

LESSON 13

Robot Car Claw

60 minutes (2 periods)

Integration Projects

WHAT THIS LESSON DOES

In this step-by-step lesson, you'll learn how to program a Robot Car Claw. Starting with setting up your Move Motor Car, you'll create a new project on the Microbit website, add the necessary extension, and understand how servos work. You'll then program the claw to close and open, and adjust the claw's grip. Finally, you'll be encouraged to explore other ways to program and use the claw.

WHAT STUDENTS SHOULD BE ABLE TO DO

- Understand the setup and requirements for a Move Motor Car with a Move Motor Klaw and a Microbit.
- Develop skills in creating a new Microbit project using the provided website.
- Gain knowledge about the kitronik-move-motor extension and how to add it to the project.
- Comprehend the functioning of servos and their application in controlling the Move Motor Klaw.
- Acquire the ability to program the claw to open and close using specific code blocks.

BEFORE THE BELL

- Count out Move Motor Cars with the Move Motor Klaw already attached and a micro:bit seated in each. One car per pair is enough. Charge batteries and check cars the day before.
- Confirm each Klaw is plugged into the SERVO 1 socket on the car (the code drives servo 1). Label any that are on SERVO 2 so you can fix them quickly.
- Open <https://makecode.microbit.org/> on the school network and confirm students can create a project and download .hex files.
- Have USB cables ready for flashing, and a spare micro:bit and cable for swaps if a download fails.

Robot Car Claw

HOW THE LESSON RUNS

Setup

Create a new project

Add the extension

How servos work

Close the claw

Open the claw

Open it on start

Close it gradually

Open it gradually

Explore

TEACHING MOVES

- 1. Setup. Do a quick kit check before anyone codes: Klaw fixed on, micro:bit fully seated, battery on. A car that is not ready wastes the whole build.
- 2. Create a new project. Send everyone to makecode.microbit.org. Have them start a new micro:bit project and give it a clear name so downloads do not get mixed up.
- 3. Add the extension. Show Extensions, search kitronik-move-motor, and add it once on the board. Wait until every pair can see the Move Motor blocks before you continue.
- 4. How servos work. Say it simply: a servo turns its shaft to an angle you send as a number. On this Klaw, that angle opens or closes the two pinchers.
- 5. Close the claw. Build write servo 1 to 0 on button A together. Download to the car, power on, press A. Do not move on until the claw actually closes on hardware.
- 6. Open the claw. Add write servo 1 to 150 on button B. Flash again and test B on the car. Simulator success is not enough; the physical claw must open.
- 7. Open it on start. Create the servoValue variable and set it on start so the claw opens to a known gape when the car powers up. Stress why a variable beats hard-coded one-shot angles.
- 8. Close it gradually. Rework A to subtract 10 from servoValue inside the if-else that stops at 0. Ask why a floor matters before they flash.
- 9. Open it gradually. Rework B to add 10 with a ceiling of 150. After download, have pairs press A and B repeatedly and watch the claw step, not jump.
- 10. Explore. Ask aloud what else they could drive with the claw or which inputs they might try next. Keep ideas spoken; nothing to type or hand up.

WHAT TO WATCH FOR

- Students treat a successful download as proof the claw works, without powering the car and pressing the buttons. Make the checkpoint physical: car on, press A or B, watch the pinchers. No next step until the hardware moves.
- The Klaw is on SERVO 2 while the blocks write to servo 1, so nothing moves and they blame the code. When a car does not respond, check the socket label first. Move the plug to SERVO 1 before changing any blocks.
- They change servoValue with no lower or upper limit, so the servo is sent angles outside 0 to 150 and the claw stalls or judders. Point back to the if-else floors and ceilings. Have them say the new value aloud before each test press.

DIFFERENTIATION

- Emerging: Seat them with a pair whose extension is already added and whose claw responds on A and B, then have them recreate the close block on their own project. Stay for the first download: confirm SERVO 1, power on, and the A press together so they see one clean success.
- Developing: Ask them to explain why A subtracts 10 only when servoValue is above 0 before they flash the gradual version. If open and close work but start position is wrong, point them back to the on start value rather than adding new blocks.
- Proficient: When gradual open and close work, push the Explore prompt: different step sizes, a tighter max gape, or a sequence that closes then opens without new kit. Have them teach another pair why 0 and 150 are the limits on this Klaw, using the working car as the demo.