



Activity 2: Locating the TARDIS

MicroPython

Teacher note – Curriculum links:

- Designing, developing and adapting programs to achieve specific goals

In our **Doctor Who and the micro:bit Live Lesson**, we'll be asking you to help us locate the TARDIS in time and space using your BBC micro:bit.

In order to join in with this activity, you'll need to flash the hex file to your micro:bit before or during the Live Lesson.

Step 1: Download the hex file and join in live

Right-click on the **Locating the TARDIS hex file** link on the Live Lessons website and 'save as' to download the hex file. Plug in your BBC micro:bit, hit 'compile' and drag the hex file onto your BBC micro:bit.

You will be asked to calibrate the compass by drawing a circle using your micro:bit. Move the dot on the LED screen to the edges of the screen until you've drawn a full circle around the LEDs. This may take you some time!

Once this has been calibrated you will see a smiley face appear on the LED screens.

You can now join in with our **Locating the TARDIS** activity alongside our Live Lesson. Simply turn around on the spot and press button A to try and find the TARDIS.

If you manage to find it, the message 'TARDIS' will be displayed on the LEDs!

If you need help finding it, press button B instead – this will display a message saying 'Higher' or 'Lower'. If it's 'higher', you will need to turn clockwise and try again, and if it's 'lower', you'll need to turn anti-clockwise.

Step 2: Understanding the code

After the Live Lesson, you may want to investigate how the program actually works. On the Live Lesson website, click on **Locating the TARDIS – MicroPython** to view the code on the micro:bit website.

Here is a quick explanation of how each part of the code works:

```
from microbit import *
```

```
compass.calibrate()
```

Calibrating the compass

This instruction asks the micro:bit to calibrate the compass, so it can tell which direction you're facing .

```
tardiscompass = 0
```

Setting up the tardiscompass variable

Here, we've also introduced a variable called **tardiscompass**, and gave it an initial value of 0.

```
while True:
```

```
    if button_b.was_pressed():
        tardiscompass = compass.heading()
        if tardiscompass >= 210:
            display.scroll("Lower")
        elif tardiscompass <= 190:
            display.scroll("Higher")
```

'While... true'

In this part of the program, we've introduced a 'while... true' loop that goes on forever, which means that whatever is in the loop keeps happening unless something else in the program interrupts it.

```
    if button_a.was_pressed():
        tardiscompass = compass.heading()
```

```
    if 190 < tardiscompass < 210:
        display.scroll("TARDIS")
    else:
        display.scroll("?")
```

Pressing button B

Inside this loop, we've introduced three **conditional statements**. The first checks if button B is pressed, and if it is, it changes the value of the variable **tardiscompass** to the value of the compass heading (between 0 and 360 degrees).

Within this, we've introduced a **nested conditional statement**, which checks if **tardiscompass** is more than or equal to 210. If it is, the message "Lower" is displayed. Otherwise, if **tardiscompass** is less than or equal to 190, the message "Higher" is displayed.

Pressing button A and finding the TARDIS

The second conditional statement checks if button A is pressed, and if it is, it changes the value of the variable **tardiscompass** to the value of the compass heading (between 0 and 360 degrees).

The third conditional statement checks if **tardiscompass** is between 190 and 210. If it is, then the message "TARDIS" is displayed. Otherwise (else), the message "?" is displayed.

Finding the TARDIS

As we're searching for the TARDIS – pressing button A as we turn around – we are changing the variable **tardiscompass** to the new compass heading of the direction we're facing in.

So, when we're facing in the compass bearing 190 to 210 degrees and pressing A, the second condition in the loop will be satisfied, and we will have 'found the TARDIS'!

Step 3: Test and play

Now that you understand the code, you can adapt it to create your own location hunting game.

A simple thing to experiment with is changing the messages that your BBC micro:bit displays. Simply click on the **display.scroll** lines with the messages and change the messages within the quotation marks to whatever you like.

Another thing you can do easily is change the difficulty of the game. You do this by changing the conditional statement. Currently, we're looking for a compass bearing between 190 and 210. You can make the game easier by making the range wider or make the game harder by making the range smaller.