

LESSON 33

Microbit Traffic Lights

40 minutes

Designing and Building for the Future

WHAT THIS LESSON DOES

In this step-by-step lesson, you'll create a Microbit project, add the Stopbit extension, and test all the lights. You'll learn about sequences in coding and program your traffic lights to display a sequence using on/off and state methods. You'll test your sequences to ensure they're working correctly.

WHAT STUDENTS SHOULD BE ABLE TO DO

- Understand and apply the process of creating a new Microbit project.
- Learn to add and utilise the Stopbit extension for programming traffic lights kit.
- Gain skills in testing and troubleshooting the functionality of the lights.
- Comprehend the concept of 'sequence' in coding and apply it to program traffic lights.
- Develop proficiency in programming the sequence of traffic lights using on/off and state methods.

BEFORE THE LESSON

- Confirm devices can open makecode.com and that the stopbit extension is not blocked by the school filter.
- Have Microbits and Kitronik traffic-lights kits ready, connected, and with mounting screws already checked.
- Plan one kit and device per pupil or per pair so everyone can download and test.

Microbit Traffic Lights

HOW THE LESSON RUNS

- Create a new Microbit project
- Add the Stopbit extension
- Test all the lights
- What is a sequence?
- Program the sequence using on/off
- Program the sequence using state

TEACHING MOVES

- 1. Create a new Microbit project. Open makecode.com on a shared screen and walk the class through starting a new Microbit project so nobody lands in the wrong editor.
- 2. Add the Stopbit extension. Show the search for stopbit and the traffic-lights kit picture. Wait until every toolbox shows the Kitronik STOP:bit category before moving on.
- 3. Test all the lights. After they send the test code, have them press A, B and A+B. If a light flickers, tighten the screws before treating it as a code fault.
- 4. What is a sequence? Read the sock, shoe, laces example aloud, then the green 5s, yellow 1s, red 2s timings, so order is clear before anyone builds blocks.
- 5. Program the sequence using on/off. Remind them to delete the test code first. When the forever sequence is on the Microbit, check green, yellow and red timings together as a class.
- 6. Program the sequence using state. Name the four states (Stop, Get Ready, Go, Ready to Stop) and what each turns on. Recheck the same green, yellow, red timings after they send the new code.

WHAT TO WATCH FOR

- Pupils treat a flickering or dead light as a coding error when a screw is loose or a connection is poor. Before rewriting blocks, retest with the A, B and A+B checks and tighten the kit screws.
- Pupils mix up the order or the pause lengths so yellow or red appears at the wrong time. Have them say the target sequence aloud (green 5, yellow 1, red 2) and match each wait to that list.
- Pupils think on/off blocks and state blocks do different light patterns, so they change timings when switching method. Keep the same forever timings and only swap how each step sets the lights, then compare side by side.

DIFFERENTIATION

- Emerging: Pair them with a partner who already has the stopbit blocks visible and work through the light test one button at a time. Stay with them while they delete old code and snap the first forever sequence so they are not stuck on setup.
- Developing: Ask them to talk through each block against the green, yellow, red timings before they download. If the pattern is wrong, have them check wait lengths and which light is set on or off at each step.
- Proficient: Ask them to explain how the on/off version and the state version both produce the same sequence. Challenge them to match the four named states to what they see on the kit without reading the blocks aloud.