Project Structure

Microcontroller

All microcontroller code is located in the repository's "main" folder. The entry point of the program is the file main.ino, which calls all other parts of the microcontroller codebase.

Constants and Globals

There are 2 files dedicated to initializing variables and/or macros: globals.h and nanolux_types.h. Globals contains mostly global variables that aren't utilized in main.ino, in addition to defining the Pattern struct and its accompanying array. nanolux_types contains nearly macros utilized as constant values by the codebase.

Audio Analysis

main.ino calls the audio analysis pipeline, which consists of audio processing functions from the files core_analysis.h and ext_analysis.h. The core_analysis files perform the basic audio analysis that most patterns pull from. The ext_analysis files perform audio analysis processes that are used by only a few patterns, but are more complex.

Utility

The nanolux_util files contain a variety of "helper" functions. The functions in these files are used all throughout the program and deal with hardware and Nanolux-specific custom mathematical operators, among other actions.

Patterns

The files patterns.h and patterns.cpp contain the actual code to run the pattern algorithms themselves. Their function pointers are handled by the Pattern array in globals.h, so this file is rarely directly addressed in the code. These files pull from the palettes.h to get color palettes for a few pattern algorithms.

Storage

The files storage.h and storage.cpp contain the code needed to save and load system setting configurations, patterns, and strip configurations. As stated before, storage.h contains the three data structures used for containing these three data categories.

Networking

Nanolux network communications are managed in the files api.h and webServer.h. The file api.h contains the request handlers for controlling pattern and strip settings, while webServer.h contains the code initializing the web app, bridging to local network connections, starting the built-in router, and manages the API for updating networking-specific device settings. You can find more on the API on its accompanying wiki page.

Web App

TBD