technology

Data with a micro:bit

Lesson at a glance

Open by naming everyday gadgets that sense the world, then model a MakeCode weather-station project on the IWB. Pairs build step by step: two variables, Buttons A/B/A+B and a loud-sound trigger to set mode, then a forever loop that reads temperature, light, sound or movement. They test on a physical micro:bit or the simulator, watching readings change when they cover the light sensor, clap or shake. Close with a display-only Code talk on bugs fixed and one pattern noticed.

Learning objectives

- Program a micro:bit to read a temperature or light sensor
- Gather a small dataset of sensor readings
- Read a simple pattern from the data and say what it shows

Before the bell — prep

Before the lesson, open MakeCode on the class devices and confirm login works. Charge and pair one micro:bit per pair with its USB lead; the on-screen simulator is the fallback if kit is short. Have one demo board ready on the IWB cable.

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 sound tests sensible — no shouting close to a classmate's ears. Handle physical micro:bits and USB leads carefully; no yanking cables from devices.

Lesson 26

Teaching moves

- **What we're building:** Keep the hook short: name weather apps and fitness trackers, then show the micro:bit as today's mini weather station. Confirm every device is on MakeCode and tell pairs who drives first before you model.
- **Create a New Project:** Model New Project on the IWB and name it 'Weather Station'. Scan for pairs still inside an old project or leaving it untitled — a clear name is how they find the work later. Remind them to swap driver after each block of steps.
- **Initialize Variables:** Add reading and mode on the board and say plainly: mode chooses the sensor, reading holds the number. Reassure pairs who see a blank board — this is set-up only. Ask why a program might need to remember a value.
- **Configure Button A:** Wire Button A so mode becomes 1. Watch for the block not fully snapped. Have slower pairs test just this button on the simulator before moving on.
- **Configure Button B:** Let a confident pair predict the Button B code before you show it. Circulate specifically for the bug of dropping the new block inside the Button A handler — the two must sit as separate on-button blocks.
- **Configure Buttons A and B:** Explain A+B as one combined press, a third choice. Ask how that differs from pressing A then B. Check pairs chose the single A+B block, not two separate handlers.
- **Configure Loud Sound Detection:** Model the loud-sound trigger; on physical boards a clap is enough, on the simulator show where to click sound. Remind the class: sensible sound only — no shouting near a classmate's ears.
- **Create a Forever Loop:** This is the key step. Build the forever loop slowly and name it as the bit that keeps checking mode. The main bug is mode checks left outside the loop so nothing repeats — stop and re-voice 'why must this run again and again?' if you see it.
- **Test Your Program:** Pairs press each control in turn and watch the display. Prompt them to breathe on the board, cover the light sensor, clap or give a gentle shake so readings move. Nothing is written here — they say to their partner what changed. A blank bar graph usually means the reading is too low; louder sound or a shake lights LEDs. Handle boards and leads carefully.
- **Code talk — what worked and what we'd change:** Display-only discussion, no typing. Invite one pair to name a bug and exactly how they fixed it; revoice their steps for the class. Draw out one pattern someone saw in the readings and one change they would make next time.

What it should show

Button A should show a temperature number (around room temperature in °C); Button B a light level that drops when the sensor is covered; A+B and loud-sound modes show bar graphs that rise with movement or claps. A blank display in sound or shake mode usually means the reading is too low to light LEDs — prompt a louder sound or firmer shake. A mode that never changes usually means the forever checks sit outside the loop.

Misconceptions & interventions

- **Nothing is happening — the variables must be wrong.** — Point at the board and say this is set-up only; the display comes alive once the forever loop reads mode. Have them finish the loop, then press A and watch a number appear.
- **I put Button B inside Button A, so only one mode works.** — On their screen, trace the two handlers and show they must sit as separate on-button blocks. Drag the nested one out and re-test A then B.
- **The sound or shake mode is broken because the LEDs stay dark.** — Explain a low reading lights no LEDs. Ask for a louder clap or a gentle shake and watch the bar graph fill.

Differentiation

Emerging	Developing	Proficient
• Sit the pair at the teacher table and build only Button A plus the forever temperature branch first; add B and sound once that displays a number. • Allow full use of the simulator so they can click buttons without managing a physical board and lead.	• After a successful run, ask them to cover the light sensor and say aloud what the reading did and why. • Prompt one deliberate bug-fix cycle: break mode on purpose, then talk through how they spotted and fixed it.	• Ask which mode shows a number and which shows a bar graph, and why that choice fits the sensor. • In Code talk, push for one concrete improvement (e.g. a clearer way to switch mode) without adding new kit.

Cross-curricular hook

Link to Maths Data — pupils describe the pattern in the sensor readings they watched, naming high, low and change.