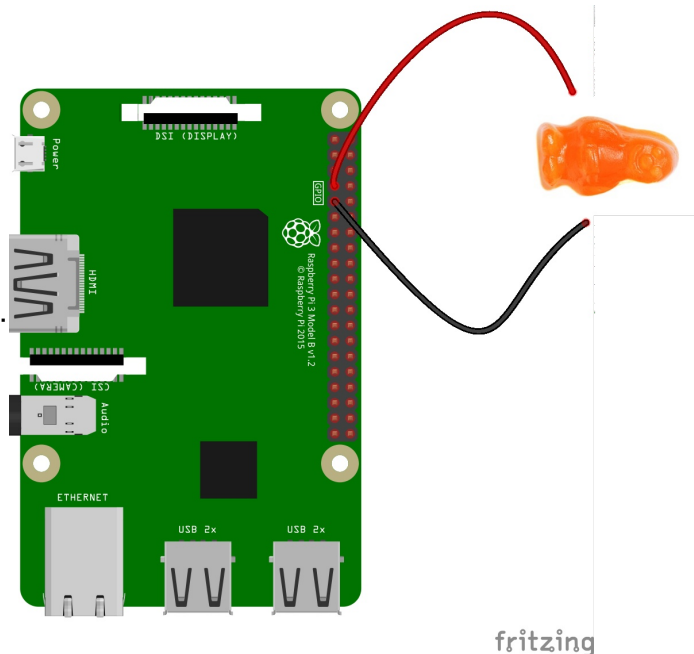


# SCREAMING JELLY BABIES

This workshop will guide you through creating a "Screaming Jelly Baby" using your Raspberry Pi computer, some simple electronics and a Jelly Baby. You will need either headphones or speakers for this one!

## THE ELECTRONICS

You should have two jumper wires (both female-to-female, holes in both ends) and two paperclips. Straighten out the paperclips. Push a paperclip into one end of each jumper wire. Connect the other end of one jumper wire to GPIO 4. Next, connect the other end of the other jumper wire to ground (negative, GND). As shown in this diagram. Now poke the paperclips into the jelly baby diagonally, so they cross over in an X pattern. You want them to overlap very closely, but not quite touch.



## THE PROGRAM

Power up your Raspberry Pi. From the menu, select "Programming - Python 3 (IDLE)". Then use File, New Window to create a new program. Type in the following program:

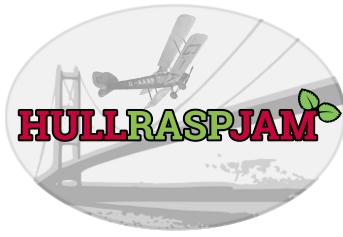
```
from time import sleep
from os import system
from gpiozero import Button

button = Button(4)
print ("Don't hurt me!")
while True:
    button.wait_for_press()
    print ("Ouch!")
    system("omxplayer scream-c.wav")
```

Use File, Save to save this program as scream.py in the python/jellybaby folder (it must be saved in the same folder as your .wav files) and then run it with Run menu, Run Module. The .wav files have already been saved to your SD Cards for the Jam but if you want to repeat this at home all the files are available to download from the HullRaspJam GitHub area:

[https://github.com/HullRaspJam/HullRaspJam\\_Worksheets/tree/master/Screaming\\_Jelly\\_Babies](https://github.com/HullRaspJam/HullRaspJam_Worksheets/tree/master/Screaming_Jelly_Babies)

Now squish your jelly baby! As the two paperclips touch, you should hear a little scream. To stop the program, select Shell menu, Restart Shell.



# SCREAMING JELLY BABIES

## NOT WORKING?

Check you've got:

1. The correct GPIO pins
2. Four spaces at the start of lines which need spaces
3. No spelling mistakes, no missing brackets, and you've used round

brackets instead of pointy, curly or square ones.

## WHAT IS THIS PROGRAM DOING?

```
from time import sleep
from os import system
from gpiozero import Button
```

The first three lines tell the computer to learn about new things. Computers can learn from programs that other people have written; we call these other programs "libraries". Our instructions tell the computer to learn how to run other programs (such as the sound player) from the system library, and how electrical buttons work (two wires touching) from the GPIO Zero library. We're not actually using the sleep command yet, but you might later.

```
button = Button(4)
```

We then create an Button object. We say that the button is connected to GPIO 4.

```
print ("Don't hurt me!")
while True:
```

We print a message to the screen, then start a loop that repeats forever.

```
    button.wait_for_press()
    print ("Ouch!")
    system("omxplayer scream-c.wav")
```

Next we wait until the wires touch, print a message and then play the sound file.

There are other sound files. Can you get it to play Doh-Ray-Me with little screams? What about Mary Had A Little Lamb? Use sleep(1) to wait for a second between sounds.