technology

Data with a micro:bit

Lesson at a glance

The class programs a micro:bit into a small weather station, step by step, on their own devices. Two variables hold a mode and a reading; button A, button B, both buttons together and a loud sound each set a different mode; and a forever loop checks the mode and shows the matching sensor, the temperature as a number and the light and sound levels as a bar graph on the LEDs. Pupils then take a few readings around the room and say what the numbers show. Model each piece on the board, let them build and test it, and close on the pattern they 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

One micro:bit per pair if you have them, otherwise the MakeCode simulator does everything in this lesson except a real clap. Build the program yourself once beforehand so you know what each mode looks like on the LEDs. Decide in advance where pupils will take readings, a sunny window, a dark press and a corner of the room is plenty, and have a torch to hand if the day is dull.

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. If micro:bits are being connected to computers, supervise unplugging so the connectors are not pulled sideways. Nothing else on this lesson.

Lesson 26

Teaching moves

- **What we're building:** Hold up a micro:bit, or show the simulator, and ask what it could measure about this room. Take temperature, light and sound, then say they will build one device that does all three and switches between them. Do not open the editor yet.
- **Initialize Variables:** Slow down here. Mode is a number that remembers which sensor we want, and reading is where the measurement is put. Say the four modes aloud as you set them up, because everything after this depends on knowing what mode 1 through 4 mean.
- **Configure Button A:** Run it straight after this block, even though nothing visible happens yet. Ask what changed. The answer, that a number in the micro:bit's memory changed and nothing on screen did, is worth more here than any output would be.
- **Configure Loud Sound Detection:** Point out that this is a different kind of start from the buttons: nobody presses anything, the micro:bit is listening. That is the difference between a button and a sensor, and it is the whole idea of the next lesson too.
- **Create a Forever Loop:** This is the longest block of the lesson and the one to model most carefully. Read it as a loop that keeps asking which mode we are in and shows the matching sensor. Note that temperature shows as a number while light and sound show as a bar graph, so a bar filling the screen is not a fault.
- **Test Your Program:** Give this its full time and send them to the reading spots you chose. Have them write each number down with what was happening at the time, in the journal, as they go. Readings written from memory at the end are no use for the pattern.
- **Code talk:** Talk only, nothing typed and nothing saved. Ask what the highest and lowest readings were and what was happening each time, so the pattern comes off their own numbers. Then take one bug and how they found it.

What it should show

Most pupils finish with all four modes working: a temperature number on button A, a light bar on button B, a sound bar on both buttons, and a fourth mode on a loud sound. Light readings should change sharply between a window and a covered sensor, which is the clearest pattern to talk about. A pupil with two working modes and a few honest readings has met the objectives.

Misconceptions & interventions

- **Pupils expect something on screen the moment a button block is added, and think a silent run means broken code.** — Say that the button only changes a number the micro:bit is holding. Nothing shows until the forever loop is built, because that is the part that decides what to display.
- **Pupils read the bar graph as a number, so a full bar is taken to be a big temperature.** — Show the two outputs side by side. Temperature is written out as digits; light and sound are drawn as a bar out of a maximum. A full bar means as bright as it measures, not a value.
- **Pupils record a reading without recording what was happening, then cannot say anything about the data.** — Insist on the second column as they go. A number on its own says nothing; a number beside where they were standing is evidence.
- **Pupils think the micro:bit is broken when the temperature barely changes around the room.** — That is a real result and worth naming. A classroom is much the same temperature throughout, which is why light is the better sensor for spotting a pattern in one hour.

Differentiation

Emerging	Developing	Proficient
• Aim for buttons A and B with the temperature and light modes working. Two sensors are enough to show a pattern. • Let them work in the simulator if a physical micro:bit is slowing them down; every sensor in this lesson can be driven with the simulator's sliders.	• Build all four modes. Ask them to say which block decides what is shown, so the forever loop is named as the part doing the choosing. • Take readings in at least three different places so there is something to compare.	• Take a reading of the same spot twice, a few minutes apart, and say whether the difference is real or just the sensor wobbling. • Ask what a fifth mode could measure, and what button or sensor would switch to it.

Cross-curricular hook

Maths, through data. The readings are a small dataset with a highest, a lowest and a range, and the bar graph is a chart the device draws for you. Ask which spot they would predict a reading for next, and why.