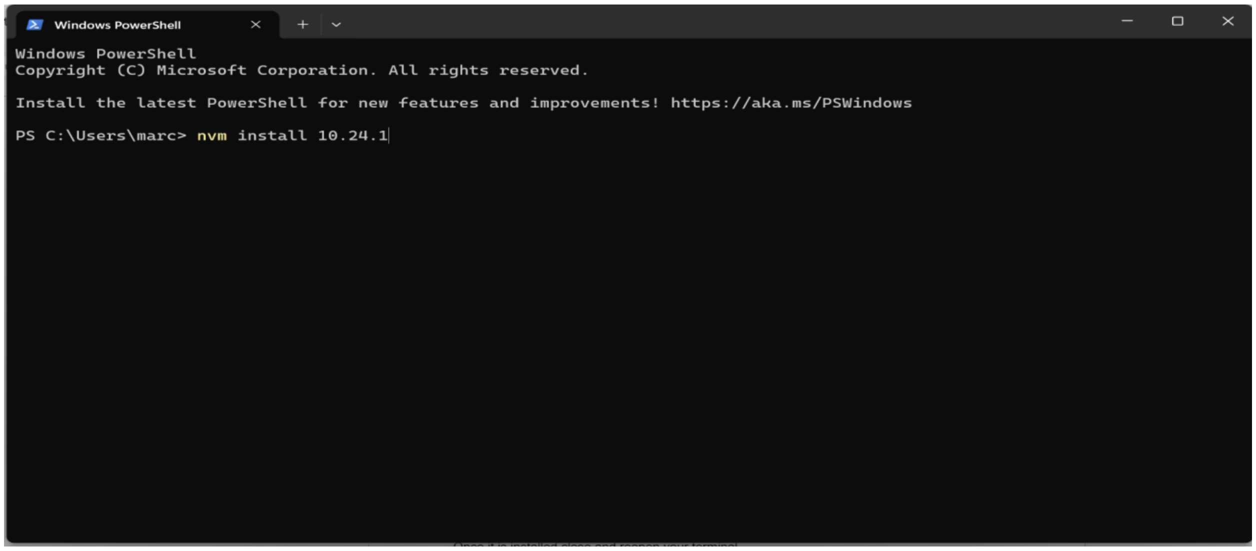
NVM for Windows

The Microsoft/npm/Google recommended Node.js version manager for *Windows*.
▶ This is not the same thing as nvm! (expand for details)

Can't sponsor?
Consider nominating @coreybutler for a Github star.

Figure 2: GitHub page for downloading Node Version Manager (NVM) with the download link highlighted.

Open the command prompt (Windows).
Type the following command to install the right version of Node.js: nvm install 10.24.1

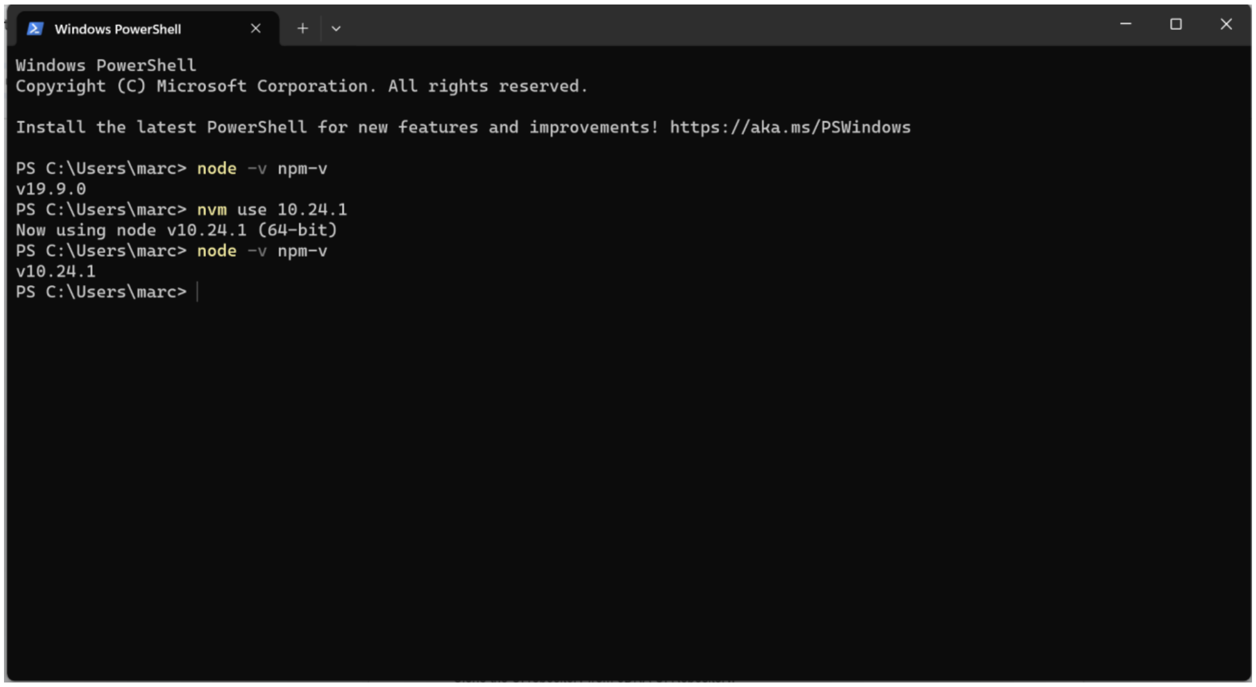


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\marc> nvm install 10.24.1
```

Figure 3: Illustrates the command prompt for installing the correct version of Node.js using NVM.

To use the installed version type: nvm use 10.24.1

You can confirm that Node.js and npm are installed correctly by running: node -v npm-v



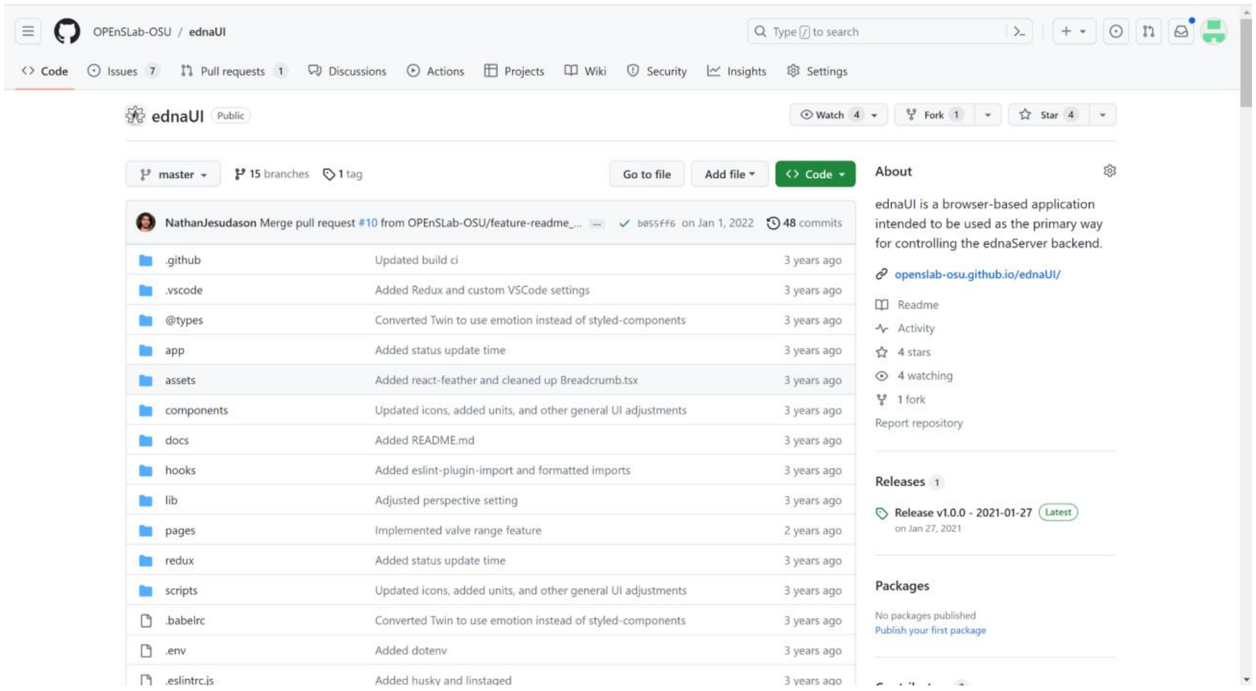
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\marc> node -v npm-v
v19.9.0
PS C:\Users\marc> nvm use 10.24.1
Now using node v10.24.1 (64-bit)
PS C:\Users\marc> node -v npm-v
v10.24.1
PS C:\Users\marc>
```

Figure 4: CLI command to check which version of node is being used

MacOS and Linux:
Install NVM by running the following command:curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.1/install.sh | bash
Once it is installed close and reopen your terminal
Run the following command to install the right version of Node.js: nvm install 10.24.1
To use the installed version: nvm use 10.24.1
You can confirm that Node.js and npm are installed correctly by running: node -v npm-v

Custom Software/Package Installation
To clone the eDNA UI repository first navigate to the link below.
https://github.com/OPEnSLab-OSU/ednaUI

The ednaUI repository should look something like this.



OPEnSLab-OSU / ednaUI

NathanJesudason Merge pull request #10 from OPEnSLab-OSU/feature-readme_... ✓ b055ff6 on Jan 1, 2022 48 commits

.github	Updated build ci	3 years ago
.vscode	Added Redux and custom VSCode settings	3 years ago
@types	Converted Twin to use emotion instead of styled-components	3 years ago
app	Added status update time	3 years ago
assets	Added react-feather and cleaned up Breadcrumb.tsx	3 years ago
components	Updated icons, added units, and other general UI adjustments	3 years ago
docs	Added README.md	3 years ago
hooks	Added eslint-plugin-import and formatted imports	3 years ago
lib	Adjusted perspective setting	3 years ago
pages	Implemented valve range feature	2 years ago
redux	Added status update time	3 years ago
scripts	Updated icons, added units, and other general UI adjustments	3 years ago
.babelrc	Converted Twin to use emotion instead of styled-components	3 years ago
.env	Added dotenv	3 years ago
.eslintrc.js	Added husky and linstaged	3 years ago

About

ednaUI is a browser-based application intended to be used as the primary way for controlling the ednaServer backend.

openslab-osu.github.io/ednaUI/

Readme
Activity
4 stars
4 watching
1 fork
Report repository

Releases 1

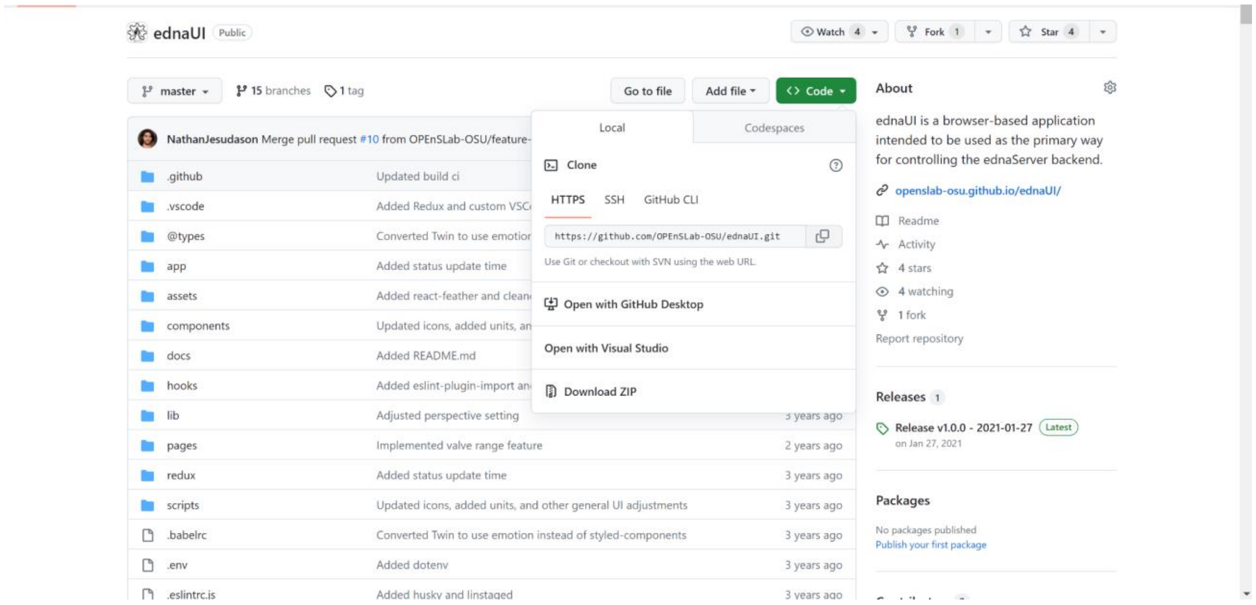
Release v1.0.0 - 2021-01-27 Latest
on Jan 27, 2021

Packages

No packages published
Publish your first package

Figure 5: ednaUI repository

Click on the green "code" button and copy the HTTPS link provided

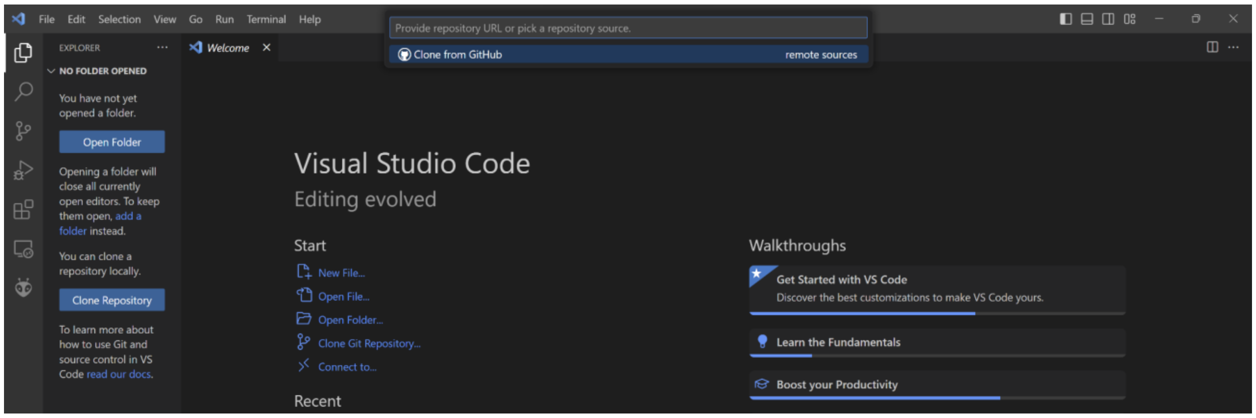


```
https://github.com/OPEnSLab-OSU/ednaUI.git
```

Use Git or checkout with SVN using the web URL.

Figure 6: The https link to be copied to your machine

Once copied, open VScode and click on “Clone Repository”. Paste the URL where prompted.



Provide repository URL or pick a repository source.

You have not yet opened a folder.

Opening a folder will close all currently open editors. To keep them open, add a folder instead.

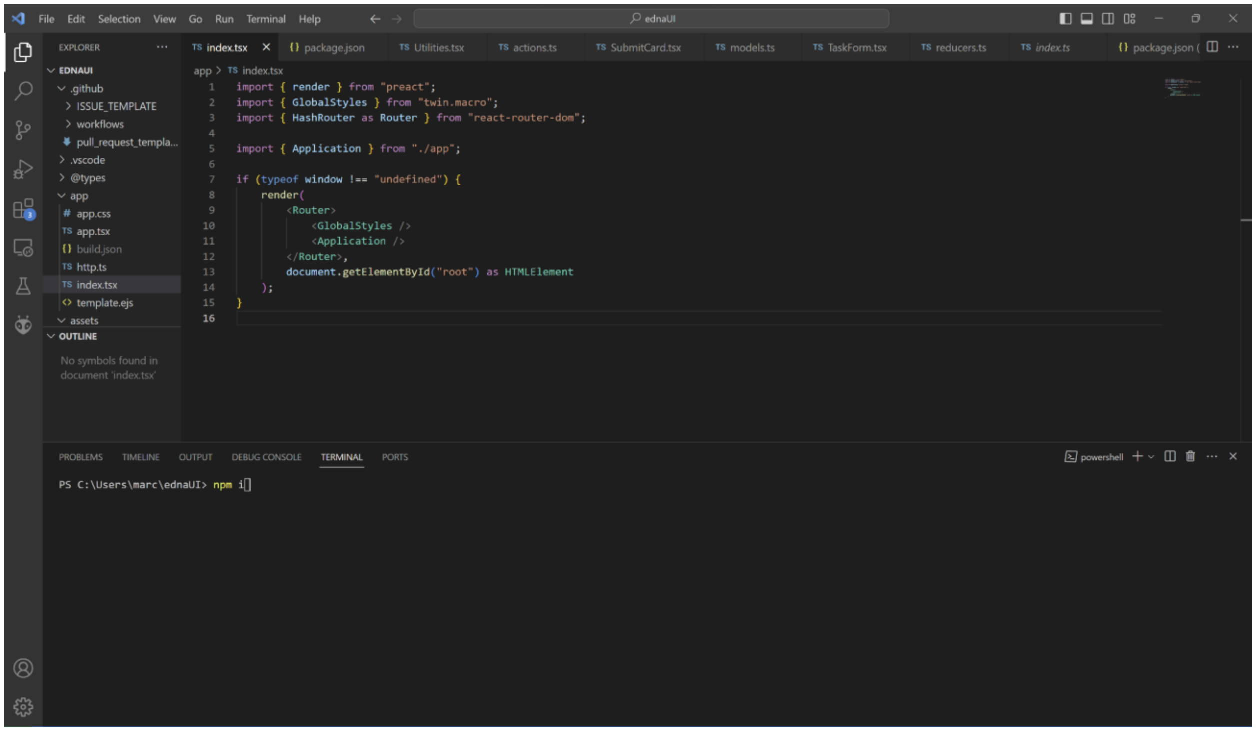
You can clone a repository locally.

To learn more about how to use Git and source control in VS Code read our docs.

Figure 7: Cloning the eDNA UI repository using VScode.

You can also copy the repository from the command line with the following command: git clone https://github.com/OPEnSLab-OSU/ednaUI.git

Navigate to the eDNA UI folder either by clicking “open folder” or using the command line. Once there run the following command: npm i



```
import { render } from "preact";
import { GlobalStyles } from "twin.macro";
import { HashRouter as Router } from "react-router-dom";

import { Application } from "./app";

if (typeof window !== "undefined") {
    render(
        <Router>
            <GlobalStyles />
            <Application />
        </Router>,
        document.getElementById("root") as HTMLElement
    );
}
```

```
PS C:\Users\marc\ednaUI> npm i
```

Figure 8: Running npm i will install the packages needed.

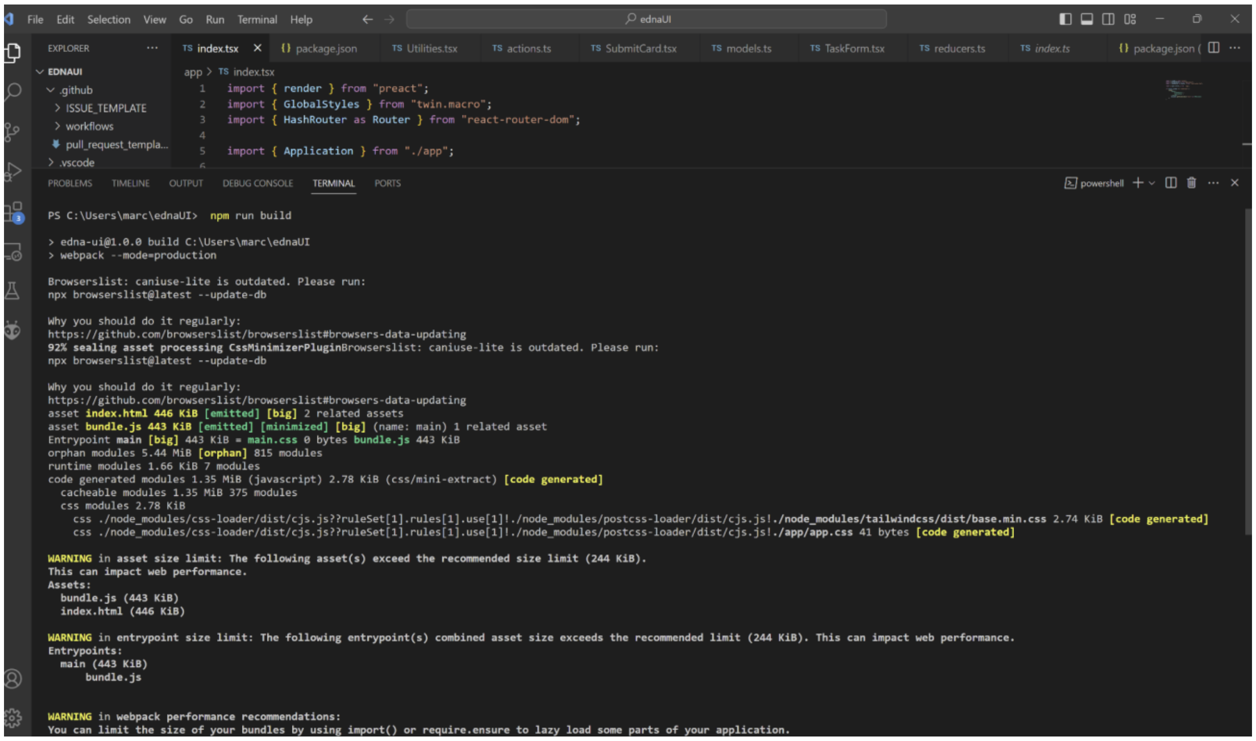
Useful NPM commands

npm run build: Builds UI files in /dist.
npm run watch: Builds development UI files in /dist and automatically rebuilds upon changes.
npm run dev: Build the UI in development mode.
npm run lint: Detects lint/syntax errors.
npm run lint-fix: Automatically fixes lint/syntax errors.
npm run start: Builds and serves UI locally, useful for seeing changes you make to the UI.

Running the Software

From VScode navigate to the folder containing the eDNA UI.
To create the UI files you will need run the following command: npm run build

Note: When you run your code, you may notice several warnings, similar to those in Figure 9. Do not be concerned; these warnings won't impact the functioning of your code.

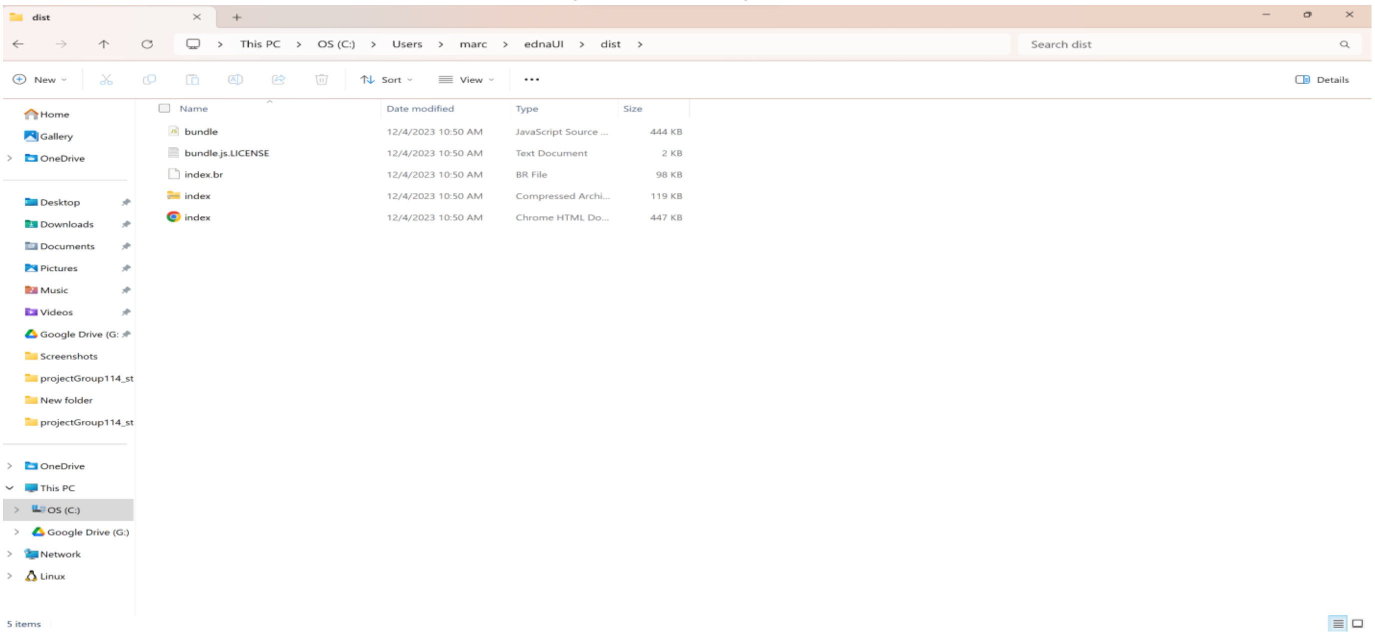


```
import { render } from "preact";
import { GlobalStyles } from "twin.macro";
import { HashRouter as Router } from "react-router-dom";

import { Application } from "./app";
```

```
PS C:\Users\marc\ednaUI>  npm run build

> edna-ui@1.0.0 build C:\Users\marc\ednaUI
> webpack --mode=production

Browserslist: caniuse-lite is outdated. Please run:
npx browserslist@latest --update-db

Why you should do it regularly:
https://github.com/browserslist/browserslist#browsers-data-updating
92% sealing asset processing CssMinimizerPluginBrowserslist: caniuse-lite is outdated. Please run:
npx browserslist@latest --update-db

Why you should do it regularly:
https://github.com/browserslist/browserslist#browsers-data-updating
asset index.html 446 KiB [emitted] [big] 2 related assets
asset bundle.js 443 KiB [emitted] [minimized] [big] (name: main) 1 related asset
Entrypoint main [big] 443 KiB = main.css 0 bytes bundle.js 443 KiB
orphan modules 5.44 MiB [orphan] 815 modules
runtime modules 1.66 KiB 7 modules
code generated modules 1.35 MiB (javascript) 2.78 KiB (css/mini-extract) [code generated]
  cacheable modules 1.35 MiB 375 modules
  css modules 2.78 KiB
    css ./node_modules/css-loader/dist/cjs.js??ruleSet[1].rules[1].use[1]!./node_modules/postcss-loader/dist/cjs.js!./node_modules/tailwindcss/dist/base.min.css 2.74 KiB [code generated]
    css ./node_modules/css-loader/dist/cjs.js??ruleSet[1].rules[1].use[1]!./node_modules/postcss-loader/dist/cjs.js!./app/app.css 41 bytes [code generated]

WARNING in asset size limit: The following asset(s) exceed the recommended size limit (244 KiB).
This can impact web performance.
Assets:
  bundle.js (443 KiB)
  index.html (446 KiB)

WARNING in entrypoint size limit: The following entrypoint(s) combined asset size exceeds the recommended limit (244 KiB). This can impact web performance.
Entrypoints:
  main (443 KiB)
      bundle.js


WARNING in webpack performance recommendations:
You can limit the size of your bundles by using import() or require.ensure to lazy load some parts of your application.
```

Figure 9: Command line after running npm run build.

This will create a folder called dist containing the following files.



Name	Date modified	Type	Size
bundle	12/4/2023 10:50 AM	JavaScript Source ...	444 KB
bundle.js.LICENSE	12/4/2023 10:50 AM	Text Document	2 KB
index.br	12/4/2023 10:50 AM	BR File	98 KB
index	12/4/2023 10:50 AM	Compressed Archi...	119 KB
index	12/4/2023 10:50 AM	Chrome HTML Do...	447 KB

Figure 10: Inside of the dist folder containing the files needed to load on the sampler

Uploading the UI to the sampler

To upload the UI to the sampler save the content of the dist folder into an SD card and insert it onto the board. The following files need to be saved:

- index
- Index.br
- bundle
- bundle.js.LICENSE,

Troubleshooting

You may find your code not running due to conflicts between dependencies. To avoid this make sure to install node v10 and not the latest version of node.

Server side information and set up instructions

Overview of Software Used:

- VS Code (Visual Studio Code): Code editor for development, supporting various programming languages.

- PlatformIO: Extension for VSCode, used for IoT development.
- Git: Version control system for tracking changes in source code.

Required Toolings

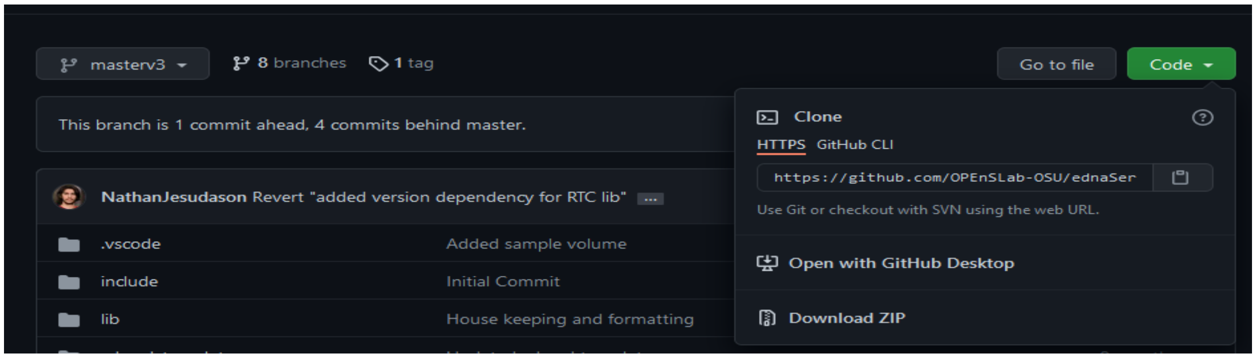
Install VSCode

Install PlatformIO extension `platformio.platformio-ide`

Download ZIP Files from GitHub

Get the GitHub repository branch link that is relevant to your version in the Sampler History Document under `Firmware`. (As of the writing of this guide masterv4.2 is the branch you will need to be using.)

Download a ZIP file for code branch masterv4.2, and then unzip.



This branch is 1 commit ahead, 4 commits behind master.

NathanJesudason Revert "added version dependency for RTC lib"

.vscode	Added sample volume
include	Initial Commit
lib	House keeping and formatting

https://github.com/OPEnSLab-OSU/ednaSer

Use Git or checkout with SVN using the web URL.

Figure 11: eDNA server GitHub page with the download link.

Download the ZIP file for the Git Submodule of the code. After unzipping this file, copy and paste the contents of the main folder inside of the `lib/Framework` folder inside of the folder you unzipped in the previous step

Open the masterv4.2 code inside of VSCode by going to `File => Open Folder...`Next, click on `platformio.ini` configuration file located in the file explorer.

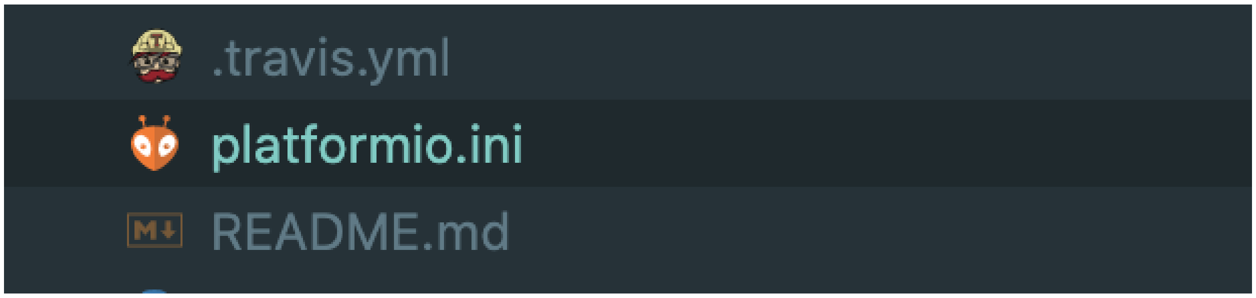


Figure 12: platformio.ini configuration file.

Next, please comment the `[env:debug]` section and uncomment `[env:live]`. The file should look like the following. Notice that we have `;` in front of the commented line.

```
;[env:debug]

 ; build_type = debug

 ;build_unflags = -std=gnu++11

 ;build_flags = -D DEBUG=1 -Wall -Wno-unknown-pragmas -std=c++14



 ;  [env:release]

 ; build_unflags = -std=gnu++11

 ; build_flags = -D RELEASE=1 -Wall -Wno-unknown-pragmas -std=c++14



 [env:live]

 build_unflags = -std=gnu++11

 build_flags = -D LIVE=1 -Wall -Wno-unknown-pragmas -std=c++14
```

`optional` Optionally you can change the server name and password by going to `src/configuration.hpp`. It is recommended if you have multiple samplers to change the `SERVER_NAME` to `"ednaServer-###"`, where `###` is the serial number on the electronics box of the sampler. Otherwise, the server will default to the following values:

```
#define SERVER_NAME     "ednaServer"

#define SERVER_PASSWORD "password"
```

You are now ready to upload the firmware to the microcontroller. After connecting the micro-controller to your computer, please click on the `right-arrow` button located at the bottom of the status bar.

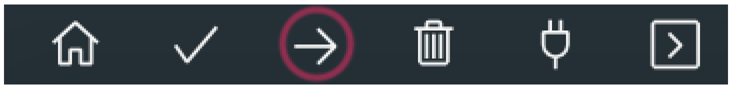


Figure 13: Status bar

Git Method (recommended for development or to users with programming experience)

Clone Git Repository

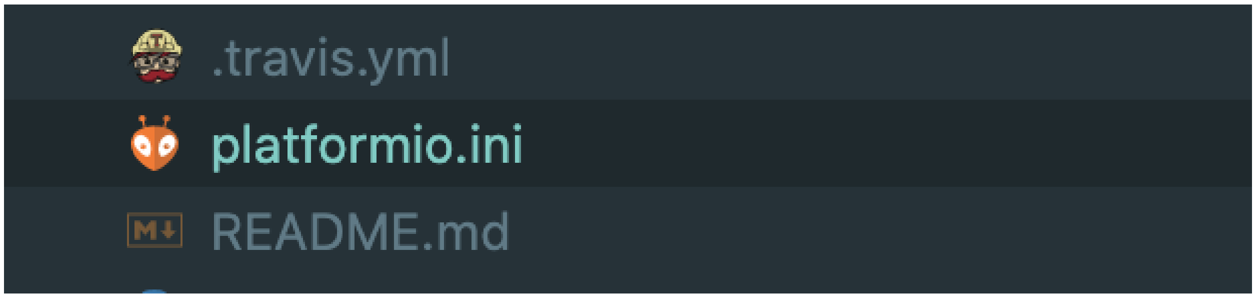
The following code clone development git repo to your local computer. Switch to the relevant branch based on the Sampler History Document. The Master branch, which the repo is on by default, is the latest version.

```
git clone https://github.com/OPEnSLab-OSU/ednaServer

cd ednaServer

git checkout <branch>
```

Initialize git submodule. This will pull in another git repo that the server relies on.

```
git submodule init

git submodule update

git submodule foreach "git checkout develop"

code .
```

You should now have VSCode opened automatically. Next, click on `platformio.ini` configuration file located in the file explorer.



Next, please comment the `[env:debug]` section and uncomment `[env:live]`. The file should look like the following. Notice that you have `;` in front of the commented line.

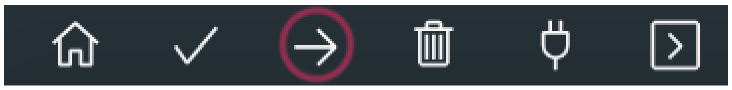
```
;[env:debug]

; build_type = debug
```

```
;build_unflags = -std=gnu++11

;build_flags = -D DEBUG=1 -Wall -Wno-unknown-pragmas -std=c++14

; [env:release]

; build_unflags = -std=gnu++11

; build_flags = -D RELEASE=1 -Wall -Wno-unknown-pragmas -std=c++14

[env:live]

build_unflags = -std=gnu++11

build_flags = -D LIVE=1 -Wall -Wno-unknown-pragmas -std=c++14
```

`optional` Optionally you can change the server name and password by going to `src/configuration.hpp`. It is recommended if you have multiple samplers to change the `SERVER_NAME` to `"ednaServer-###"`, where `###` is the serial number on the electronics box of the sampler. Otherwise, the server will default to the following values:

```
#define SERVER_NAME     "ednaServer"

#define SERVER_PASSWORD "password"
```

You are now ready to upload the firmware to the microcontroller. Please click on the `right-arrow` button located at the bottom of the status bar.



Troubleshooting

SD file corruption

What causes this?

It's possible that if power is interrupted to the sampler while it is writing to a SD file, meaning that file won't be written to properly.

What does this problem look like?

If the sampler doesn’t work on startup and seems to freeze (lights aren’t blinking, UI can’t be connected to), then it is possible that this problem is occurring

How to resolve this issue?

The easiest way to resolve this issue is to remove all files from the SD card and then put on new files. You may also run the sampler through debug mode following the firmware installation guide, and then check the serial monitor for when the sampler says there is an error.

UI redirecting to https (Firefox)

What causes this?

By default, Firefox attempts to connect websites using the HTTPS protocol, which is not supported by the eDNA sampler.

What does this problem look like?

When trying to connect to 192.168.1.1, you are automatically redirected to https://192.168.1.1, and can’t connect to the UI

How to resolve this issue?

Go to `about:config` (same as entering any web address), then search for `network.stricttransportsecurity.preloadlist` and make it set to false. Then, search for `browser.fixup.fallback-to-https` and set it to false as well.

PlatformIO (Visual Studio Code Extension) is not working

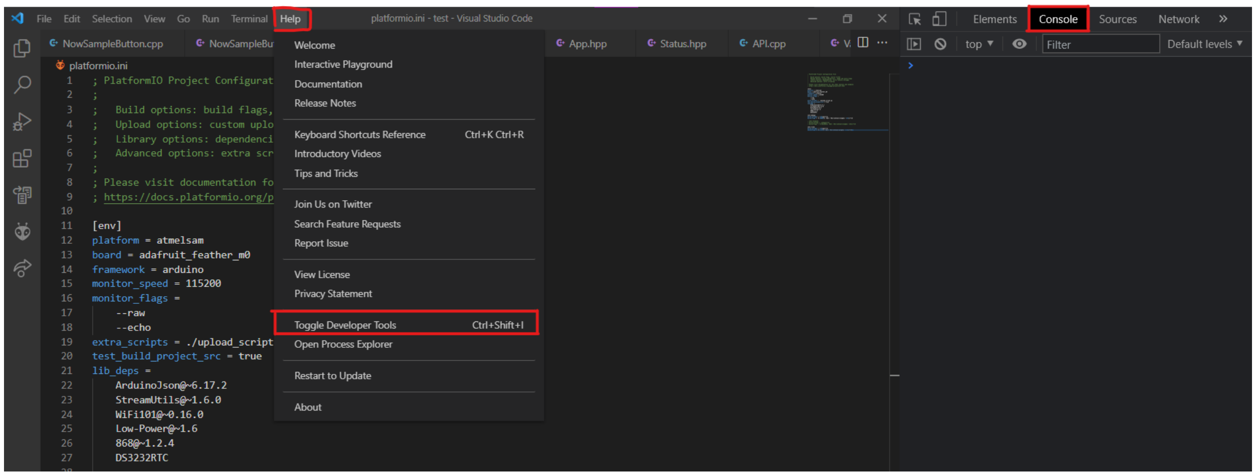
What causes this?
There could be a variety of issues that are causing this.

What does this problem look like?

There are a variety of possible error messages, but they all originated from PlatformIO. For example: when trying to upload code, PlatformIO may complain that it cannot start its home server, or it may give an error message that it failed to install PlatformIO IDE.

How to resolve this issue?

Often, the best thing to do is to open the Visual Studio Code Console and see what errors it prints. First, go to the help tab on the top menu of VSCode. Then, press Toggle Developer Tools. This will open a new panel on the right side of your screen, and you’ll want to go to the top menu on this panel and click on Console. Reading through some of the error messages will tell you what the error is. For example, it could be that there's a missing python package. Because there's a variety of possible issues, if you feel uncomfortable solving them yourself, feel free to reach out to us and we can help.



```
; PlatformIO Project Configurat
;
;   Build options: build flags,
;   Upload options: custom uplo
;   Library options: dependenci
;   Advanced options: extra scr
;
; Please visit documentation fo
; https://docs.platformio.org/p

[env]
platform = atmelsam
board = adafruit_feather_m0
framework = arduino
monitor_speed = 115200
monitor_flags =
    --raw
    --echo
extra_scripts = ./upload_script
test_build_project_src = true
lib_deps =
    ArduinoJson@~6.17.2
    StreamUtils@~1.6.0
    WiFi101@~0.16.0
    Low-Power@~1.6
    868@~1.2.4
    DS3232RTC
```

Figure 14: Developer Console in VSCode.