Hello!
I can talk to you.

https://learn.adafruit.com/make-it-talk

Last updated on 2023-08-29 03:55:47 PM EDT
Table of Contents

Overview 3
• Parts

Generating Voice Sound Clips 5
• First Try - Do Recordings Exist for Your Dialog?
• Custom Voice Files
• Method 1: Record Your Own Voice
• Method 2: Using Machine Generated Voice (text-to-speech)

Example: Making WAV Files with Text to Speech 7
• Obtaining MP3 Files from the BlueMix Text to Speech Engine
• Converting BlueMix MP3 Files to WAV Files
• If Space is Limited...

Converting Sound Files for Use 11
• Converting Your Clips

CircuitPython 12
• Simple Sound Triggering
• Expanding the Number of Words You Can Trigger

Numbers 15
• Trigger on Different Conditions
• Saying Numbers
• Going Further With Numbers

Lots of Buttons: NeoTrellis 18
• Example Tutorials

Going Further 19
• Text to Speech in Python
• Featuring Your Projects
Overview

For many years, microcontrollers were limited to blink an LED or make beeping noises to get your attention.

Today our projects can do more than the basics. Modern electronics can play songs and make complex sounds including human speech. The capability to speak is a game changer in building modern projects.

This guide will help you quickly generate the sound files to have your microcontroller the center of your voice capable project.

Parts

For this tutorial, we'll mainly be using the Express line of CircuitPython programmable boards from Adafruit. The ability to use CircuitPython provides many advances in implementing complex programs with just a few lines of code.

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

Adafruit METRO M0 Express - designed for CircuitPython
Metro is our series of microcontroller boards for use with the Arduino IDE. This new Metro M0 Express board looks a whole lot like our
https://www.adafruit.com/product/3505
Adafruit Metro M4 feat. Microchip ATSAMD51
Are you ready? Really ready? Cause here comes the fastest, most powerful Metro ever. The Adafruit Metro M4 featuring the Microchip ATSAMD51. This...
https://www.adafruit.com/product/3382

Adafruit Feather M4 Express - Featuring ATSAMD51
It's what you've been waiting for, the Feather M4 Express featuring ATSAMD51. This Feather is fast like a swift, smart like an owl, strong like a ox-bird (it's half ox,...
https://www.adafruit.com/product/3857

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 Test Lead (set of 12)
Connect this to that without soldering using these handy mini alligator clip test leads. 15" cables with alligator clip on each end, color coded. You get 12 pieces in 6 colors....
https://www.adafruit.com/product/1008
Generating Voice Sound Clips

Every project is different and every source for material is different, so to obtain voice clips and for them to be in the correct format will likely take some work. It isn't very hard, but it may take a bit of time depending on the source material and the number of word files you want to play.

First Try - Do Recordings Exist for Your Dialog?

Before generating custom dialog, does dialog exist somewhere already recorded? Probably not for specific texts but for simple words and phrases, yes there are websites that offer pre-recorded sounds. The sites may not always be on the Internet and it is likely the files need conversion to microcontroller-friendly format but it will save you time.

Here are sources for pre-recorded dialog that we have found and the licensing terms:

- **Aspect** () - Pre-Recorded Audio: A wide variety of spoken words and phrases in American male english. Numbers, alphabet, much more. [LGPL license](https://freesound.org/).
- **https://freesound.org/** () has a large number of license free sounds.
- An Internet search may find additional resources for pre-recorded sounds.
- If you are looking to read numbers, this tutorial has pre-recorded numbers for you. [See this page for more details](https://freesound.org/).

Custom Voice Files

Method 1: Record Your Own Voice

A number of programs allow you to record your voice on your PC/Mac/Linux machine.
For Mac, it is suggested you use QuickTime Player: Select File then New Audio Recording

For Windows 10, search for the Voice Recorder app.

For all platforms: you may download and record audio with the free Audacity program.

You will want a "script" your phrases: write down the words you will want to play in your project. Then you or your voice person will speak each of the phrases one at a time with some space between words (or record each word as its own sound file which may save some edit time).

You can also record sound effects, holiday sounds, etc. this way.

Method 2: Using Machine Generated Voice (text-to-speech)

While computers can generate voices, it can get convoluted, so we'll use the power of the web to use some websites to generate text to speech. Different services support different speakers (male/female) and languages.

Note that Adafruit does not endorse any 3rd party site, sometimes they restrict use or the site disappears, in such a case, go to another or use a search engine to find "Free text to speech mp3".
IBM BlueMix Text to Speech

The IBM speech engine is accessible at this web site. You can type in words, phrases, sentences and more and save the result as an MP3 file. Different speakers and different languages are supported.

See the example of how to use this on the next page.

NaturalReaders

This web site says it allows for 20 minutes of free speech conversion to MP3.

fromtexttospeech.com

Another Text to Speech engine available here, works well, check out male voice John.

Text 2 speech .org

A basic site that has US and Indian voices and you can download in WAV and MP3.

Amazon Polly

This service charges a very small fee per character but you can use the recorded audio anyway you want.

Example: Making WAV Files with Text to Speech

Here is a step by step method for making your own words and sentences via the IBM text to speech web site.

Obtaining MP3 Files from the BlueMix Text to Speech Engine

The IBM speech capability is called BlueMix. You can navigate to the site via a modern web browser to https://text-to-speech-demo.ng.bluemix.net/.
Select the voice you would like to use.

For examples later in this tutorial, I chose Allison which is a female American English voice. While there are thirteen voices in ten variants, you do not have the choice of male and female in all languages. If you would like a different voice than those listed, you should look at other text to voice services.

Type the word/phrase/sentence/text you would like converted to speech in the input box.

You can press > to play the clip in the browser or the Speak button to do the same.

When you are happy with the results, click Download to save the file on your computer. The file will be in MP3 format which is fine.

If the application is not reading your text the way you want, consider alternative spellings of words, even if the spelling looks silly since we're only looking for a good sounding clip.

You can use SSML to change the American English voices. Speech Synthesis Markup Language adds tags to tell a speech engine how you want text read. IBM has documentation here and tags here if you need this capability.

Make all the MP3 clips for your entire project and put them in locations which are easily found for the next step.
Converting BlueMix MP3 Files to WAV Files

The conversion is easily done using the open source program Audacity which is available for all operating systems. If you do not already have the program installed, you can download it from the official site and install it.

Open Audacity and you'll see the main page.

Use File -> Open and navigate to the voice MP3 file for your first conversion. Here I have opened a file 0.mp3 which contains a recording saying "zero".

You'll notice the BlueMix site generates monotone (one channel) audio at 22,050 Hz 16-bit PWM which is (usually) exactly what we want without fiddling with any changes.

Use File -> Export -> Export as WAV to convert the file to a WAV file.

Audacity opens a Save File dialog. Save the WAV file where you'll remember it on your hard drive.

Repeat the same process for all your MP3 files and you'll have a collection of WAV files suitable for use on microcontroller projects.
You will want to write your code such that your WAV files are easy to use. Adafruit tutorials put limited numbers of WAV files in the main (root) directory on the project CIRCUITPY drive. For more than a few files, it is suggested that you make a subdirectory to put the files into. For the number voice WAV files used in this guide, a subdirectory called /numbers was created to hold the files.

If Space is Limited...

All the files above were created by BlueMix to be 22,050 Hz. If you have WAV files that take up a great amount of space, your space on the board CIRCUITPY flash drive might fill up.

If you need additional space, you might consider creating files at a lower sampling rate (lower than 22,050 Hz). Perhaps use 11,025 Hz or maybe down to 8000 Hz. The fidelity of the voice will suffer as you reduce the sample rate. Before Exporting the file to WAV, you will want to Resample the file. In Audacity, use Tracks -> Resample. Type in the new rate. Then you can Export as described above.
This will result in smaller files with a reduction in sound quality. It is not suggested to
go lower than 8,000 Hz as that will reproduce voice to 4,000 Hz which will not
reproduce the high notes in voice.

Converting Sound Files for Use

The sound files you have recorded probably need some work. Our microcontroller
really works best if we provide it sound files in a standard format that it understands
easily, so that it can do other things at the same time in projects.

For our target language, CircuitPython, the format we can use is the WAV (wave) file
format. WAV files can come in a variety of types (way too many) so we will be specific
to what we want:

- 16-Bit Pulse Control Modulation (PCM)
- Sampling speed no greater than 22,050 Hz

If you have a microcontroller that can process stereo (left/right) sound like the NeoTrel
lis M4 (), you can make your files stereo. For other microcontrollers, Adafruit specifies
using mono (single channel) only.
Converting Your Clips

Adafruit has a separate guide on using software to convert sound files to the correct WAV format:

- [Microcontroller Compatible Audio File Conversion](#)

Programs like Audacity, used in that guide, can trim files to eliminate all but the desired recording. They can also trim one big file like a script into individual recordings.

Convert your files, ensuring the exact parameters noted above and in the conversion guide. Save the files on your computer with a .wav extension on the filename, such as 1.wav, hello.wav, spooky.wav, one.wav, two.wav, etc.

Next we'll code some CircuitPython to use those sound files.

CircuitPython

Currently CircuitPython is best for opening and using sound files. MakeCode does not handle files and Arduino is a steep mountain full of deep chasms.
Are you new to using CircuitPython? No worries, there is a full getting started guide here.

Adafruit suggests using the Mu editor to edit your code and have an interactive REPL in CircuitPython. You can learn about Mu and its installation in this tutorial.

Simple Sound Triggering

For this example we will use a Circuit Playground Express board as it has two buttons onboard.

The code waits for either Button A or Button B to be pressed and plays one of two wav files, 01.wav or 02.wav.

The sound will come out on the speaker on the Circuit Playground Express board. If you'd like to hook up headphones or powered speakers, make the following connections:

```python
from adafruit_circuitplayground.express import cpx

while True:
    if cpx.button_a:
        cpx.play_file("01.wav")
    if cpx.button_b:
        cpx.play_file("02.wav")
```

Copy this file to the CIRCUITPY drive as code.py.

Download these two wav files below and place them on the CIRCUITPY drive in the main (root) directory.
Here is what it sounds like with two demonstration voice files:

For other Express boards besides the Circuit Playground Express, here is a more generalized code script. This is shown using the Metro M0 Express although the code still runs on the Circuit Playground Express by uncommenting two lines for pin definitions. Use the save WAV files above for testing.

The analog output A0 goes to the jack of **headphones** or **amplified speakers**.

```python
# Generalized CircuitPython audio play file
import time
import board
import audiocore
from digitalio import DigitalInOut, Direction, Pull

# D1 = board.BUTTON_A  # uncomment these if using
# D2 = board.BUTTON_B  # a Circuit Playground Express

# Button A
button_a = DigitalInOut(D1)
button_a.direction = Direction.INPUT
button_a.pull = Pull.DOWN

# Button B
button_b = DigitalInOut(D2)
button_b.direction = Direction.INPUT
button_b.pull = Pull.DOWN

# Audio playback object and helper to play a full file
aout = audioio.AudioOut(board.A0)
```
# Play a wave file

def play_file(wavfile):
    print("Playing", wavfile)
    with open(wavfile, "rb") as f:
        wav = audiocore.WaveFile(f)
        aout.play(wav)
        while aout.playing:
            pass

while True:
    if button_a.value:
        play_file("01.wav")
    if button_b.value:
        play_file("02.wav")
    time.sleep(0.01)

Expanding the Number of Words You Can Trigger

At this point, you can consider triggering sounds a number of different ways. The Metro M0 Express has 13 digital pins and 6 analog pins which could be used as wired above.

Capacitive touch is a very popular method of getting multiple "button" triggers and making sounds. On the Circuit Playground Express, capacitive touch is available on all data pads except A0.

The Adafruit tutorial [FruitBox Sequencer: Musically Delicious Step Pattern Generator ()](https://www.adafruit.com/circuitplayground) uses capacitive touch on the Circuit Playground Express to trigger different musical sounds. You can use the same exact CircuitPython code but replace the musical wav files with your word recordings. The words could be "Peach", "Apple", "Orange", "Mango", "Lemon", "Lime", and "Tangerine". Tapping a finger on the fruit triggers the corresponding wav file containing what that fruit is named. Great demonstration for what an inexpensive microcontroller can do!

---

**Numbers**

**Trigger on Different Conditions**

You can use the multitude of sensors Adafruit carries to monitor things and trigger voice prompts based on the sensor results.

The program below expands on the code used in the Adafruit tutorial [Make It Sense ()](https://www.adafruit.com/makeit). In that guide, the various sensors on a Circuit Playground Express trigger various actions. Those actions could trigger the board to speak various words or
phrases also! We'll again use CircuitPython as MakeCode does not have wav file support.

You can also see examples of Circuit Playground Express and CircuitPython code in the Adafruit tutorial CircuitPython Made Easy on Circuit Playground Express.

For how to use external switches with microcontrollers, see the Adafruit guide Make It Switch.

An Example: Temperature

Take the sample CircuitPython code for temperature in Make it Sense. Rather than plot the temperature, you can use a voice to say something depending on the temperature. For example, > 90 degrees Fahrenheit it says "too hot!" and 32 degrees F or less it says "too cold!".

```python
import time
from adafruit_circuitplayground.express import cpx

while True:
    if cpx.temperature > 37.0:  # Celsius fever reading
        cpx.play_file("fever.wav")
        time.sleep(5)
```

For Circuit Playground Express and sensors available beyond the thermistor, see the Adafruit Circuit Playground Express guide.

Saying Numbers

When it comes to numbers, there are literally an infinite number of them. And recording an infinite combination of files would be very tough. All the commercial voice programs break a number into the numeric places and voice those. So for the number 25, you would say the word "Twenty" followed by the number "Five" instead of recording "Twenty Five".

The following program expands on the example above, saying the temperature from -299 to 299 degrees Celsius or Fahrenheit. The numbers have been recorded from the IBM BlueMix site mentioned earlier. Click Download Project Zip to get the file code.py and a subdirectory containing the WAV files.

Copy code.py to the CIRICUITPY drive on a Circuit Playground Express.
Besides the numbers, there are some additional files for projects for humidity, clocks, and the words thousand and million that are not needed for this project. The other files should be copied to the CIRCUITYP drive in the subdirectory /numbers.

```python
# SPDX-FileCopyrightText: 2018 Anne Barela for Adafruit Industries
# SPDX-License-Identifier: MIT

# CircuitPython Speaking Thermometer Example
# Coded for Circuit Playground Express but it may be
# modified for any CircuitPython board with changes to
# button, thermister and audio board definitions.
# Anne Barela for Adafruit Industries, MIT License

import time
import board
import adafruit_thermistor
import audioio
import audiocore
from digitalio import DigitalInOut, Direction, Pull

# Enables the speaker for audio output
spkrenable = DigitalInOut(board.SPEAKER_ENABLE)
spkrenable.direction = Direction.OUTPUT
spkrenable.value = True

D1 = board.BUTTON_A
D2 = board.BUTTON_B

# Button A setup (Celsius)
button_a = DigitalInOut(D1)
button_a.direction = Direction.INPUT
button_a.pull = Pull.DOWN

# Button B setup (Fahrenheit)
button_b = DigitalInOut(D2)
button_b.direction = Direction.INPUT
button_b.pull = Pull.DOWN

# Set up reading the Circuit Playground Express thermistor
thermistor = adafruit_thermistor.Thermistor(
    board.TEMPERATURE, 10000, 10000, 25, 3950)

# Audio playback object and helper to play a full file
aout = audioio.AudioOut(board.A0)

# Play a wave file
def play_file(wavfile):
    wavfile = "/numbers/" + wavfile
    print("Playing", wavfile)
    with open(wavfile, "rb") as f:
        wav = audiocore.WaveFile(f)
        aout.play(wav)
        while aout.playing:
            pass
```
# Function should take an integer -299 to 299 and say it
# Assumes wav files are available for the numbers
def read_temp(temp):
    play_file("The temperature is.wav")
    if temp < 0:
        play_file("negative.wav")
        temp = - temp
    if temp >= 200:
        play_file("200.wav")
        temp = temp - 200
    elif temp >= 100:
        play_file("100.wav")
        temp = temp - 100
    if (temp >= 0 and temp < 20) or temp % 10 == 0:
        play_file(str(temp) + ".wav")
    else:
        play_file(str(temp // 10) + "0.wav")
        temp = temp - (temp // 10) * 10
        play_file(str(temp) + ".wav")
    play_file("degrees.wav")

while True:
    if button_a.value:
        read_temp(int(thermistor.temperature))
        play_file("celsius.wav")
    if button_b.value:
        read_temp(int(thermistor.temperature * 9 / 5 + 32))
        play_file("fahrenheit.wav")
    time.sleep(0.01)

Going Further With Numbers

See the pre-made WAV files files located in the GitHub repository and see if you'd like to make a clock, a humidity display, or reading another type of number. Feel free to make your own phrases to go along with the ones we have provided.

Lots of Buttons: NeoTrellis

If you want a lot of buttons, you might want to check out the Adafruit NeoTrellis M4 Express. 32 buttons of multicolored glory and 8 megabytes of file storage provides for
lots of capability. The flash size means each button cannot read a whole book but words and phrases certainly will fit with room to spare.

If you do not have 32 sounds, load up the number you have.

Now you can use NeoTrellis as a sound box. Say someone is having voice issues. Load up the words you want, print out a guide to which button does what, and you have a communications device!

Example Tutorials

- **ABC Soundboards with NeoTrellis** - uses the techniques in this guide to make two English alphabet soundboards - one for an out of copyright animal book and one for the Adafruit "E is for Electronics" alphabet color book. See a video demonstration below.
- **Star Trek Soundboard with NeoTrellis** - Play sound effects from the Original Series

Going Further

Text to Speech in Python

While CircuitPython does not have native or library text to speech (as of yet), there are services for Python 3 if you are running on a single board computer (SBC) like Raspberry Pi.

Google Text to Speech

gTTS, *Google Text-To-Speech* is a Python library from Pierre-Nick Durette that converts a string into an audio file. It does not automatically speak, rather it creates an mp3 file that we can then play.

[See this post by Les](https://learn.adafruit.com) on how to set up gTTS.

Amazon Polly

Polly is Amazon's text to speech engine. You will need to sign up for a AWS account to use.
Featuring Your Projects

If you build voice-enabled projects, post video or links to them on Twitter tagging @Adafruit, we’d love to see what you are making.