MP3 Player

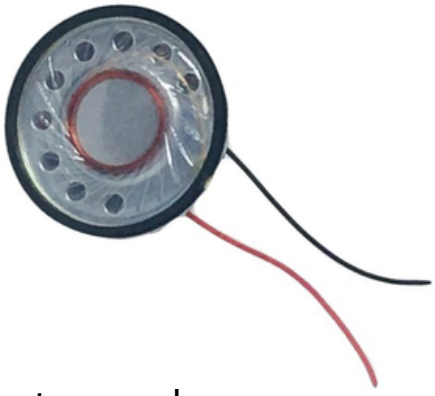
*Definitions of words in **bold** are on the last page*

CircuitPython

- CircuitPython is an open source fork of MicroPython.
- The audio libraries we need to use are only available with CircuitPython.
- We can freely install either interpreter onto our Pi Picos.

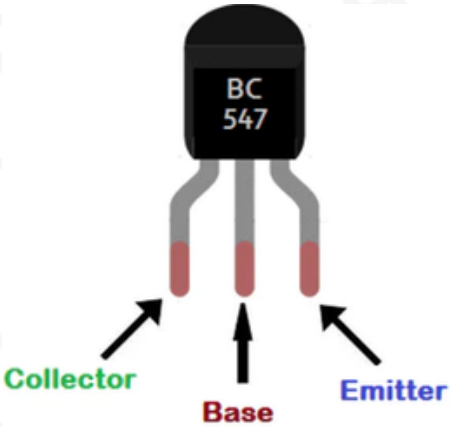
Speaker

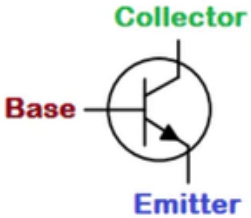
- Uses an electro magnet to vibrate a diaphragm to make sound waves.
- Takes an analog input, we can imitate this with a PWM signal



Transistor

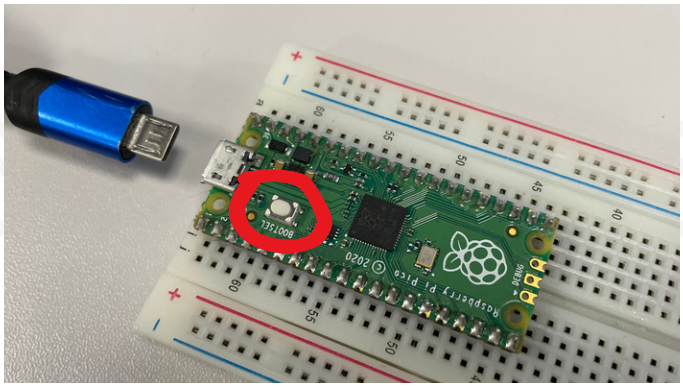
- These take two inputs:
  - A **collector** or “Power”
  - A **base** or “Data” signal
- And have one output, the **Emitter**.
- By connecting an audio signal to the collector and power to the base, we can make a very basic amplifier.



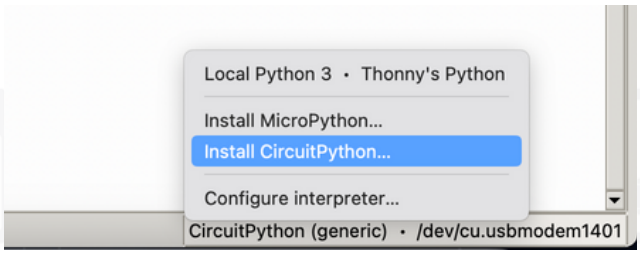


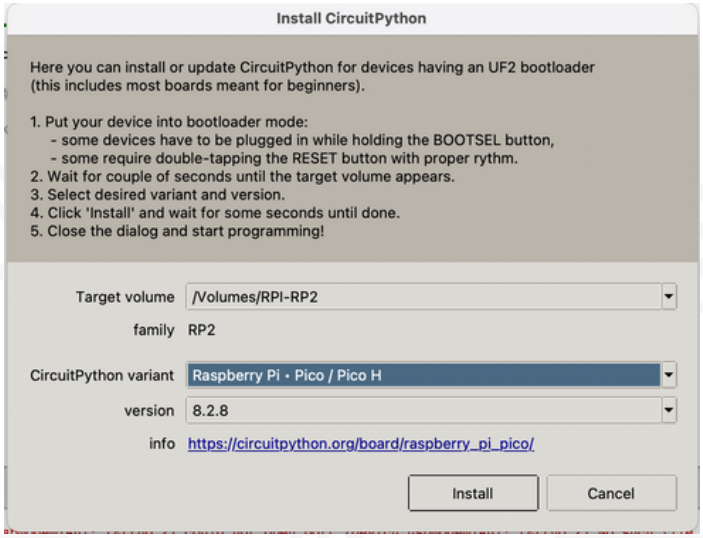
Install CircuitPython on the Pico

- To start we need to put the Pico into boot mode:
  - Unplug the USB cable.
  - Hold down the reset button (circled in red)
  - Plug the USB cable back in.
  - Release the reset button

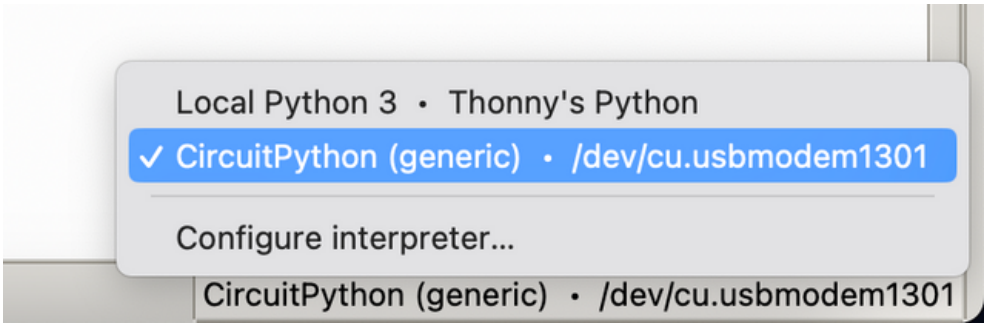


- Click in the lower right hand corner of Thonny, where it tells you what device your code is running on.
- Select **Install CircuitPython...**
- A new window should pop up
- For the CircuitPython variant select **Raspberry * Pi Pico / Pico H**
- Press install





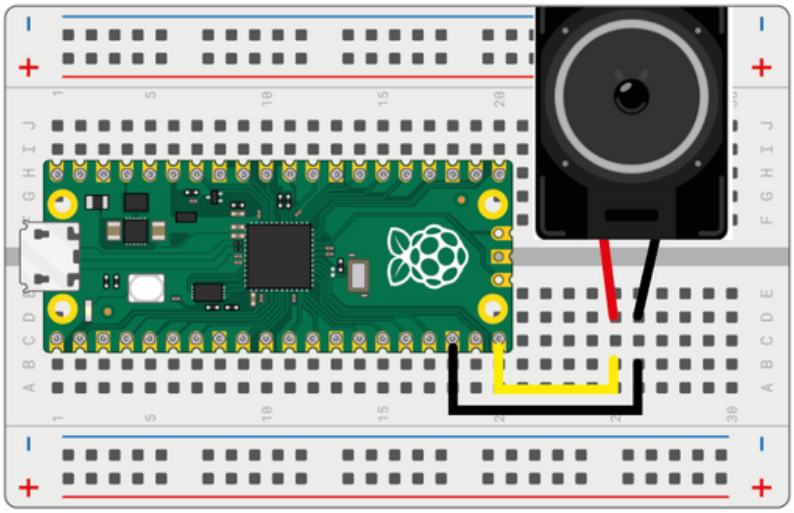
- If all goes well, after copying it should say **Done!**
- Close the install window
- Select your CircuitPython device in the lower right corner

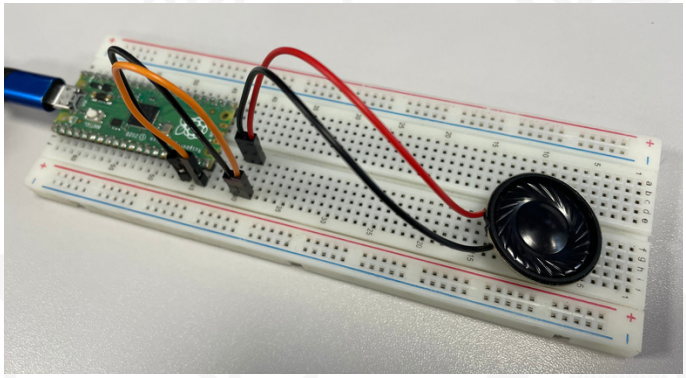


Wire up the speaker

- Connect one pin of the speaker to **Pin 15**
- Connect the other pin to **GND**

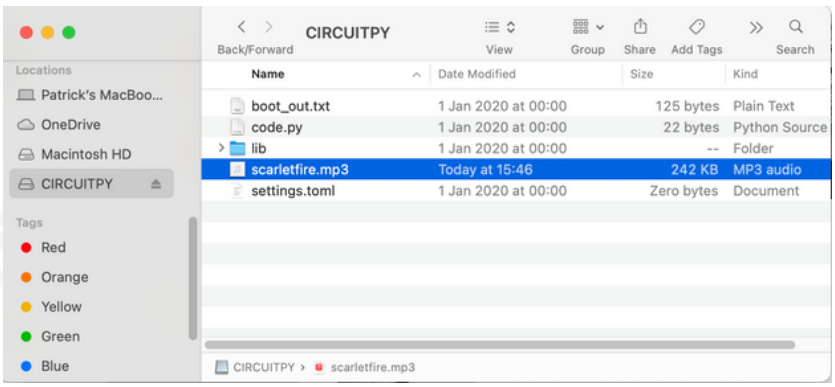
Note: It should not matter which way around the pins to the speaker are.





Load some music on the Pico

- You will need to download **scarletfire.MP3**. This can be found in the discord or on a USB stick.
- Other MP3 files may work, but they need to be small. Picos only have 2MB of storage and 256KB of RAM.
- Your Pico should show up as a USB device, copy **scarletfire.MP3** onto it.



Audio Code

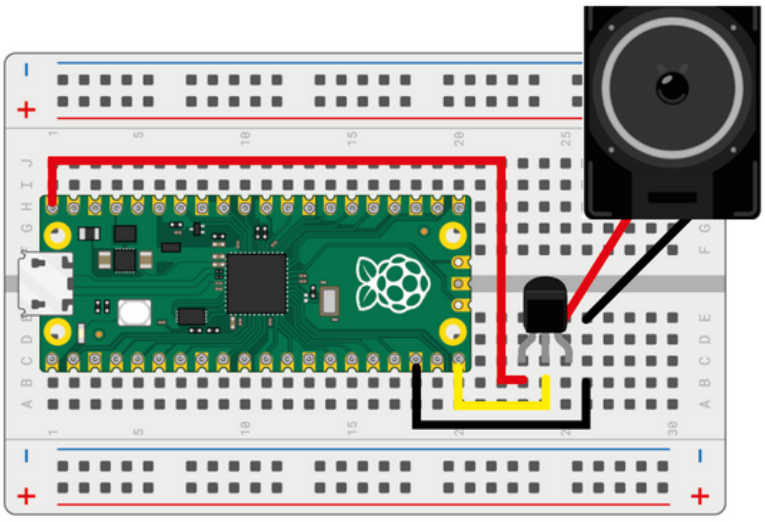
This code should be all you need to play the MP3 file.
You might have to listen carefully as it will be very quiet.
To make it louder we will have to build an amplifier...

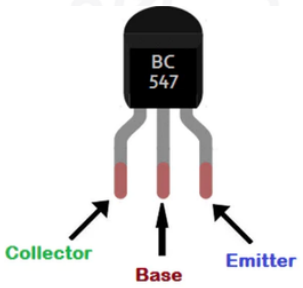
```
import board
import audiomp3
import audiopwmio

audio = audiopwmio.PWMAudioOut(board.GP15) # Set the pin for audio to play from

mp3file = open("scarletfire.mp3", "rb") # Open the mp3 track we want

decodedMp3 = audiomp3.MP3Decoder(mp3file)

audio.play(decodedMp3) # Start playing the track

while audio.playing: # Keeps the program running while audio is playing
    pass

print("Done playing!")
```

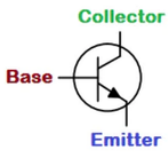
Amplifier

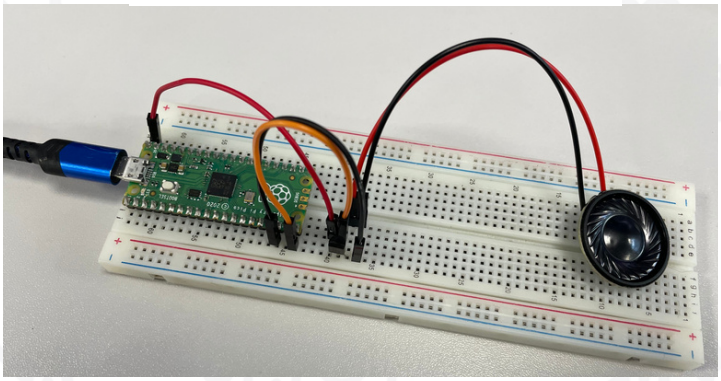
We can make a very basic amplifier out of a single **transistor**.

- **VBUS** will be connected to the as **Collector.** VBUS is the 5v power directly from the USB connection
- Pin 15, the audio output will be connected to the **Base**.
- The speaker will connect to the **Emitter** on one pin and **GND** on the other









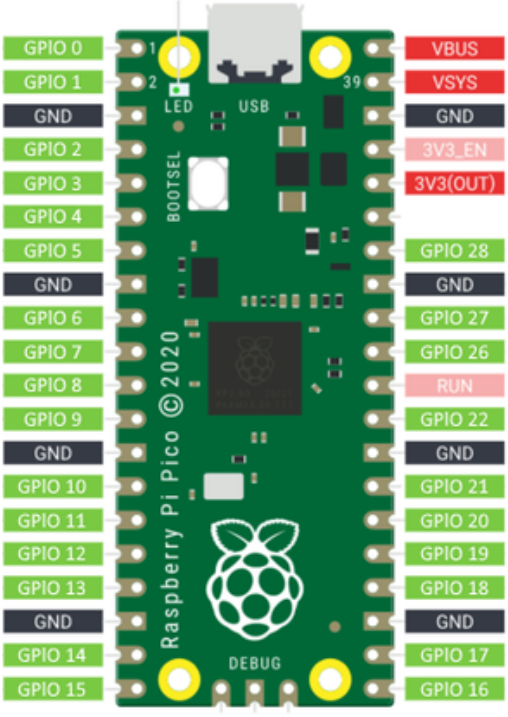
The music should be slightly louder now. This is still quite a weak amplifier so you shouldn't expect to keep the neighbours up

Review and extra challenges:

Today we covered the basics of how audio devices work. You learnt a bit more about the setup of the Pi Pico and how we have more options when it comes to software.

Challenge: If you want to add volume control, try adding a potentiometer.

Thinking point: You may have noticed the way we setup the output pin is different in CircuitPython than MicroPython. How might a simple blink program look different too?



Pi Pico Code Docs

Worksheet produced by Patrick Culmer for use by HackSussex
https://github.com/supersand21/Robotics-Workshop