Lesson 2: Introducing the micro:bit

45 minutes

Overview

This lesson gives students their first opportunity to write App Lab programs that incorporate the micro:bit physical computing device. After first inspecting the micro:bit boards, investigating some of its physical features and hypothesizing about potential uses, students move online to create applications that use App Lab inputs to control the micro:bit's main output, its LED display, to show simple pictures, words and numbers.

Question of the Day: How can I control the main output on the BBC micro:bit?

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming
- ▶ **CS** - Computing Systems

Agenda

Warm Up (5 minutes)
- **What is a micro:bit?**

Activity (35 minutes)
- **Connecting The Board**

Wrap Up (5 minutes)
- **Only 25 LEDs**

Objectives

Students will be able to:

- Connect and troubleshoot external micro:bit devices
- Control the micro:bit's LED display with code: turn individual LEDs on and off, scroll words and numbers

Preparation

- Make sure that student computers have the drivers and software necessary to connect Code.org App Lab to the micro:bit (**details here**)
- Prepare a micro:bit and USB cable for each pair of students
- Check the "Teacher's Lounge" forum for verified teachers to find additional strategies or resources shared by fellow teachers

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the teachers

- **Introducing the micro:bit** - Slides
  ▼ Make a Copy

For the students

- **LED Planning Sheet** - Resource
- **The micro:bit LED Display** - Resource
- **The micro:bit: Overview** - Resource

Vocabulary

- **LED** - Light Emitting Diode, lights that use very little energy and which are

found on many electronic devices
- **Output** - the information computers give to users, devices, or other computers

Teaching Guide

Warm Up (5 minutes)

Remarks

> These boards are called BBC micro:bits. In this unit we're going to learn how to use some of the micro:bit's built-in inputs (sensors and buttons) and outputs in App Lab programs. In this lesson, we'll start by focusing on the micro:bit's main **output**, its LED display.

What is a micro:bit?

Distribute: Give each pair of students a micro:bit board and USB cable. Let them know that they should not plug the micro:bits into their computers yet, just spend one minute inspecting them carefully.

Prompt: What do you think the micro:bit is? What do you think it might be used for?

Discussion Goal: Prompt students to make links between features of the micro:bit and other devices they may know, like phones or game controllers that may also have inputs like buttons, outputs like displays. Record student ideas for using the micro:bit on the board. Ideas may not be accurate or practical, but that's fine if they are based on evidence from inspecting the micro:bit board. You and your students can revisit these ideas and refine them as you work through the unit.

Question of the Day: How can I control the main output on the BBC micro:bit?

Activity (35 minutes)

Connecting The Board

Transition: Ask students to plug their boards in and head to **the Maker Toolkit setup page** to confirm that the software has been correctly configured.

Transition: Send students to Code Studio

1 **Setup**

Teaching Tip

Text-to-Speech Options: The instructions panel includes two options that can support comprehension for students.

- **Text to Speech** which reads aloud the instructions for students
- **Microsoft Immersive Reader** which opens a new panel for the instructions and gives controls to change the text size, contrast, or translate to another language.

Click here to learn more about these options

Troubleshooting Devices: If student devices aren't able to connect, they can click the Help and Tips tab and then access the Maker Setup page (click the image to see more). This will let them re-reconnect their board to App Lab.

If students continue to have issues, refer to the Troubleshooting section of our **Maker Support Article**

2 Predict

Teaching Tip

Facilitating Predict Levels: Predict levels are a great opportunity for students to think critically about code and engage in class discussions. Consider having students think of their own prediction and discuss with a partner before typing in their response. Once they're run the code, bring the class together for a full-group discussion to discuss how their predictions were similar or different from the resulting program. Use this as an opportunity to address any misconceptions that students may have had about the code initially.

Digging Deeper: For more tips about programming levels, see the **Guide to Programming Levels**. This document includes strategies and best-practices for facilitating programming levels with students.

Discussion Goal: When the button on the screen is clicked, the code will turn on a small red LED slightly offset from the corner of the ledScreen. Students may have expected the LED light in the corner to light up, expecting `(1, 1)` to represent the top-left corner. In the following levels and discussion, students will discover that `(0, 0)` is the upper-left corner of the ledScreen.

3-5 Skill Building

3 4 5

Teaching Tip

Facilitating Skill Building Levels: Skill Building levels are designed to continue teaching new skills and blocks through exploration, trial-and-error, and using worked examples from pre-supplied code. Students are still getting familiar with the concepts in the lesson and will need strong support throughout these levels to build confidence, debug their code, and cement their understanding.

Consider having students complete Skill Building levels in pairs using **Pair Programming**, which has students use one computer and trade between being a Driver or a Navigator. This process is highlighted in **this video**, which you can show to the class. You can have students switch roles based on a timer, or switch every time they complete a level.

Digging Deeper: For more tips about programming levels, see the **Guide to Programming Levels**. This document includes strategies and best-practices for facilitating programming levels with students.

6 Predict

Discussion Goal: This code will scroll the text "Hello World" across the ledScreen. Students may not be sure what it does initially, but should be able to make sense of the block once they see the result on their micro:bit.

7-8 Skill Building

7 8

Teaching Tip

Using Resources: Below you can find recommendations for using the many resources students are introduced to in the lesson. You could consider creating a "Resource Chart" to keep track of these options and support students to be self-sufficient as they progress through levels.

- **Videos:** Watched as a class, but students can always return to them.
- **Help and Tips Tab:** This tab contains all of the relevant videos and reference guides for a particular level.
- **Reference Guides:** Contain text and diagrams explaining content. These are intended as helpful student resources, not class readings. They are a good place to go for review after learning content or when students get stuck in levels. You may decide to print these and have them available for students as they work through levels.
- **Documentation and Examples:** Hovering over a block will show a short description of what the block does. Clicking the "See Examples" link will open the documentation for that block.
- **Level Instructions:** Instructions may introduce small pieces of new content. Each level features a "Do This" section explaining what students are supposed to do in that level. Set the expectation early that reading these instructions, not just the "Do This" section, is important.

9 Practice

Teaching Tip

Facilitating Practice Levels: Practice levels are designed for students to apply their knowledge from the previous levels and develop fluency in using the new blocks of code to solve problems. Students can choose which practice levels they would like to complete, and it's not necessary for a student to complete each practice level before continuing.

Students tend to be more engaged and respond better when they have an authentic choice about how to continue their learning. Allow students to choose practice levels according to their interests and level of comfort, and consider providing opportunities for students to demonstrate and explain their solutions to the practice levels they chose to the entire class.

Digging Deeper: For more tips about programming levels, see the **Guide to Programming Levels**. This document includes strategies and best-practices for facilitating programming levels with students.

10 Assessment

Teaching Tip

Facilitating Assessment Levels: Assessment levels contain a single task that requires applying the skills and concepts from the level in order to solve. Students should complete these levels individually and you can use your judgment of how much external help students should have. Assessment levels also contain a rubric that can be used for formative assessment and a box to provide feedback to students - **click here to learn more about using rubrics and giving feedback to students**.

Digging Deeper: For more tips about assessing student work, see the **Guide to Assessment**.

11 **Challenge**

Teaching Tip

Facilitating Challenge Levels: Challenge levels are designed as extensions to the concepts and skills students learn throughout a lesson. Challenge levels tend to focus on more open-ended tasks for students to complete, or opportunities to combine several skills from previous lessons together into one program.

Challenge levels do **not** need to be completed for students to meet the core objectives of a lesson. Instead, every task in a challenge level is meant to supplement and enrich the learning objectives of a lesson, but are not required for future lessons. Students can still demonstrate mastery of the objectives of a lesson without completing any of the challenge levels.

Digging Deeper: For more tips about programming levels, see the **Guide to Programming Levels**. This document includes strategies and best-practices for facilitating programming levels with students.

Wrap Up (5 minutes)

Only 25 LEDs

Remarks

> You can think of the micro:bit's LED display as a simple version of a phone or computer screen. Even though it only has 25 pixels (picture elements) rather than thousands, we can still use it to show simple emojis, pictures, words and numbers. Working on such a limited canvas gives your students the chance to practice **abstraction**, simplifying a problem by only keeping its most essential elements, such as the mouth and eyes needed to show a happy or sad face.

Discuss: What did we learn using just 25 LEDs?

Discussion Goal: Encourage students to talk about uses of LED matrix displays like the one on the micro:bit, and in particular how they found the experience of using it. What was positive? Did they find the limitations and simplicity freeing? Did they enjoy holding a physical display in their hands? Did it help cement their understanding of how numbered coordinates work? What was frustrating? For example, they may have found it hard to draw pictures on the display using coordinates. In later lessons in this unit, we will look at other ways of creating pictures on the LED display they may find quicker to use.

Journal: If time allows, it's a good idea to get students to record their observations in a learning journal which they can update and revisit throughout the unit.