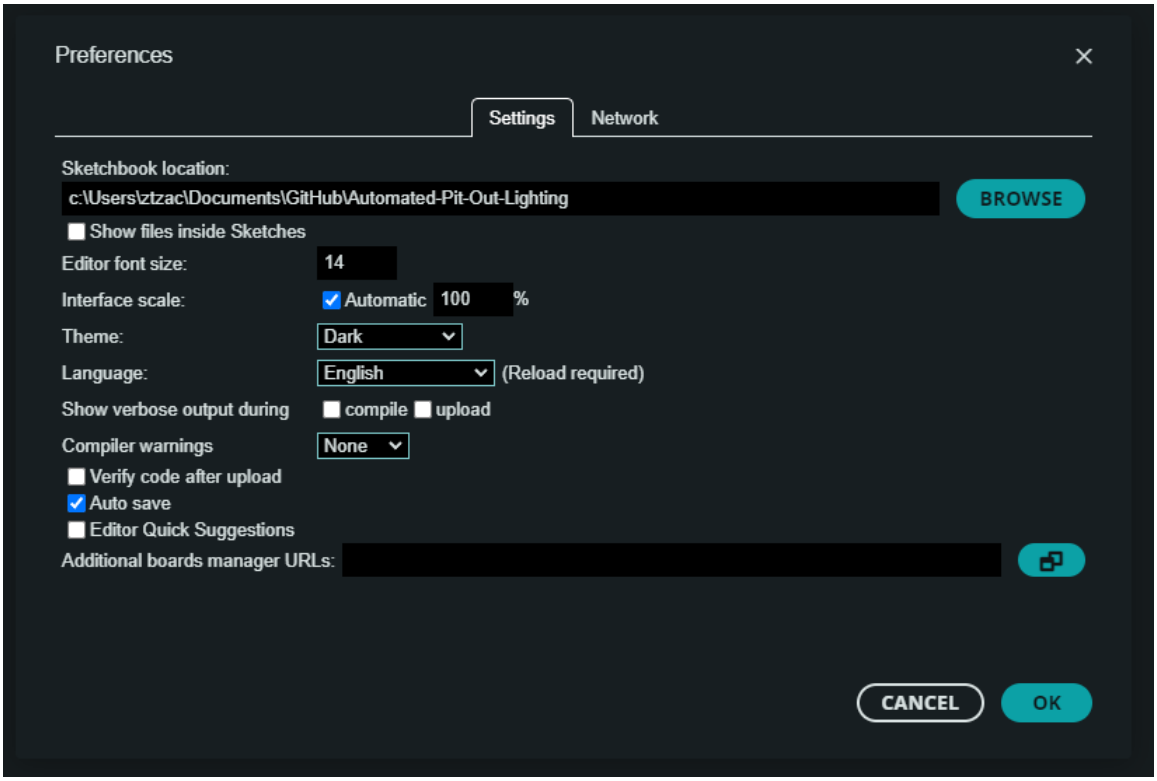
Software Environment Setup & Build Instructions

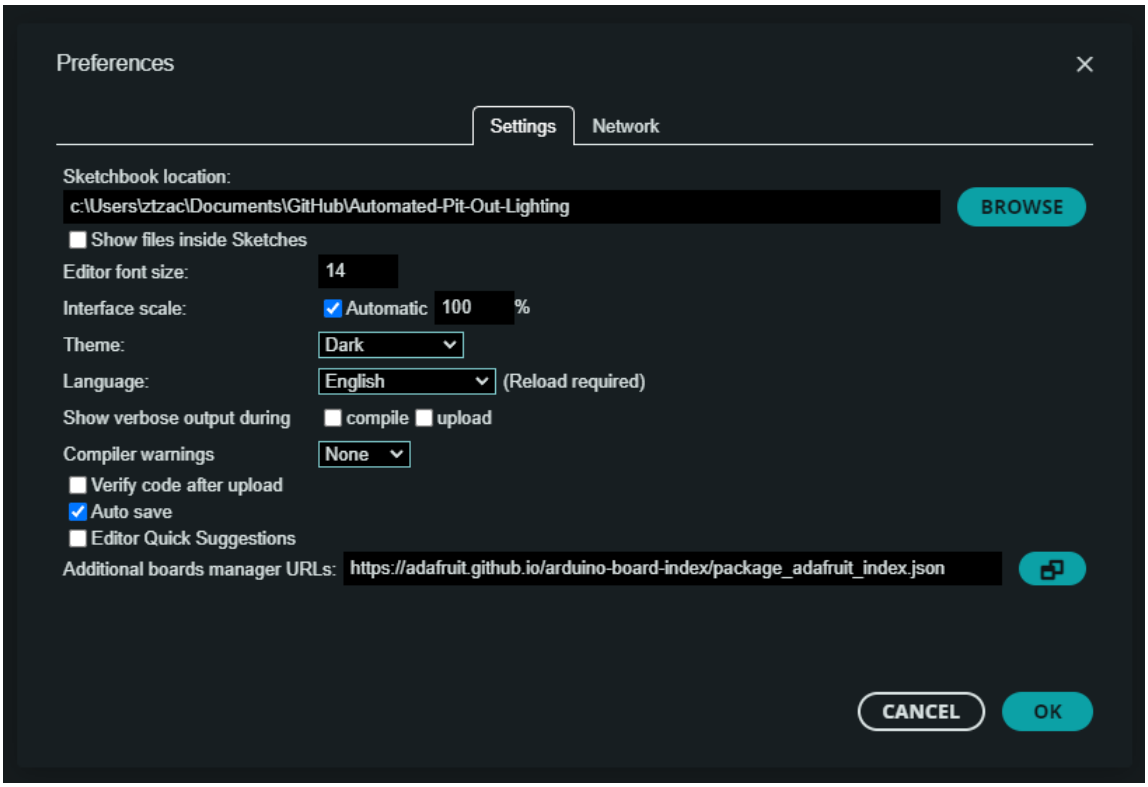
Table of Contents

Arduino IDE Setup

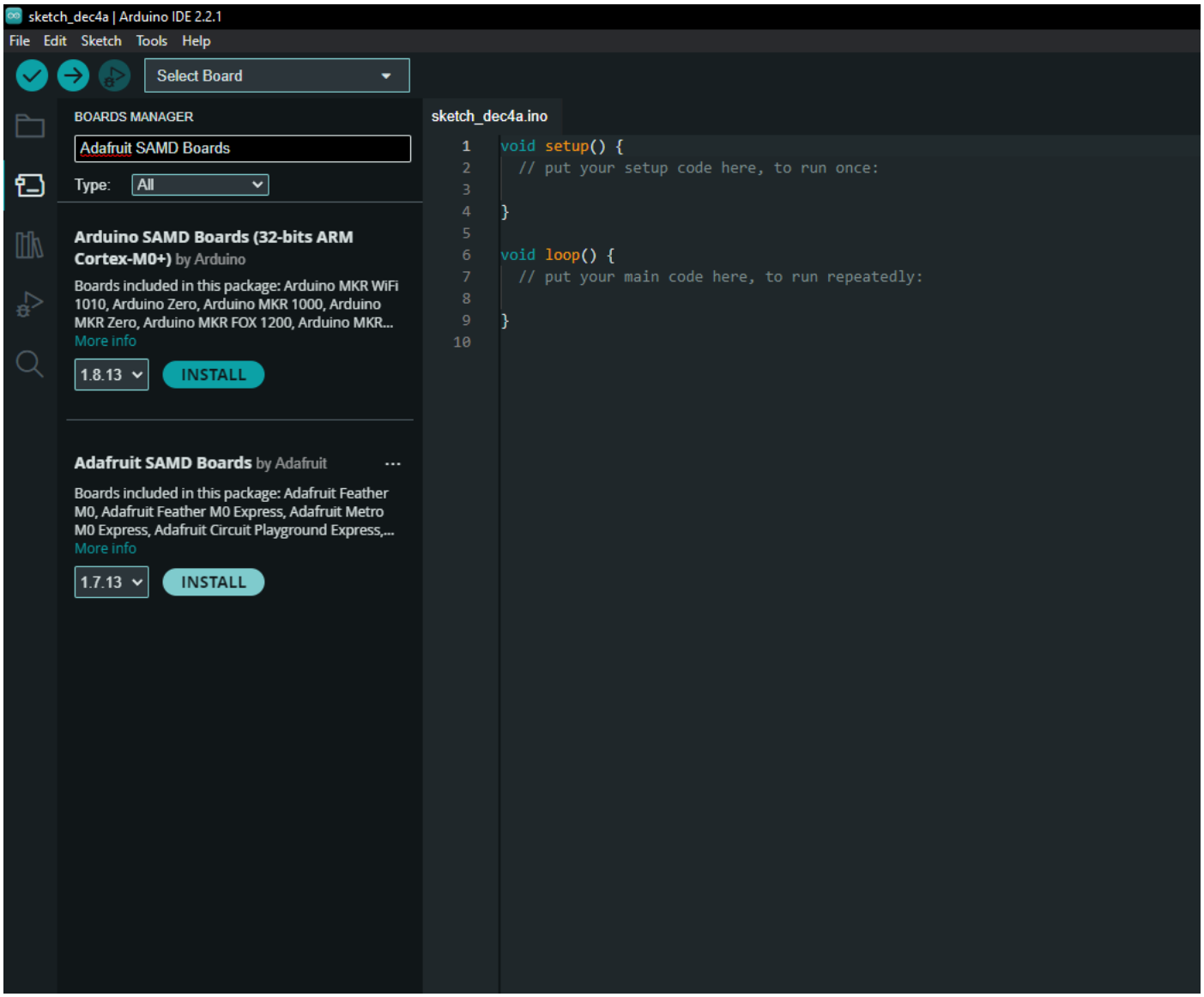
1. Download and install the Arduino IDE (the installer can be accessed here here
2. In the Arduino IDE, click File -> Preferences
3. In the Preferences menu, change the sketchbook location to the base of the project directory tree (this will cause the Arduino IDE to include the libraries in the /libraries/ folder when compiling).
    - Alternatively, copy the libraries from the project /libraries folder into your Arduino IDE libraries folder.



4. In the preferences menu, add the following link to the "Additional Boards Manager URLs" field: https://adafruit.github.io/arduino-board-index/package_adafruit_index.json



5. Install the Adafruit SAMD boards through the board manager



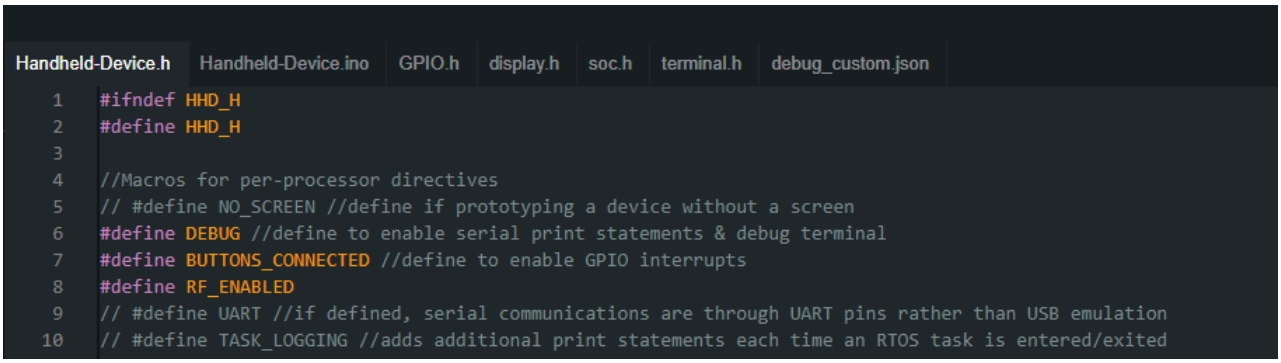
WARNING: if "Adafruit SAMD Boards" does not show up in the search results, you may not have completed the prior step correctly.

Setting up file structure

Coming soon!

Customizing Software Build

Some software features have been implemented such that they are configurable at build time. This configuration is accomplished through the use of macros. In the "`<project name>`.h" file, there are a list of `#define <MACRO_IDENTIFIER>` statements used to define these macros. If a macro is defined, the code corresponding to that software feature will be included at compilation time. In order to exclude a software feature, the macro definition statement must be commented out through the use of two consecutive slash characters `//` added to the beginning of that line of code. Below is an example of macro definition and commenting out macro definitions.



The file being edited is the Handheld Device, so the macro definitions are found in "Handheld-Device.h" file. In the example above, the macros `DEBUG`, `BUTTONS_CONNECTED`, and `RF_ENABLED` were defined; however, the macros `NO_SCREEN`, `UART`, and `TASK_LOGGING` were not defined as they were commented out using two slashes.

Below are a series of tables showing what configurable software features different macros correspond to for each subsystem.

> **Warning: Some macros may not have been maintained and updated as software development progressed. If issues are experienced, convert back to the default macro definitions provided at time of delivery.**

Handheld Device

Macro	Feature	Purpose
BUTTONS_CONNECTED	Includes code related to button input handling.	Allows software to be built with button handling
DEBUG	Enables serial print statements and debug terminal.	Used for debugging issues experienced.
IDLE_ENABLED	Includes code that puts the subsystem into sleep mode.	Configuring power saving options.
NO_SCREEN	Excludes code that initializes the screen and displays information to it.	Allows one to debug the software through the use of a Feather M0 controller that is not connected to a screen.
RF_ENABLED	Enables the Lo-Ra modem driver code.	Used for debugging to isolate issues.
TASK_LOGGING	Logs when FreeRTOS tasks are entered and exited.	For debugging purposes.
UART	Redirects serial outputs to the UART pins on the board rather than through the serial converter attached to the USB port.	Configuring serial output.

Pit Out Light

Macro	Feature	Purpose
BUTTONS_CONNECTED	Includes code related to button input handling.	Allows software to be built with button handling
DEBUG	Enables serial print statements and debug terminal.	Used for debugging issues experienced.
IDLE_ENABLED	Includes code that puts the subsystem into sleep mode.	Configuring power saving options.
LIGHTS_CONNECTED	Includes code related to light driving.	Allows software to be built without light driving code.
NO_SCREEN	Excludes code that initializes the screen and displays information to it.	Allows one to debug the software through the use of a Feather M0 controller that is not

Macro	Feature	Purpose
		connected to a screen.
RF_ENABLED	Enables the Lo-Ra modem driver code.	Used for debugging to isolate issues.
TASK_LOGGING	Logs when FreeRTOS tasks are entered and exited.	For debugging purposes.
UART	Redirects serial outputs to the UART pins on the board rather than through the serial converter attached to the USB port.	Configuring serial output.

Vehicle Detection Device

Macro	Feature	Purpose
BUTTONS_CONNECTED	Includes code related to button input handling.	Allows software to be built with button handling
DEBUG		
NO_SCREEN	Excludes code that initializes the screen and displays information to it.	Allows one to debug the software through the use of a Feather M0 controller that is not connected to a screen.
RF_ENABLED		
TASK_LOGGING		
UART		

Additionally, there are macros which define key values of the system. These macros have an additional value following the definition. Below is an example of these kinds of macros from "Pit-Out-Light.h."

```
//Macros for system parameters
#define DURATION_MAX 24
#define DURATION_MIN 1
#define DURATION_INC 5
#define BAUD_RATE 115200
#define PULSE_DELAY 1000
#define IDLE_START_MILLISECONDS 30000 //30 Seconds
```

Below are a series of tables showing what configurable system parameters different macros correspond to for each subsystem.

Handheld Device

Macro	Value	Unit	Purpose
MAX_QUEUED_REQUESTS	How many packets can be queued at any given time for transmit.	# of queued requests	Configurable for memory saving purposes.
REFRESH_DELAY	How much time will elapse before the display updates.	Milliseconds	Configurable for power management and optimization purposes.
BAUD_RATE	Sets the serial baud rate.	Baud/s	Allows for reconfiguration of the serial baud rate.

Macro	Value	Unit	Purpose
MAX_TRANSMIT_ATTEMPTS	How many times the device will attempt to transmit a packet before it drops the packet.	# of attempts	For optimizing speed and power consumption vs. reliability
SOC_MONITORING_DELAY	How frequently the battery voltage will be sampled.	Milliseconds	For power optimization purposes.
IDLE_START_MILLISECONDS	How much time will elapse prior to the subsystem being put into idle mode.	Milliseconds	For configuring to user preference and power optimization.

Pit Out Light

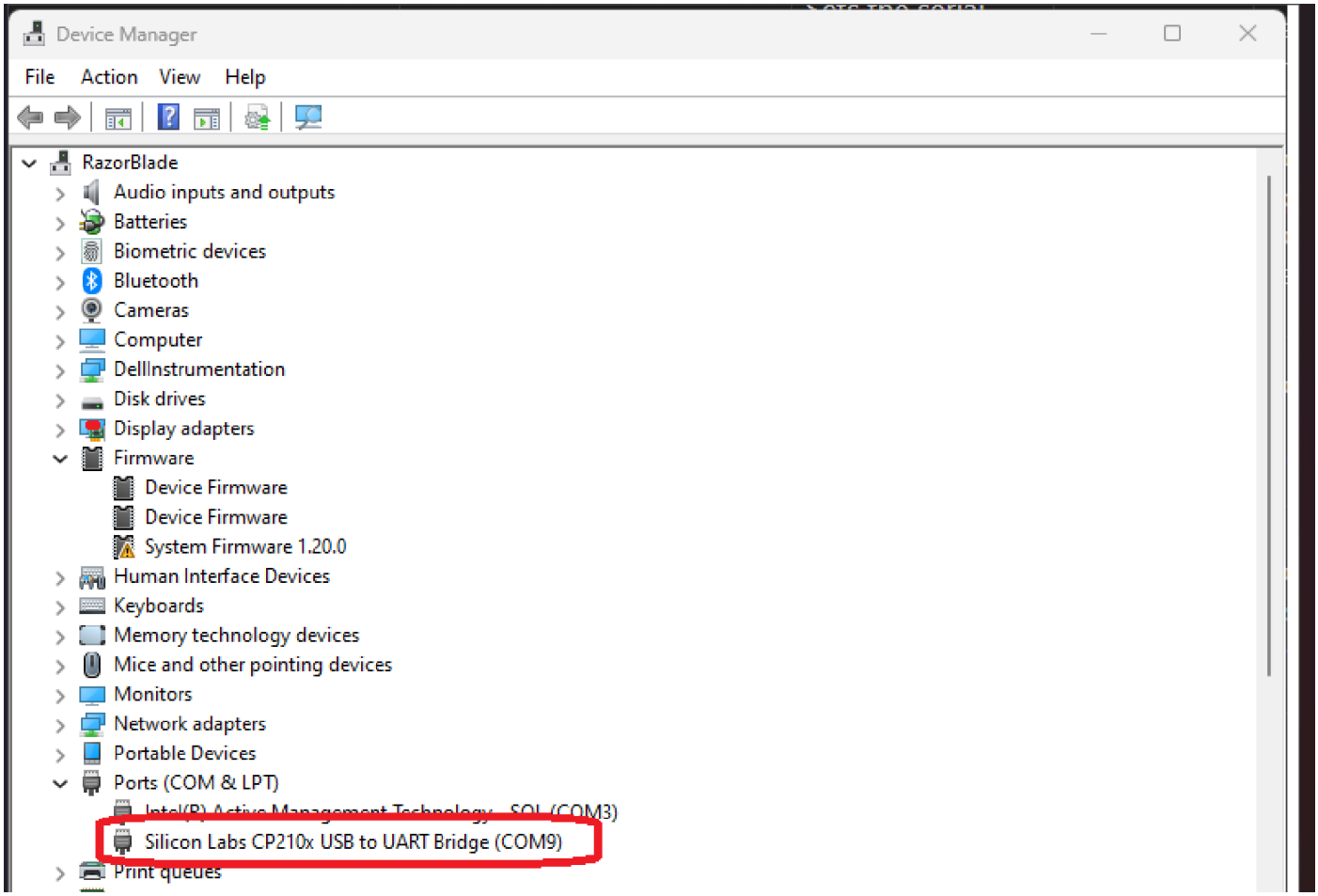
Macro	Value	Unit	Purpose
DURATION_MAX	Maximum override timer value.	Integer multiple of the DURATION_RESOLUTION	For reconfiguring the override timeout.
DURATION_MIN	Maximum override timer value.	Integer multiple of the DURATION_RESOLUTION	For reconfiguring the override timeout.
DURATION_INC	The override timer increment.	Milliseconds	Allows for reconfiguration of the serial baud rate.
BAUD_RATE	Sets the serial baud rate.	Baud/s	Allows for reconfiguration of the serial baud rate.
PULSE_DELAY	How much time the green light is pulsed on for.	Milliseconds	For optimizing speed and power consumption vs. reliability
IDLE_START_MILLISECONDS	How much time will elapse prior to the subsystem being put into idle mode.	Milliseconds	For configuring to user preference and power optimization.

Vehicle Detection Device

Macro	Value	Unit	Purpose
MAX_QUEUED_REQUESTS	How many packets can be queued at any given time for transmit.	# of queued requests	Configurable for memory saving purposes.
REFRESH_DELAY	How much time will elapse before the display updates.	Milliseconds	Configurable for power management and optimization purposes.
BAUD_RATE	Sets the serial baud rate.	Baud/s	Allows for reconfiguration of the serial baud rate.
MAX_TRANSMIT_ATTEMPTS	How many times the device will attempt to transmit a packet before it drops the packet.	# of attempts	For optimizing speed and power consumption vs. reliability
SOC_MONITORING_DELAY	How frequently the battery voltage will be sampled.	Milliseconds	For power optimization purposes.
IDLE_START_MILLISECONDS	How much time will elapse prior to the subsystem being put into idle mode.	Milliseconds	For configuring to user preference and power optimization.

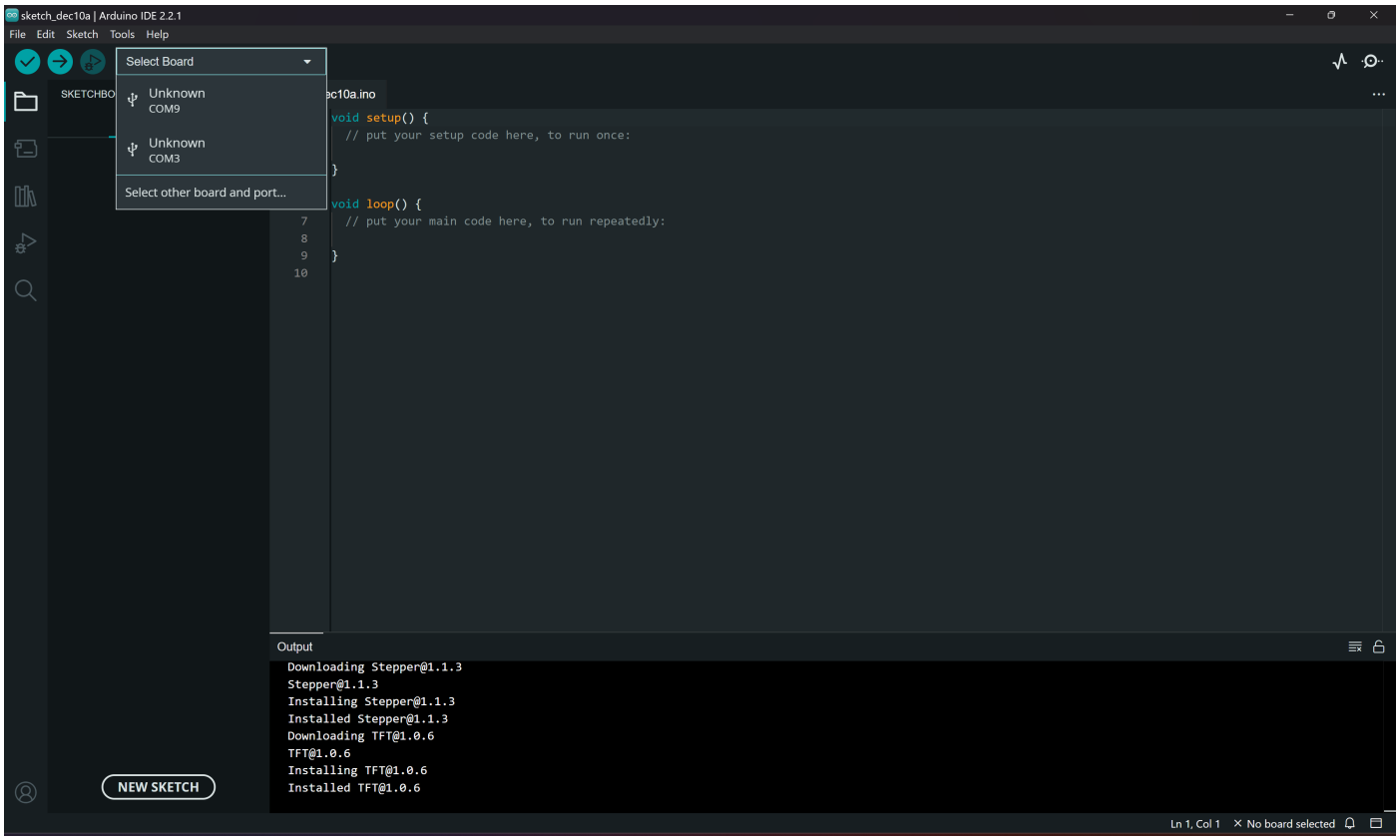
Software Upload

1. Find the Feather device in Windows' "Device Manager" application to find the COM port of the device.

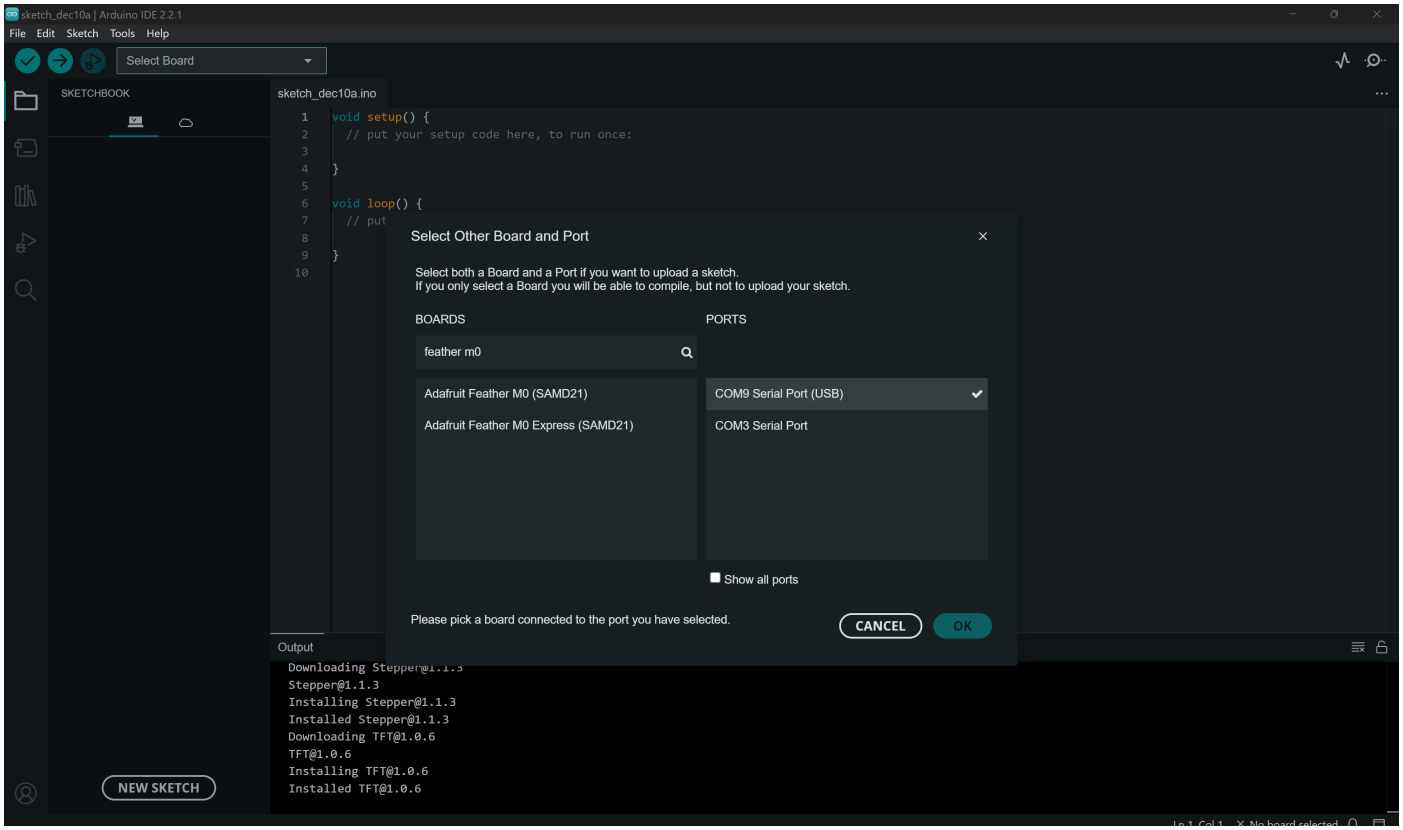


⚠ **The name and COM port that appear on your computer may vary from what is shown above. The screenshot simply demonstrates where the device will show up in the Device Manager.**

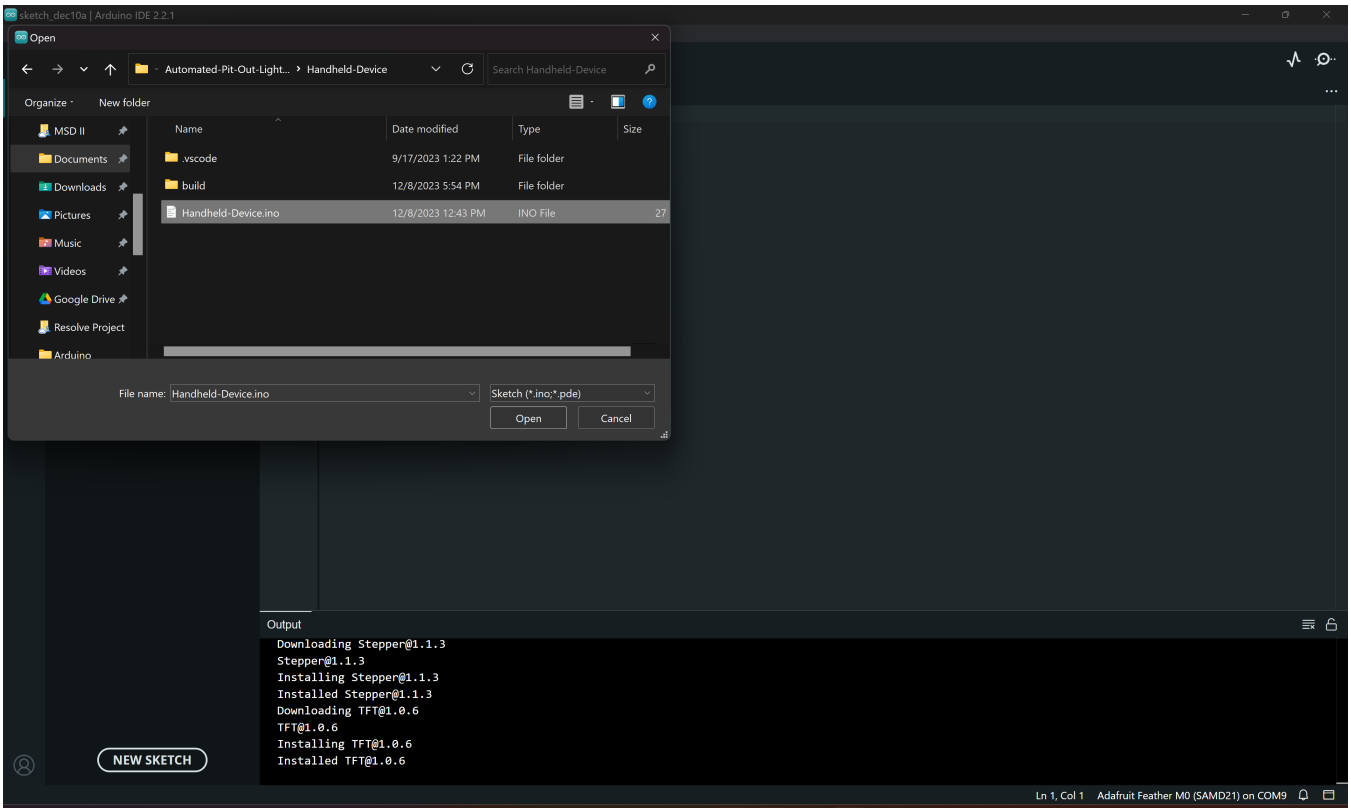
2. In the Arduino IDE, click select board, and select the COM port identified in the prior step.



3. Type "Feather M0" in the window the pop-up window.



4. Select the "Adafruit Feather M0 (SAMD21)" board and click "OK".
5. In the Arduino IDE, click "File" -> "Open", select the folder of the device software you want to upload, and click on the folder's ".ino" file.



6. Click the "Upload" button