

Environment Detective: Exploring Our World with micro:bit

Microbit

Beginner

45 Mins

1 Overview

Ever wondered how technology can help us understand the world around us?

In this project, you are going to become an environment detective using your micro:bit and its built-in sensors.

Just like your senses help you explore the world, the micro:bit can tell you:

- How warm or cold it is using its temperature sensor
- How bright or dark it is using its light sensor
- Which direction you are facing using its compass

You will make the readings appear on the micro:bit's LED matrix.



What you'll learn

- Read the temperature using the micro:bit's temperature sensor
- Use the LED matrix to show numbers, pictures and letters
- Detect light levels and make the micro:bit react
- Use the compass to find a direction
- Use buttons and a shake gesture to control your code
- Use blocks based on Python code

Are you ready? Let's begin our detective adventure!

2 What you'll need



What you'll need

- micro:bit
- EduBlocks coding editor
- Micro USB cable
- Battery pack (optional)

3 Key words



Key words

micro:bit

The micro:bit is a tiny programmable computer, about the size of a matchbox, made for learning to code. It has two buttons, a grid of little red LED lights and built-in sensors that can detect movement, light and temperature.

A bit like a pocket-sized computer you can program yourself, a micro:bit can become a step counter, a thermometer or even a simple game.

Temperature sensor

A temperature sensor is a little part that measures how warm or cold it is around it, then turns that into a number your code can read. As the room heats up or cools down, the number changes too.

Like the thermometer that tells you if you've got a temperature when you're poorly, a temperature sensor measures the warmth around it and gives your code a number to work with.

Light Sensor

A light sensor is a little part that measures how bright or dark it is around it, then turns that into a number your code can read. As the light changes, so does the number.

Like the automatic headlights on a car that switch on when it gets dark, a light sensor notices the light dropping and lets your code react.

Compass

A compass senses which way you are facing and turns it into a number your code can read, from 0 all the way round to 359. North is 0, east is 90, south is 180 and west is 270.

Like the little arrow on a map app that swings round as you turn, a compass tells your code which way you are pointing.

LED Matrix

The LED matrix is the grid of little red lights on the front of the micro:bit. By switching them on and off in patterns, your code can show numbers, letters and tiny pictures.

Like a mini scoreboard made of dots, the LED matrix lights up to spell out your step count.

Variable

A variable is something that can change. Think of it like a labelled box: you pop something inside, and the label tells you what it is. You can swap what's inside whenever you like.

A variable called score might start at 0, then change to 1, 2 and 3 as you collect points in a game.

If statement

An if statement checks whether something is true, and only runs its instructions when it is. It's how code makes decisions.

"If it's cold, put on a coat." If it's warm, you skip the coat, the action only happens when the check is true.

Comparison Operators

A comparison operator checks how two things measure up, for example whether they are equal or one is bigger than the other. It gives back a simple true or false answer.

Asking "is 6 == 6?" gives back true, but "is 6 == 2?" gives back false, so your code knows whether the two numbers match.

4 Set up EduBlocks

First, let's create a new EduBlocks project.

1. Open your favourite web browser. Google Chrome or Microsoft Edge works well.
2. Go to app.edublocks.org.
3. Find **Create New Project**.
4. Select **micro:bit**.
5. Name your project **Environment Detective**.
6. Make sure **Blocks** is selected as the project type.
7. Select **Create**.

The EduBlocks micro:bit editor will now open.





Create Project

Project Name *

Environment Detective

Mode *

micro:bit

Type

Blocks
Great for beginners

Text only
Ideal for advanced users

Cancel

Create



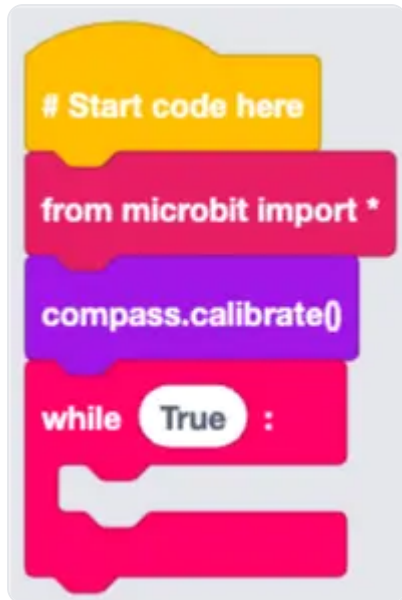
EduBlocks lets you build Python code using colourful blocks. The blocks snap together and help you see how the finished Python code is organised.

5 Prepare the code area

We need to import the micro:bit tools, prepare the compass and add a loop that keeps checking the sensors.

1. Open **Basic**.
2. Select **Imports**.
3. Drag a `from microbit import *` block onto the `# Start code here` block.
4. Open **Compass**.
5. Drag a `compass.calibrate()` block underneath the import block.
6. Open **Basic**.
7. Select **Loops**.
8. Drag a `while True:` block underneath `compass.calibrate()`.

Your starting code should look like this:



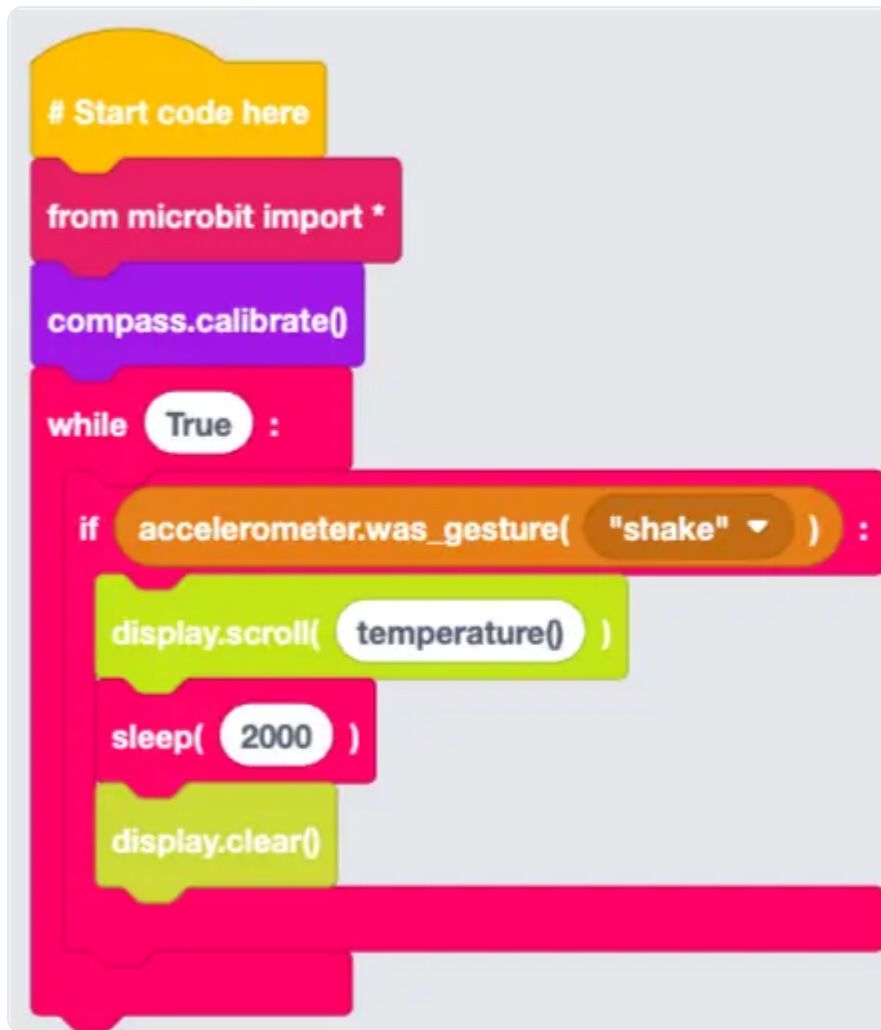
The `while True:` block creates a loop. Everything placed inside it will keep running while the micro:bit is switched on.

6 Sense the temperature

First, we will make the micro:bit display the temperature when you shake it.

1. Open **Basic**.
2. Select **Logic**.
3. Drag an `if True:` block inside the `while True:` block.
4. Open **Accelerometer**.
5. Drag `accelerometer.was_gesture("shake")` onto the `True` section of the `if` block.
6. Open **Display**.
7. Drag `display.scroll(0)` inside the shake block.
8. Change the `0` to `temperature()`.
9. Open **Basic**.
10. Select **Statements**.
11. Add `sleep(1000)` underneath `display.scroll(temperature())`.
12. Change `1000` to `2000`.
13. Open **Display**.
14. Add `display.clear()` underneath `sleep(2000)`.

Your code should look like this:



Here's what this bit of code is doing:

- The micro:bit waits until it detects a shake.
- It reads the temperature.
- It scrolls the temperature across the LED matrix.
- It waits for two seconds.
- It clears the display.



The micro:bit displays temperature in degrees Celsius.

7 Sense the light level

Next, we will use button A to check whether the room is dark.

1. Open **Basic**.
2. Select **Logic**.
3. Add another `if True:` block underneath the whole temperature section.
4. Open **Buttons**.

5. Drag `button_a.was_pressed()` onto the `True` section.
6. Add another `if True:` block inside the button A block.
7. Open **Basic**, then **Logic**.
8. Drag a `0 == 0` comparison onto `True`.
9. Change `==` to `<`.
10. Add `display.read_light_level()` before the `<` symbol.
11. Change the second `0` to `100`.

Your comparison should now say:

```
display.read_light_level() < 100
```

Show a picture when it is dark

1. Open **Display**.
2. Add `display.show(10)` inside the light-level check.
3. Add an `Image.HEART` block to replace `10`.
4. Change `HEART` to `HAPPY`.
5. Open **Basic**, then **Logic**.
6. Add an `else:` block underneath the light-level check.
7. Open **Display**.
8. Add `display.clear()` inside `else:`.
9. Open **Basic**, then **Statements**.
10. Add `sleep(1000)` underneath the light section.
11. Change `1000` to `500`.

Your light-sensing code should look like this:



When you press button A:

- The micro:bit checks the light level.
- If the light level is below `100`, it shows a happy face.
- If the light level is `100` or higher, it clears the display.

The code uses an if statement and a comparison operator to make this decision.



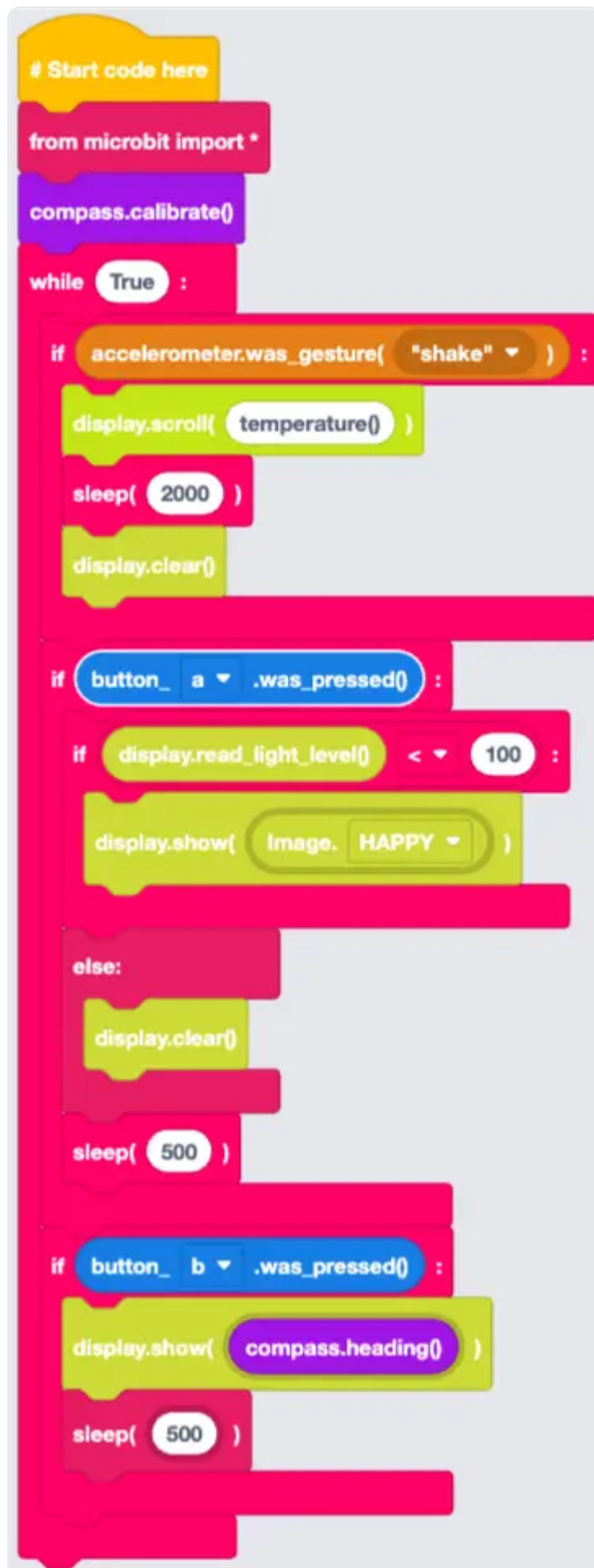
Cover the micro:bit with your hand and press button A. Then move it somewhere brighter and try again.

8 Build the compass code

Now we will use button B to display the compass direction as a number.

1. Open **Basic**.
2. Select **Logic**.
3. Add another `if True:` block underneath the whole button A section.
4. Open **Buttons**.
5. Drag `button_b.was_pressed()` onto `True`.
6. Open **Display**.
7. Add `display.scroll(0)` inside the button B block.
8. Open **Compass**.
9. Drag `compass.heading()` onto the `0`.
10. Open **Basic**, then **Statements**.
11. Add `sleep(1000)` underneath the display block.
12. Change `1000` to `500`.

Your code should look like this:



When you press button B, the micro:bit will display a number between 0 and 359.

Direction	Compass heading
North	0
East	90
South	180
West	270



The compass works like a circle. After 359 degrees, it returns to 0 degrees.

9 Show compass letters

Numbers are useful, but letters are easier to recognise quickly.

We will now create a second button B section that shows **N**, **E**, **S** or **W**.

Create the direction variable

1. Add another `if True:` block underneath the compass-number section.
2. Open **Buttons**.
3. Add `button_b.was_pressed()` to the `True` section.
4. Open **Variables**.
5. Select **Create a Variable**.
6. Name the variable `direction`.
7. Add `direction = 0` inside the button B block.
8. Open **Compass**.
9. Replace `0` with `compass.heading()`.

The block should now say:

```
direction = compass.heading()
```

A variable stores information that can change. In this project, the `direction` variable stores the current compass heading.

Add the north check

1. Open **Basic**, then **Logic**.
2. Add an `if True:` block underneath `direction = compass.heading()`.
3. Add a `True and True` block onto `True`.

4. Change `and` to `or`.
5. Add a comparison block to the first `True`.
6. Build the comparison `direction > 315`.
7. Add another comparison to the second `True`.
8. Build the comparison `direction < 45`.
9. Open **Display**.
10. Add `display.show("Hello")` inside the `if` block.
11. Change `Hello` to `N`.

The first check should now say:

```
if direction > 315 or direction < 45:
```

Add east

1. Open **Basic**, then **Logic**.
2. Add an `elif True:` block underneath the north section.
3. Build the comparison `direction < 135`.
4. Add `display.show("Hello")` inside it.
5. Change `Hello` to `E`.

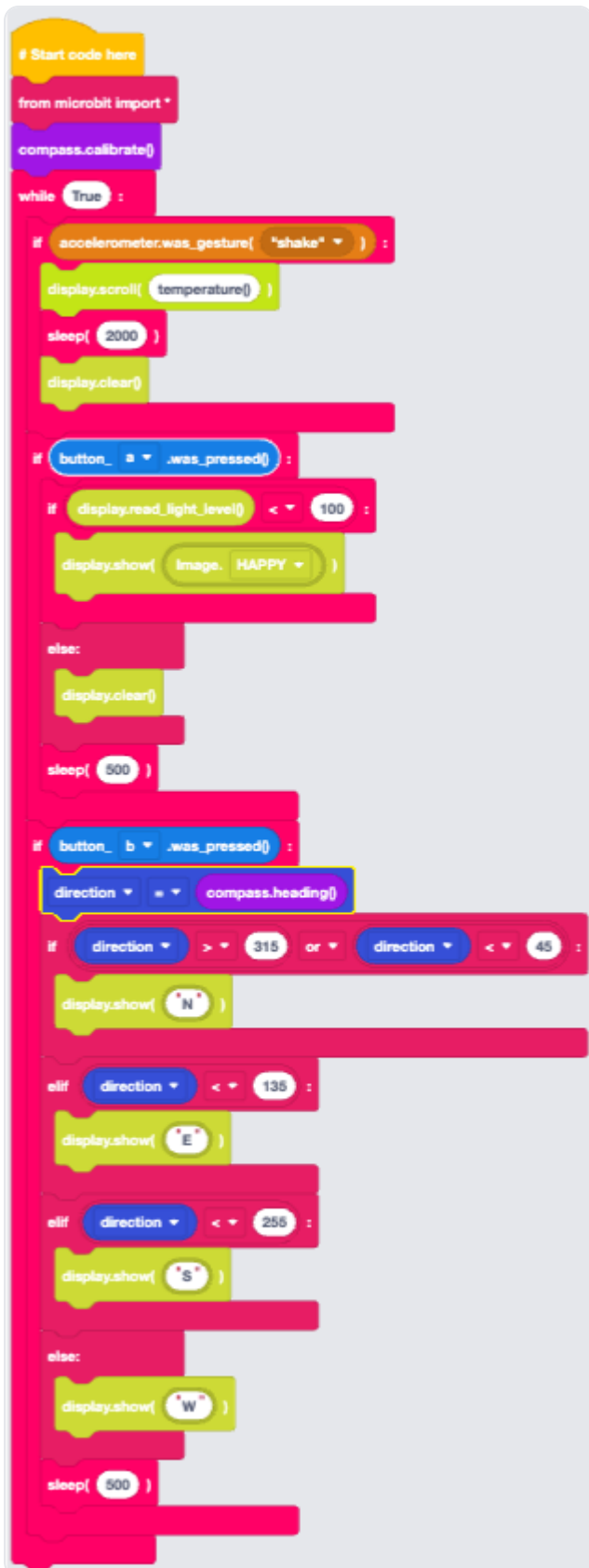
Add south

1. Add another `elif True:` block.
2. Build the comparison `direction < 225`.
3. Add `display.show("Hello")` inside it.
4. Change `Hello` to `S`.

Add west

1. Add an `else:` block underneath the south section.
2. Add `display.show("Hello")` inside it.
3. Change `Hello` to `W`.
4. Add `sleep(500)` underneath the completed compass section.

Your finished compass code should look like this:



How your code decides

Imagine the compass is a pizza cut into four slices. The code checks which slice the compass number has landed in, then shows the letter for that direction.

10 Download your code

Your environment detector is ready to transfer to the micro:bit.

1. Connect the micro:bit to your computer using the micro USB cable.
2. In EduBlocks, select **Connect**.
3. Follow the instructions on the screen to pair your micro:bit.
4. If your browser asks for permission, choose the micro:bit from the list.
5. Select **Connect**.
6. Select **Flash** to send the code to the micro:bit.



If EduBlocks cannot find the micro:bit, try another USB cable. Some cables can only charge devices and cannot transfer code.

11 Test your environment detector

When the micro:bit first starts, it may ask you to tilt it in different directions until all the LEDs have lit up.

This is called calibrating. It helps the compass work out which direction is north.

Test the temperature

1. Shake the micro:bit.
2. Watch the temperature scroll across the LED matrix.
3. Move to a different room or go outside and try again.

Test the light level

1. Cover the micro:bit with your hand.
2. Press button A.
3. A happy face should appear if the light level is below 100.
4. Move the micro:bit somewhere brighter and press button A again.

Test the compass

1. Face one direction.
2. Press button B.
3. Look at the letter on the LED matrix.
4. Turn around and press button B again.

You should see **N**, **E**, **S** or **W**, depending on which direction the micro:bit is facing.



If the compass gives an unexpected answer, calibrate it again and move away from large metal objects or magnets.

12 Try it yourself



Try it yourself

Challenge: Build an environment warning system

Can you make the micro:bit warn you when the room gets too cold?

Inside the `if accelerometer.was_gesture("shake"):` section, add another `if` block.

Make it check whether `temperature()` is below a number you choose, such as `15`.

If it is too cold, make the micro:bit show:

- A sad face
- A snowflake
- A warning picture of your own

Can you create another warning for the light sensor?

Try changing the button A code so the micro:bit shows a warning picture when the room gets too dark.

13 Stuck? Quick fixes

Problem	Try this
Nothing happens when I shake the micro:bit	Check that <code>display.scroll(temperature())</code> is inside <code>if accelerometer.was_gesture("shake") :</code> . Give the micro:bit one firm, clear shake.
The temperature looks too warm	This is normal. The sensor is on the micro:bit itself, so it can warm up after the board has been switched on or held in your hand.
The happy face does not appear	Cover the micro:bit with your hand. Check that your comparison says <code>display.read_light_level() < 100</code> .
Button B shows the wrong direction	Complete the compass calibration by tilting the micro:bit until every LED has lit up.
The compass always shows W	Check the numbers in your comparisons: <code>315</code> and <code>45</code> for north, <code>135</code> for east and <code>225</code> for south.
I cannot find a block	Check the correct category: Basic, Display, Buttons, Accelerometer, Compass, Logic, Variables or Statements .
My code will not download	Try another USB cable or USB socket. Check that the micro:bit is paired with EduBlocks.



Don't worry if it doesn't work first time. That's how coding works! Check one block at a time and compare it with the pictures.