Wave Farm
Toolkit
Development System Manual

November 17, 2024

(1) Things you will need to download

(1a) Arduino IDE

https://www.arduino.cc/en/software

We are using the "Legacy" 1.8.19 IDE, but the newer v2 IDE should also work.
Scroll down the page to find the 1.8.19 download section.

(1b) Serial Device Driver

https://www.wch-ic.com/downloads/CH341SER_ZIP.html

Arduino works by loading the code into the Toolkit over a serial port. There's a little USB to serial gadget in the package Jen sent you. You need a driver for it. It is similar to a FTDI driver, but a little bit different.

(1c) Filesystem Uploader that you can find here:

https://randomnerdtutorials.com/esp32-web-server-spiffs-spi-flash-file-system/

The Filesystem Uploader is a plugin for Arduino. There are instructions for how to install it for Arduino 1.8.19 and Arduino v2. Install the Arduino IDE first!

(2) Setting up the Arduino IDE

With Arduino, everything related to a project is stored in a "Sketch" folder. In this case we call it "InternetRadioSketch". You can download that from the github.

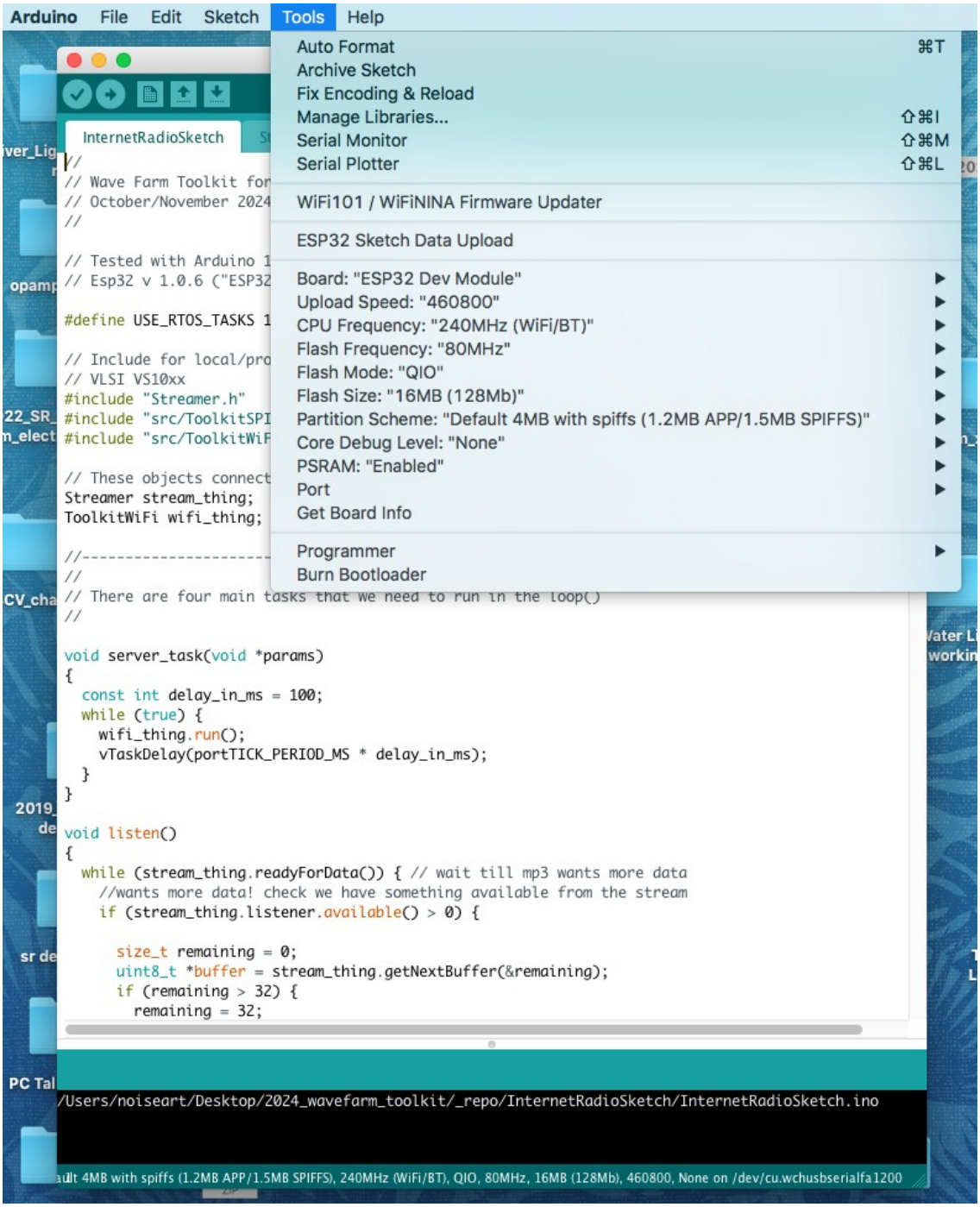
https://github.com/theotherjenkutler/InternetRadio

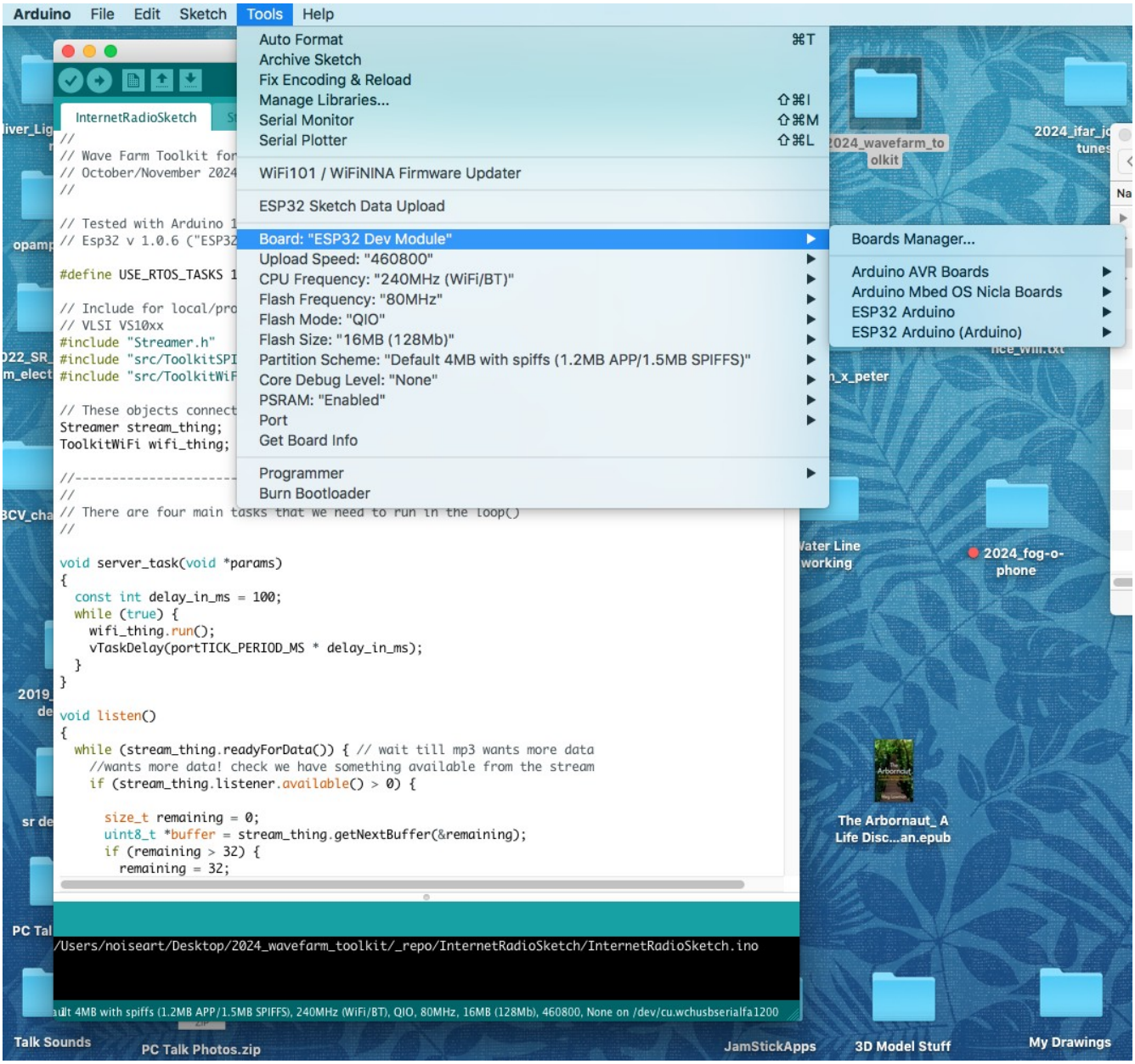
Inside the InternetRadioSketch folder there is a project/code file "InternetRadioSketch.ino".
Open this in the IDE and then you can configure the settings for it.

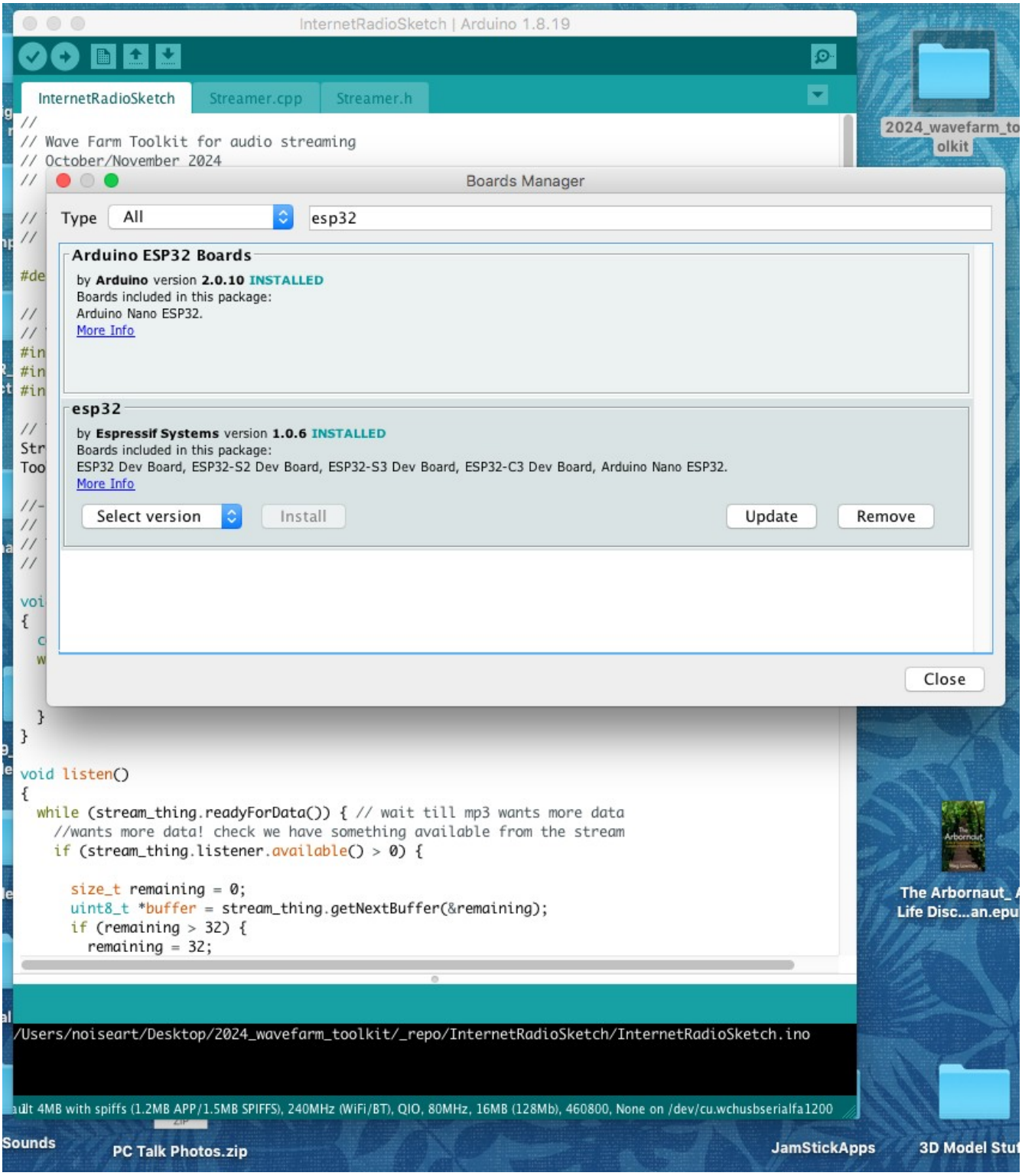
You will need to configure the Arduino IDE to talk to your Toolkit Board.

(2a) In the Tools menu, you will see a line called "Board: ..". This has a submenu that will take you to the "Boards Manager". Once you open the Board Manager you can type "esp32" into the search box (top right) and you get a listing of two option. The first is for one of Arduino's esp32 boards. The second is for all of the esp32 version that are supported by Espressif. You will need to install the boards for Espressif. The most recent versions are 3.0.x. These will work on newer operating systems. I am using 1.0.6 (because I have an old OS).

Program code and files are downloaded to the Toolkit using Python scripts. The python libraries that are used for this are included in the with the "board manager" files, and they only run on specific versions of Mac OS. If you are using Windows or Linux, it should work regardless.




Arduino File Edit Sketch Tools Help
Auto Format ⌘T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... ⇧⌘I
Serial Monitor ⇧⌘M
Serial Plotter ⇧⌘L
WiFi101 / WiFiNINA Firmware Updater
ESP32 Sketch Data Upload
Board: "ESP32 Dev Module"
Upload Speed: "460800"
CPU Frequency: "240MHz (WiFi/BT)"
Flash Frequency: "80MHz"
Flash Mode: "QIO"
Flash Size: "16MB (128Mb)"
Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)"
Core Debug Level: "None"
PSRAM: "Enabled"
Port
Get Board Info
Programmer
Burn Bootloader
Boards Manager...
Arduino AVR Boards
Arduino Mbed OS Nicla Boards
ESP32 Arduino
ESP32 Arduino (Arduino)
InternetRadioSketch
// Wave Farm Toolkit for
// October/November 2024
// Tested with Arduino 1
// Esp32 v 1.0.6 ("ESP32
#define USE_RTOS_TASKS 1
// Include for local/pro
// VLSI VS10xx
#include "Streamer.h"
#include "src/ToolkitSPI
#include "src/ToolkitWiF
// These objects connect
Streamer stream_thing;
ToolkitWiFi wifi_thing;
// There are four main tasks that we need to run in the loop()
void server_task(void *params)
{
const int delay_in_ms = 100;
while (true) {
wifi_thing.run();
vTaskDelay(portTICK_PERIOD_MS * delay_in_ms);
}
}
void listen()
{
while (stream_thing.readyForData()) { // wait till mp3 wants more data
//wants more data! check we have something available from the stream
if (stream_thing.listener.available() > 0) {
size_t remaining = 0;
uint8_t *buffer = stream_thing.getNextBuffer(&remaining);
if (remaining > 32) {
remaining = 32;
/Users/noiseart/Desktop/2024_wavefarm_toolkit/_repo/InternetRadioSketch/InternetRadioSketch.ino
ult 4MB with spiffs (1.2MB APP/1.5MB SPIFFS), 240MHz (WiFi/BT), QIO, 80MHz, 16MB (128Mb), 460800, None on /dev/cu.wchusbserialfa1200
2024_wavefarm_to
olkit
2024_ifar_jc
tunes
2024_fog-o-
phone
The Arbornaut_ A
Life Disc...an.epub
Talk Sounds
PC Talk Photos.zip
JamStickApps
3D Model Stuff
My Drawings

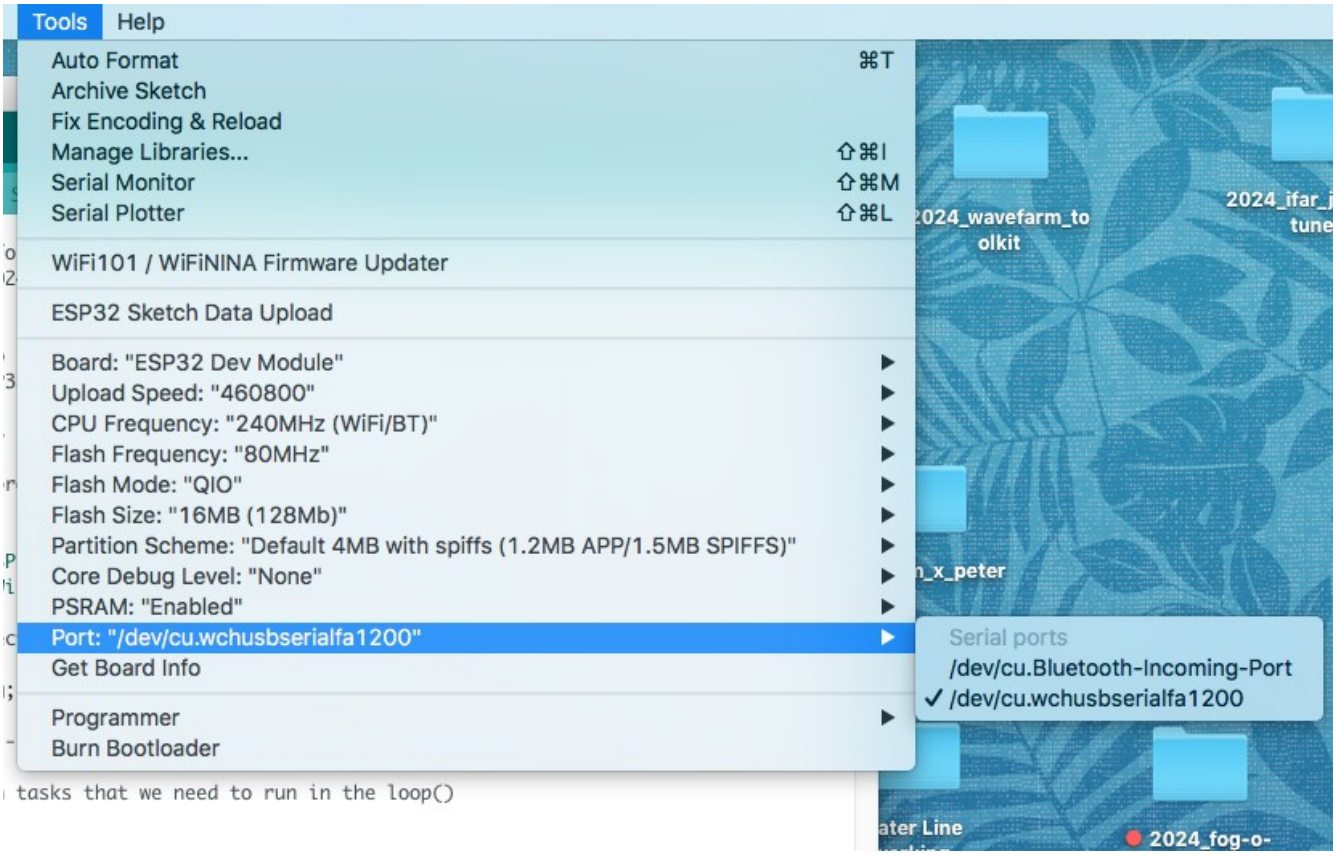

InternetRadioSketch | Arduino 1.8.19
InternetRadioSketch
Streamer.cpp
Streamer.h
// Wave Farm Toolkit for audio streaming
// October/November 2024
Boards Manager
Type
All
esp32
Arduino ESP32 Boards
by Arduino version 2.0.10 INSTALLED
Boards included in this package:
Arduino Nano ESP32.
More Info
esp32
by Espressif Systems version 1.0.6 INSTALLED
Boards included in this package:
ESP32 Dev Board, ESP32-S2 Dev Board, ESP32-S3 Dev Board, ESP32-C3 Dev Board, Arduino Nano ESP32.
More Info
Select version
Install
Update
Remove
Close
void listen()
{
while (stream_thing.readyForData()) { // wait till mp3 wants more data
//wants more data! check we have something available from the stream
if (stream_thing.listener.available() > 0) {
size_t remaining = 0;
uint8_t *buffer = stream_thing.getNextBuffer(&remaining);
if (remaining > 32) {
remaining = 32;
/Users/noiseart/Desktop/2024_wavefarm_toolkit/_repo/InternetRadioSketch/InternetRadioSketch.ino
PC Talk Photos.zip
JamStickApps

In the first image you can see all the settings for the ESP32 board:

For 1.0.6 it is "ESP32 Dev Module"
For the newer version it is "ESP32 Wrovere Kit (all versions)"

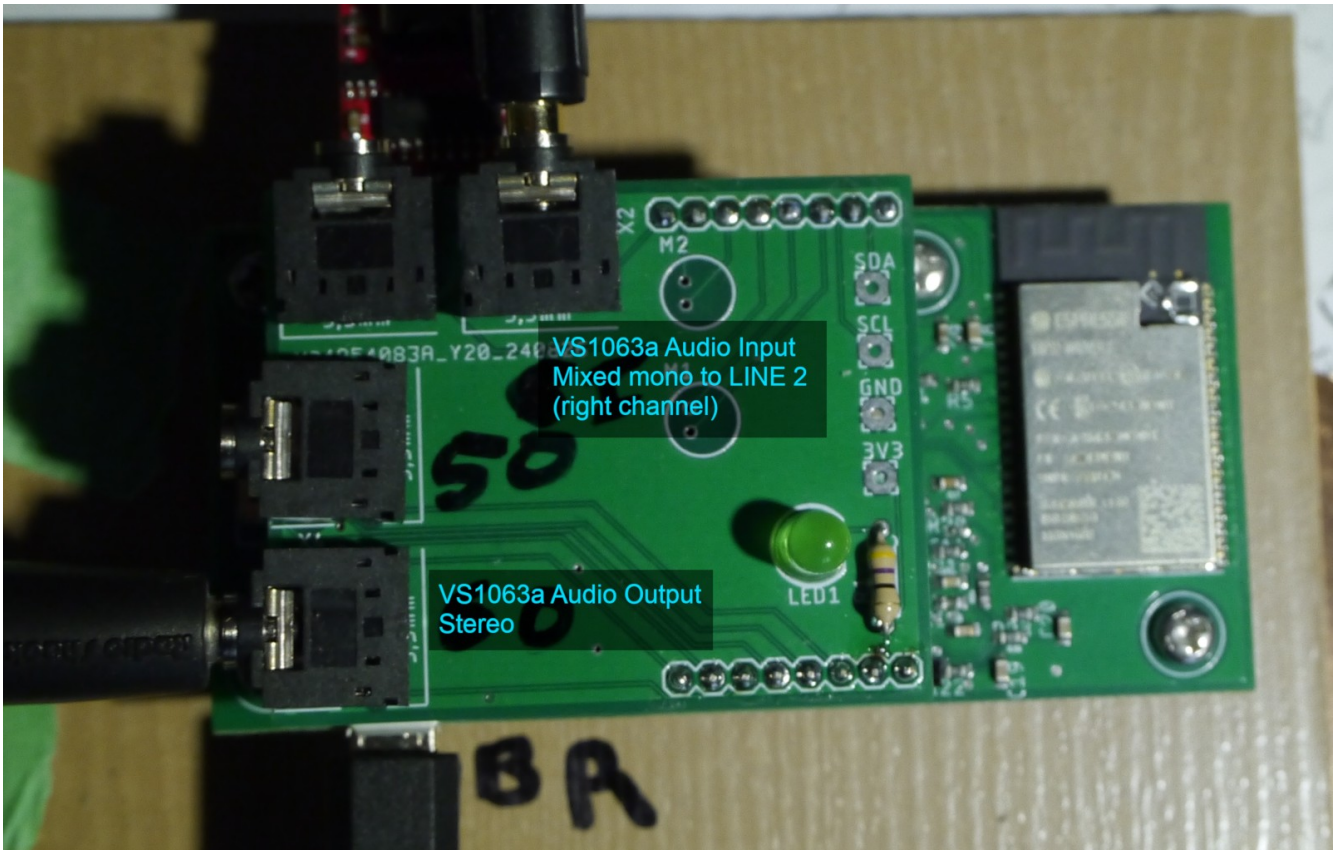
Some board settings will get added to the tools menu:
Set Flash Size to 16MB
Set Upload Speed to 460800
Set PSRAM to enabled

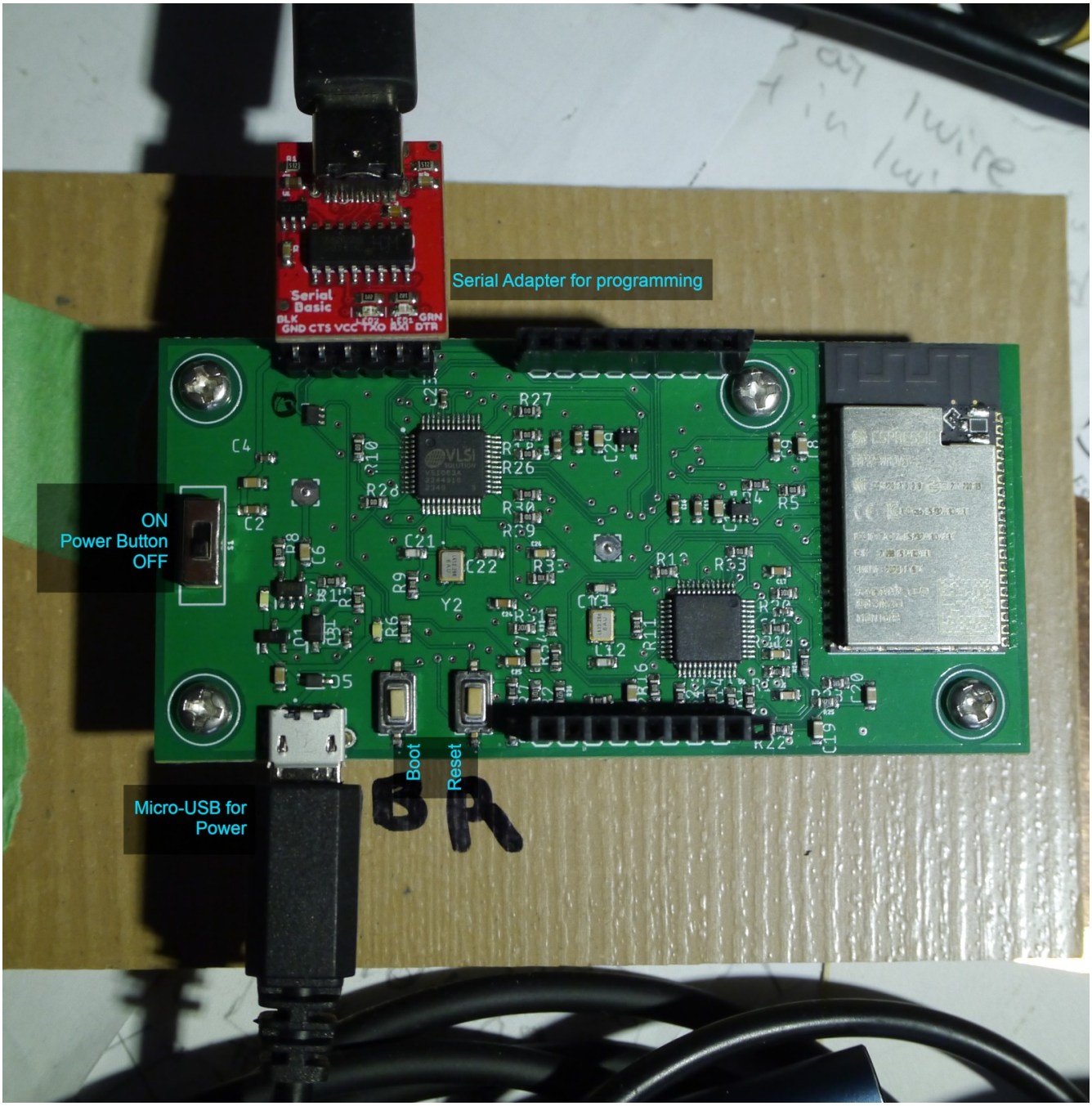
You also need to set the serial port that you are going to use to talk to the Toolkit. Connect the serial gadget to your computer using a USB-C cable. After a few seconds it should appear in the "Port" submenu.



(3) Setting up the Toolkit Board

The Toolkit has two circuit boards the connect together. The top one is the audio connections and the bottom one is the esp32 board with the VLSI audio chips on it.





The Toolkit has:
- a micro-USB port for 5 Volt power. This will also charge the battery when the power switch is Off.
- a 6 pin connector that goes to the serial port gadget. I have soldered a right-angle header onto my Toolkit. You can also use the pogo-pressure-pin-thing that Jen sent you. Make sure that Ground on your serial device is on the correct side of the connector.
- a sliding power button
- 2 little buttons near the micro-USB power connector. The one closest to the usb is "boot" the other is "reset".

(4) Talking to the Toolkit from the Arduino IDE.

When you want to download something onto the Toolkit, you hold down the "boot" button and then "reset". As long as the "boot" button is down when the "reset" is released, then the board will enter download mode.

Once your USB is hooked up to the Toolkit and your Serial port settings are correct)you can click the upload button (top left corner of the Arduino window). The IDE will compile the source code and then upload it to the Toolkit. It will take some time to compile everything, and it should throw out 1 warning about the WiFi headers. But there shouldn't be any other errors or warnings.

Arduino has a "Serial Monitor" which you can open up from the Tool menu. This acts like a console terminal between Arduino and the Toolkit. When the Toolkit is running it will send messages to the serial monitor.

NOTE: if you open the Serial Monitor while you are uploading code you will crash the upload process. If the serial monitor is open when you are using the Filesystem Uploader then the filesystem upload will fail.

Once the upload process is complete, you will have to manually reset the board - either to make it run, or to place it back into download mode.

Once you have uploaded the compiled source code to the Toolkit, then you have to upload the settings and html/css/js files to the Toolkit's built in flash drive. Use the Filesystem Uploader to do this. Restart the Toolkit in download mode (boot button + reset). Then you can select "ESP32 Sketch Data Upload" from the Tools menu. You should see the upload messages at the bottom of the Arduino IDE window.

(5) Running the Toolkit

Open the tools/serial monitor in Arduino. Make sure it is running at 115200 baud (bottom of the serial monitor window). Restart the toolkit (without the boot button) and you should see a bunch of stuff on the screen.

```
Wave Farm Toolkit!
--------------------------
SETTINGS:
wifi_router_SSID = TELUS7838-2.4G
wifi_router_password =
wifi_toolkit_hostname = PetersToolkit
wifi_toolkit_AP_SSID = PetersToolkit_AP
listen_icecast_url = audio.wavefarm.org
listen_icecast_port = 8000
listen_icecast_mountpoint = /wgxc.mp3
listen_volume = 0.87
remote_icecast_url = us1.internet-radio.com
remote_icecast_port = 8078
remote_icecast_user = source
remote_icecast_password = [redacted]
remote_icecast_mountpoint = live
mic_not_line = 0
channels = 1
bitrate = 128
sample_rate = 44100
agc_not_manual = 1
manual_gain_level = 1.0
agc_maximum_gain = 16.0
startup_auto_mode = transmitter
--------------------------
VS1053 status: 4
volume 0.87 -> 1/2 dBs -13.00 -> vlsi 13
VS1063 status: 6
Loaded 7384 words of patches code
Setting clock: B000
VS1063 clock: B000
volume 0.87 -> 1/2 dBs -13.00 -> vlsi 13
Connecting to SSID TELUS7838-2.4G
.............WiFi connected to local router
IP address:
192.168.1.76
Soft AP SSID: PetersToolkit_AP, IP address: 192.168.4.1
DNS Server started on Access Point.
Starting in transmitter mode
ICY connecting .. us1.internet-radio.com:8078
PUT /live HTTP/1.1
HOST: us1.internet-radio.com:8078
Authorization: Basic c291cmNlOmZyMzNkb21yYWRpbw==
Accept: */*
Transfer-Encoding: chunked
Content-Type: audio/mpeg
Ice-Public: 1
Ice-Name: Toolkit Stream
Ice-Description: Toolkit stream description.
Ice-URL: http://wavefarm.org
Ice-Genre: Toolkit Art
Expect: 100-continue
```

The Toolkit currently runs in one of 3 modes - Listening, Transmitting, Waiting.

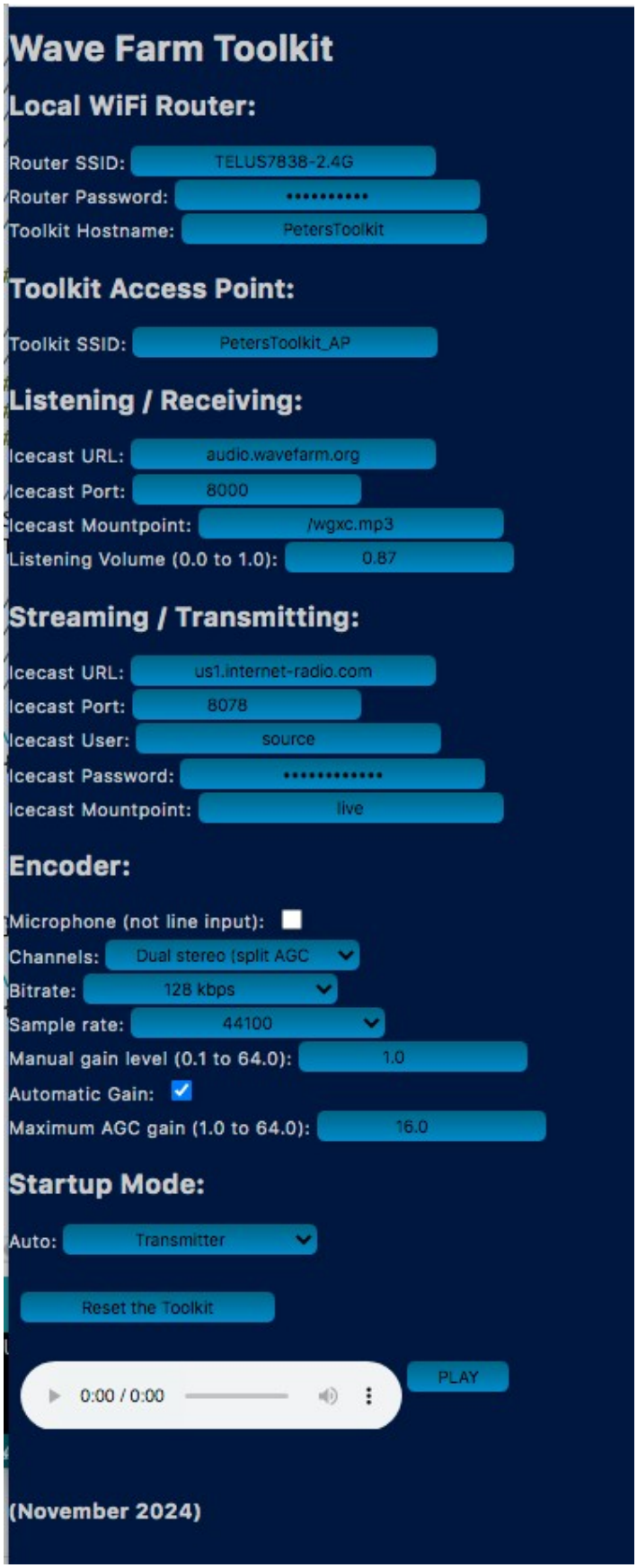
Listening connects to a remote icecast server and streams the mp3 data from the server to the audio output of the Toolkit. It also relays the mp3 data to the Toolkits "captive portal" or "configuration" web page.

Transmitting connects to a remote icecast server as a "source" client and sends mp3 data from the Toolkit's audio input to the icecast server. It also sends the mp3 data to the Toolkit's local web page.

Waiting doesn't run any audio in or out of the Toolkit, but it runs the configuration web page.

To access the web page you can lookup the Toolkit's local IP address and use that in you desktop browser. Or you can go the WiFi settings on your phone and select the Toolkit's SSID. You phone should then take you to a "captive portal" browser page where you can see the Toolkit's settings.

Sometimes the phone won't load the css and/or js files. Sometimes it will load them but not run the scripts. You can go back and reconnect to the WiFi to get it running. There is a 2 second time delay before the page loads the settings to get around some "captive portal" issues.


Wave Farm Toolkit
Local WiFi Router:
Router SSID: TELUS7838-2.4G
Router Password: ••••••••••
Toolkit Hostname: PetersToolkit
Toolkit Access Point:
Toolkit SSID: PetersToolkit_AP
Listening / Receiving:
Icecast URL: audio.wavefarm.org
Icecast Port: 8000
Icecast Mountpoint: /wgxc.mp3
Listening Volume (0.0 to 1.0): 0.87
Streaming / Transmitting:
Icecast URL: us1.internet-radio.com
Icecast Port: 8078
Icecast User: source
Icecast Password: ••••••••••••
Icecast Mountpoint: live
Encoder:
Microphone (not line input):
Channels: Dual stereo (split AGC
Bitrate: 128 kbps
Sample rate: 44100
Manual gain level (0.1 to 64.0): 1.0
Automatic Gain:
Maximum AGC gain (1.0 to 64.0): 16.0
Startup Mode:
Auto: Transmitter
Reset the Toolkit
0:00 / 0:00
PLAY
(November 2024)