

---

# **Adafruit Soundboard Library Documentation**

*Release 0.1.1*

**Mike Mabey**

**Aug 29, 2017**



---

## Contents

---

<b>1</b>	<b>Installation</b>	<b>3</b>
1.1	CircuitPython Instructions . . . . .	3
1.2	MicroPython Instructions . . . . .	3
<b>2</b>	<b>Quick Start</b>	<b>5</b>
<b>3</b>	<b>API Reference</b>	<b>7</b>
3.1	adafruit_soundboard . . . . .	7
<b>4</b>	<b>Contributing</b>	<b>9</b>
<b>5</b>	<b>License</b>	<b>11</b>



The [Adafruit Soundboards](#) are an easy way to add sound to your maker project, but the [library](#) provided by Adafruit only supports Arduino.

If you've wanted to use one of these boards with a [MicroPython](#) or [CircuitPython](#) microcontroller (MCU), this is the library you've been looking for.



This driver depends on either [MicroPython](#) or [CircuitPython](#) and is intended for use to control one of the [Adafruit Audio FX](#) boards via UART.

Make sure to get the latest version of the code from [GitHub](#).

### CircuitPython Instructions

First, you'll need to get [Adafruit CircuitPython](#). Then, please ensure all dependencies are available on the CircuitPython filesystem. This is easily achieved by downloading [the Adafruit library and driver bundle](#).

### MicroPython Instructions

At this time, you have to install the driver by copying the `soundboard.py` script to your MicroPython board along with your `main.py` file. At some point in the future it may be possible to `pip install` it.





First, you'll need to decide which UART bus you want to use. To do this, you'll need to consult the documentation for your particular MCU. In these examples, I'm using the original `pyboard` (see [documentation here](#)) and I'm using UART bus 1 or XB, which uses pin X9 for transmitting and pin X10 for receiving.

Then, create an instance of the `Soundboard` class, like this:

```
sound = Soundboard('XB')
```

I *highly* recommend you also attach the RST pin on the sound board to one of the other GPIO pins on the MCU (pin X11 in the example). Also, my alternative method of getting the list of files from the board is more stable (in my own testing) than the method built-in to the sound board. Also, I like getting the debug output and I turn the volume down to 50% while I'm coding. Doing all this looks like the following:

```
SB_RST = 'X11'  
sound = Soundboard('XB', rst_pin=SB_RST, vol=0.5, debug=True, alt_get_files=True)
```

Once you've set up all of this, you're ready to play some tracks:

```
# Play track 0  
sound.play(0)  
  
# Stop playback  
sound.stop()  
  
# Play the test file that comes with the sound board  
sound.play('T00     OGG')  
  
# Play track 1 immediately, stopping any currently playing tracks  
sound.play_now(1)  
  
# Pause and resume  
sound.pause()  
sound.unpause()
```

You can also control the volume in several different ways:

```
# Raise volume by 2 points (0 min volume, 204 max volume)
sound.vol_up()

# Turn down volume until lower than 125
sound.vol_down(125)

# Get the current volume
sound.vol

# Set volume to 56 (out of 204 maximum)
sound.vol = 56

# Set volume to 75% of maximum volume
sound.vol = 0.75
```

## CHAPTER 3

---

### API Reference

---

`adafruit_soundboard`



## CHAPTER 4

---

### Contributing

---

Contributions are welcome! Please read our [Code of Conduct](#) before contributing to help this project stay welcoming.



## CHAPTER 5

---

License

---

This project is licensed under the [MIT License](#).