

Making Technological Fireworks

With BBC micro:bit

Equipment needed

- 1 x micro:bit
- 1 x battery pack
- 2 x AAA batteries

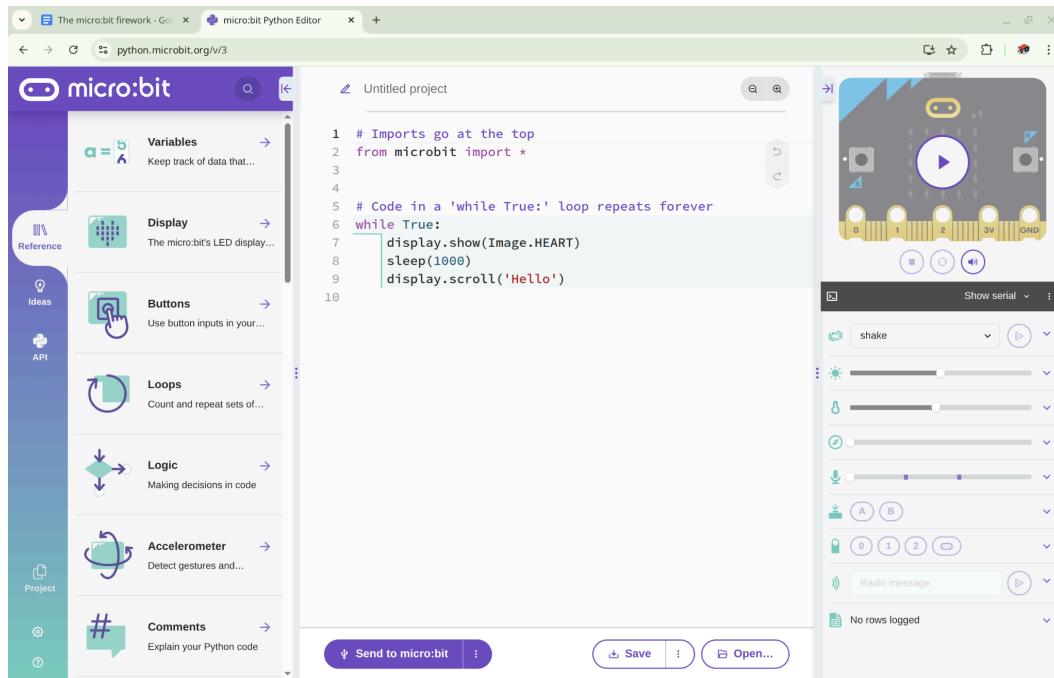
Brief

Seeing the fireworks on bonfire night or New Year are always spectacular. Hearing the rush of the rocket as it launches into the sky followed by the big bang or a loud fizz as it lights up the night sky. You are going to have a go at simulating a firework using the micro:bit and a good old fashioned throw.



Challenges

1. Go to the [micro:bit Python editor](#) (You can use the block-based MakeCode editor if you prefer)



2. When the screen (above) loads, click on the words at the top that say “Untitled project” and change the name of the project to “Firework”.
Below the `from microbit import *` line add this line: `import speech`
3. Above the `while True:` loop. Add the following code:

```
display.show(Image.HAPPY)
sound_triggered = False
```

The above code creates a happy face on the micro:bit display. We also create a variable called `sound_triggered` which we assign the value `False`. This will help us not have the sound repeat.

4. We will now change the code in the `while True:` loop. Delete the code from the loop and create a variable that will keep track of the z value of the micro:bit (The effect of gravity)

```
7 # Code in a 'while True:' loop repeats forever
8 while True:
9     z = accelerometer.get_z()
```

5. We now need to check if the `z` variable is over a certain value. You can do this by adding the following code (Make sure you have the correct indents)

```
8 # Code in a 'while True:' loop repeats forever
9 while True:
10     z = accelerometer.get_z()
11     if z > 500 and not sound_triggered:
12         sound_triggered = True
13         audio.play(Sound.SAD)
```

The if rule checks if the `z` value is over 500 and if the `sound_triggered` is false. If those conditions are met then `sound_triggered` turns to True and a sad sound is played (closest sound to whizz I could find)

6. Download your code onto your micro:bit once it has loaded (It will show a happy face), disconnect it from your computer and attach the battery pack. Gently throw your micro:bit in the air and it should make a whizzing sound. If it doesn't make the sound (**It will only place once**) connect it back to your computer and change the 500 to a smaller value.

7. Now to add the bang. Now I could not find a bang so I used the text-to-speech function. Add the following 3 lines within your if block

```
8 # Code in a 'while True:' loop repeats forever
9 while True:
10     z = accelerometer.get_z()
11     if z > 500 and not sound_triggered:
12         sound_triggered = True
13         audio.play(Sound.SAD)
14         speech.say("bang!")
15         sleep(1000)
16         sound_triggered = False
```

8. Load your program to the micro:bit. Once it has loaded (It will have a happy face on it) disconnect it from your computer and attach the battery pack. Try throwing your micro:bit (gently) and you should hear the whizz and the word bang!

Super Challenges

Have a go at extending your program by doing the following:

1. Add displays so that the bang can show a star or exploding animation
2. Replace the text-to-speech bang with different sounds, music, or make your own sound (Use the reference section to help you)
3. Try adding the data logger to your code so that you can track how the z value changes as the micro:bit works. You can check the values by connecting your micro:bit back to the computer, opening the file exploring, going to micro:bit, and opening the file MY_DATA