Lesson 28

technology

micro:bit responds to an input

Lesson at a glance

The class programs a micro:bit into a clap-controlled light, step by step, on their own devices: a variable that remembers whether the light is on, a sound threshold that decides how loud counts, a block that listens for a loud sound, and an if-then-else that turns the LEDs on if they are off and off if they are on. It is tested in the simulator by dragging the sound level up, then sent to a real micro:bit and tested by clapping. The idea is a device that reacts to the world without anyone pressing a button.

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

One micro:bit per pair if you have them, otherwise the simulator, which has a sound slider that does everything except the clap. Build the program yourself once beforehand and clap at it, because the threshold almost always needs adjusting for a real room and it is much easier to help once you have done it. Agree a signal for quiet before the testing starts, thirty children clapping at once triggers every device in the room.

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

Normal care with devices and cables, and supervise plugging and unplugging micro:bits. Agree a limit on how loud the testing gets, and be aware that sustained clapping is uncomfortable for some pupils. Nothing else on this lesson.

Lesson 28

Teaching moves

- **What we're building:** Ask what could switch a light on without anyone touching a switch. Take a security light and a phone that wakes when you lift it, then say they will build one that listens. Do not open the editor yet.
- **Create a variable:** Say what the variable is for before you make it: the micro:bit needs to remember whether the light is currently on, because a clap has to do the opposite of whatever it is doing now. Without that memory there is no way to turn it off again.
- **Sound threshold:** Name this as the decision about how loud counts as a clap. Say now that the number will probably need changing in a real classroom, so that adjusting it later feels like part of the job rather than a failure.
- **Add the if then else:** Read it aloud as a question with two answers: if the light is not on, turn it on, otherwise turn it off. Ask what would happen with only the first half, and let someone answer that it would never switch off. That is why the else is there.
- **Test it in the simulator:** Everyone can do this step, with or without a device. Drag the sound slider up past the threshold and watch the LEDs come on, then again to turn them off. Have them do it twice so the toggle is obvious rather than accidental.
- **Send the code:** Expect the threshold to need changing once it is in the room, and treat that as the interesting part. Ask what number they tried, what happened, and what they changed it to. That is a real engineering decision made from evidence.
- **Code talk:** Talk only, nothing typed and nothing saved. Ask what set the light off that they did not intend, a chair scraping or a shout, and what that would mean for a real device. Then take one bug and how they found it.

What it should show

Most pupils finish with LEDs that come on at a clap and go off at the next one, working in the simulator, and on a real micro:bit if you have them. Expect the threshold to need adjusting in a noisy room, and expect some false triggers, both of which are worth talking about rather than fixing away. A pupil whose light toggles reliably in the simulator has met the objectives.

Misconceptions & interventions

- **Pupils think the light should stay on while the sound continues, like a microphone level.** — It reacts to the moment the sound crosses the threshold, not to how long it lasts. Have them clap once and watch, then hold a long noise and watch nothing more happen.
- **Pupils leave out the else and cannot understand why the light will not switch off.** — Put the two versions side by side. Without the else the program only ever knows how to turn it on. Ask what instruction is missing before you add it.
- **Pupils treat a false trigger as a bug in their code.** — It is not a bug, it is the threshold. Ask what else in the room is as loud as a clap, then change the number and test again. That is the fix, and it is a design decision.
- **Pupils believe the variable is the light, rather than a note about the light.** — Point at the two separate things: the LEDs are what you see, the variable is what the micro:bit remembers about them. If the note is wrong the next clap does the wrong thing.

Differentiation

Emerging	Developing	Proficient
• Aim for the light coming on at a loud sound. The toggling off is a second idea and can wait. • The simulator's sound slider is easier to control than a real clap, and every part of the learning is there.	• Build the full toggle. Ask them to say what the variable is remembering and why the program needs it. • Change the threshold once, deliberately, and say what difference it made.	• Find the threshold that responds to a clap but not to talking, by testing rather than guessing, and write down the number they settled on. • Ask what else could switch it instead of sound, and what would have to change in the program.

Cross-curricular hook

SPHE and everyday technology. Devices that listen are all around them, and it is worth asking what a device that is always listening would know about a room. Keep it to what this micro:bit does, which is react and remember nothing.