**Barefoot meets micro:bit**

Recommended for

**ages 9-11**

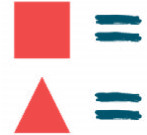
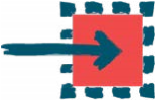
Litter Hunt - Lesson 1

Duration:

**1**

**hour**

Concepts and approaches covered



Inputs Variables

# Overview

In this lesson, pupils develop their understanding of the micro:bit by using the buttons and sensors on the device to make different images display. Pupils are introduced to the concept of a variable to store data, then create a program to alter the value of a variable using the buttons and sensors. Pupils go on to create a ‘push the button’ game, where they race against another player to see who can press the button the fastest.

# Pupil objectives

■ I can explain what the buttons and sensors on the micro:bit do

■ I can program sensors and buttons using the MakeCode editor ■ I can store data with variables using the MakeCode editor

# Before you start

Check the ‘Technical requirements’ section of the ‘Lesson overview’ document to ensure the micro:bit works successfully with your school.

Pupils can work individually or in pairs. micro:bit devices should be available from the beginning of the lesson for pupils to examine.

A whiteboard and pen should be available to introduce pupils to variables, along with printing out the images of button A and button B from the ‘UKS2 - Lesson 1 - Buttons’ file

Pupils will need a way to time 20 seconds, such as a stopwatch, clock, or timer on a phone / tablet

Pupils will work in pairs to complete the ‘UKS2 - Lesson 1 - Code Creation sheet’, which should be printed out for each pair

You might choose to complete the LKS2 micro:bit lessons before teaching this lesson. If pupils are unfamiliar with the device, the LKS2 support videos or initial tasks within LKS2 lesson one could be used to introduce pupils to the micro:bit.

The videos included with these resources can be used as a guide for how to use the micro:bit. The videos can also be shown to pupils within the lessons, as indicated in the lesson plan below.

# Resources

■ Desktop or laptop computer with a web browser that can access the micro:bit MakeCode editor - <https://makecode.microbit.org/>and an available USB port, which can be used to transfer files to the micro:bit. Further details are outlined in the ‘Technical requirements’ section of the ‘Lesson overview’ document.

■ micro:bit devices

■ Code creation sheet printed out for each pair of pupils

■ Pens / pencil to complete the code creation sheet

■ A stopwatch, clock, or timer on a phone / tablet for each pair of pupils to use ■ The following support videos: [UKS2 Buttons](https://vimeo.com/686296303/2afa5eb452)

# Introduction 5 minutes

Explain to pupils they will be using the micro:bit device, which is a special type of computer they can program. Provide pupils with a micro:bit and ask them if they can identify the following parts of the device, which they will be using during these lessons - LEDs, USB connector, Buttons, Accelerometer (Slide 3).

Ask pupils to discuss whether the LEDs are inputs and outputs, and elicit that we say they are outputs, as they are displaying information from the computer. (Please note - the LEDs are also used as an input to detect light, although this is beyond the scope of this lesson.) Explain the buttons and accelerometer are inputs, as they take information from the outside world. Ask pupils to also identify other parts of the device and discuss their functionality using the guidance within the ‘Lesson overview’ as required.

Explain to pupils they will be using the micro:bit to produce a button race game, which will enable them to see who can press the micro:bit buttons the fastest. Add that before we go on to make our game, pupils will ensure they can remember how to create a program using MakeCode.

Show pupils MakeCode - <https://makecode.microbit.org/> and ask them to share (Slide 4):

■ How to create a new project

■ How to place “show icon” blocks into the programming area

■ How to download the program

■ How to transfer the program file to the micro:bit

Remind pupils of how to keep safe when using the micro:bit (Slide 5). Explain to pupils they are to create a program containing at least one image, which they should transfer to the micro:bit (Slide 6). Once pupils have finished the task, they should assist other pupils until everyone in the class has completed it successfully.

# Main activity 45 minutes

## Using the buttons - 15 mins

Explain to pupils the micro:bit’s buttons can be used to start different parts of our programs, or to make different things happen. Show pupils the image of the micro:bit and the various code blocks (Slide 7). Ask them to match up the different code blocks with the relevant button and sensor as follows:

■ on button A pressed - Button A

■ on button B pressed - Button B

■ on button A+B pressed - Buttons A and B

■ on shake - Accelerometer

Demonstrate to pupils, using MakeCode, that the blocks related to buttons and sensors can be found in the “Input” section of the editor (Slide 8). Demonstrate to pupils how to produce a program using the ‘on button A pressed’ and ‘on shake’ blocks, such as displaying different images, as shown on slide 8. Emphasise that no image is displayed on the LEDs until the button is pressed or the device is shaken.

Ask pupils to make and transfer a program to display different images for button A and when shaken, as shown on slide 9. Once complete, ask pupils to attempt the other tasks on the screen. Support pupils as required and remind them each new program needs to be transferred to the micro:bit if required.

## Introducing variables - 15 mins

Explain to pupils we are now going to use the micro:bit to store data, such as to record how many times a button has been pressed (Slide 10). Explain that we can use a variable to store data within our programs. Using the whiteboard, explain to pupils you are going to demonstrate how a variable functions. Write the word ‘counterA’ at the top of the whiteboard and explain this is the name of the variable. Explain to pupils it is important we give the variable an appropriate name, as we need to be able to understand what it does and remember what it is storing.

Show pupils the printed images of buttons A and B from the ‘UKS2 - Lesson 1 - Buttons’ file and explain they are going to see how the program on slide 11 functions. Ask one pupil to hold the image of button A and another pupil to hold button B. Explain to pupils you are now going to start the program and write ‘0’ on the whiteboard. Emphasise that you have ‘set’ the value of ‘counterA’ to zero, which is known as initialising the variable and it is important we do this at the start of our program to ensure we start from the correct value. Explain to pupils the ‘show number’ block is used to display the value of the variable, as it would not otherwise be shown.

Ask a pupil to press the image of button A, then erase ‘0’ from the whiteboard and write ‘1’. Explain to pupils you have ‘changed’ the value of counterA by 1. Ask another pupil to press the image of button A, then erase ‘1’ from the whiteboard and write ‘2’. Explain you have again ‘changed’ counterA by 1. Repeat this process with 3-5 more pupils and change the number each time.

Ask a pupil to press the image of button B, then erase the current value on the whiteboard and write ‘0’. Explain to pupils you have ‘set’ the value of the counterA variable to 0, rather than ‘changing’ it. Increase the value of counterA again using the button A image, then set it back to zero using the button B image.

Check for pupil understanding using the following questions (slide 12):

■ Which blocks initialise the value of counterA?

■ Which blocks display the value of counterA on the LEDs?

■ Which block increases the value of counterA by 1? ■ Which block resets the value of counterA to 0?

Ask pupils to make and transfer a program to change and display the value of the variable, as shown on slide 13. Once complete, ask pupils to attempt the other tasks on the screen. Support pupils as required and remind them each new program needs to be transferred to the micro:bit if required.

## Time for a race - 15 mins

Explain to pupils that they are now going to create a program to see who can press a button the most times in 20 seconds. Show the algorithm on slide 14 and explain to pupils they should use the Code Creation sheet to match the required code blocks to each part of the algorithm. Working in pairs, ask pupils to complete the sheet.

Once complete, as a class, ask pupils to share the blocks they might use for each part of the algorithm and produce the program on slide 15 on the board. An example program can be found here - <https://makecode.microbit.org/_c7fKfy5zbPJ7> Ensure a range of pupils give suggestions as you make the program. (Note: This pedagogical approach, where all pupils work together as a class to produce a program, is known as ‘shared programming’.). Explain to pupils the order the ‘on button A pressed’, ‘on button B pressed’ and ‘on shake’ blocks are arranged is not important for the program to work, as the instructions are constantly checked whilst the program is running.

Keeping the program produced as a class on the board, ask pupils to create their own button pressing game and to transfer it to the micro:bit. Using the timers, ask pupils to race their peers to see how many times they can press the button and who is the fastest. Emphasise to pupils they must both hold the device in two hands to ensure the micro:bit is not accidentally dropped. Explain to pupils that they can restart their program using the reset button on the back of the micro:bit if required.

## Plenary 5 minutes

Ask pupils to remind each other what a variable is used for in programming (Slide 16). Ask pupils to also share ways their racing game could be improved, such as including a timer within the program, messages or images during the game, or a rest button.

# Differentiation

## Support

Pupils could be provided with the required blocks of code for each task, then asked to rearrange the blocks in order

## Stretch

More confident pupils could replace the ‘on button A pressed’ block with an ‘if, then’ block within a ‘forever’ block. The ‘UKS2 - Lesson 1 - Stretch - Activities’ file can be provided for pupils, with possible solutions included in the ‘UKS2 - Lesson 1 - Stretch - Solutions’ file.

# Assessment opportunities

Informal teacher assessment of understanding through lesson and assessment of pupils’ programs. Key understanding to assess:

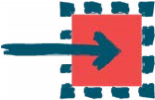
■ Can pupils produce programs including a range of instructions based on pressing the buttons or sensors?

■ Can pupils explain the instructions they used to produce each program?

# Teaching notes

Concepts and approaches

## Variables

Pupils use variables to store a range of values, including a counter of button presses within their game

## Inputs

The buttons and sensors both take information from the outside world into the micro:bit for processing, so are classified as inputs.

# Curriculum links

Please refer to the resource overview page on the website, to understand how the learning objectives covered in this lesson relate to the curriculum in your country.

**We love seeing our resources being used in the classroom.**

Please share your comments and pictures via our social channels:

  **@BarefootComp /barefootcomputing @barefootcomputing**

enquiries@barefootcomputing.org | barefootcomputing.org

Supported by With help from