Make It Sound

Created by Anne Barela


https://learn.adafruit.com/make-it-sound
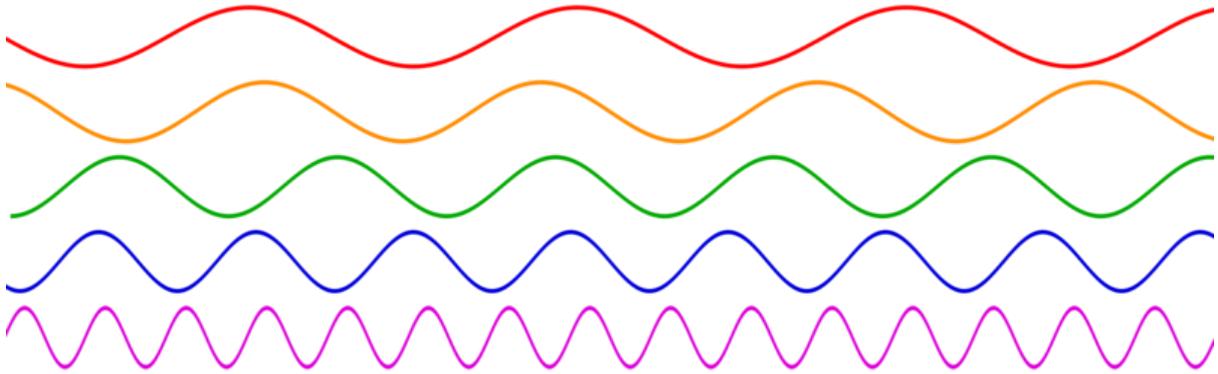
Last updated on 2024-06-03 02:26:25 PM EDT

Table of Contents

Overview

Making and reacting to sound is fundamental to daily life. Your next project can use sound quickly and easily to provide indications something is happening or as the perfect accompaniment to your projects use.

Using either Microsoft MakeCode or CircuitPython, you can use sound in your project.

The code will use the Adafruit Circuit Playground Express board. If you have a project that is robotic, you can add the Adafruit Crickit robotics board to Circuit Playground Express which provides the benefit of an amplifier for much better sounds than that provided on the Circuit Playground Express on-board speaker.

Part List
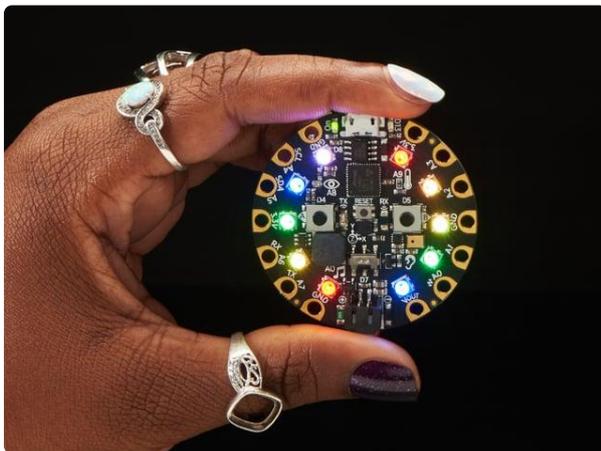
Circuit Playground Express
Circuit Playground Express is the next step towards a perfect introduction to electronics and programming. We've taken the original Circuit Playground Classic and...
https://www.adafruit.com/product/3333

To Go Further
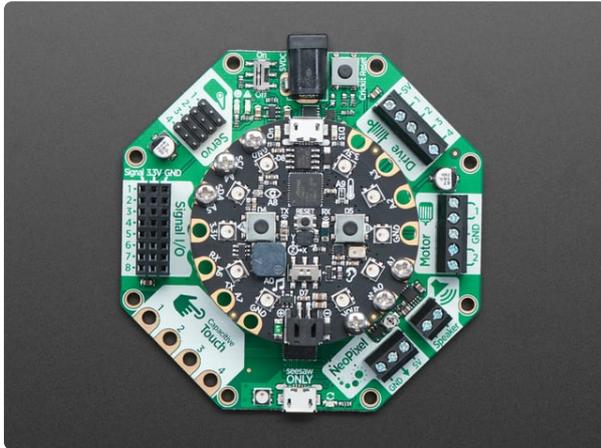
Adafruit CRICKIT for Circuit Playground Express

Sometimes we wonder if robotics engineers ever watch movies. If they did, they'd know that making robots into servants always ends up in a robot rebellion. Why even go down that...

https://www.adafruit.com/product/3093

Mini Metal Speaker w/ Wires - 8 ohm 0.5W

Listen up! This tiny 1" diameter speaker cone is the perfect addition to any small audio project where you need an 8 Ω impedance and will be using 0.5W or less of power. We...

https://www.adafruit.com/product/1890

Music and Sound in MakeCode

If you are new to Microsoft MakeCode, you can learn the basics of MakeCode here (https://adafru.it/wWd).

Currently MakeCode has a whole block group, **MUSIC**, devoted to making sounds:
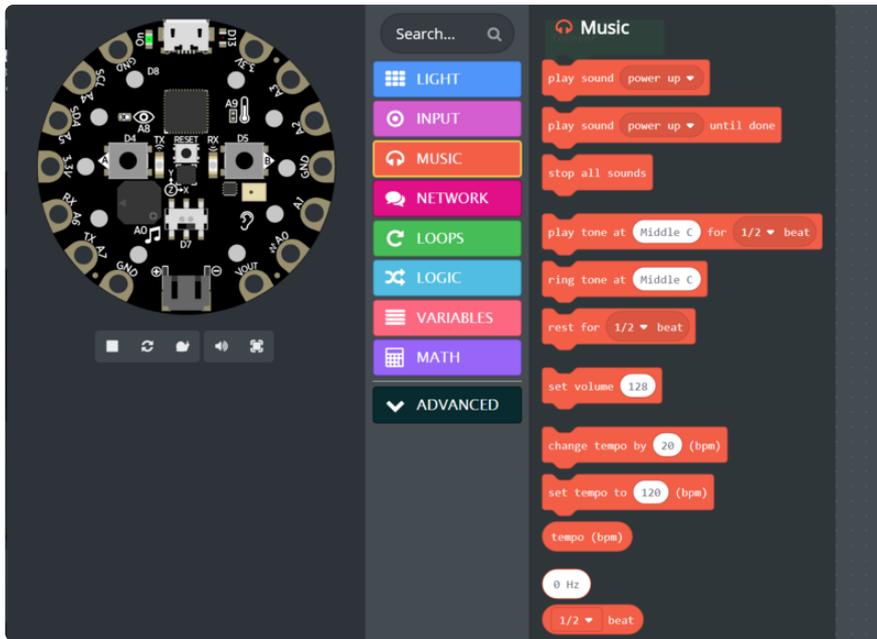
The main groups of blocks provide the following:

- Setting the volume (loudness) of your tones
- Play sounds at startup or stop playing all sounds
- Play any frequency tone
- Play a musical note
- Play music faster or slower
- Note duration (beat) and pause between notes (rest)

The following code will play a `220` Hz tone when you press the A button and a `440` Hz tone when you press the B button. Within the **INPUT** group are the loops used for button presses. Select two `on button` blocks and add to each a `play tone at` block from the **MUSIC** group.

Change the value of the button A play tone at to `220` and it will replace the number with the value `Low A`. Replace the value of the Button B play tone to `440` and MakeCode will change the number to `Middle A`. MakeCode recognizes certain sound frequencies as musical notes and will change those accordingly. Alternatively MakeCode will put up a small keyboard when you work with the play tone at block's value to help you select musical notes if you like.

Click here to open this project in MakeCode

https://adafru.it/BVY
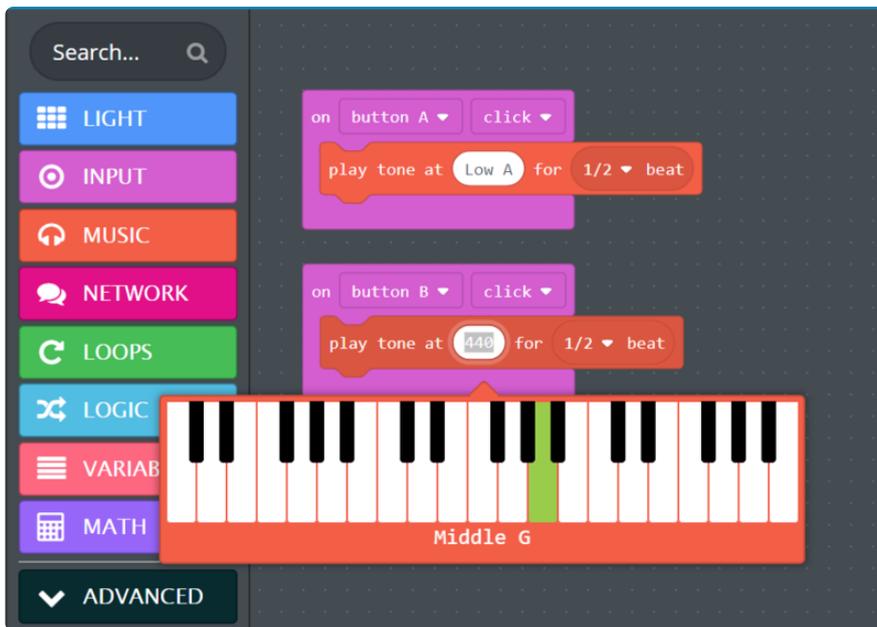
Pull up this example and press the Download button to save it to your Circuit Playground Express. Press the A and B buttons. You get two different sounds! Feel free to change them up.

Predefined Sounds

MakeCode comes with several predefined sounds that have several musical notes. Change the Button A and Button B contents both to use the `play sound until done` block. Change them so each button plays a different sound:

Click here to open this project in MakeCode

https://adafru.it/BVZ
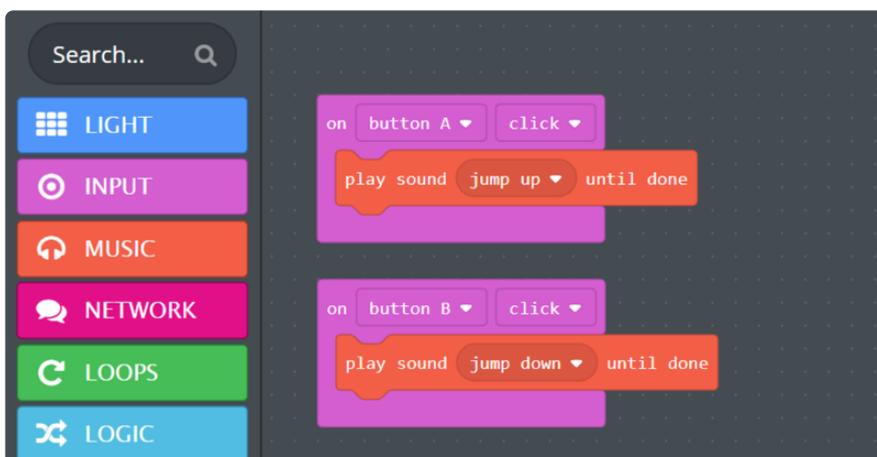
Push the buttons and you get two melodies. Feel free to use the predefined sounds in your code.

Music

Typing in your own musical melody is possible but you need to know a bit about musical notation. I have found that some Internet sites are good at providing the notes, some are not. You can look to someone who knows how to read music to help if all you find is sheet music.

The following program plays a well known tune. See how many notes it takes to do the short portion? The code also uses the A and B buttons to change how fast or slow the music plays.

Click here to open this project in MakeCode

https://adafru.it/BV-
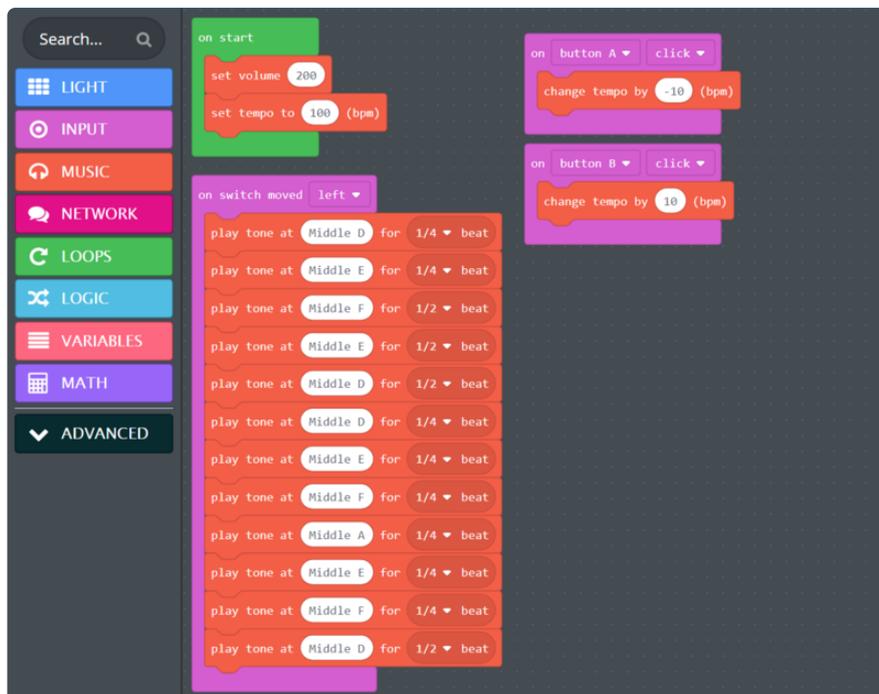
Using MakeCode, it is a bit tedious to put in lots of custom sounds. On the next page, CircuitPython will be used which also allows use of pre-recorded sound files which MakeCode cannot use at present.

Music and Sound in CircuitPython
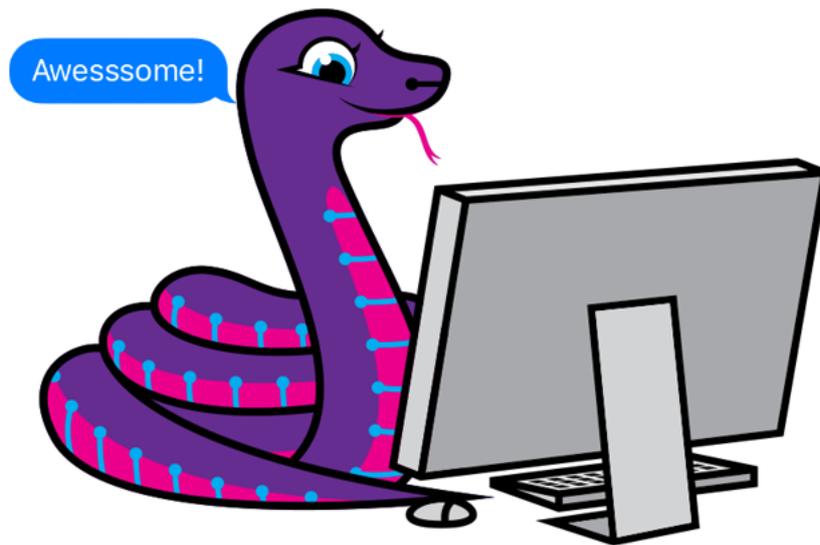
CircuitPython has similar capabilities to MakeCode for playing tones. CircuitPython, though, doesn't understand music so you would have to provide your own understanding of which musical notes correspond to which sound frequencies.

You can look up which musical notes correspond to which frequencies on a number of websites including this chart on Wikipedia (https://adafru.it/BW0).

Simple Tones

Adafruit provides the `adafruit_circuitplayground` library to provide high level access to the various parts of the board in CircuitPython. The functions CircuitPython provides for using the speaker:

- `cpx.play_tone(frequency, duration)`
- `cpx.start_tone(frequency)`
- `cpx.stop_tone()`
- `cpx.play_file("laugh.wav")`

Most often for tones, it is known how long you might want to play the tone, for a second or so or perhaps until you manually turn it off (tones that play a long time are annoying unless you are doing it on purpose like for an alarm).

The two tone program using buttons A and B for two versions of the musical note A is:

```
from adafruit_circuitplayground.express import cpx
```

```
while True:
    if cpx.button_a:
        cpx.play_tone(220, 0.5)
    elif cpx.button_b:
        cpx.play_tone(440, 0.5)
```

In this case we play the frequencies 220 Hz and 440Hz each for 1/2 second.

Continuous Tones

For an alarm, say your soil is too dry and you need to water or your family member tripped your alarm. We'll use a capacitive touch on Circuit Playground Express pad A1 to trigger an alarm. The Circuit Playground Express slide switch will silence and reset the alarm:

```
    from adafruit_circuitplayground.express import cpx

cpx.adjust_touch_threshold(100)  # pretty sensitive

while True:
    if cpx.switch and cpx.touch_A1:
        cpx.start_tone(2000)
    if not cpx.switch:
        cpx.stop_tone()
```

Wave Files

The strength of CircuitPython lies with its ability to play sound files in the wave or WAV format. Not 100% of wave files on the Internet are compatible though. The parameters for CircuitPython playable files are:

- A sample rate less than or equal to 22,050 hertz
- 16-bit
- mono / single channel

Adafruit has a guide to help you with your audio files here (https://adafru.it/s8f).

Using wave files for your project is wonderful! You have nearly unlimited sounds to sample and play on whatever command you set.

Here is a simple program that will play one wave file when button A is pressed and another if button B is pressed.

Here's the two files we're going to play:

<div align="center">

rimshot.wav

https://adafru.it/zFK

laugh.wav

https://adafru.it/zFL

</div>

Click to download, and save them onto your **CIRCUITPY** drive, alongside your **code.py** program file.
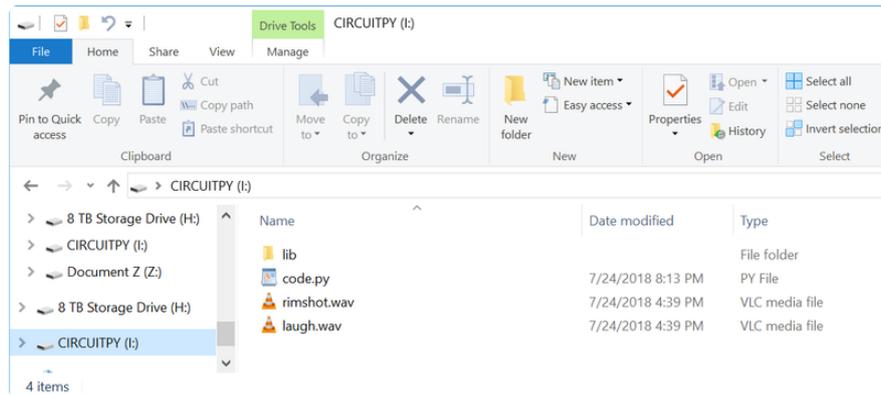

```python
from adafruit_circuitplayground.express import cpx

while True:
    if cpx.button_a:
        cpx.play_file("rimshot.wav")
    if cpx.button_b:
        cpx.play_file("laugh.wav")
```

It is that easy!

If you want to try out some projects which use wave files, here are some good ones in the Adafruit Learning System:

- FruitBox Sequencer: Musically Delicious Step Pattern Generator (https://adafru.it/BW1)
- Sparky the Blue Smoke Monster Automaton (https://adafru.it/BNL)