# micro:bit campfire music

## Overview

This activity meets **requirement three** of **stage two** of the **Digital Maker Staged Activity Badge** (‘Make a simple digital creation that uses code to interact with the wider world through inputs (such as buttons or typing on a keyboard) and outputs (such as a computer screen, sound, or lights)’). It shows people how to turn a micro:bit into a campfire music player that plays ‘Campfire’s Burning’.

45 minutes

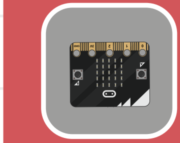
Flexible (ideally two young people at each computer)

If you’re running this activity without internet access, you’ll need to download the software and starter project ahead of time. You’ll probably also need to print handouts.

Wherever you have access to computers

## You will need:

- Computers – tablets or other mobile devices may work as well.
- Activity handouts (at least one for each group)
- micro:bit MakeCode editor
- For each group:
  - One micro:bit
  - One USB A to micro USB B cable
  - Two cables with crocodile clips (from websites such as Amazon, eBay, or CPC)
  - One pair of headphones or a portable speaker



### If your meeting place has internet

Run the micro:bit MakeCode editor in a web browser: **rpf.io/makecode**.

### If your meeting place doesn’t have internet

Download the micro:bit editor software before you begin – check out the micro:bit guide at **rpf.io/scouts-microbit** for instructions.

## Key messages

- Sound (or music) is just one output of a computing device (other examples are light, movement, and storing data). Music’s a sequence of notes played in order, just like computer code is a sequence of instructions performed in order.
- Often, sequences of notes repeat in a tune. A computer program can loop, so we don’t have to write out the whole tune in code. Instead, we can tell the program to loop the repeated sequences of notes.
- MakeCode is a simple drag-and-drop programming language that you can use to make a computer carry out instructions.

Scouts Raspberry Pi

# micro:bit campfire music

## Leader instructions

1 Chat about how computers can be programmed to make sounds and musical notes.

2 Explain that the challenge is to program a micro:bit to play the tune 'Campfire's Burning'.

3 If people haven't used micro:bits before (or if they need a reminder), introduce the micro:bits. Explain that they have two buttons and a small LED display (for input and output). They can also output an analogue signal that sounds like musical notes when you hear it through headphones or a speaker.

4 Explain that everyone should use cables and crocodile clips to connect the micro:bit to headphones or a speaker.

5 Show everyone how to open the micro:bit MakeCode editor.

6 Everyone should get into groups. Each group will need a computer and at least one copy of the handout.

### Community and sharing

Young people should have the chance to tell everyone else about their code (or how they found or fixed a bug).

### Alternatives

- ○ If you don't have micro:bits, you could just use the micro:bit code editor to complete the coding section of the activity and create the tune.

### Safety

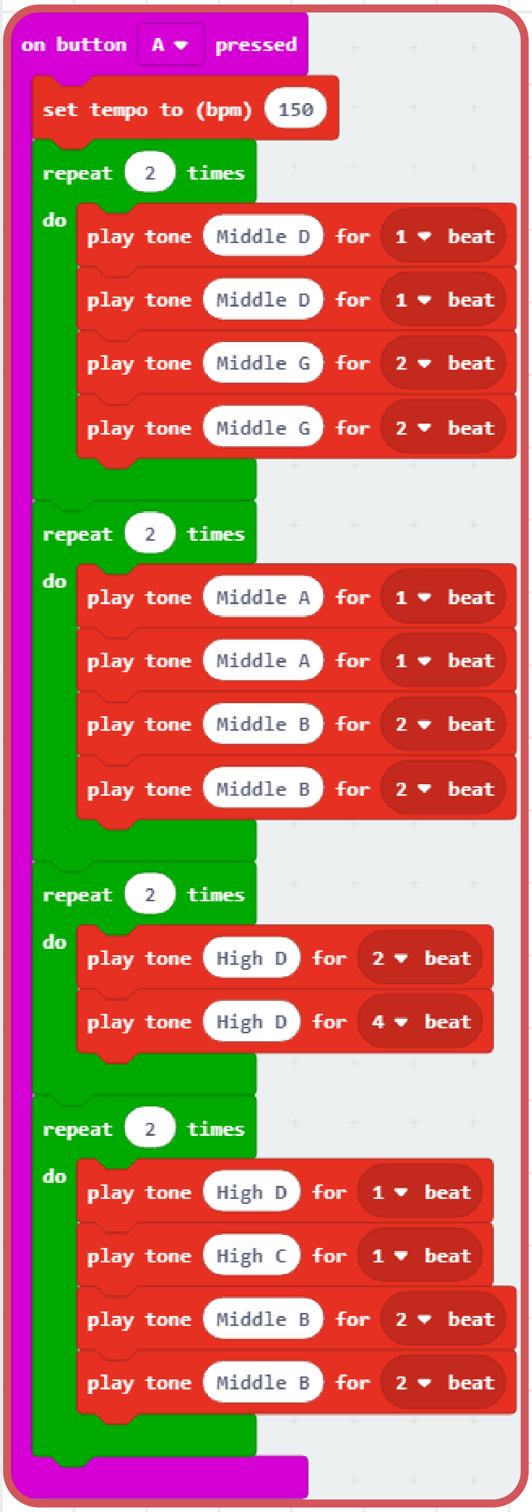
If young people are working online, they should ask for permission before viewing any other websites. It's a good idea to set up parental controls – you can find instructions on the NSPCC website (**rpf.io/scoutsnspcc-online**).You should also give each young person a 'Stay safe' leaflet (**rpf.io/scouts-staysafe**).

### Adaptability

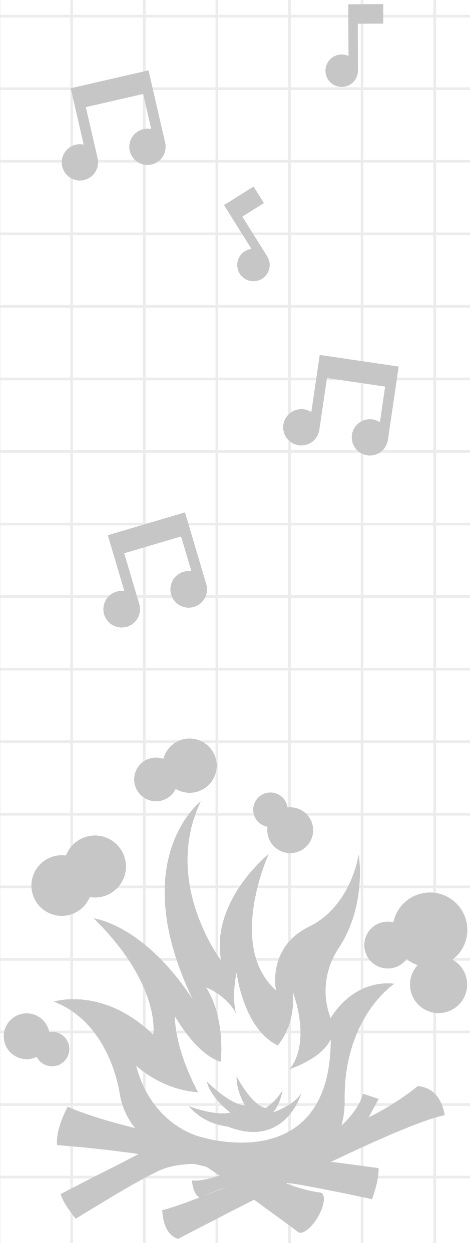
- ○ You can share the complete program if anyone needs a helping hand. Find it at **rpf.io/campfirecode**.
- ○ It can be tricky to connect the crocodile clips to the micro:bit. Help anyone who's struggling.

Scouts Raspberry Pi

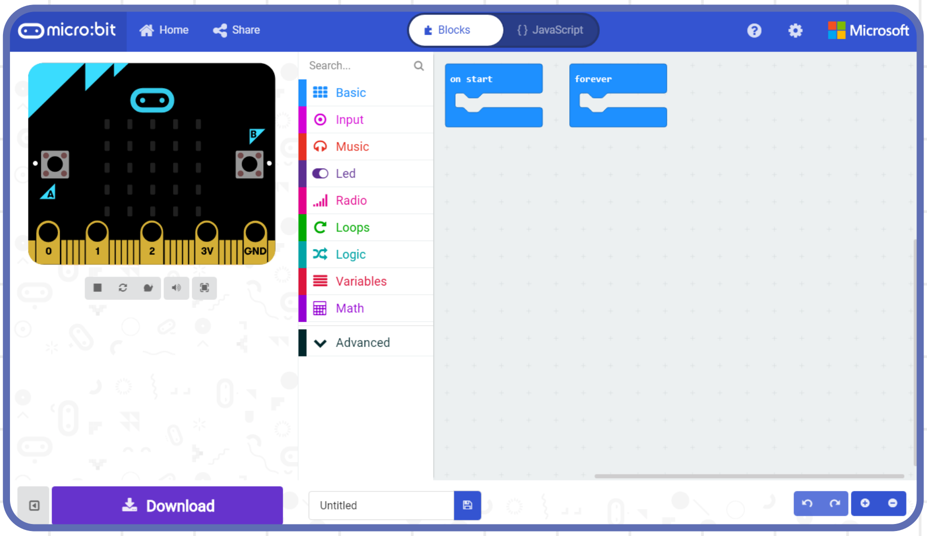
## Complete program



```
on button A pressed
    set tempo to (bpm) 150
    repeat 2 times
    do
        play tone Middle D for 1 beat
        play tone Middle D for 1 beat
        play tone Middle G for 2 beat
        play tone Middle G for 2 beat
    repeat 2 times
    do
        play tone Middle A for 1 beat
        play tone Middle A for 1 beat
        play tone Middle B for 2 beat
        play tone Middle B for 2 beat
    repeat 2 times
    do
        play tone High D for 2 beat
        play tone High D for 4 beat
    repeat 2 times
    do
        play tone High D for 1 beat
        play tone High C for 1 beat
        play tone Middle B for 2 beat
        play tone Middle B for 2 beat
```



1 Open the micro:bit MakeCode editor at **makecode.microbit.org** and click **New Project** to start.

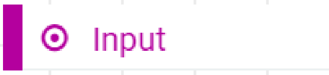


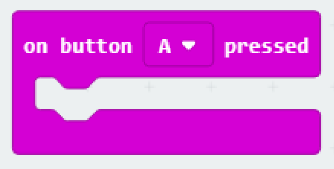
You'll see a virtual micro:bit on the left, which will react when you run your code. The menu in the middle is where you'll find the blocks to build your program in the right panel. First off, make your micro:bit make a sound when button A is pressed.

## Trivia

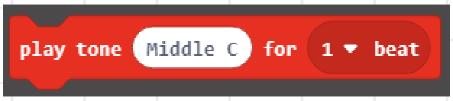
Did you know that the tune of 'Campfire's Burning' is taken from the nursery rhyme 'London's Burning'? The rhyme's said to be about the Great Fire of London in 1666, where the city burned for five whole days.

2 Click on the **Input** section of the menu, then click on **on button A pressed**.

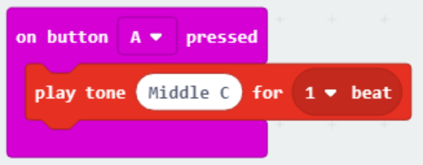




3 In the **Music** section of the menu, click on **play tone Middle C for 1 beat**.

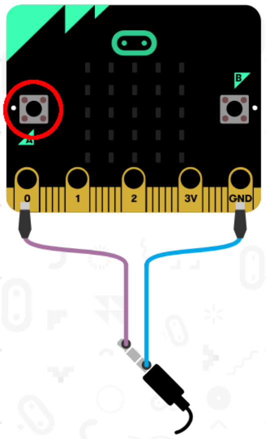


4 In the right panel, drag the `play tone` block so it is attached to the `on button A pressed` block. Now, whenever **button A** is pressed, the tone **middle C** will be played for **1 beat**. The **tone** is the note which gets played, and the **beat** says how long the note is played for.

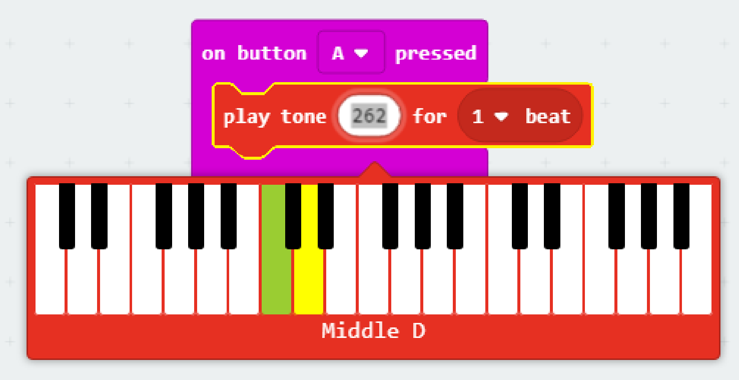


```
on button A ▾ pressed
  play tone Middle C for 1 ▾ beat
```

5 On the left, click **button A** of the virtual micro:bit to run your program and hear the tone being played.



6 Try a different note by clicking on **Middle C** and selecting a new note on the piano that appears.



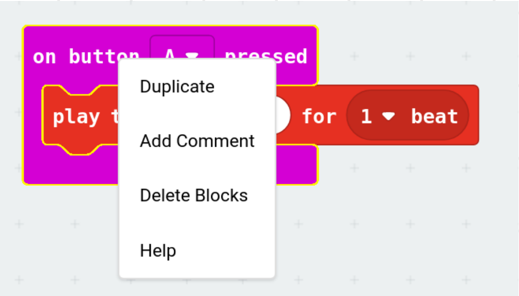
```
on button A ▾ pressed
  play tone 262 for 1 ▾ beat
```

7 Change the number of the beat too, then press **button A** to hear what difference it makes to the sound.

A tune is a sequence of notes played one after each other in order. 'Campfire's Burning' is made up of the following sequence of notes:

D D G G
D D G G
A A B B
A A B B
D D D D
D C B B
D C B B

8 Add three more play tone blocks to your program, and attach them together. The easiest way to add more play tone blocks is to right-click on one of them and choose ‘Duplicate’ in the menu.

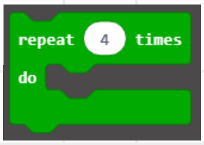


9 Change the tone and beats to play the first four notes (D D G G), each for the right amount of time:

| Tone | Beat |
|---|---|
| Middle D | 1 |
| Middle D | 1 |
| Middle G | 2 |
| Middle G | 2 |

10 Press **button A** to play your tune, though you probably won’t recognise it yet. The next section of the tune repeats the same four notes. You could add in four more play tone blocks, but it’s much easier to use a repeat block that makes your program run the same code more than once.

11 Go to the Loops section of the menu, click on the repeat 4 times block, and attach it to the bottom of the on button A pressed block.

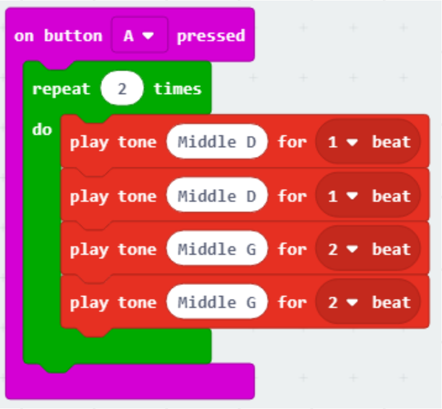


```
repeat 4 times
do
```

12 Arrange your code so that the four play tone blocks are inside the repeat block.

13 You only want these four notes to be played twice, so change the repeat 4 times to 2 times.

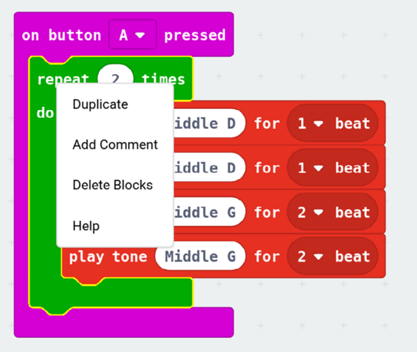
14 You can play your tune again, but it might still be difficult to recognise it.



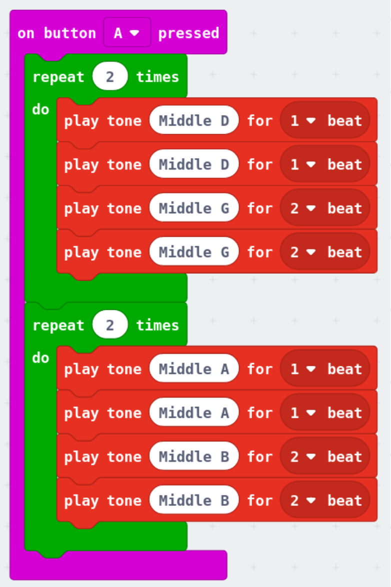
```
on button A pressed
  repeat 2 times
  do
    play tone Middle D for 1 beat
    play tone Middle D for 1 beat
    play tone Middle G for 2 beat
    play tone Middle G for 2 beat
```

15 Now that you understand how to play notes and repeat them, you should be able to create the next section of the tune, **A A B B**, which repeats twice. You can duplicate the entire `repeat` loop, to save some time.

| Tone | Beat |
|---|---|
| Middle A | 1 |
| Middle A | 1 |
| Middle B | 2 |
| Middle B | 2 |



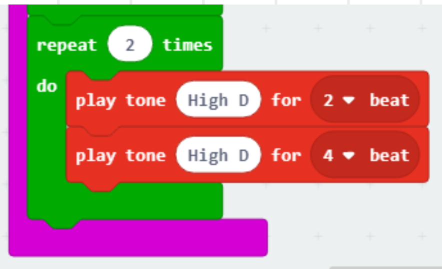
```
on button A ▾ pressed
  repeat 2 times
  do
    ...iddle D for 1 ▾ beat
    ...iddle D for 1 ▾ beat
    ...iddle G for 2 ▾ beat
    play tone Middle G for 2 ▾ beat
```



```
on button A ▾ pressed
  repeat 2 times
  do
    play tone Middle D for 1 ▾ beat
    play tone Middle D for 1 ▾ beat
    play tone Middle G for 2 ▾ beat
    play tone Middle G for 2 ▾ beat
  repeat 2 times
  do
    play tone Middle A for 1 ▾ beat
    play tone Middle A for 1 ▾ beat
    play tone Middle B for 2 ▾ beat
    play tone Middle B for 2 ▾ beat
```

16 The third section of the tune, **D D D D**, is slightly different: the four notes are played slower and for twice as many beats. That means you can also use a `repeat` block.

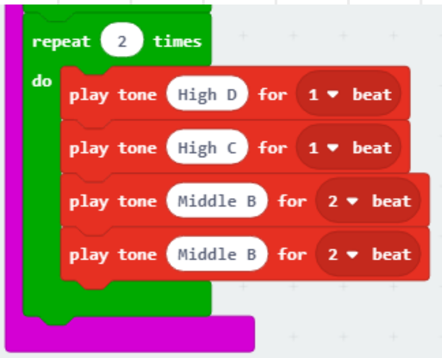
| Tone | Beat |
|---|---|
| High D | 2 |
| High D | 4 |



```
repeat 2 times
do
  play tone High D for 2 ▾ beat
  play tone High D for 4 ▾ beat
```

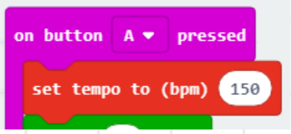
17 The final section of the tune, **D C B B**, works just like the first two sections. Use a `repeat 2 times` block and four `play tone` blocks.

| Tone | Beat |
|---|---|
| High D | 1 |
| High C | 1 |
| Middle B | 2 |
| Middle B | 2 |



```
repeat 2 times
do
  play tone High D for 1 ▾ beat
  play tone High C for 1 ▾ beat
  play tone Middle B for 2 ▾ beat
  play tone Middle B for 2 ▾ beat
```

18 The tune’s now complete, so you should recognise it when you play it. If you think it’s too slow, change the bpm (beats per minute) of the tune to 150 by adding a `set tempo to (bpm)` block to the top of your program.

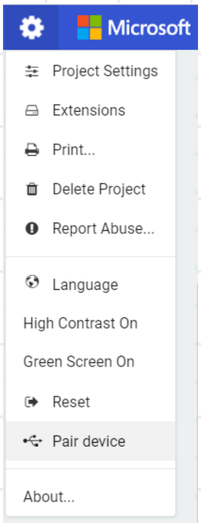


```
on button A ▾ pressed
    set tempo to (bpm) 150
```

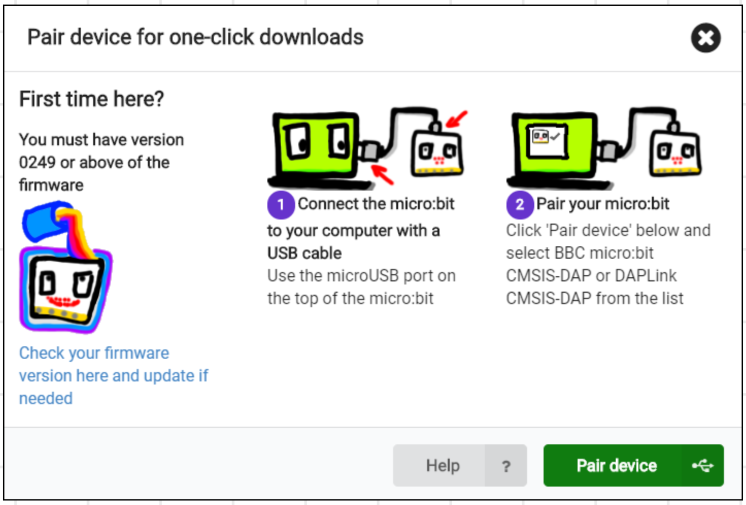
Next, you’ll need to download the program and transfer to your micro:bit.

19 Connect the micro:bit to your computer with a USB cable or via Bluetooth.

20 Click the ‘Settings’ button and then choose ‘Pair device’.



21 In the next dialogue box, click on ‘Pair device’.



Pair device for one-click downloads

First time here?

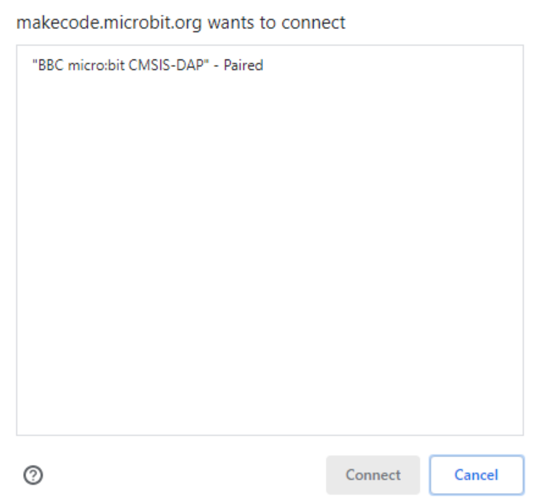
You must have version 0249 or above of the firmware

Check your firmware version here and update if needed

1 Connect the micro:bit to your computer with a USB cable
Use the microUSB port on the top of the micro:bit

2 Pair your micro:bit
Click 'Pair device' below and select BBC micro:bit CMSIS-DAP or DAPLink CMSIS-DAP from the list

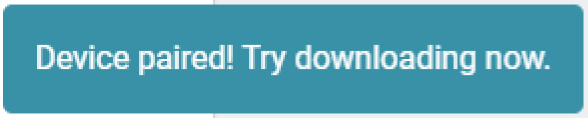
22 Choose the micro:bit that you’re connecting to, then click on ‘Connect’.



makecode.microbit.org wants to connect

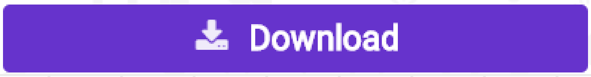
"BBC micro:bit CMSIS-DAP" - Paired

23 If your computer connects successfully, you’ll see the following message:



Device paired! Try downloading now.

24 Now that your micro:bit has been paired, you can click the ‘Download’ button to transfer your code to your micro:bit.



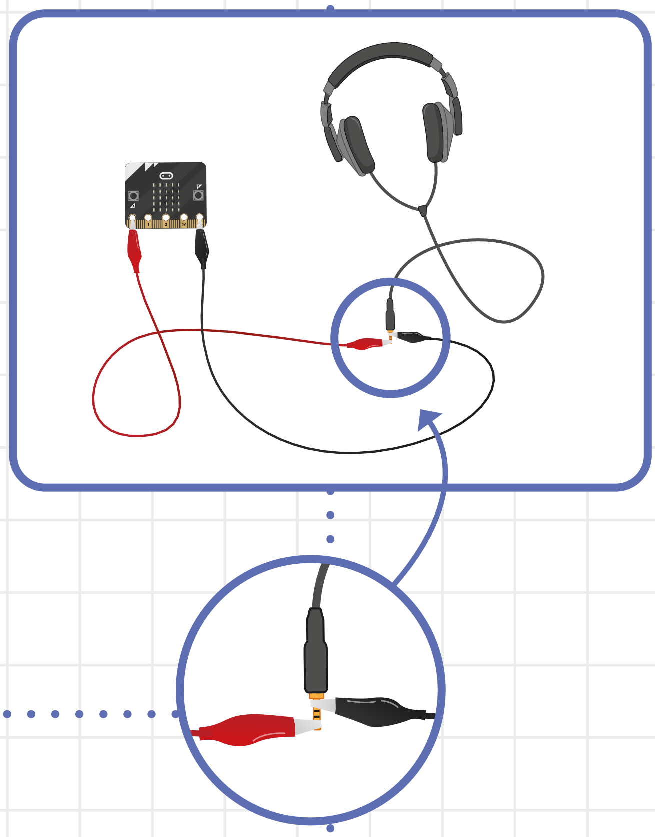
**25** Finally connect your headphones or speaker to your micro:bit using cables and crocodile clips so you can hear your tune:

a. Attach one end of a crocodile clip to the base of the speaker or headphone jack and the other end to **GND** connector on the micro:bit.

b. Attach the other crocodile clip to the tip of the jack and the connector **0** of the micro:bit.

### Be careful

The micro:bit might play the tune very loudly.



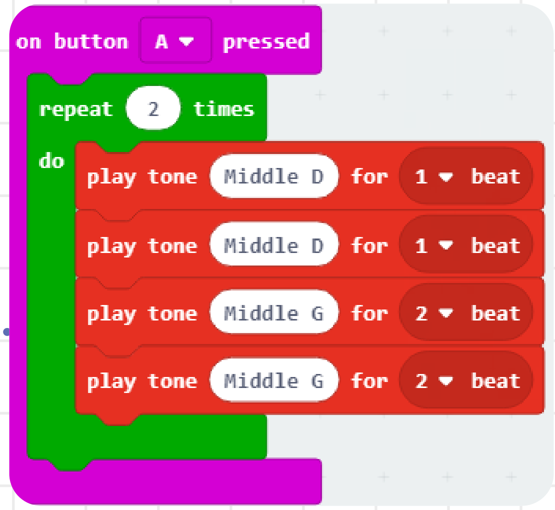
**26** Press **button A** on your micro:bit, and you should hear your tune through one of the headphones.

**27** If you're using speakers, you could play the song in a round by teaming up with others and starting your tune in time.

**Connect a battery pack to your micro:bit and try this out at your next campfire.**

## Discuss

You used **repeat** blocks in your code so you didn't have to create as many code blocks. Look at this section of your code:



```
on button A ▾ pressed
  repeat 2 times
  do
    play tone Middle D for 1 ▾ beat
    play tone Middle D for 1 ▾ beat
    play tone Middle G for 2 ▾ beat
    play tone Middle G for 2 ▾ beat
```

How could you have used additional **repeat** blocks to use fewer **play tone** blocks?