Lesson Plan

Lesson 3: Emotion badge

****

|  |  |
| --- | --- |
| **Ages:** 7 – 11 |  |
| **Programming language:** MakeCode blocks |
| **Topics**: Input/output (Computer systems), Selection (Programming)  **Outcomes:** Students make an emotion badge to show how they feel, using the micro:bit’s button inputs and LED display output. |

### Key learning in this lesson

* Code the micro:bit to make different outputs happen depending on different inputs. (This is a very simple kind of selection. We look at selection in more detail in lesson 5, Nightlight.)
* Understand that inputs and outputs involve the flow of data in and out of computers.
* Apply this knowledge using the micro:bit’s button inputs and display output.

### Learning objectives

* I can make the micro:bit show different pictures on the LED display output depending on which button input is pressed.
* I can explain that **inputs** are data sent **to** a computer.
* I can explain that **outputs** are data sent **from** a computer.

# Preparation: before the lesson

### What you need

* BBC micro:bits and micro-USB cables – at least one for every two students
* At least one computer (laptop or desktop) for every two students, with internet access to the Microsoft MakeCode editor: <https://makecode.microbit.org/>
* Alternatively, you can use iPads with the micro:bit app installed. See our guide: <https://mbit.io/lessons-mobile>
* micro:bit battery packs (optional) – one per micro:bit
* PowerPoint presentation – whole class teaching slides
* Code blocks student handout (optional)
* LED planning sheet (optional for extension task)

****The lesson download also includes an optional ‘.hex’ program file of the completed project, which may be useful if you have limited internet access. You can drag and drop this direct onto the MICROBIT drive when you connect a micro:bit to your computer. You can also drag ‘.hex’ files into the MakeCode editor to examine the code and test it in the simulator.

### Differentiation ideas

* If this is one of your first coding lessons with the micro:bit, it may be hard to know which students will need more support. You can use the extension ideas in the teaching section below for students who complete the task more quickly than others.

### Decide how to deliver the ‘Create’ coding activity

You’ll share the completed code on screen with your whole class from the slides. You can choose any of these methods that suit your classroom and teaching style:

* You (or selected students) model building and testing the code yourself on a large screen. The completed code is in the lesson plan and slide deck.
* Give students printed code blocks handouts to follow or cut out and assemble.
* Share a step-by-step YouTube coding video with the whole class, or individual students.
* If YouTube is blocked in your school, we also provide an animation in the slides showing how to assemble the code.
* Students can individually follow an online step-by-step tutorial.
* ****You can also choose to manage the whole class coding activity and save every student’s code using **micro:bit classroom**. Find out more at <https://mbit.io/lessons-classroom>

### Decide how to deliver the ‘Evaluate’ activity

Students download their code to real micro:bits and test the project.

You may want your students to answer the evaluation questions:

* on paper
* verbally with partners
* as part of a whole-class discussion.

### Glossary

|  |  |
| --- | --- |
| **data**: | information collected for use elsewhere |
| **icon**: | a graphic representation of something on a computer screen |
| **input**: | data sent to a computer for processing such as button presses and sensor readings |
| **LED**: | light emitting diode - the micro:bit display is made of 25 LEDs |
| **output:** | data sent from a computer such as information shown on the LED display |
| **selection**: | making different things happen based on different conditions |

# Teaching: during the lesson

### Screenshot of slide 2. Recap prior learning (slide 2)

Ask your students what they discovered last time, for example:

* we used sequences of instructions to code animations on the micro:bit’s LED display **output**.
* we used the ‘forever’ loop block to keep the animation running.
* the micro:bit’s display, even though it only has 25 LED lights, can show pictures that represent real things like hearts or animals.

## Think: starter activity

### Screenshot of slide 3. Sharing feelings (slide 3)

* Explain that today we’re making micro:bit badges to show how we feel.
* Discuss what emotions we might want to share and why this can be a good idea.
* Some people find it hard to express or read emotions and a badge like this can be helpful for them.

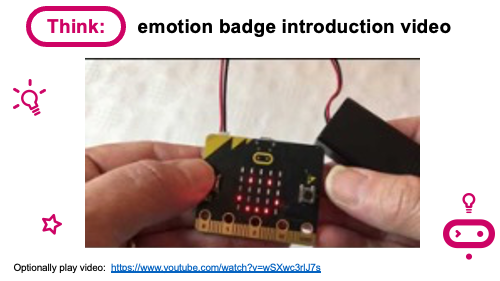
### Screenshot of slide 4. Learning objective (slide 4)

* I can make the micro:bit show different pictures on the LED display output depending on which button input is pressed.
* I can explain that inputs are information computers get from users.
* I can explain that outputs are the information users get from computers.

Ask your students what other inputs and outputs they can think of. Keys on a keyboard, a touch screen or buttons on a game controller are inputs. The display on a phone or game console, or sound from a phone or game console speakers are examples of outputs.

**micro:bit buttons video (slide 5)**

Optionally play a short, animated YouTube video introducing the micro:bit buttons: <https://mbit.io/lessons-buttons-video>

**Emotion badge introduction video (slide 6)**

Optionally play the project introduction video: <https://mbit.io/lessons-emotion-intro-video>

## Create: coding activity

### Screenshot of slide 7. Examine code with students (slide 7)

Explain:

* The ‘on button A pressed’ block makes something happen when you press input button A.
* The ‘show icon’ block makes an image appear on the LED display output. You can choose different built-in images.
* The ‘on button B pressed’ block makes something happen when you press input button B.
* So, you can make different outputs happen on the LED display when you press different button inputs.

You can also follow the link in slide 7 to open the completed code in the editor and model testing it in the simulator: <https://mbit.io/lessons-emotion-code>

### Screenshot of slide 8. Model building the code (slides 8-11)

* You can open a new MakeCode project and model building the code. The blocks you need are in the ‘Input’ and ‘Basic’ sections.
* Optionally share the YouTube coding video on slide 9 with your class: <https://mbit.io/lessons-emotion-code-video>
* Or share the coding animation on slide 10 if YouTube is blocked in your school.

Students recreate the code, testing it in the simulator. They can:

* Copy the code from slide 7.
* Follow printed code block handouts.
* Individually follow a step-by-step online tutorial:  
  <https://mbit.io/tutorial-emotion-badge> - you can share the link from slide 11.
* If you are using [micro:bit classroom](https://classroom.microbit.org/), start a new session and ask your students to join your lesson. You can also open a session with completed code to edit and share with students: <https://mbit.io/lessons-emotion-classroom>

## Screen shot of slide 12Evaluate: (slide 12)

Students transfer code to their micro:bit and test.

**Questions:**

* Does it work as you expect?
* If not, do you need to debug the code and download it again?
* How good is the project?
* Would you recommend it to a friend?
* How could you improve it?
* Could it have other uses?
* How does it work?
* Encourage students to think about how it works when holding it in their hands.

## Screen shot of slide 13Extend: (optional, slide 13)

* Add more icons when you press A + B input buttons together.
* Design your own emotion images on paper. Code them using the ‘show LEDs’ block. (You can use the LED grid planning sheet download or any paper.)
* Can you make another emotion image appear if you shake your micro:bit? (Use the ‘on shake’ block in the Input section).

## Share: revisit learning objectives (slide 14)

* I can make the micro:bit show different pictures on the LED display output depending on which button input is pressed.
* I can explain that **inputs** are data sent **to** a computer.
* I can explain that **outputs** are data sent **from** a computer.

 **Ask**:

* How did they make their micro:bits show different images? (Using different buttons and the ‘on button pressed’ input code blocks)
* What are inputs? (Data sent to a computer for processing such as button presses and sensor readings. Examples of inputs include the buttons on a game controller or the touch screen on a phone).
* What are outputs? (Data sent from a computer such as information shown on a display. Other examples of outputs include phone or game screens or sound from speakers)

## Screen shot of slide 15.Next steps (slide 15)

Today we used some of the micro:bit’s **inputs**, to make different pictures appear when we press different buttons.

Next time, we’re going to use the input that can sense when you shake your micro:bit to make a step counter: the **accelerometer**.

# Assessment: after the lesson

When assessing students’ work you may find it helpful to ask these questions:

|  |  |
| --- | --- |
|  | Did they make the micro:bit work in practice by creating relevant code and transferring it to the micro:bit? |
|  | What is their understanding of inputs? |
|  | What is their understanding of outputs? |

**Here are some guiding criteria you might want to include when assessing your students’ work:**

### WORKING TOWARDS the learning objective

|  |  |
| --- | --- |
|  | Student created some code, either in the simulator or on a real micro:bit, that showed a built-in emoji image on the LED display. |
|  | The student may be able to give some examples of inputs. They can use an ‘on button pressed’ code block to cause an outcome when they press a button but may not be able to distinguish between different input events using different buttons. |
|  | The student may be able to give examples of outputs on the micro:bit or other devices but may not be able to describe them in terms of the flow of data (information) to and from a computer. |

### MEETING the learning objective

|  |  |
| --- | --- |
|  | The student created code to make different built-in emoji images appear when they pressed different buttons and transferred it to their micro:bit. |
|  | They can explain that inputs are data (information) sent into a computer and give an example such as pressing a button to choose which emotion picture to show. |
|  | They can describe outputs as data (information) sent from a computer and give an example such as a picture that represents how they are feeling. |

### EXCEEDING the learning objective

|  |  |
| --- | --- |
|  | The student created code and transferred it to a micro:bit to show their own designs for emoji images, or created animated emoji sequences that appeared when they pressed different buttons, or buttons A+B together, or when they shook the micro:bit. |
|  | They can explain that inputs are data sent into a computer and give multiple examples such as pressing a button to choose which emotion picture to show, typing on a computer keyboard, or using a game controller. |
|  | They can describe outputs as data sent from a computer and give examples of inputs and outputs both on the micro:bit and other devices.  Additionally:  They can explain that the different ‘on button pressed’ code blocks cause different outcomes when you press different buttons and describe this choice as a kind of selection. |