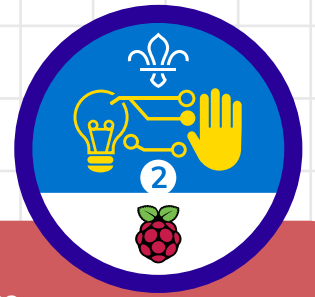


# micro:bit treasure hunt

1



## Overview

This activity fulfills **Stage 2, Requirement 1, 2, and 3** of the **Digital Maker Staged Activity Badge** ('Create a piece of interactive or animated digital art using software; create a simple computer program to help with a scouting activity, and try it out; 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)'). Young people will create a micro:bit project that will assist them in playing a treasure hunt game letting them send 'warmer/colder' animated graphics between two micro:bits.



45 – 90 minutes



Up to four young people per computer (pairs are ideal)



If you're running this activity without access to WiFi, you will need to download the software ahead of time; you may also wish to print handouts.



Wherever you have access to computers; the complete project can be used outdoors

## You will need:



- Laptops or desktop computers; tablets or other mobile devices may potentially work as well
- Activity handouts (at least one per group)
- micro:bit MakeCode editor
- Per pair:
  - Two micro:bits
  - Two micro:bit battery packs
  - One USB A to micro USB B cable



### If your meeting place has WiFi

Run the micro:bit MakeCode editor in a web browser: [rpf.io/makecode](https://rpf.io/makecode).

### If your meeting place doesn't have WiFi

Download the micro:bit code editor software beforehand; refer to the micro:bit guide at [rpf.io/scouts-microbit](https://rpf.io/scouts-microbit) for instructions.

## Key messages



- Computers can receive inputs, such as a button being pressed or something being typed on a keyboard, and they can send outputs, such as showing something on a screen, playing sound, or turning on a light. Radio messages are another example of inputs (when received) and outputs (when sent).
- Computers can select instructions to follow, based on inputs they get.
- Computers can repeat instructions.
- MakeCode is a simple drag-and-drop programming language that you can use to make a computer carry out instructions.

## Leader instructions



- 1** Discuss that many digital devices (e.g. mobile phones) can use radio transmissions to communicate with each other, and that this is just another method of receiving input or sending output.
- 2** Discuss how digital devices can improve accessibility. For instance, they can help people who have a hearing impairment or a language barrier.
- 3** In pairs, the young people should play a quick treasure hunt game:
  - a. One person should hide an object around the room while the other person isn't looking.
  - b. The other person has to then find the object.
  - c. The person who hid the object can only use the words 'warmer' and 'colder' to guide their partner to the object.
- 4** Explain that they are going to create a digital device that can help people play the game, even when they can't talk to each other.
- 5** If they have not used them before, introduce the young people to the micro:bit. Explain that micro:bits not only have two buttons (for input) and a small LED display (for output), but they are also able to communicate with each other using radio waves.
- 6** Demonstrate how to open the micro:bit MakeCode editor.
- 7** With the instructions in the printout, the young people will code micro:bits so that they can use them for a treasure hunt.
- 8** Once their project is complete, the young people can use their paired micro:bits for another treasure hunt, but this time, instead of using the words 'warmer' and 'colder', they'll use their micro:bits to send 'warmer/colder' messages.

## Alternatives



- If you do not have access to micro:bits or only to a couple, young people can still complete the coding section of this activity using the micro:bit simulation in the MakeCode editor. Physical micro:bits are only needed for playing the treasure hunt at the end.

## Community and sharing



Young people should have the opportunity to explain their code, or to explain how they found or fixed a bug, to the rest of their team.

## Adaptability



- If any young people get stuck, show them the completed code so they can see what their code should look like. You can find the completed project at [rpf.io/treasure-hunt](https://rpf.io/treasure-hunt).
- Consider pairing young people who have no knowledge of the MakeCode editor with those that do.

## Safety





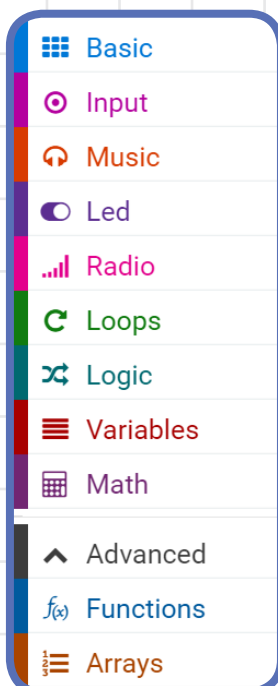
If the young people are working online, tell them to ask for permission before viewing any other websites. It's a good idea to set up parental controls — you can find instructions for this on the NSPCC website ([rpf.io/scouts-nspcc-online](https://rpf.io/scouts-nspcc-online)). You should also give each young person a Stay Safe leaflet ([rpf.io/scouts-staysafe](https://rpf.io/scouts-staysafe)).

**You are going to make a digital device to help you run a treasure hunt. The device will respond to a signal from another player to show animations telling you whether you are getting 'warmer' or 'colder'. This is great for playing this game when you need to keep quiet, when a player has a hearing impairment, or when you don't speak the same language as the person you're playing with.**

## Part 1: Create an image

**1** Go to the website [makecode.microbit.org](https://makecode.microbit.org), or open the MakeCode editor if your computer is not connected to the internet.

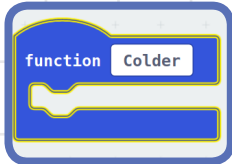
**2** Start by making the animation that will be displayed when the seeker is getting 'colder'. In the menu in the centre, click on  Advanced and then on  Functions




### Trivia

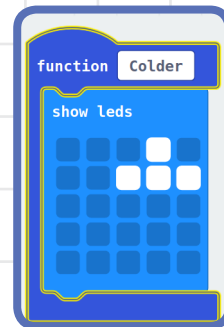
Did you know that Bluetooth radio, used by many mobile devices, was named after the tenth-century Danish king Harald Bluetooth, who united the Danish tribes into a single Kingdom? The name was chosen because Bluetooth radio unites communication methods between computers.


- 3** Create a new function by clicking on **Make a Function**. Call the function Colder. A function is just a way of grouping lots of code together.



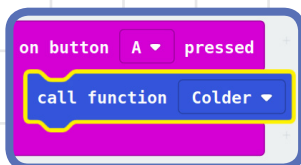
- 4** You're going to add your animation into this function. Start by clicking on the  **Basic** button in the menu. Click and drag the show leds block into your Colder function.

- 5** You can click on the squares to "turn them on" and draw a simple design to represent 'colder'. This one is meant to be a cloud, but if you like, you can come up with your own image for 'colder'.

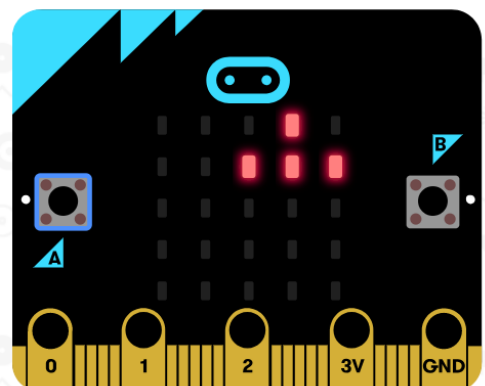


- 6** To show your image, you can use a button press to **call** (run) your function. In the  **Input** menu, click on the on button A pressed block.

- 7** In the **Functions** section of the menu, move the call function Colder block so it is linked underneath the on button A pressed block.




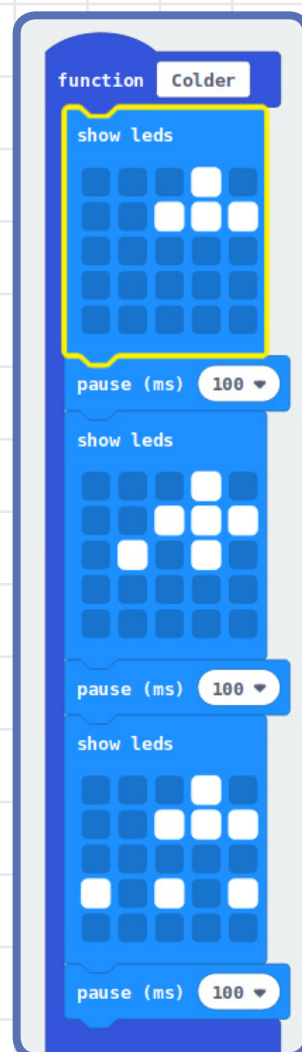
- 8** On the image of the micro:bit that's on the left side of the editor window, click on **button A**, and your icon should be displayed.




## Part 2: Animate your image

Now you can animate your image so it looks like it's moving. By adding pause (ms) blocks between show leds blocks, you can create a short animation.

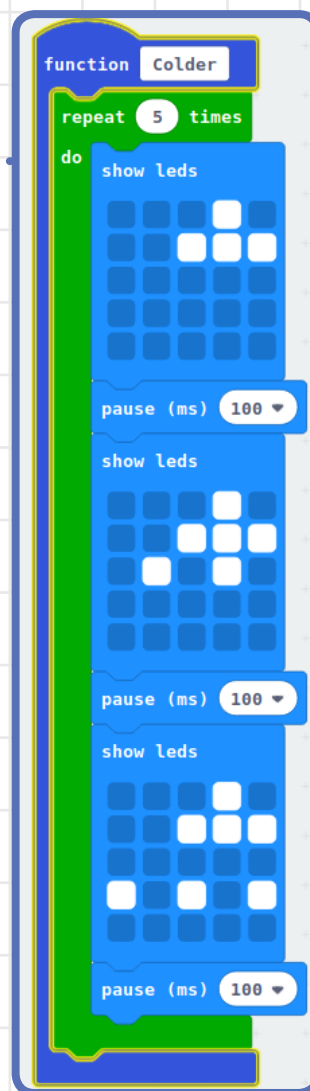
- 1 Go to the  **Basic** section, and select the pause (ms) block and insert it below your current show leds block.
- 2 Connect another show leds block at the bottom, and create a slightly different image. Then attach another pause (ms) block beneath it. You can add as many of these as you want to make an animation that signals 'colder'. Your code will end up looking something like this:
- 3 Click on **button A** in the image of the micro:bit again to watch your animation.



**4** Right now, the animation only plays once, but it might look a little better if it repeats. Use a repeat 4 times block, found in the  Loops section, to repeat the animation a few times.


**5** Click on the **button A** again to view your animation.

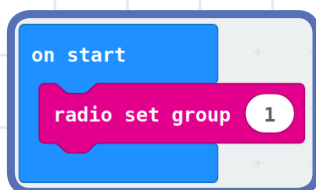
**6** Now you need a 'warmer' animation. This one should be activated by pushing **button B**. Create a new function called Warmer, and use it to loop an animation of your own design. To do this, just go through the steps you followed in Part 1 and 2 to create the Colder function and animation.





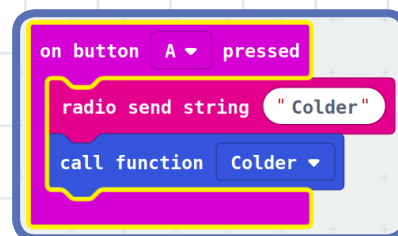
## Part 3: Make two micro:bits talk to each other



Next you'll use the radio feature of the micro:bit to send instructions from one device to another.

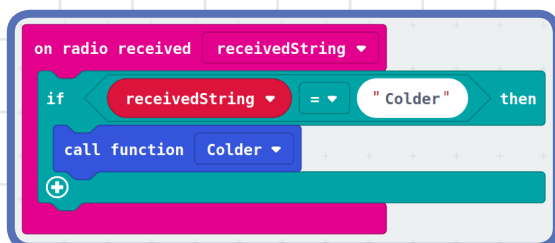
- 1 To begin with, you need to set the **radio number** of the micro:bits, as two micro:bits can only communicate with each other when they are in the same group. Look in the  **Radio** section for the block you need:



- 2 When **button A** is pressed, the controller micro:bit should send a message to the receiver micro:bit. The message can just be the word "Colder". You will need blocks from the  **Radio** and  **Text** sections to create this code.



- 3 Whenever the receiver micro:bit gets a message, it needs to check what the text is. **If** the text is "Colder", **then** the micro:bit should call the Colder function. Look in the  **Logic** and  **Variables** sections for the extra blocks you need.



- 4 Test your code by clicking on **button A** in the image of the micro:bit. Another micro:bit should appear in the window, and both should show the animation.



**5** Now repeat steps 15 and 16 for the 'Warmer' animation. When the **button B** is pressed, the controller micro:bit should send the string "Warmer". If the receiver micro:bit gets a message with the string "Warmer", then it should call the Warmer function. (Tip: stack if/then blocks on top of each other here.)


**7** Plug your micro:bit into your computer using the USB cable. It should appear as a new **drive** in your file manager. The letter assigned to it might not be the same as the one shown in the picture, but it should be called **MICROBIT**.

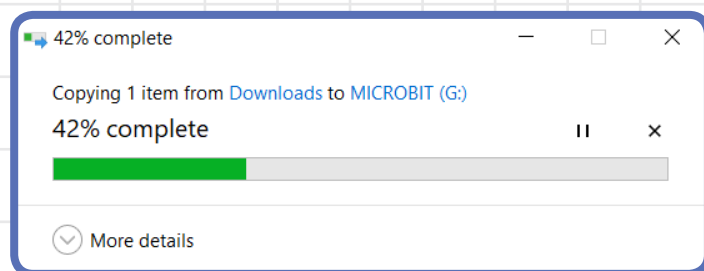
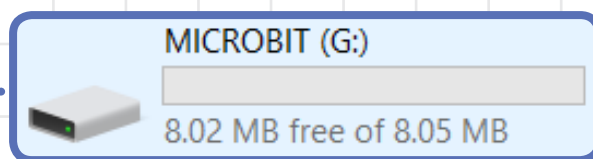
**8** Click on the **.hex** file you've downloaded to drag and drop it into this drive.

**9** The orange LED on the back of the micro:bit should flash, showing that it is receiving the program.

**10** Repeat this process with the second micro:bit, so that both have a copy of the program.

**11** Plug both micro:bits into battery packs.

**6** You have completed your code! Now you need to transfer it to the micro:bit. Click on the  **Download** button. This will place a **.hex** file in your Downloads folder.



**You're finished — now enjoy your treasure hunt!**