LESSON 13

# Digital vs. Analogue Input

40 minutes | Foundations of Computer Science | Outcome 2.14 | Outcome 3.11 | Outcome 3.12 | Outcome 3.13 | Outcome 3.14 | Outcome 1.22 | Outcome 1.9

## WHAT THIS LESSON DOES

In this lesson, you'll explore the differences between digital and analogue inputs in computing. Understand their unique characteristics, identify real-world examples, and demonstrate both types using MakeCode for micro:bit in practical simulations.

## CURRICULUM OUTCOMES

| Outcome | What the specification asks for |
|---|---|
| **2.14** | describe the difference between digital and analogue input |
| **3.11** | use and control digital inputs and outputs within an embedded system |
| **3.12** | measure and store data returned from an analogue input |
| **3.13** | develop a program that utilises digital and analogue inputs |
| **3.14** | design automated applications using embedded systems |
| **1.22** | read, write, test, and modify computer programs |
| **1.9** | use modelling and simulation in relevant situations |

## WHAT STUDENTS SHOULD BE ABLE TO DO

- Grasp the fundamental concepts of digital and analogue inputs in computing.
- Identify the key differences between digital and analogue data processing.
- Recognise real-world applications of digital and analogue inputs.
- Apply knowledge by simulating digital and analogue inputs using MakeCode for micro:bit.
- Understand the importance of input types in designing computing systems.

### BEFORE THE LESSON

- Open makecode.microbit.org yourself and confirm it loads through the school filter, no micro:bit hardware is needed as all three programs run in the on-screen simulator.
- Check the light-level slider is visible in the simulator on a class machine, students adjust it to see analogue values change.

# Digital vs. Analogue Input

**HOW THE LESSON RUNS**

- Introduction
- What is Digital Input?
- What is Analogue Input?
- Key Differences
- Real-World Examples
- Practical: Digital Input with micro:bit
- Practical: Analogue Input with micro:bit
- Combining Inputs: A Simple Example
- Wrap Up

**TEACHING MOVES**

- 2. What is Digital Input? Anchor digital to the word digits: countable, separate, all-or-nothing states. Use the keyboard key in front of them as the example, pressed is 1, released is 0, with nothing between.
- 3. What is Analogue Input? Contrast directly with the previous step: a dimmer switch has infinite in-between positions where a light switch had two. Stress continuous means a smooth range, not just more options.
- 4. Key Differences. Draw the two-column table on the board as you talk: nature, processing, noise, examples. Flag that analogue must pass through an ADC before a computer can use it, this is the key exam point.
- 5. Real-World Examples. Ask students to sort a few devices into digital or analogue before you reveal the answers. The smart-home example, door sensor digital and temperature sensor analogue, sets up the combined program later.
- 6. Practical: Digital Input with micro:bit. Have them type the button code rather than paste it. Emphasise the event runs only on the press, a discrete moment, then click A in the simulator to trigger the Yes icon.
- 7. Practical: Analogue Input with micro:bit. Point out forever loops so the number keeps updating. As they drag the light slider, the displayed value moves through a whole range from 0 to 255, that continuous change is the analogue idea.
- 8. Combining Inputs: A Simple Example. Name the pattern explicitly: the digital button decides when to sample, the analogue sensor decides what value comes back. This is how real systems avoid reading constantly.

### WHAT TO WATCH FOR

- Students think analogue simply means old or non-computer, and digital means modern, rather than continuous versus discrete. Show the light-sensor reading is analogue data used by a very modern device, and the button is digital. The difference is the nature of the signal, not the age of the technology.
- Students assume a computer can read analogue values directly, missing the conversion step. Return to the ADC point from Key Differences. The micro:bit already converts the light level to a number for you, but a raw analogue signal must be digitised first.
- Believing the light level has only a few settings because the simulator slider snaps as they drag. Have them move the slider slowly and watch many different numbers appear, contrasting with the button which only ever gives one action per press.

### DIFFERENTIATION

- Emerging: If a student cannot find the button or light blocks, point them to the Input category and pair them with a neighbour to locate blocks by colour. Let them run just the digital button program successfully before attempting the analogue one.
- Developing: Ask them to predict what number the display shows before moving the slider, then check. Have them explain in one sentence why the button code sits in an on-button-pressed block but the light code sits in forever.
- Proficient: Challenge them to modify the combined program so the icon shown depends on whether the light level is above or below a threshold, introducing a comparison on analogue data. Ask them to describe how an ADC would convert a continuous voltage to a number, as evidence they could reuse in an Applied Learning Task on inputs.