LESSON 100

# Testing and Documentation

160 minutes (4 periods) | Applied Learning Tasks (ALTs) | Outcome 1.11 | Outcome 1.19 | Outcome 1.23 | Outcome 2.19 | Outcome 2.20 | Outcome 2.21 | Outcome 2.22

## WHAT THIS LESSON DOES

In this lesson, you'll explore how to test and document your embedded systems project. Learn to conduct usability tests, gather peer feedback, structure a detailed report, and create a digital portfolio. Use the self-assessment rubric to evaluate your work.

## CURRICULUM OUTCOMES

| Outcome | What the specification asks for |
|---|---|
| **1.11** | discuss the complex relationship between computing technologies and society including issues of ethics |
| **1.19** | identify features of both staged and iterative design and development processes |
| **1.23** | reflect and communicate on the design and development process |
| **2.19** | test solutions and decisions to determine their short-term and long-term outcomes |
| **2.20** | identify and fix/debug warnings and errors in computer code and modify as required |
| **2.21** | critically reflect on and identify limitations in completed code and suggest possible improvements |
| **2.22** | explain the different stages in software testing |

## WHAT STUDENTS SHOULD BE ABLE TO DO

- Understand the importance of usability testing in ensuring embedded systems are user-friendly and functional.
- Develop skills in gathering and applying feedback through peer reviews to improve project outcomes.
- Master the structure and content of a comprehensive project report, including ethical considerations.
- Acquire the ability to organise and present project work effectively in a digital portfolio.
- Enhance self-assessment capabilities using a rubric to evaluate and reflect on personal project performance.

### BEFORE THE LESSON

- Confirm every team has a working embedded system to test, and that a few devices with cables and power are set up so peers can actually try each project.
- Check students can reach the digital portfolio destination the class uses and are logged in, so upload time is not lost to accounts.
- Have the self-assessment rubric visible on screen or printed so students can score against it during the final activity.

# Testing and Documentation

### HOW THE LESSON RUNS

| |
|---|
| Introduction |
| Usability Testing |
| Peer Review Activity |
| Report Structure Guidance |
| Digital Portfolio Upload |
| Self-Assessment Rubric |

### TEACHING MOVES

- 2. Usability Testing. Insist students watch a peer use the system without coaching them. The point is to see where a real user hesitates or the sensor misbehaves, not to defend the design. Have them note faults before fixing anything.
- 3. Peer Review Activity. Structure the feedback so it is specific and kind: one thing that works, one usability issue, one ethical or safety point. Vague praise helps no one. Ask reviewers to name the exact input or output that failed.
- 4. Report Structure Guidance. Walk through the report sections aloud, stressing that the design process and ethics carry marks. Remind students this is Applied Learning Task work, formative practice for the report skills they need in the assessed Coursework Project.
- 5. Digital Portfolio Upload. Have students check each file opens after upload, not just that it appears. A portfolio with a corrupted or empty file reads worse than a missing one. Ask them to name files clearly so evidence maps to the report.
- 6. Self-Assessment Rubric. Make students justify each score with one sentence of evidence, not a number alone. Push honest low scores: a self-identified weakness with a fix is stronger than an unearned five across the board.

### WHAT TO WATCH FOR

- Students think a system that works when they operate it has passed usability testing. Point out that they know exactly which button to press and how hard. Testing means someone who has never seen it, so a genuine peer trial reveals problems the builder is blind to.
- Students treat ethics as a paragraph to bolt on at the end rather than something in the design. Ask concrete questions about their actual build: could a sensor fault harm a user, can everyone reach the controls, what happens on power loss. Ethics becomes specific and easy to write once tied to the device.

**DIFFERENTIATION**

- Emerging: Give a fixed feedback template with three prompts so a student who is unsure what to look for still gathers usable review points. Provide the report section headings as a checklist so getting started is filling in structure, not facing a blank page.
- Developing: Ask them to trace one piece of peer feedback all the way through to a change in the system and a line in the report, so they see the iterative loop end to end.
- Proficient: Have them write the ethics section as they would for the assessed Coursework Project, weighing a real trade-off in their design rather than listing generic concerns. Ask them to review a weaker peer's portfolio against the rubric and give a marked-up score, which sharpens their own self-assessment.