Grand Central Soundboard in Ten Minutes

Created by Anne Barela

https://learn.adafruit.com/grand-central-soundboard-in-ten-minutes

Last updated on 2024-06-03 02:37:22 PM EDT

# Table of Contents

Overview

The Adafruit Grand Central is arguably one of the most feature-rich boards Adafruit produces. Adafruit is planning some great tutorials to showcase using the Grand Central in many ways. But what about a simple "I just got my board and want to do a quick demo to kick the tires?"

Rather than code some "blink the LED" or "Hello World" project, here is a quick project only using a few parts which will have you building a fully functional soundboard in about ten minutes.

The ability to do this partially lays with the powerful processor on the Grand Central but has everything to do with being able to run CircuitPython. This easy to use yet powerful code is loaded onto the board by saving a file onto the flash drive, which shows up when you plug the Grand Central into your PC / Mac / Linux / Chromebook via a USB cable. It is that easy, no software framework to set up, no compiling, no compatibility issues. It really allows you to plug, copy, and go!

This project will get you started with the code and eight public domain/creative commons sound files and you can make this project your own by changing things up.

Parts
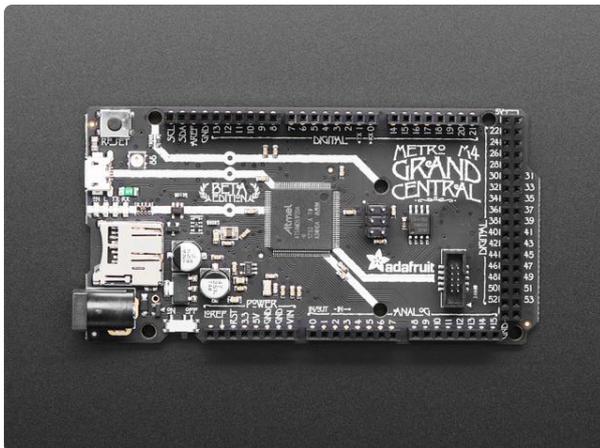
Adafruit Grand Central M4 Express featuring the SAMD51

Are you ready? Really ready? Cause here comes the Adafruit Grand Central featuring the Microchip ATSAMD51. This dev board is so big, it's not...

https://www.adafruit.com/product/4064

Membrane 3x4 Matrix Keypad + extras

Punch in your secret key into this numeric matrix keypad. This keypad has 12 buttons, arranged in a telephone-line 3x4 grid. It's made of a thin, flexible membrane material with an...

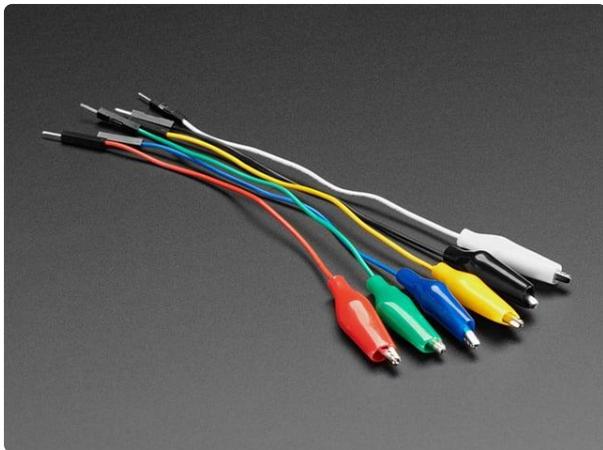https://www.adafruit.com/product/419

USB Powered Speakers

Add some extra boom to your audio project with these powered loudspeakers. We sampled half a dozen different models to find ones with a good frequency response, so you'll get...

https://www.adafruit.com/product/1363

Small Alligator Clip to Male Jumper Wire Bundle - 6 Pieces

When working with unusual non-header-friendly surfaces, these handy cables will be your best friends! No longer will you have long, cumbersome strands of alligator clips. These...

https://www.adafruit.com/product/3448

USB cable - USB A to Micro-B

This here is your standard A to micro-B USB cable, for USB 1.1 or 2.0. Perfect for connecting a PC to your Metro, Feather, Raspberry Pi or other dev-board or...

https://www.adafruit.com/product/592

Power

For a demo project, USB power for the speakers and the Grand Central is perfect. If you really want to take this on the go, an optional a cell phone recharge battery with two USB ports would help.

USB Battery Pack for Raspberry Pi - 10000mAh - 2 x 5V outputs
A large-sized rechargeable battery pack for your Raspberry Pi (or Arduino, or
https://www.adafruit.com/product/1566

or maybe two of these:

USB Battery Pack - 2200 mAh Capacity - 5V 1A Output
A smaller-sized rechargeable battery pack for your Raspberry Pi or Raspberry...
https://www.adafruit.com/product/1959

Software

The Grand Central comes preloaded with CircuitPython. CircuitPython allows you to program a wide variety of microcontroller boards and single board computers without installing software on your computer. All you need is a plain text editor, found on all computers since before IBM made the PC.

If you would like a Python-friendly editor which has a serial connection to the board, a plotter, and more, Adafruit recommends Mu, a free editor made for ease of use. See this page on installing Mu (https://adafru.it/ANO) if you like.

Download Library Files

Plug your Grand Central board into your computer via a USB cable. Please be sure it is a good power+data cable so the computer can talk to the Grand Central board.

A new disk should appear in your computer's file explorer/finder called **CIRCUITPY**. This is the place we'll copy the code, sound files, and code libraries.

Create a new directory on your Grand Central's **CIRCUITPY** drive named **lib**.

Download the latest CircuitPython drivers to your computer using the green button below. Save to your computer's hard drive where you can find it.

> **Go to GitHub to get the latest CircuitPython library bundle**

https://adafru.it/zB-

With your file explorer/finder, browse to the bundle and open it up. Copy the following file from the library bundle to your **CIRCUITPY lib** directory you made earlier:

- **adafruit_matrixkeypad.mpy**

All of the other necessary code is baked into CircuitPython!

Download Code and Sounds

Below is the code for this project. To grab the entire package with the sample train sound files, select **Download: Project Zip** to your computer hard drive. This will contain **code.py** and a subdirectory called **sounds** which contains the wav files.

With your computer file explorer/finder, browse to the location where you saved the code zip file, open the zip file, open the folder **Grand_Central_Soundboard** directory and copy the files **code.py** and the **sounds** directory to your **CIRCUITPY** drive. The **sounds** directory will contain 8 wav files and a license text file.

```python
# SPDX-FileCopyrightText: 2019 Anne Barela for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import time
import os
import digitalio
import audioio
import audiocore
import board
import adafruit_matrixkeypad

# Membrane 3x4 matrix keypad - https://www.adafruit.com/product/419
cols = [digitalio.DigitalInOut(x) for x in (board.D3, board.D2, board.D1)]
rows = [digitalio.DigitalInOut(x) for x in (board.D7, board.D6, board.D5, board.D4)]

keys = ((1, 2, 3),
        (4, 5, 6),
        (7, 8, 9),
        ('*', 0, '#'))

keypad = adafruit_matrixkeypad.Matrix_Keypad(rows, cols, keys)

wavefiles = [file for file in os.listdir("/sounds/")
             if (file.endswith(".wav") and not file.startswith("._"))]
if len(wavefiles) < 1:
    print("No wav files found in sounds directory")
else:
    print("Audio files found: ", wavefiles)

# audio output
gc_audio = audioio.AudioOut(board.A0)
```

```
audio_file = None

def play_file(filename):
    global audio_file  # pylint: disable=global-statement
    if gc_audio.playing:
        gc_audio.stop()
    if audio_file:
        audio_file.close()
    audio_file = open(filename, "rb")
    wav = audiocore.WaveFile(audio_file)
    gc_audio.play(wav)

while True:
    keys = keypad.pressed_keys
    if keys:
        print("Pressed: ", keys)
        button = keys[0]
        if button > 0 and button < 9:
            soundfile = "/sounds/0"+str(keys[0])+".wav"
            play_file(soundfile)
        if button == 0:
            gc_audio.stop()
            if audio_file:
                audio_file.close()
    time.sleep(0.1)
```
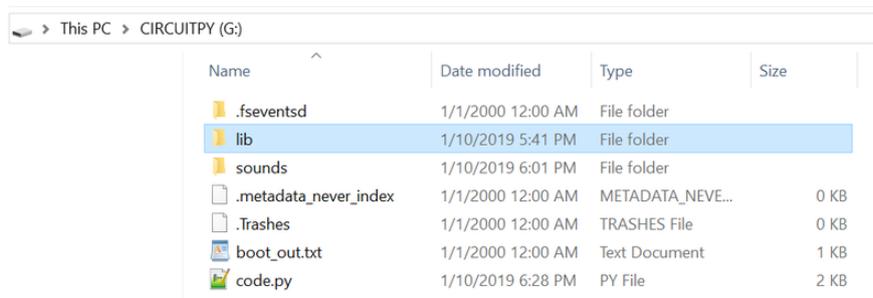
Coding the project in CircuitPython makes this project truly easy. Copy the code and go! You can see the libraries that facilitate using the Grand Central and the matrix keypad are already coded - the program defines what we're using (keypad, audio), grabs the sound files, then checks to see when you press a key to trigger the appropriate effect.

Files Double Check

A listing of the **CIRCUITPY** drive in your file explorer/finder should look similar to the one below. The extra files starting with a period (if you see them) are for Mac file handling, you don't need to deal with those. You should just ensure you have **code.py**, a **sounds** directory (containing wav files) and a **lib** directory containing the matrix keypad library.


On your Grand Central **CIRCUITPY** flash drive, if you are missing the **code.py**, the **lib** directory or the **sounds** directory (with the respective files in the directory files), please go back and download them. They'll all be needed for the final project.

Connections and Operation

This project is built to use a minimum of parts to get started quickly. It can easily be expanded in multiple ways if desired.
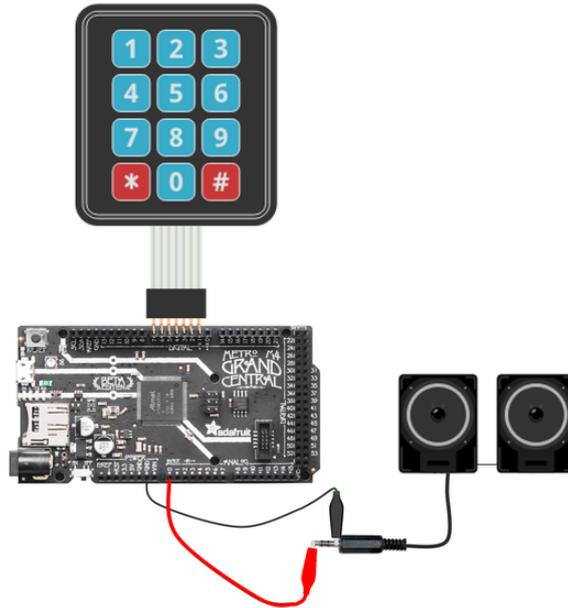
The matrix keypad from Adafruit comes with some male-male pins allowing the female connector to mate with a female breadboard or circuit board. Place the header onto the keypad connector and plug the pins into the Grand Central pins **1** to **7**. The colored face of the keypad should be up as shown in the diagram.

Place a **black** header male pin to alligator clip from **GND** on the Grand Central to the amplified speaker TRS connector, the **sleeve** closest to the plastic of the plug. Take a **red** jumper, plug into the Grand Central **A0** pin and clip the alligator clip to the **tip** of the speaker plug. This will make a mono channel amplifier connection to one of the speakers of the stereo set.

Ensure the Grand Central is plugged into power, either 9 volts on the barrel connector or maybe connected to the USB port to draw 5 volts (such as a computer connection or a 5V "cell phone recharge" battery.

Power the speakers by placing the speaker USB connector to a suitable 5 volt USB jack.

Operation

Turn on the power for the speakers if necessary and adjust the volume so sound can be heard when operating.

Press one of the keys on the keypad, from 1 to 8. You should head a sound effect related to trains. Pressing another key will interrupt the playing sound to play the new sound. You can also use the zero (0) key to stop a playing sound.

If the sound cannot be heard, check your speaker volume and connections. Be sure the speakers and Grand Central are powered and that the keypad is plugged in correctly.

Have fun with all the train sounds on your new soundboard!

The attribution for the sounds used are in the file **Sound-licenses.txt** in the **/sounds** directory. Two are public domain, the rest are CC3 attribution.

Going Further

You are not limited to these 8 train related sounds. You can take any sound, convert it to a wav file, name it from 01.wav to 08.wav and have it triggered. Use this guide (https://adafru.it/BvU) to convert a file to the appropriate format for playing on microcontrollers: mono, 16-bit PCM, 22,050 Hz (or less), WAV format.

You may consider many variations on this project. Some code changes and all 12 keys can play sounds. A bigger keypad or multiple keypads would allow selection of more sound effects. With all the input pins on the Grand Central, a very elaborate soundboard could be created. The limiting factor would then be the amount of space on the Grand Central flash drive for sound files. In that case, the SD card slot could read a large SD-micro card with sounds on it, leading to some truly Grand effects.