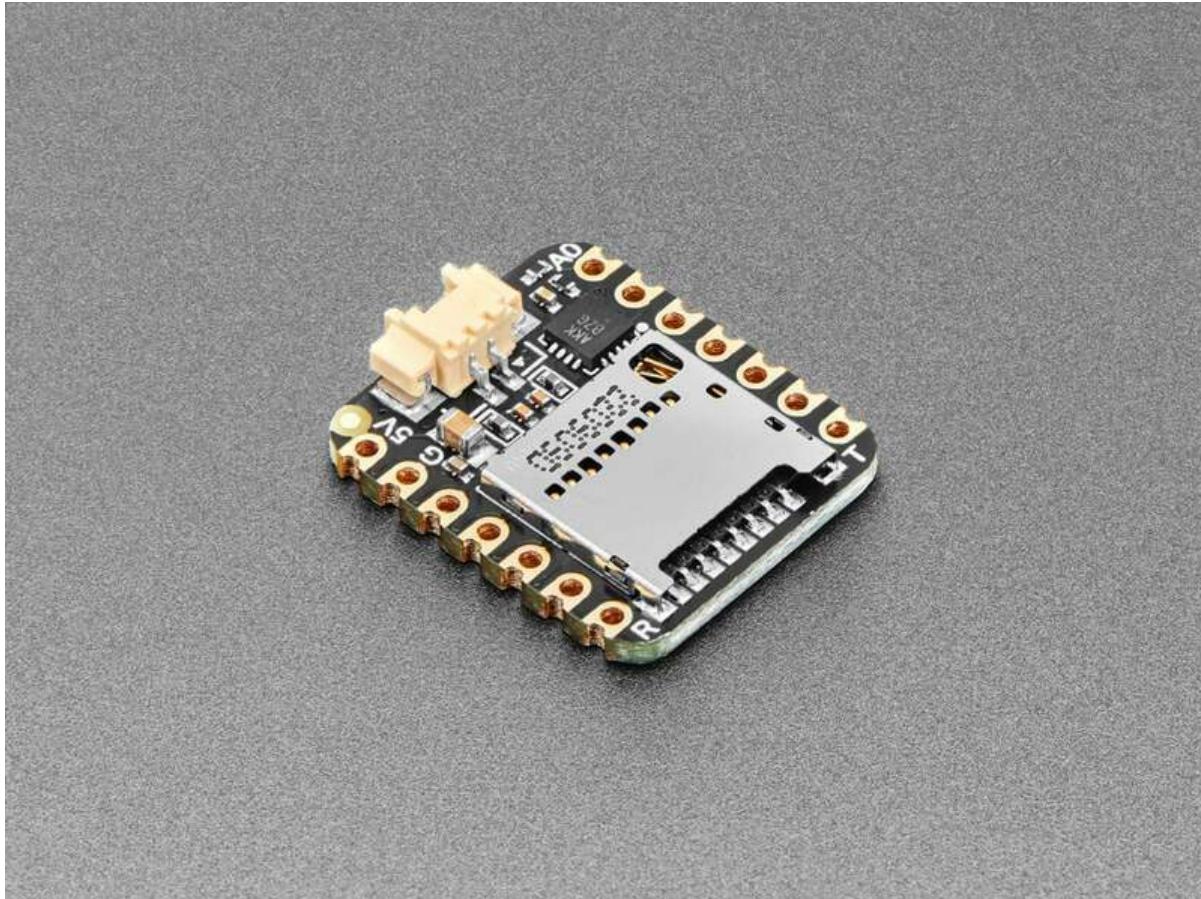




Adafruit Audio BFF

Created by Liz Clark



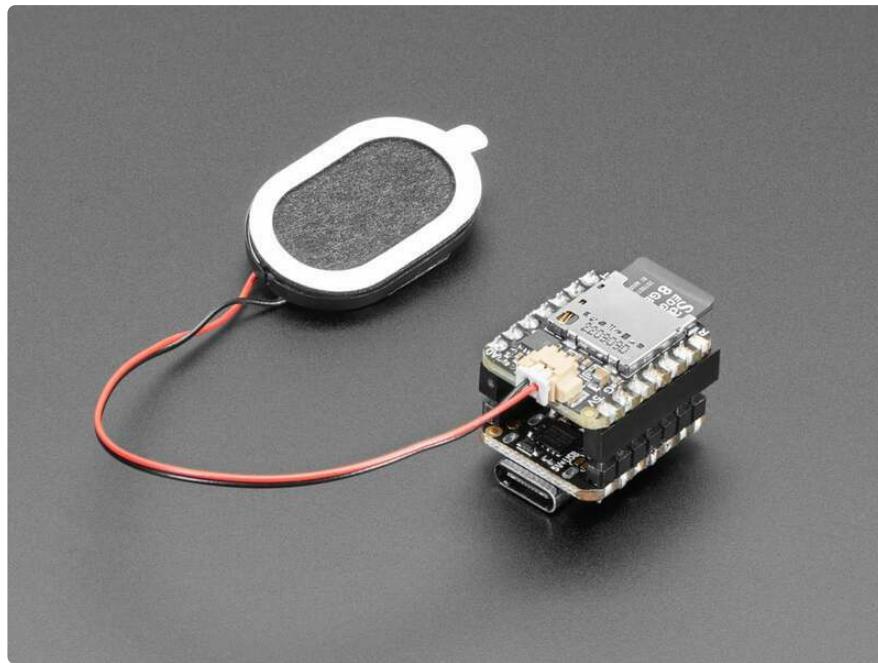
<https://learn.adafruit.com/adafruit-audio-bff>

Last updated on 2024-09-18 10:15:13 AM EDT

Table of Contents

Overview	3
Pinouts	6
• I2S Amplifier	
• Default I2S Pins	
• Default SD Card Pins	
• I2S Gain Jumper	
• Speaker Connector	
• micro SD Card Slot	
CircuitPython	8
• micro SD Card Prep	
• CircuitPython Microcontroller Wiring	
• CircuitPython Usage	
• Example Code	
Python Docs	12
Arduino	13
• micro SD Card Prep	
• Wiring	
• Library Installation	
• Example Code	
Arduino Docs	19
Downloads	19
• Files	
• Schematic and Fab Print	
• 3D Models	

Overview



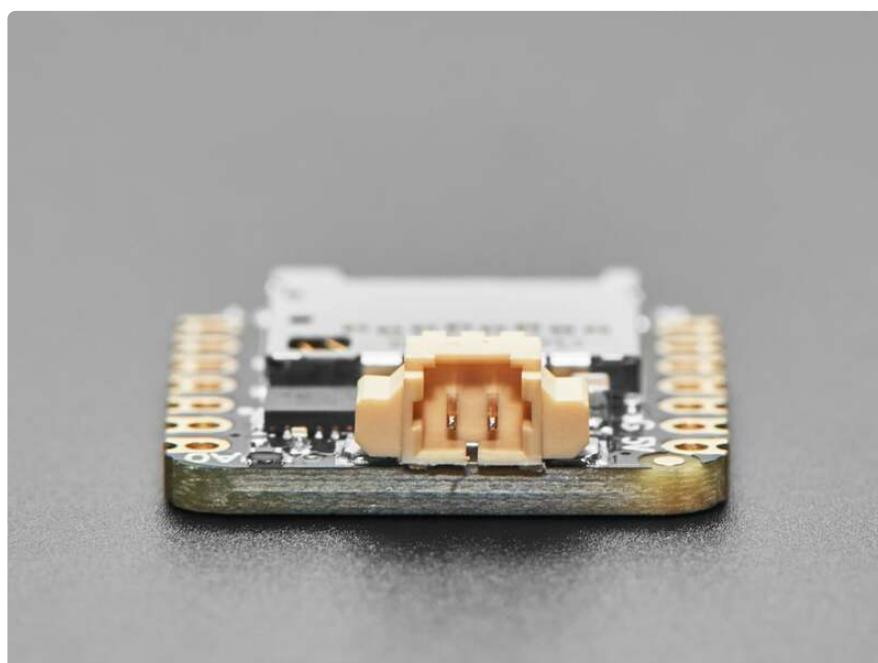
Our QT Py boards are a great way to make very small microcontroller projects that pack a ton of power - and now we have a way for you to turn many QT Py boards into powerful audio play projects that are super small!



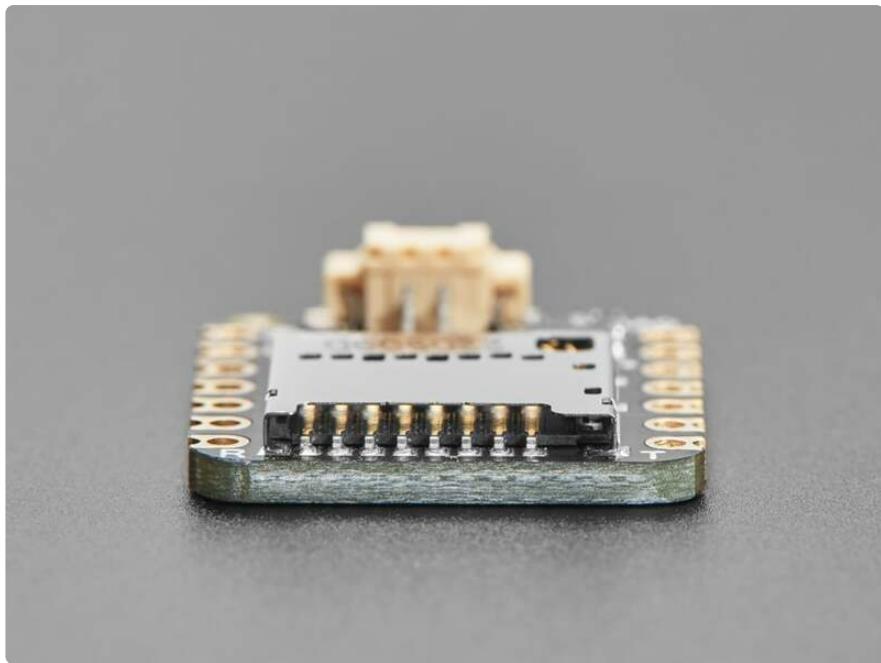
We call this the **Adafruit Audio BFF** - a "Best Friend Forever". When you were a kid, you may have learned about the "buddy" system; well, this product is kinda like that! A board that will watch your QT Py's back and give it more capabilities.



This PCB is designed to fit onto the back of any QT Py or Xiao board, it can be soldered into place or use pin and socket headers to make it removable. Onboard is a MAX98357 audio amplifier and picoblade-compatible connector for plugging in a 4 or 8 ohm speaker. We use **A1** for the audio data, **A2** for wordselect clock, and **A3** for bitclock. The SD card connects over the SPI port: **MOSI**, **MISO** and **SCK** plus **A0** for card select.



This pinout will work with ESP32 series, nRF52840, and RP2040 chipset boards. It won't work with the [ATSAMD21 'original' QT Py because those pins on the SAMD21 are not I2S capable \(<https://adafru.it/OeY>\)](#). However, you could cut and rewire the traces to connect to the I2S pads if desired - personally, we recommend just upgrading to an RP2040 QT Py instead.



We include some header that you can solder to your QT Py. [You can also pick up an Itsy Bitsy short female header kit to make it removable but compact \(<http://adafru.it/4174>\)](#), you'll just need to trim down the headers to 7 pins long.

- Comes as an assembled and tested PCB
- For any QT Py or Xiao boards
- [Contains a MAX98357 3 Watt audio amplifier \(<http://adafru.it/3006>\)](#) pre-configured for 'stereo' mix output and 9 dB gain which will work great for any project.
- Use any micro SD card that supports SPI mode with one CS pin.
- [Connect to the speaker output with a picoblade-compatible 2-pin cable \(<http://adafru.it/3922>\)](#). [We recommend this 3W 4 ohm speaker \(<http://adafru.it/3968>\)](#)
- There are various Arduino / CircuitPython / MicroPython libraries can be used to talk to the SD card and I2S amplifier.

Speaker, microSD memory card, and QT Py are not included.

Pinouts



This breakout is meant to be used with a QT Py or Xiao form factor board. Using with other form factor boards may be difficult.

I2S Amplifier

- The grey square located towards the top right of the BFF is the **MAX98357 I2S amplifier chip**. It supports digital audio.

Default I2S Pins

These pins are compatible with ESP32-series, nRF52840 and RP2040 microcontroller boards.

- **A1** - This is used for audio data, or **DAT**.
- **A2** - This is used for wordselect clock, or **LR**.
- **A3** - This is used for bitclock, or **BCLK**.

These pins will not work with the QT Py SAMD21 microcontroller board.

Default SD Card Pins

The SD Card is connected to the default SPI pins on the QT Py. **SDCS** is connected to pin **A0**.

- **M0** - This is the SPI MOSI (**Microcontroller Out / Serial In**) pin. It is used to send data from the microcontroller to the SD card. It is connected to the default **MOSI** pin on the QT Py.
- **M1** - This is the SPI MISO (**Microcontroller In / Serial Out**) pin. It's used for sending data from the SD card to the microcontroller. It is connected to the default **MISO** pin on the QT Py.
- **SCK** - This is the SPI clock input pin. It is connected to the default **SCK** pin on the QT Py.
- **A0** - This is the SD chip select pin (**SDCS**).

I2S Gain Jumper

On the back of the board, towards the top, are the I2S gain jumper pads. They are outlined in white on the board silk and are labeled **Gain**. The default gain for the amplifier is 9 dB.

- To set the gain to 6 dB, solder the center jumper, labeled **9**, to the jumper above labeled **6**. This connects the **GAIN** pin on the I2S amplifier to **5V**.
- To set the gain to 12 dB, solder the center jumper, labeled **9**, to the right jumper below labeled **12**. This connects the **GAIN** pin on the I2S amplifier to **GND**.

Speaker Connector

- This two-pin **PicoBlade-compatible speaker connector**, located at the center of the front of the BFF, is meant for plugging in a **4-8Ω, 3W-or-less speaker**.
- To connect a speaker with bare wires, you'll want to pick up [this cable](http://adafru.it/3922) (<http://adafru.it/3922>).
- If you need a speaker, [this speaker](http://adafru.it/3968) (<http://adafru.it/3968>) is an excellent place to start.

micro SD Card Slot

- On the front of the board, towards the bottom, is the micro SD card slot. You can use any micro SD card that supports SPI mode with one CS pin.
-

CircuitPython

As of CircuitPython 9, a mount point (folder) named /sd is required on the CIRCUITPY drive. Make sure to create that directory after upgrading CircuitPython.

Follow these steps to create the /sd directory

<https://adafru.it/19ei>

It's easy to use the **Audio BFF** with CircuitPython, the builtin `audiobusio` module and the [Adafruit_CircuitPython_SD](#) (<https://adafru.it/zwC>) module. These modules allow you to easily write Python code that lets you read from an attached SD card and play audio.

micro SD Card Prep

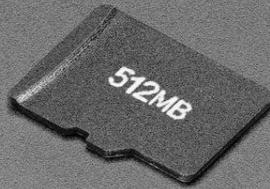
There are five **.wav** audio files available for download below in a **.zip** file. After downloading the **.zip** file, extract the contents:

- **adabot.wav**
- **hello.wav**
- **interesting.wav**
- **uhoh.wav**
- **whhaatt.wav**

Then, drag and drop the five **.wav** files onto a micro SD card that you'll use with the Audio BFF.

Audio_BFF_Example_.WAV_Files.zip

<https://adafru.it/18Qa>



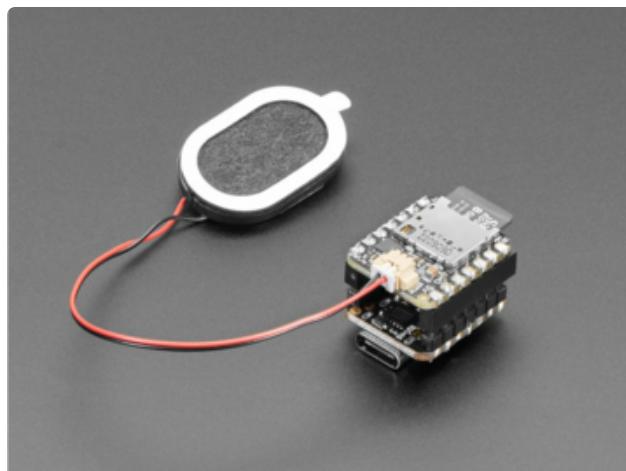
512MB micro SD Memory Card

Add storage in a jiffy using this 512MB microSD card. Preformatted to FAT32, so it works out of the packaging with our projects. Works great with any device in the...

<https://www.adafruit.com/product/5252>

CircuitPython Microcontroller Wiring

Plug an Audio BFF into your QT Py or Xiao form factor board exactly as shown below. Here's an example of connecting a QT Py RP2040 to the BFF.



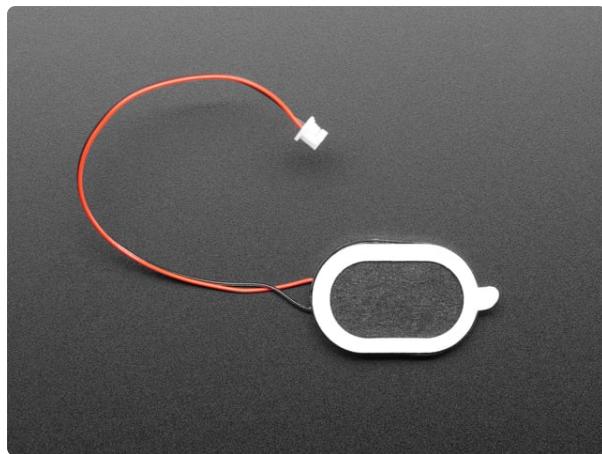
Connect the QT Py RP2040 with plug headers into the Audio BFF with socket headers. They should be plugged in with the backs of the boards facing each other.

For more information on soldering socket headers, [check out this Learn Guide \(https://adafru.it/18gf\)](#).

Then, plug in a speaker to the speaker input. Finally, insert the micro SD card with the audio files that you downloaded above into the micro SD card slot.

How to Solder Headers Learn Guide

<https://adafru.it/18gf>



Mini Oval Speaker - 8 Ohm 1 Watt

Hear the good news! This wee speaker is a great addition to any audio project where you need 8 ohm impedance and 1W or less of power. We particularly like...

<https://www.adafruit.com/product/3923>

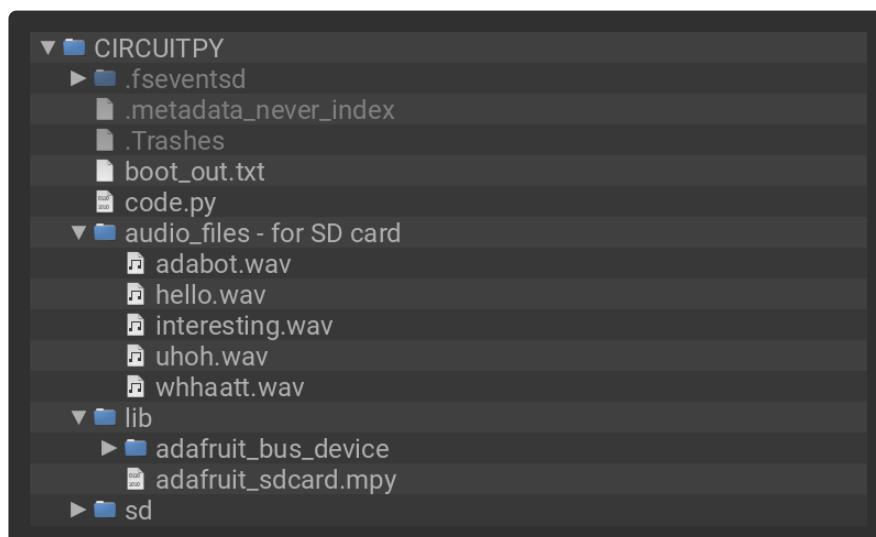
CircuitPython Usage

To use with CircuitPython, you need to first install the `adafruit_sdcard` library, and its dependencies, into the **lib** folder on your **CIRCUITPY** drive. Then you need to update `code.py` with the example script.

Thankfully, we can do this in one go. In the example below, click the **Download Project Bundle** button below to download the necessary libraries and the `code.py` file in a zip file. Extract the contents of the zip file, and copy the **entire lib folder** and the `code.py` file to your **CIRCUITPY** drive.

Your **CIRCUITPY/lib** folder should contain the following folders and files:

- `/adafruit_bus_device`
- `adafruit_sdcard.mpy`



Example Code

Once everything is saved to the **CIRCUITPY** drive, [connect to the serial console \(<https://adafru.it/Bec>\)](#) to see the data printed out!

```
# SPDX-FileCopyrightText: 2023 ladyada for Adafruit Industries
#
# SPDX-License-Identifier: MIT

# Demo audio player that plays random wav files from internal storage or
# SD card. Default pinout matches the Audio BFF for QT Py S2, S3 and RP2040

import os
import random
import audiocore
import board
import audiobusio
import audiomixer
import adafruit_sdcard
import storage
import digitalio

card_cs = digitalio.DigitalInOut(board.A0)
card_cs.direction = digitalio.Direction.INPUT
card_cs.pull = digitalio.Pull.UP
sdcard = None

DATA = board.A1
LRCLK = board.A2
BCLK = board.A3
audio = audiobusio.I2SOut(BCLK, LRCLK, DATA)
mixer = None

button = digitalio.DigitalInOut(board.BUTTON)
button.switch_to_input(pull=digitalio.Pull.UP)

wave_files = []
for filename in sorted(os.listdir("/")):
    filename = filename.lower()
    if filename.endswith(".wav") and not filename.startswith("."):
        wave_files.append(filename)

def open_audio():
    n = random.choice(wave_files)
    print("playing", n)
    f = open(n, "rb")
    w = audiocore.WaveFile(f)
    return f, w

wavefile = 0

while True:
    if not sdcard:
        try:
            sdcard = adafruit_sdcard.SDCard(board.SPI(), card_cs)
            vfs = storage.VfsFat(sdcard)
            storage.mount(vfs, "/sd")
            print("Mounted SD card")
            wave_files = ["/"+file for file in os.listdir('/') if
file.endswith('.wav')]
            wave_files += ["/sd/"+file for file in os.listdir('/sd') if
file.endswith('.wav')]
            print(wave_files)
        except OSError:
```

```

    pass

if not button.value:
    if mixer and mixer.voice[0].playing:
        print("stopping")
        mixer.voice[0].stop()
        if wavefile:
            wavefile.close()
    else:
        wavefile, wave = open_audio()
        mixer = audiomixer.Mixer(voice_count=1,
                                sample_rate=wave.sample_rate,
                                channel_count=wave.channel_count,
                                bits_per_sample=wave.bits_per_sample,
                                samples_signed=True)
        mixer.voice[0].level = 0.1
        audio.play(mixer)
        mixer.voice[0].play(wave)

while not button.value:
    pass

```

In the code, the SD card is mounted and the available .wav files are added to an array and printed to the Serial Console. Then, when you press the **BOOT** button, a random audio file will play from the array. If you press the **BOOT** button while a file is playing, the file will stop.

```

CircuitPython REPL
Auto-reload is on. Simply save files over USB to run them or enter REPL to disable.
code.py output:
Mounted SD card
['/sd/hello.wav', '/sd/interesting.wav', '/sd/uhoh.wav', '/sd/whhaatt.wav', '/sd/adabot.wav']
playing /sd/adabot.wav
stopping
playing /sd/adabot.wav
playing /sd/interesting.wav
playing /sd/uhoh.wav
playing /sd/uhoh.wav
playing /sd/uhoh.wav
stopping
playing /sd/adabot.wav

```

If you encounter any errors, such as `RuntimeError: Mount point directory missing` or an `OSError`, make sure that you have created an `/sd` directory on your CIRCUITPY drive.

Python Docs

[Python Docs \(https://adafru.it/18uf\)](https://adafru.it/18uf)

Arduino

Using the Audio BFF with Arduino involves plugging the breakout into your Arduino-compatible QT Py or Xiao form factor board, installing the [Adafruit WavePlayer library](https://adafru.it/18Qf) (<https://adafru.it/18Qf>), and running the provided example code.

This example has been tested with the QT Py RP2040. A different chip family (ESP32, SAMD, etc) may require different example code.

micro SD Card Prep

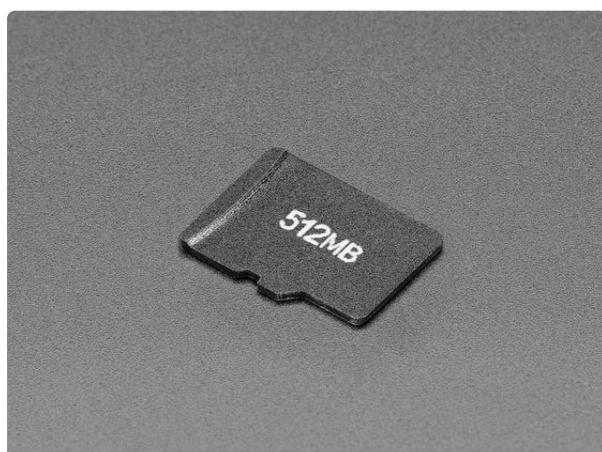
There are five **.wav** audio files available for download below in a **.zip** file. After downloading the **.zip** file, extract the contents:

- **adabot.wav**
- **hello.wav**
- **interesting.wav**
- **uhoh.wav**
- **whhaatt.wav**

Then, drag and drop the five **.wav** files onto a micro SD card that you'll use with the Audio BFF.

Audio_BFF_Example_.WAV_Files.zip

<https://adafru.it/18Qa>



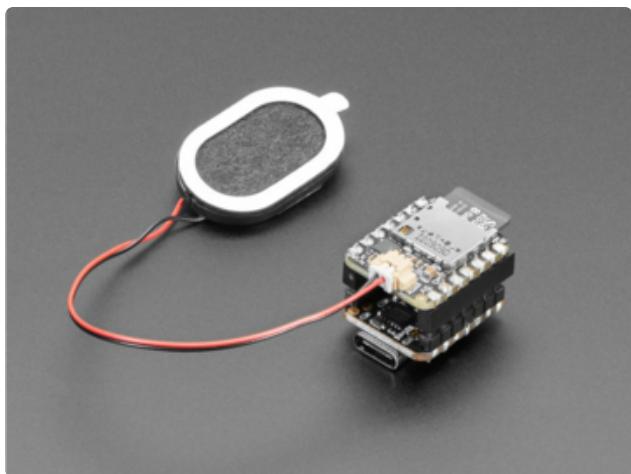
512MB micro SD Memory Card

Add storage in a jiffy using this 512MB microSD card. Preformatted to FAT32, so it works out of the packaging with our projects. Works great with any device in the...

<https://www.adafruit.com/product/5252>

Wiring

Plug an Audio BFF into your QT Py or Xiao form factor board exactly as shown below. Here's an example of connecting a QT Py RP2040 to the BFF.



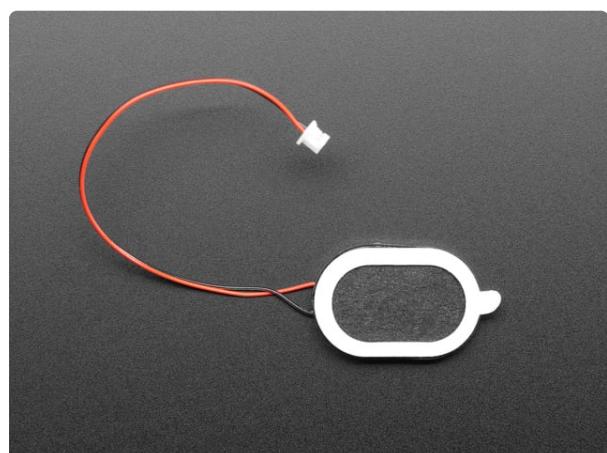
Connect the QT Py RP2040 with plug headers into the Audio BFF with socket headers. They should be plugged in with the backs of the boards facing each other.

For more information on soldering socket headers, [check out this Learn Guide](https://adafru.it/18gf) (<https://adafru.it/18gf>).

Then, plug in a speaker to the speaker input. Finally, insert the micro SD card with the audio files that you downloaded above into the micro SD card slot.

How to Solder Headers Learn Guide

<https://adafru.it/18gf>



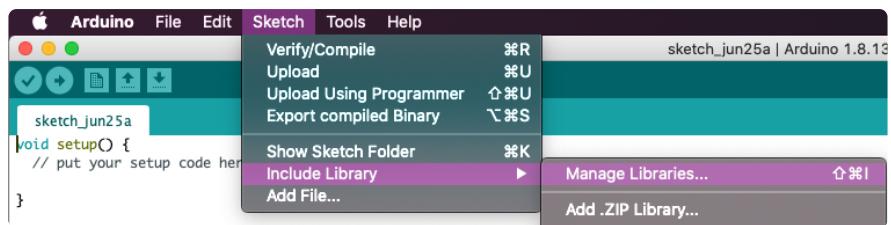
Mini Oval Speaker - 8 Ohm 1 Watt

Hear the good news! This wee speaker is a great addition to any audio project where you need 8 ohm impedance and 1W or less of power. We particularly like...

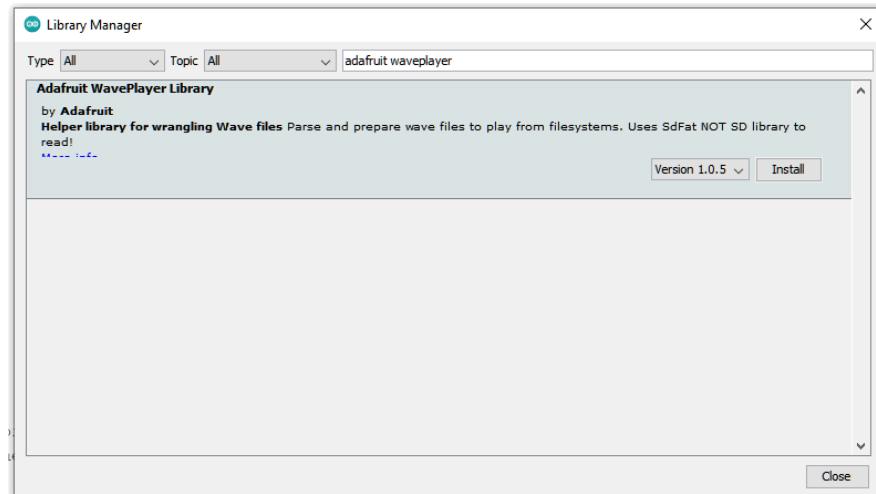
<https://www.adafruit.com/product/3923>

Library Installation

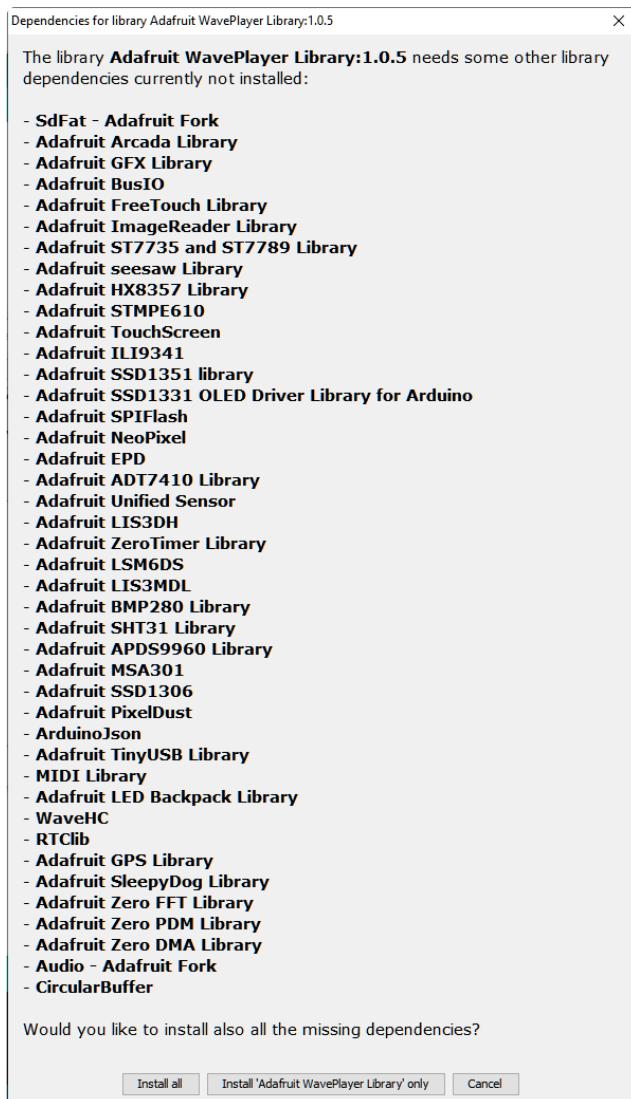
You can install the **Adafruit WavePlayer** library for Arduino using the Library Manager in the Arduino IDE.



Click the **Manage Libraries ...** menu item, search for **Adafruit WavePlayer** and select the **Adafruit WavePlayer** library:



If asked about dependencies, click "Install all".



If the "Dependencies" window does not come up, then you already have the dependencies installed.

If the dependencies are already installed, you must make sure you update them through the Arduino Library Manager before loading the example!

Example Code

```
// Audio BFF example for QT Py RP2040. Loops through all WAV files
// in specified directory of the SD Card. Files are
// not alphabetized and will usu. play in order they were installed.
// Requires SdFat - Adafruit Fork and Adafruit_WavePlayer libraries.

#include "SdFat.h"
#include <Adafruit_WavePlayer.h>
#include <I2S.h>

I2S          i2s(OUTPUT);           // I2S peripheral is...
Adafruit_WavePlayer player(false, 16); // mono speaker, 16-bit out.

volatile bool      playing    = false; // For syncing cores.
```

```

volatile bool      load      = false;

#if SD_FAT_TYPE == 0
SdFat sd;
File dir;
File file;
#elif SD_FAT_TYPE == 1
SdFat32 sd;
File32 dir;
File32 file;
#elif SD_FAT_TYPE == 2
SdExFat sd;
ExFile dir;
ExFile file;
#elif SD_FAT_TYPE == 3
SdFs sd;
FsFile dir;
FsFile file;
#else // SD_FAT_TYPE
#error invalid SD_FAT_TYPE
#endif // SD_FAT_TYPE

// I2S GPIO pin numbers
#define pBCLK A3 // QT Py Audio BFF default BITCLOCK
#define pWS    A2 // QT Py Audio BFF default LRCLK
#define pDOUT A1 // QT Py Audio BFF default DATA

#define SD_CS_PIN A0 // QT Py Audio BFF SD chip select
#define SPI_CLOCK SD_SCK_MHZ(50)

#if HAS_SDIO_CLASS
#define SD_CONFIG SdioConfig(FIFO_SDIO)
#elif ENABLE_DEDICATED_SPI
#define SD_CONFIG SdSpiConfig(SD_CS_PIN, DEDICATED_SPI, SPI_CLOCK)
#else // HAS_SDIO_CLASS
#define SD_CONFIG SdSpiConfig(SD_CS_PIN, SHARED_SPI, SPI_CLOCK)
#endif // HAS_SDIO_CLASS

void setup() {

    // debug output at 115200 baud
    Serial.begin(115200);
    while (! Serial) delay(10);
    // setup SD-card
    Serial.print("Initializing SD card...");
    if (!sd.begin(SD_CONFIG)) {
        Serial.println(" failed!");
        return;
    }
    Serial.println(" done.");
}

void setup1() {
    // Configure I2S, enable audio amp
    i2s.setDATA(pDOUT);
    i2s.setBCLK(pBCLK);
    i2s.setBitsPerSample(16);
}

uint8_t num;

// Core 0 main loop - scans folder for WAV files to play
void loop() {

    if (dir.open("/")) {
        while (file.openNext(&dir, O_RDONLY)) {
            char filename[EXFAT_MAX_NAME_LENGTH + 1];
            file.getName(filename, sizeof filename);

```

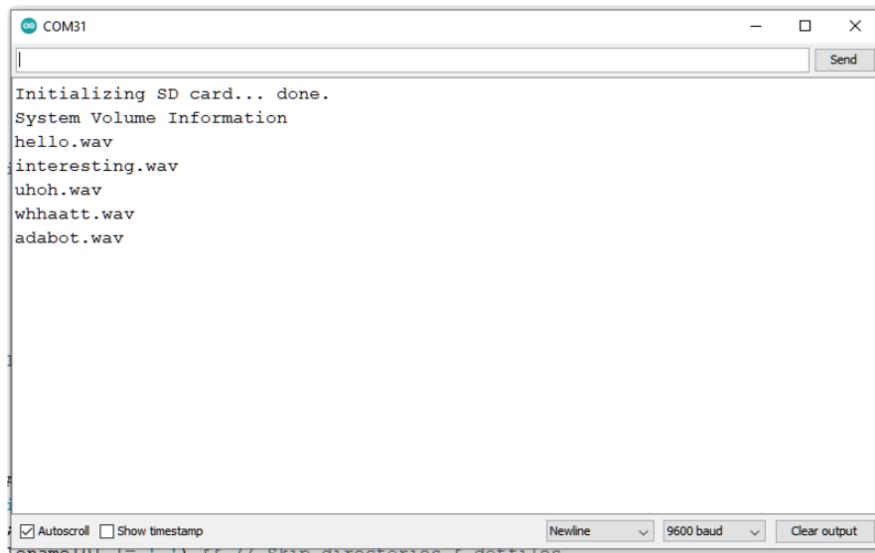
```

Serial.println(filename);
if (file.isFile() && (filename[0] != '.') && // Skip directories & dotfiles
    !strcasecmp(&filename[strlen(filename) - 4], ".wav")) {
    play_i2s(file);
    delay(1000);
} // end WAV
file.close();
} // end file open
} // end dir open
} // end loop()

// Core 1 main loop handles I2S audio playback. This allows concurrency
// when pre-loading the next audio buffer without getting into hairy DMA.
// Global 'load' flag is set here when core 0 should load next chunk.
void loop1() {
    if (playing) {
        wavSample sample;
        switch (player.nextSample(&sample)) {
            case WAV_LOAD:
                load = true; // No break, pass through...
            case WAV_OK:
                i2s.write((int32_t)sample.channel0 - 32768);
                i2s.write((int32_t)sample.channel1 - 32768);
                break;
            case WAV_EOF:
            case WAV_ERR_READ:
                playing = load = false;
        } // end switch
    } // end playing
} // end loop1()

// Play WAV file (already opened above). Runs on core 0.
void play_i2s(File file) {
    digitalWrite(LED_BUILTIN, HIGH);
    uint32_t rate;
    wavStatus status = player.start(file, &rate);
    if ((status == WAV_OK) || (status == WAV_LOAD)) {
        if (i2s.begin(rate)) {
            playing = load = true;
            while (playing) {
                if (load || (status == WAV_LOAD)) {
                    load = false;
                    status = player.read();
                    if (status == WAV_ERR_READ) playing = false;
                } // end load
            } // end playing
            i2s.write((int16_t)0);
            i2s.write((int16_t)0);
            i2s.end();
        } else {
            Serial.println("Failed to initialize I2S!");
        } // end i2s
    } // end WAV_OK
    digitalWrite(LED_BUILTIN, LOW);
} // end play()

```



Upload the sketch to your board and open up the Serial Monitor (**Tools -> Serial Monitor**) at 115200 baud. Once the SD card is initialized, it will print to the Serial Monitor. You will hear the audio files play through the speaker. As they play, the file names will be printed to the Serial Monitor.

Arduino Docs

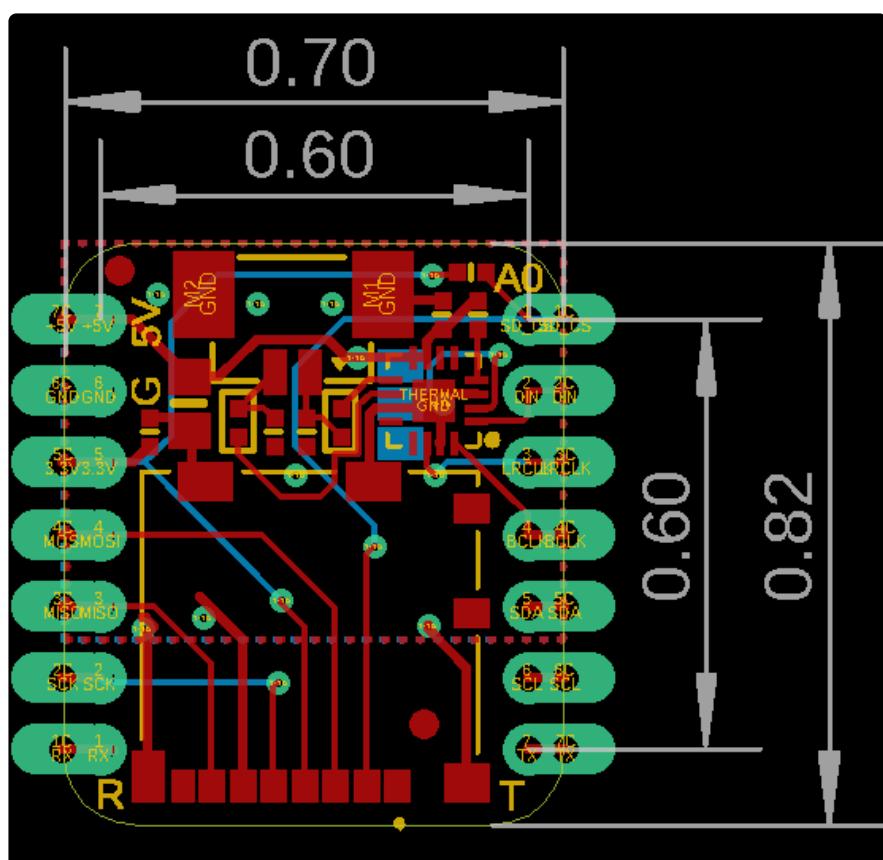
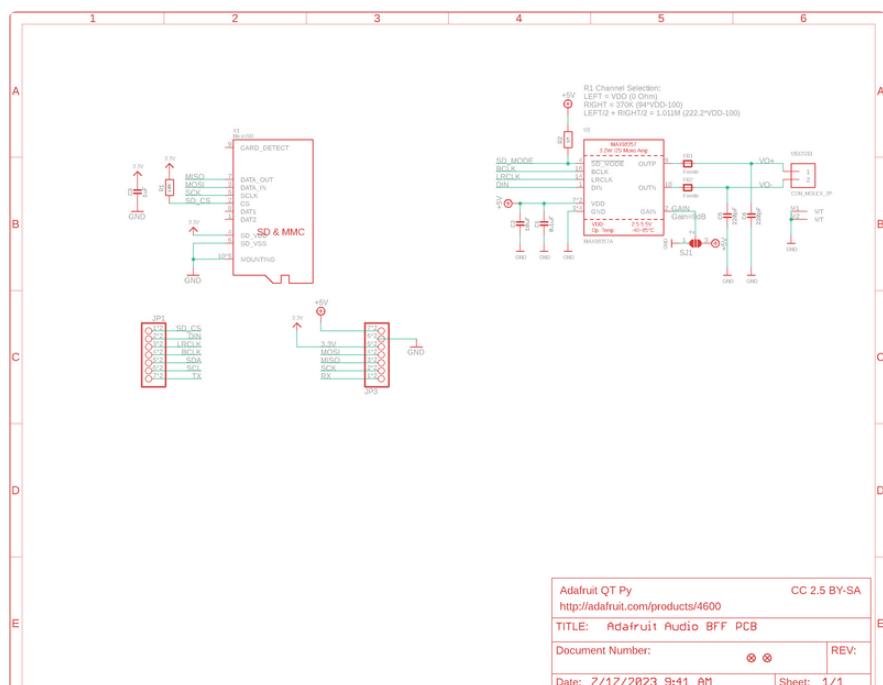
[Arduino Docs](https://adafru.it/18QB) (<https://adafru.it/18QB>)

Downloads

Files

- [MAX98357 Datasheet](http://adafru.it/30069835798357) (<http://adafru.it/30069835798357>)
- [EagleCAD PCB files on GitHub](https://adafru.it/18QC) (<https://adafru.it/18QC>)
- [3D files on GitHub](https://adafru.it/19hE) (<https://adafru.it/19hE>)
- [Fritzing object in the Adafruit Fritzing Library](https://adafru.it/18QF) (<https://adafru.it/18QF>)

Schematic and Fab Print



3D Models

