LESSON 9

Autonomous Car

60 minutes (2 periods) Programming Car Behaviours

WHAT THIS LESSON DOES

Discover the world of ultrasonic sensors with your Move Motor car. Learn how these sensors measure distance, and apply this knowledge to create a project that enables your car to follow an object, reverse, and roam freely while avoiding obstacles. Enhance your car's capabilities by adding lights and improving its sensor range.

WHAT STUDENTS SHOULD BE ABLE TO DO

- Understand the function and operation of ultrasonic sensors.
- Develop skills in creating a new project using the kitronik-move-motor extension.
- Acquire the ability to program a sensor to measure distance.
- Learn to code a car to maintain a specific distance from an object.
- Enhance problem-solving skills by programming the car to reverse and maintain distance.

BEFORE THE BELL

- Count out one Move Motor car with ultrasonic sensor, one charged micro:bit, and a USB cable per pair the day before. Power on each car briefly and confirm the sensor is seated.
- Open https://makecode.microbit.org on the school network and confirm you can create a project and add the kitronik-move-motor extension.
- Clear floor space and gather small solid objects (books, boxes, bottles) for distance tests and a simple obstacle course.

Autonomous Car

HOW THE LESSON RUNS

Introduction
Create a new project
Measure distance
Follow an object
Reverse
Free roaming
Improve and add lights

TEACHING MOVES

- 1. Introduction. Hold up a Move Motor car and point to the front sensor. Explain: it sends a high-frequency sound pulse, times the echo, and turns that into distance in centimetres. It does not see shape or colour.
- 2. Create a new project. Get every pair onto makecode.microbit.org, new project, then add the kitronik-move-motor extension. Do not move on until each pair can see the Move Motor blocks.
- 3. Measure distance. Build the distance variable and show-number loop together. Download, power the car, and have pairs hold an object at different distances while calling out the reading on the micro:bit.
- 4. Follow an object. Change the code so the car drives only when distance is over 10 cm and stops at 10 cm. Test with a hand or book pulled slowly away, then held still.
- 5. Reverse. Add reverse when the reading is under 10 cm so the car holds the gap both ways. Have students move the object back and forth and watch for overshoot.
- 6. Free roaming. Code the under-10 cm path: stop, pause 500 ms, reverse 500 ms, turn right 200 ms, stop, then forward if clear. Run one demo car through a small obstacle area before pairs test.
- 7. Improve and add lights. Ask what still fails: side hits, reverse crashes, repeated right turns. Challenge pairs to randomise left or right, add lights, and retest. Link to why road cars use several sensors.

WHAT TO WATCH FOR

- Students treat the ultrasonic sensor like a camera that recognises objects, not a timer that only returns distance. Cover the sensor briefly, then hold different objects at the same distance. Same number means distance only, not identity.
- Students assume a successful download means the behaviour is correct, so they stop testing. Ask them to prove the 10 cm rule with a ruler: move the object and say whether the car matched the gap before changing more code.
- Students expect the sensor to protect the back and sides the same way it protects the front. On a reverse or angled approach, pause the class and show the single forward-facing sensor. That is why free-roam still clips obstacles.

DIFFERENTIATION

- Emerging: Stay with the pair until the extension is added and the live distance number appears on the micro:bit before any driving code. Pair them with a group that already has distance displaying and have that group talk through one block at a time.
- Developing: Have them change and test only one behaviour at a time (stop at 10 cm, then reverse, then turn) rather than pasting the full free-roam stack. Ask them to read the distance value aloud during a test so they link the number to the car's move.
- Proficient: Once free-roam works, have them randomise left or right turns, add lights, and compare two versions on the same obstacle layout. Ask them to explain to you why a single front sensor still fails on reverse and what a real car adds.