

LESSON 7

Microbit Voting System

60 minutes (2 periods)

Intermediate Projects

WHAT THIS LESSON DOES

In this lesson, you'll build a voting system with microbits, using multiple voting units and one central unit. Start by creating two projects on the MakeCode Microbit website to programme the distinct functionalities for each type of microbit.

WHAT STUDENTS SHOULD BE ABLE TO DO

- Develop a microbit voting system with separate voting and central microbits.
- Programme the voting microbits to cast a single 'Yes' or 'No' vote.
- Configure the central microbit to receive votes, count them, and reset the voting system.
- Implement a reset function on the voting microbits to allow for multiple rounds of voting.
- Enhance the voting system by adding a security feature to ensure the integrity of the votes.

BEFORE THE BELL

- Open MakeCode for micro:bit on the school network and confirm students can create and download two separate projects.
- Count and charge micro:bits the day before: ideally several voting units plus one central unit, with USB cables for each. If stock is tight, plan pairs and use the simulator for extra voters.
- Have a simple seating plan: which micro:bit is central, which are voters, so downloads do not get mixed up.

Microbit Voting System

HOW THE LESSON RUNS

- Create a Project
- Set up the Voting Buttons
- Set up the Vote Receiver and Reset
- Receive the Reset Signal
- Display the Vote Results
- Test Your Enhanced Voting System
- Enhance your project

TEACHING MOVES

- 1. Create a Project. Get every student or pair to create two named MakeCode projects, Voting and Central, before any blocks. Say aloud: voters only send and lock; the central only counts and resets.
- 2. Set up the Voting Buttons. On the Voting project, build `voted = false`, radio group 1, and show ?. Gate both A and B so a second press does nothing once voted is true. Read the send strings aloud: yes and no.
- 3. Set up the Vote Receiver and Reset. On the Central project only, add `yesCount` and `noCount` at 0 and the same radio group 1. On receive, branch on the string yes or no. Wire A+B to zero the counts and broadcast reset.
- 4. Receive the Reset Signal. Back on every Voting project, handle received reset: set voted false and show ? again. Download that build to all voting micro:bits before anyone tests.
- 5. Display the Vote Results. On the central micro:bit, run a forever display of Yes and No counts so the room can see the tally update live. Download only to the central device.
- 6. Test Your Enhanced Voting System. Run a live vote: each voter presses once, central tallies, then A+B resets. Call out failures: second votes, wrong group, or reset not clearing ?. Simulator broadcast is fine if kit is short.
- 7. Enhance your project. Ask: who could cheat this system? Pick one idea from the lesson (PIN, receive check, or vote-total display) and give ten quiet minutes to add it, then demo one secure build.

WHAT TO WATCH FOR

- Students put Voting and Central code in one project, or use different radio groups, so messages never arrive. Before download, hold up two project tabs and chant group 1 on both. Do one send-receive smoke test with a single pair first.
- They treat a button press as a successful vote even when the central count does not change, or allow a second vote because voted is never set true. Watch the central display, not the voter LED. Trace: press, set voted true, send. A second press must hit the voted gate and do nothing.
- Reset clears the central counts but voting micro:bits stay locked, or the reverse. On A+B, require both: central zeros counts and broadcasts reset; each voter must show ? again before the next round.

DIFFERENTIATION

- Emerging: Pair them with a working build and have them own only the voted gate on A and B while the partner handles radio. Use the simulator broadcast of yes, no, and reset so they can see receive logic without juggling multiple handsets.
- Developing: Give a quick checklist: same group, single send, voted lock, central count, reset restores ?. They tick it during the class test. Ask them to explain one failed second-vote attempt out loud before they change any code.
- Proficient: Challenge them to add one security feature from the lesson list and show that a double vote or spoofed message is harder. Have them run two full rounds and report whether the displayed totals always match the number of physical voters.