

Synthesis @ ASU M5StickC Setup & Use Guide

Table of Contents

1 Introduction

This guide is meant as a resource for those intending to use the M5Stickc device with the Synthesis @ ASU firmware. Depending on what type of configuration you intend to do, you may want to jump to the later sections of this guide.

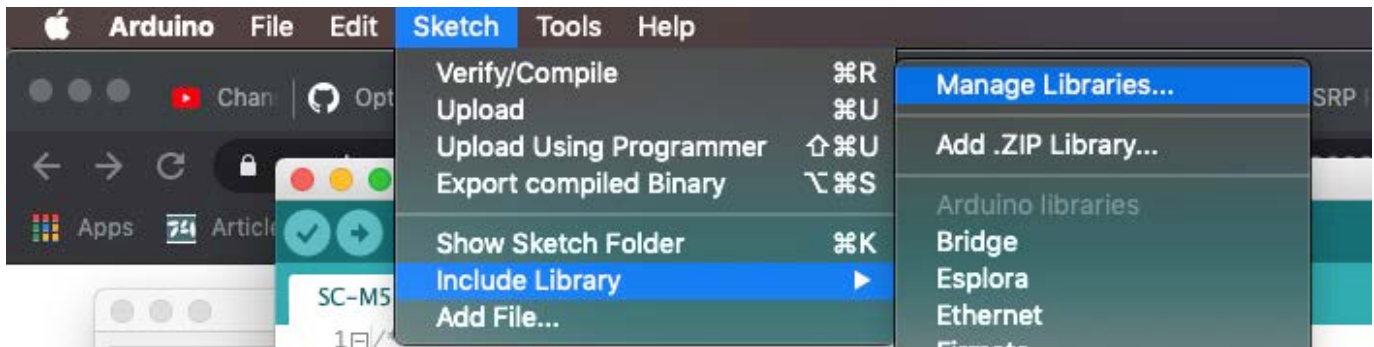
Where to start?

- Freshly bought M5Stickc with factory firmware: **Section 2, Download Device Firmware**
- Device already has the Synthesis @ ASU firmware installed and you want to configure the network settings: **Section 4, Configure Network Settings Via USB**
- Writing your own software to parse output from the device: **Section 5, Data Parsing Explanation and Examples**

2 Download Device Firmware & Example Software

In this section we will detail which prerequisite software packages and libraries are necessary to update device firmware and configure the network parameters. Additionally, we will direct you to the example source files for handling the data output from the M5Stickc device. For many who are following this guide for the purpose of using Synthesis @ ASU software, these items will be all you need. In the case that you plan on using the M5Stickc device with environments outside of Max/MSP/Jitter, please see Appendix II for links to documentation for handling OSC input in other languages.

The firmware update procedure for the M5Stickc devices requires that users first download the Arduino environment. The Arduino software functions as an Integrated Development Environment for device firmware on the M5Stickc, Arduino's range of microcontrollers, and others. You can download the Arduino software from their website. Once you have the Arduino software downloaded and installed, you will need to download a two libraries, M5StickC & OSC. You can download these libraries by opening the Library Manager in Arduino under "Sketch>Include Library>Manage Libraries...".

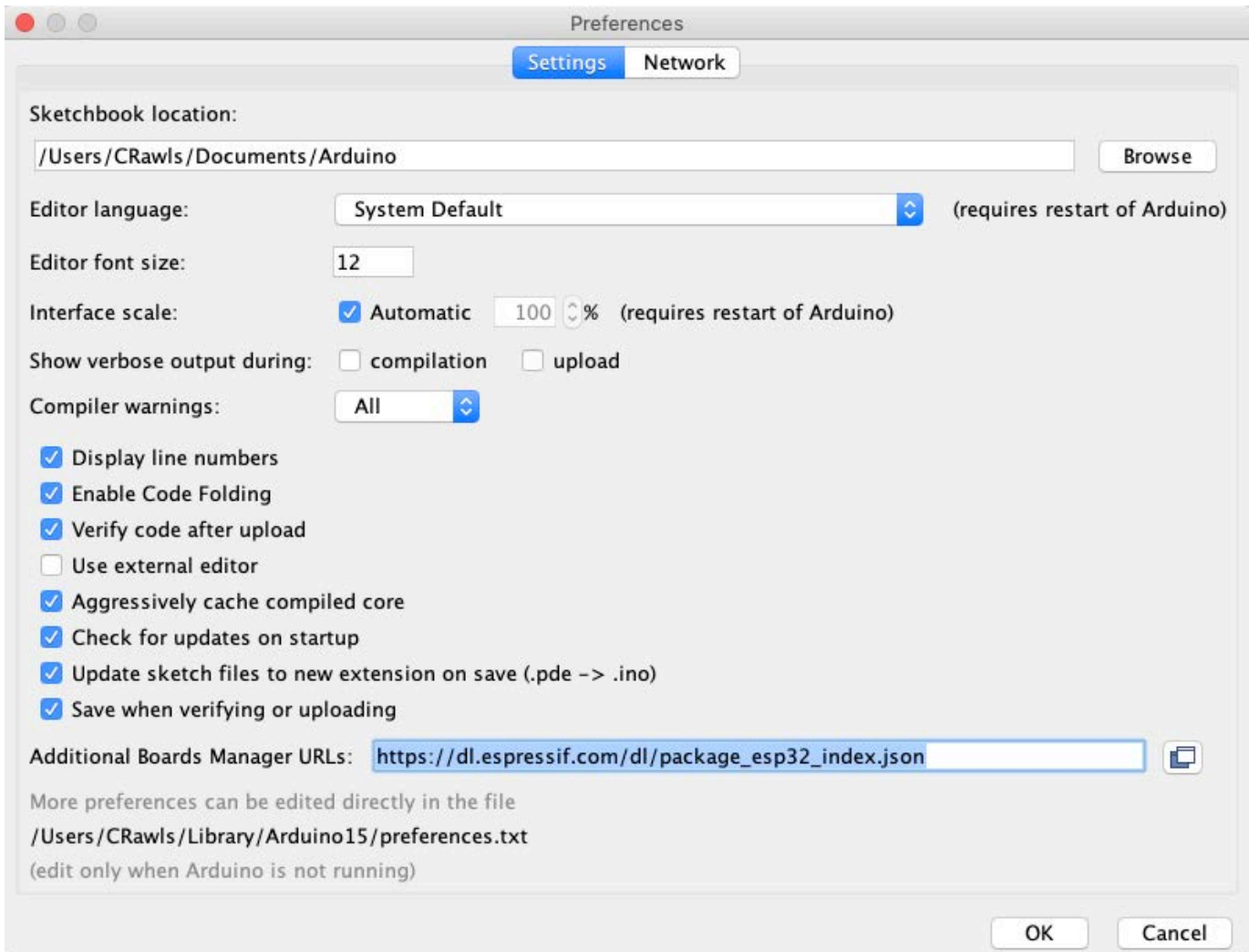


Menu Path in the Mac Environment, Same Path for Windows Users.

In addition to these libraries, you will also need to include the board manager for the M5StickC device. The board manager is the set of instructions that are run when you upload the firmware to the device that ensure that the incoming software is placed correctly in device memory. If incorrectly placed, the device may fail to run or become inoperable; using the correct manager is paramount to proper functioning.

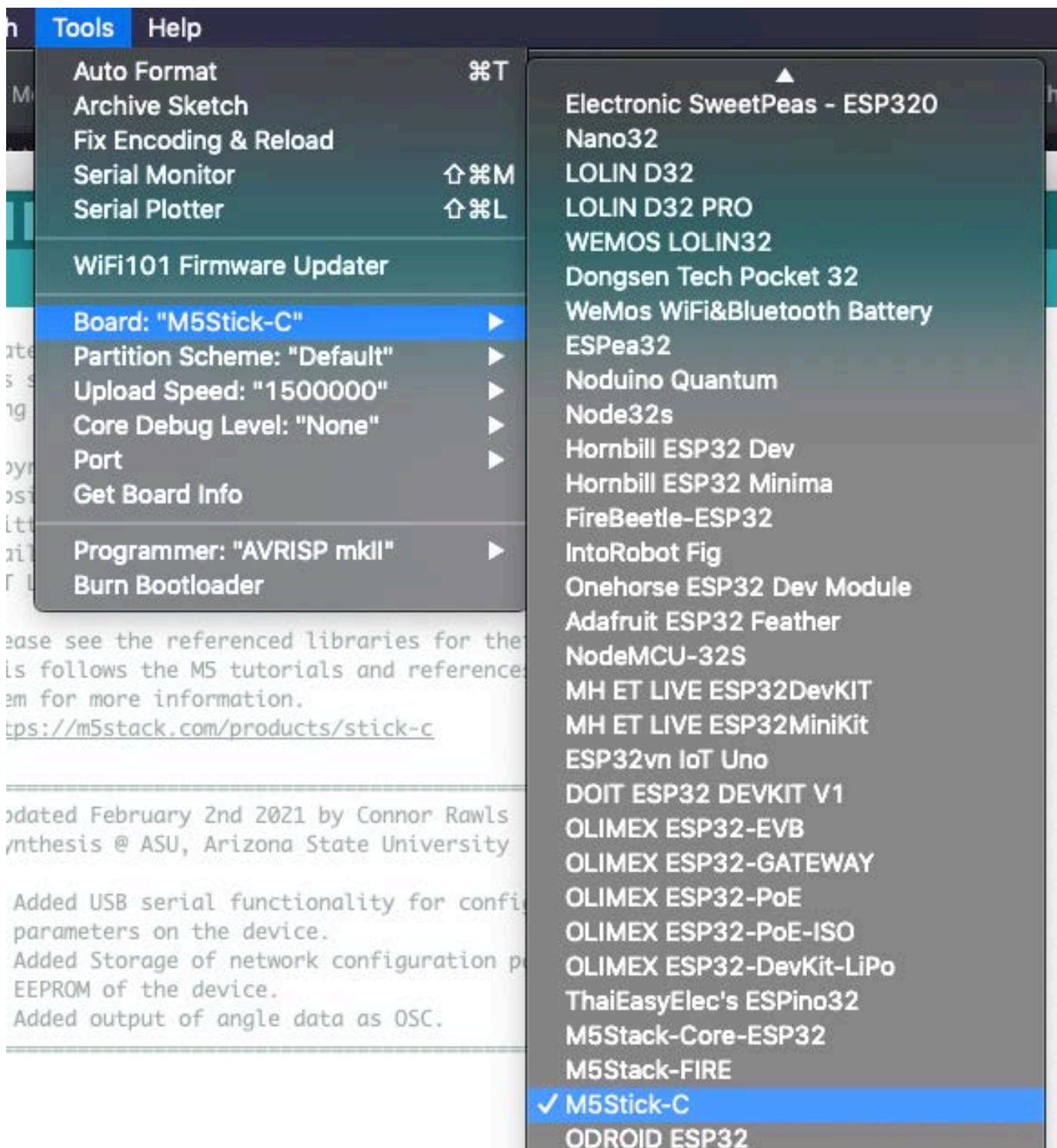
To access the board manager download, you will first need to open the Arduino preferences window and add a URL to the "Additional Boards Manager URLs" box. The URL is for the ESP-32 microprocessor (which the M5 products are based on) which gives you access to the M5Stick-C board manager.

URL: https://dl.espressif.com/dl/package_esp32_index.json



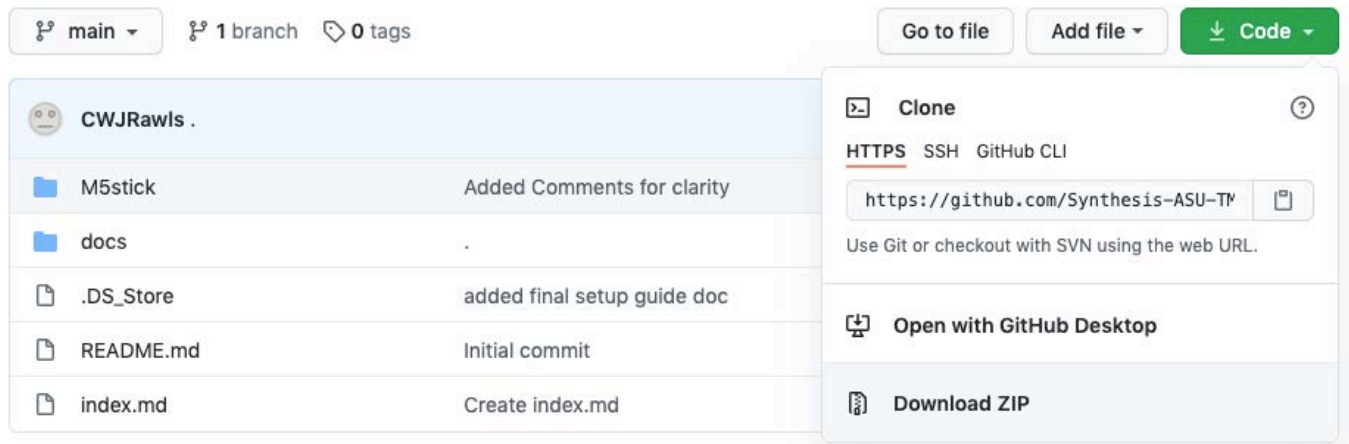
Arduino Preferences Window with Board Manager URL highlighted. MacOS look & feel, layout should be similar across operating systems.

Once you have added the URL to the box in the preferences window, accept your changes and close the window. The next step is to download the Board Manager package for the ESP-32 set. You can add managers under "Tools>Board>Boards Manager...". Find the esp32 package in the Boards Manager window. At the time of writing, the version we have tested on is 1.0.4. You can try out newer versions of the esp32 boards manager, but compatibility is not guaranteed. Once the esp32 managers have been downloaded, restart Arduino. You should now have the option under "Tools>Board" to select "M5Stick-C".



Selecting M5Stick-C Board Manager on MacOS, Windows and Linux users should find the option under the same menu.

Now that Arduino is configured, we will download the Synthesis @ ASU firmware, configuration software, and example Max patches. You can find all of these at <https://github.com/Synthesis-ASU-TML/Mated-Objects>. We would suggest downloading the entire repository as a ZIP file, and just keeping the M5stick folder once decompressed.



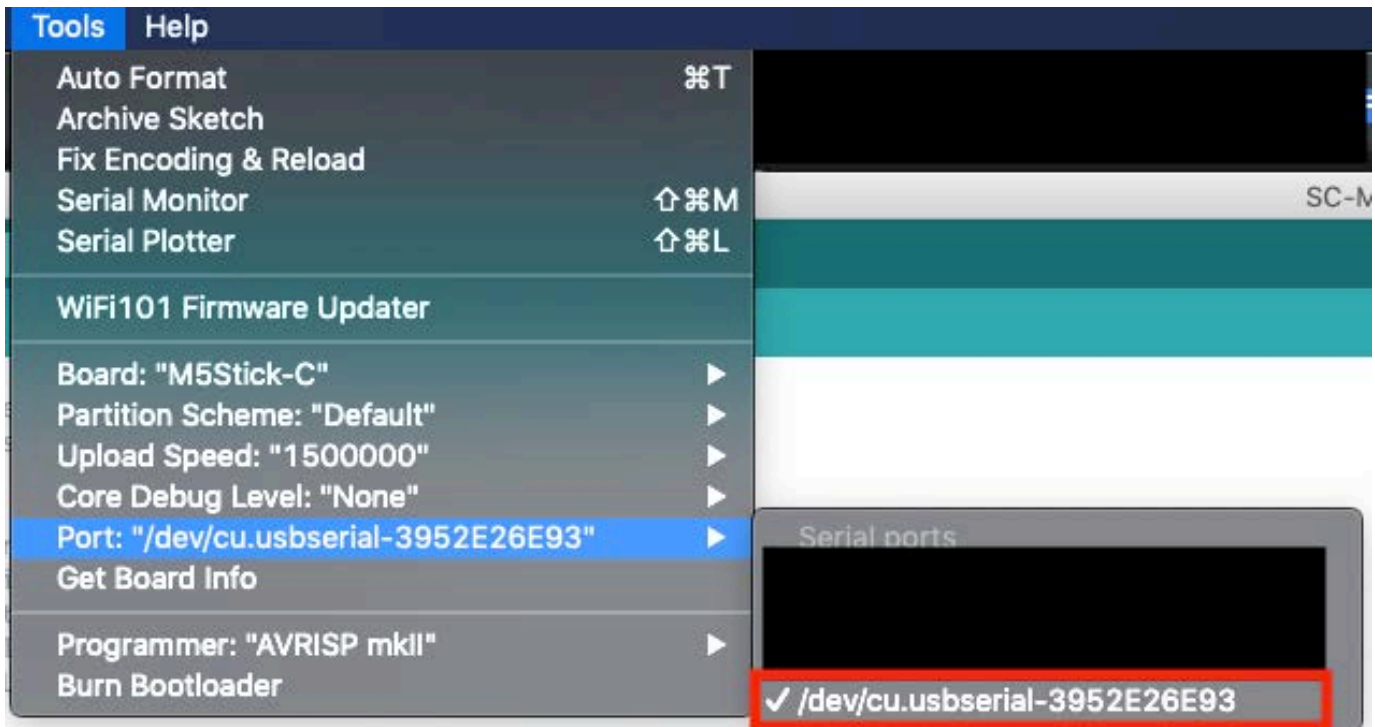
The web view of the Mated-Objects repository with the download menu opened from the green code button.

You can place the M5stick folder wherever is convenient on your file system. We suggest that you keep the contents of the folder together to avoid issues with Arduino compatibility.

4 Installing Firmware to M5StickC Device

Now that the necessary software and packages have been downloaded and installed, we will begin the process of putting the Synthesis @ ASU firmware on the M5StickC device. Start by opening SC-M5-Stick.ino in the Arduino IDE. Next, select the M5Stick-C board manager (see page 6 for details). Most of the programming details should update automatically.

After you have set the board manager to M5Stick-C, you need to select the port of the device. Depending on your operating system, the ports will appear differently in the Arduino IDE. On MacOS the name will look similar to `"/dev/cu.usbserial-hexstring"`. On Windows the device will be listed by COM port number. Since the devices will appear under generic names on both systems, it is recommended to only connect one at a time to avoid confusion.



The view of the "Port" menu in Arduino with the M5StickC device selected. Shown on MacOS. For Windows users, the "Port" menu will give you a list available COM ports. Will appear similar to "COM #".

Once you have you board and port configured, press the right-facing arrow button to compile and upload the firmware to the M5StickC device. This process will take a few minutes and will display "Done Uploading" in the bottom of the window when done.



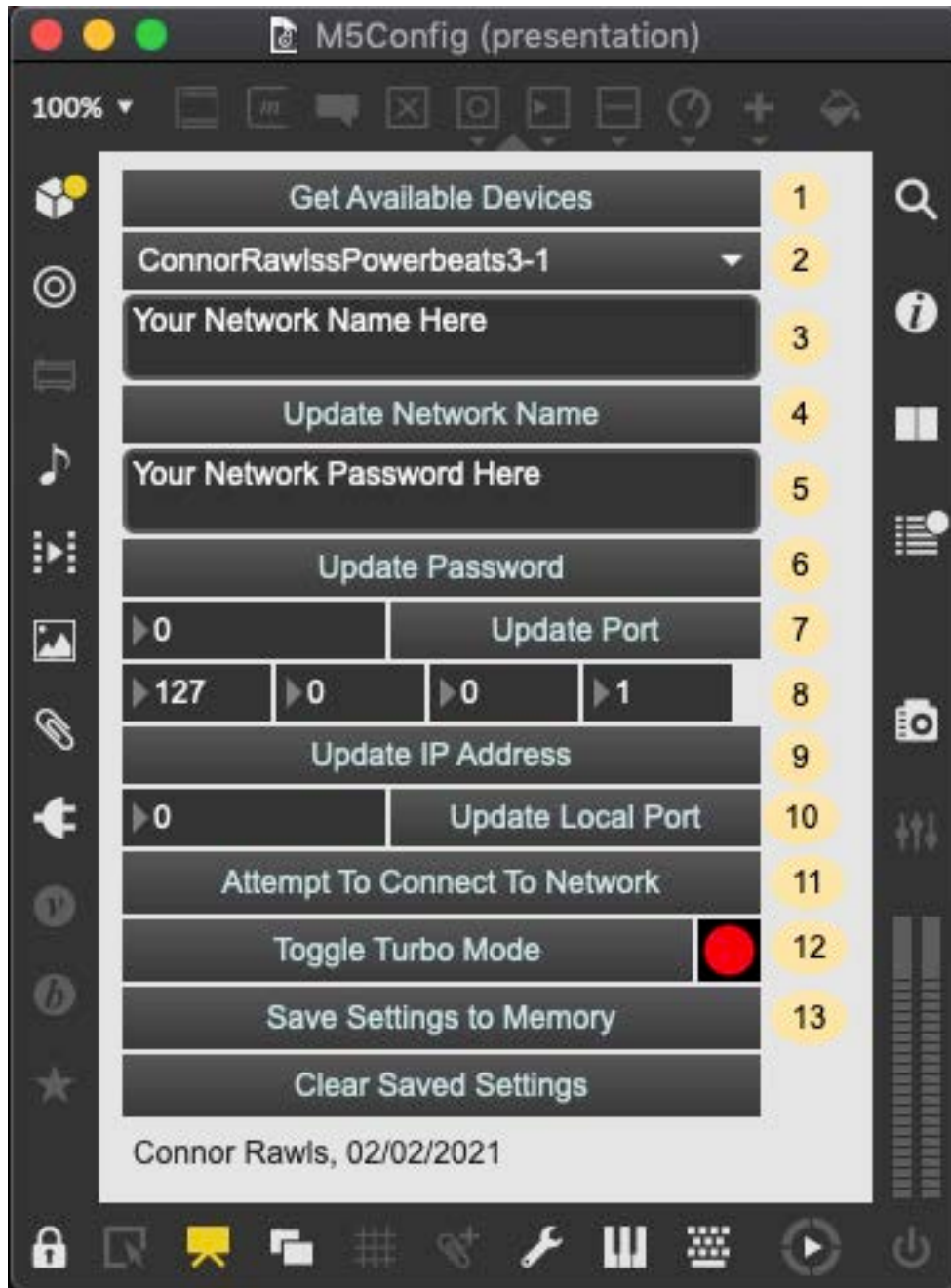
The Upload button in the Arduino IDE highlighted. Shown in MacOS, layout is the same on Windows.

In the case that you encounter an error in uploading, double check that the settings for Port and Board are correct. If you have any other M5StickC devices or other programmable micro-controllers disconnect them to avoid port confusion.

Configuring Network Settings

In this section we will cover how to use the provided M5Config Max patch to set how the M5StickC device connects to a WiFi network and setting the IP Address and port for sending data. Anytime you change target computer, or the computer's IP Address changes you will need to repeat at least a portion of this process.

Before opening the M5Config Max patch, make sure to quit the Arduino IDE entirely. Not doing so can result in the IDE blocking communication to the M5StickC device from other applications. Next find M5Config.maxpat in the M5stick folder and open it with Max 8. If you do not have Max installed on your computer, check Appendix I if you want to write your own configuration software. Otherwise, we will provide a compiled version of M5Config in the future. Once the patch has opened, see below for descriptions of each step.



The M5Config Max patch, shown on MacOS layout is the same on Windows systems.

M5Config Descriptions:

1 - Get Available Devices

Updates drop down menu with the updated list of all serial devices currently connected to the computer. Displays the port similarly to the Arduino IDE.

2 - Select Device

Using the drop down menu, select the M5StickC device that you want to configure.

3 - Network Name Box

Enter the name of the WiFi network that you want the M5StickC device to connect to.

4 - Update Network Name Button

Sends the network name to the M5StickC device.

5 - Network Password Box

Enter the password for the WiFi network that you want the M5StickC device to connect to.

6 - Update Network Password Button

Sends the network password to the M5StickC device.

7 - Port Update

Set the port number to use on the left and then press the button on the right to send the new network port to the M5StickC device. We suggest that you use the port 8001, ports above 1000 tend to be open in the case that you want to use a different port.

8 - Enter Target IP Address

Enter the four parts of the IP Address from the computer that you want to send to. See Appendix III for how to find the IP Address of your computer.

9 - Update IP Address Button

Sends the updated target IP Address to the M5StickC device.

10 - Local Port Update

Sets the network port for return traffic to the M5StickC device. In future updates there may be the possibility of configuration over WiFi for some operation parameters. By default this port is set to 7500. You can change this value, but currently has no effect.

11 - Attempt To Connect To Network

Confirm that the connection details are correct before pressing this button. Once pressed, the button tells the M5StickC to attempt to connect to WiFi using the data sent to it. Check the network display screen on the device to see if the M5StickC has been successful in connecting to WiFi.

12 - Toggle Turbo Mode

Pressing the button toggles the device between standard operation and turbo operation. Turbo mode sets the M5StickC to run as fast as possible, otherwise the device runs at 25FPS. When turbo mode is active, the circle to the right of the button will turn green.

13 - Save Settings To Memory

If you are happy with the configuration of the M5StickC, press this button to tell the device to commit your settings to internal memory. Doing this will make the settings available to the device on reset rather than reverting to default values.

Clear Saved Settings

Only use this button if you intend to erase ALL of the previously saved settings. You can overwrite specific values with the controls above.

