technology

micro:bit responds to an input

Lesson at a glance

Open by naming everyday gadgets that sense sound or movement, then set the goal: a clap-controlled micro:bit light. Model each MakeCode step on the IWB — variable, sound threshold, on-loud-sound trigger, if-then-else toggle — before pairs build it. They test first in the simulator by dragging the sound slider, then download to the board and clap. Close with a display-only code talk on bugs fixed and one improvement.

Learning objectives

- Program a micro:bit to react to a sensor input such as shake, tilt or sound
- Test, find and fix a bug in the response program
- Propose one improvement to how the device responds

Before the bell — prep

Charge and pair one micro:bit plus USB cable per pair before the lesson. Open makecode.microbit.org on the IWB and confirm browsers load it. Test one board's microphone in the room so you know a sensible starting threshold near 128.

Materials

Item	Qty	Per	Source	Low-cost substitute
micro:bit (with USB lead)	1	pair	school devices	run the lesson on the IWB using the MakeCode simulator

Safety watch-point

Keep USB cables tidy and clear of walkways while pairs plug in and test micro:bits; no trailing leads between desks.

Lesson 28

Teaching moves

- **What we're building:** Frame pupils as programmers building a device that senses a clap and reacts. Ask what everyday gadgets react to sound or movement, then reassure them bugs are expected and fixing them is the job.
- **Create a new project:** Show opening makecode.microbit.org and starting a fresh empty project on the IWB. Circulate and check no pair has reopened an old project before you move on.
- **Create a variable:** Explain the light needs to remember its current on/off state, then model creating lightOn and setting it to false on start. Ask why the micro:bit must remember whether the light is already on, and check every pair's on-start block.
- **Sound threshold:** Model the set-loud-sound-threshold block at 128. Ask what happens if the value is too high or too low, and note they will likely retune it later for the real room.
- **Detect a clap:** Place the on-loud-sound block and stress it is the trigger — code inside runs only on a loud sound. Watch for later blocks dropped outside it, which means nothing happens on a clap.
- **Add the if then else:** Model the if-then-else and read it aloud as 'if the light is NOT on, do this; otherwise do that.' Link 'not lightOn' back to the on/off memory box so the toggle logic clicks.
- **Turn it on:** Drag show-LEDs and lightOn = true into the first branch. Ask why the variable must change as well as the lights, and check the blocks sit inside that branch.
- **Turn it off:** Model clearScreen and lightOn = false in the else branch. Have pairs read each branch label aloud to catch swapped on/off code before they test.
- **Test it in the simulator:** Show dragging the simulator sound slider across 128. Ask what should happen each time it crosses the threshold. If nothing toggles, send them back to variable and branch placement before any download.
- **Send the code:** Guide download to the physical micro:bit, then clap to test. Prompt pairs to nudge the threshold up or down for the classroom. Keep USB cables off walkways while they work.
- **Code talk — what worked and what we'd change:** No typing. Invite one pair to share a bug and how they tracked it, revoice the debugging steps for the class, then draw out one concrete improvement idea. Keep it short and conversational.

What it should show

In the simulator, crossing the sound slider above 128 should toggle full LEDs on, then clear them off, alternating each crossing. On the real board, a firm clap does the same. If it never triggers, threshold is too high or code sits outside on-sound; if it flickers or won't toggle, lightOn is not updated in both branches or the branches are swapped.

Misconceptions & interventions

- **Pupils think showing the LEDs is enough and forget to set lightOn to true or false, so the toggle breaks after one clap.** — Point at both branches and ask what the micro:bit will 'remember' next clap if the variable never changes. Have them add the set-variable block beside the LED change and re-test.
- **Pupils put the if-then-else or LED blocks outside the on-loud-sound block, so clapping does nothing.** — Trace with a finger: 'When does this code run?' Show the blocks nesting inside on-sound on the IWB and have them match that structure.
- **Pupils think one threshold works everywhere and don't adjust 128 when the room is noisy or very quiet.** — Ask what the number means (0–255 loudness). Have them try a slightly higher or lower value and clap again until the board responds cleanly.

Differentiation

Emerging	Developing	Proficient
• Pair works at the teacher table copying each IWB block in lockstep; teacher checks lightOn starts false and both branches sit inside on-sound before any download.	• After a working toggle, prompt: try one new threshold value and note whether claps need to be louder or softer in this room.	• Stretch: propose and justify one response improvement (e.g. a short pause so one clap cannot double-trigger, or a different LED pattern) ready to share in the code talk.

Cross-curricular hook

Links to Maths algebra thinking — a variable as a named value that changes, and boolean true/false as two states.