Physical Computing

Lesson Plan & Overview | Rutgers EIR: Extending the CS Pipeline

Grade Level	Middle School
Duration	5-7 days
CS Standards Addressed	8.1.8.CS.2, 8.1.8.CS.3, 8.1.8.CS.4, 8.1.8.AP.1, 8.1.8.AP.2, 8.1.8.AP.3

Introduction

This five-to-seven day lesson package introduces middle school students to the foundations of physical computing through hands-on activities using Micro:bits. Throughout the unit, students explore how computers can interact with the physical world through inputs, outputs, sensors, and user interaction. By combining coding with physical devices, students begin developing an understanding of how technology responds to people and environments in real-world systems.

Students begin by exploring the components and capabilities of the Micro:bit while learning the basics of physical computing and device interaction. As the unit progresses, students create and test interactive programs, practice debugging and revising code, and examine how algorithms provide step-by-step instructions for solving problems. Through collaborative activities such as the “mystery algorithm” challenge, students compare multiple problem-solving methods while considering concepts like accuracy, efficiency, and constraints in computing systems.

The unit culminates in an open-ended creative physical computing project in which students design and build their own Micro:bit-based creation. By the end of the unit, students will have developed foundational coding skills while also strengthening creativity, collaboration, computational thinking, and problem-solving abilities through authentic hands-on learning experiences.

This lesson package includes:

- Lesson Plan & Overview
- PPT Slides for Classroom Instruction
- Introduction to Micro:bit - Student Worksheet
- Code & Create Project - Student Worksheet

Lesson Agenda

Class 1 – Introduction to Physical Computing & Micro:bits

Objective: Students will define physical computing and identify how devices like the Micro:bit use inputs and outputs to interact with the world. Students will explore the basic components of the Micro:bit and learn how to connect them to the computer.

Time	Activity	Notes
5 mins	Warm-Up: Question	Class question (ask or show slide): "What kinds of technology respond when you touch, move, or interact with them?" • Encourage students to think about examples like game controllers, phones, smart watches, and automatic doors.
10 mins	Presentation: What is Physical Computing?	1. Introduce physical computing and explain the differences between inputs and outputs. 2. Show the slide with example images (slide 7), and ask the class: • How are each of these computing devices? • How do they interact physically? • What are some other examples you can think of? • **What are their inputs & outputs?** 3. Introduce the Micro:bit and briefly introduce its basic parts, including buttons, LEDs, speaker, and sensors.
15 mins	Activity: Parts of the Micro:bit	Students will explore the Micro:bit and complete the introductory questions from the student worksheet about buttons, LEDs, and standalone functionality. Circulate and assist students with identifying hardware features as they work.
5 mins	Demonstration: Connecting the Micro:bit to the Computer	Show students how to connect the Micro:bit to the computer using the Microsoft/Micro:bit "MakeCode" website. This is just a demonstration, as students will be doing it themselves in class 2. Note: Micro:bit is also compatible with Scratch, another block-based programming language. If your class is already using scratch you may want to use that platform instead of Microsoft/Micro:bit "MakeCode". • For help connecting a Micro:bit to scratch, reference their support page here : support.microbit.org/support/solutions/articles/19000026080-using-the-micro-bit-with-scratch
5 mins	Wrap-up: Class Discussion	Review the concepts of inputs and outputs. Ask students: "What are some inputs on the Micro:bit? What are some outputs?" Students may continue any unfinished worksheet questions for homework if needed.

Class 2 - Coding with Micro:bits

Objective: Students will use Micro:bit extensions to create and test simple physical computing programs. Students will apply sequencing, testing, and debugging skills while completing guided coding activities.

Time	Activity	Notes
5 mins	Warm-Up: Question	Class question (ask or show slide): "What happens if a computer program has incorrect instructions?" • This naturally introduced de-bugging. Possible discussion around: glitches, errors, freezing, wrong actions
5 mins	Recap of Class 1 & Debugging Instruction	1. Briefly review physical computing, inputs/outputs, and connecting the Micro:bit to the MakeCode site (reference image on slide 14). 2. Explain the concepts of both coding and debugging (slide 15)
10 mins	Presentation: Guided Tutorial Options	Introduce the tutorial choices from the student worksheet, including Step Counter, Roll a Die, Rock-Paper-Scissors, and Coin Toss. Review expectations for testing and downloading programs to the Micro:bit. • Tutorials found at: makecode.microbit.org/
20 mins	Independent/Paired Coding Work	Students will select one tutorial project to build, test, revise, and finalize. Encourage experimentation and troubleshooting. Circulate to support debugging and device connections.
5 mins	Wrap-up: Reflection & Share-Out	Students briefly share what their programs do and discuss challenges they encountered while coding. Remind students to complete worksheet documentation and screenshots.

Class 3 - Algorithms & Counting People: Design Challenge

Objective: Students will explain what an algorithm is and compare multiple algorithms designed to solve the same problem. Students will evaluate algorithms based on accuracy and efficiency while designing creative solutions using Micro:bits.

Time	Activity	Notes
15 mins	Whole-Class Activity: Counting People Algorithm Challenge	Students will participate in the "Mystery Algorithm" classroom activity (reference slide 20). 1. Begin with all students standing and holding the number 1 on a sticky note or white board. 2. Find another person standing up. 3. Add your two numbers together. 4. Write this sum on one person's note / board, and the other person should sit down. 5. Repeat this until there is only one person left standing. This activity physically models an algorithm, removes computers initially, and helps students realize algorithms are not "magic". • **When finished, ask the class "what does this number (their total count) mean"?**
10 mins	Activity Warm-up: Question & Brief Instruction	Class question (ask or show slide): "Can there be more than one correct way to solve a problem?" • Engage the class in a brief instruction/discussion surrounding accuracy vs. efficiency for algorithms (reference slide 23).

20-30 mins	Small Group Activity: Introductory Micro:bit Design Challenge	1. Ask the class to think critically about other algorithms they can use to count the number of people in the room. 2. Each group (3-5 students) should be given one Micro:bit device 3. Task: using only your Micro:bit tools, design an algorithm to count the number of people in the room (total number of students). Some examples of other algorithms include (for help/discussions): • Counting one-by-one • Counting in pairs (what they just did) • Counting by rows or tables • Estimation (looks like ~25 people) This activity will likely take the remainder of the class period.

Class 4 - Creative Physical Computing Project

Objective: Students will apply physical computing concepts to design and create an original Micro:bit project and will use problem-solving and iterative design practices to test and improve their programs.

Note: The physical computing project which begins in this lesson will likely take more than two class periods to complete. Be mindful of the need for 1-2 additional days of work before moving on to the final class.

Time	Activity	Notes
5 mins	Warm-Up: Question	Class question (ask or show slide): "What makes technology fun, useful, or engaging to use?" • Encourage multiple answers ○ Some examples include games, music apps, fitness trackers, smart devices, etc.
10 mins	Presentation: Introducing the Code & Create Project	The educator will introduce the open-ended Code & Create Micro:bit project and review expectations. Students may create games, tools, interactive displays, or other creative programs using prior knowledge from the unit. • Projects may: solve a problem, entertain users, display information, react to inputs, use lights/sounds/buttons. • Requirements: mut use the Micro:bit, it must function without being connected to the computer, it must be tested and revised, and it must include a completed worksheet.
20 mins	Independent/Paired Project Work	• Students will design and begin creating their own Micro:bit project and completing the "Code & Create" student worksheet included in this lesson package. ○ Encourage students to sketch ideas, test frequently, and revise their code as needed. Circulate to support troubleshooting and creativity. Note: students do not need highly advanced or complex projects. The focus is on applying physical computing concepts creatively and thoughtfully.
5 mins	Wrap-Up: Midpoint Reflection	Students briefly describe what their project is intended to do and identify one challenge they are currently working through.

Final Class - Finalize & Showcase Physical Computing Projects

Objective: Students will finalize and present their physical computing projects while reflecting on the design process, computational thinking, and problem-solving strategies used throughout the unit.

Time	Activity	Notes
5 mins	Warm-Up: Question	Class question (ask or show slide): "What is one thing you learned about coding or technology during this unit?"
20 mins	Final Project Work Time	Students will finalize, test, and revise their projects. Remind students to complete worksheet reflections and screenshots.
10-15 mins	Classroom Showcase	Students will demonstrate their projects through short presentations or a gallery walk format. Encourage students to explain what their program does, how it works, and any challenges they solved during development.
10 mins	Wrap-Up: Debrief Discussion & Reflection	Lead a class discussion about creativity, algorithms, debugging, and physical computing. Encourage students to reflect on how physical devices and code work together to solve problems and interact with people.