

GREEN SHOE GARAGE

TRADIGITAL DESIGN + ENGINEERING

FIELD GUIDE / INSTRUCTOR EDITION 1.0



PRODUCT DEVELOPMENT BOOT CAMP

Instructor manual

A three-day course covering design, mechanical development, electronics, software, testing, and preparation for low-volume production.

3 DAYS / 18 INSTRUCTIONAL HOURS

Live online or in person. Recommended cohort of six to eight.

This manual provides the teaching sequence, practical exercises, worked answers, participant BenchBook forms, and assessment criteria for delivering the course. Use it with the companion teaching pack and a rehearsed reference hardware kit.

COURSE OUTCOME

An integrated learning prototype and a documented plan for the next stage of development.



Contents

Course delivery	3
Instructor preparation	4
Classroom and online delivery	5
Practical lab boundaries	6
Equipment and materials	7
Reference product and requirements	8
Electrical connections	9
Reference build commissioning	10
Day 1 morning product definition	11
Day 1 afternoon mechanical development	13
Day 2 morning electronics development	15
Day 2 afternoon software and integration	17
Day 3 morning testing and improvement	19
Functional test procedures	20
Fault cards and corrective action	21
Day 3 afternoon first batch preparation	22
Production handoff and readiness	23
Worked first batch cost example	24
Assembly flow and supplier preparation	25
Assessment and completion	26
Troubleshooting and fallback guide	27
Course closeout	28
Participant forms W01 through W12	29
Knowledge check and discussion answers	41
Worked answers and assessment examples	42
Technical references and terminology	43

Participant forms are reusable pages. Instructor answers follow the forms. Timings include two 15 minute breaks and a 60 minute lunch; these are excluded from the 18 instructional hours.



Using this manual

This manual equips a Green Shoe Garage instructor to deliver a three-day product development boot camp online or in person. The course follows one USB-powered workshop monitor through product definition, mechanical design, electronics, firmware, integration, testing, and planning a first batch of 25 units. Teach the connections between those activities as well as the individual techniques.

Course parameters

Parameter	Teaching baseline
Duration	18 instructional hours across three days
Daily schedule	9:00 am to 4:30 pm in the published course timezone
Breaks	10:30 to 10:45 am; lunch 12:15 to 1:15 pm; 2:45 to 3:00 pm
Cohort	Six to eight adults; one lead instructor and a recommended lab assistant
Online delivery	Three full days, preferably one week apart; same instructional hours
Entry level	Comfortable with files and installing software; introductory CAD and coding support supplied
Completion	Participation and evidence criteria on the assessment page; certificate of completion only

What participants should be able to do

- Write measurable requirements and trace them to a test or inspection.
- Explain a product architecture and the interfaces between mechanical parts, electronics, software, and the user.
- Modify an enclosure, assemble a prepared circuit, and change a small firmware behavior.
- Investigate an observed failure using evidence and repeat a test after a change.
- Prepare an initial bill of materials, assembly sequence, acceptance test, and first-batch estimate.
- Describe the unresolved work between a classroom prototype and a product released for sale.

How the teaching materials fit together

The lesson pages specify timing, instructor actions, prompts, expected evidence, and recovery options. Forms W01 through W12 are reproducible participant BenchBook pages. The answer pages are instructor references. The companion teaching pack contains the reference firmware, editable enclosure source, simulated datasets, and a preparation record. Use those files in the lessons; they are not a substitute for the instructor's hardware rehearsal.

Use the common project for assessed practical work. Let participants apply each day's decisions to their own idea in W12. Keep proprietary details out of group discussion unless the owner chooses to share them. A company-specific course can use a different project after the instructor adapts and rehearses every dependent exercise.

Teaching boundary. Participants finish a learning prototype and a draft production plan. Commercial release requires additional engineering, qualification, sourcing, and compliance work appropriate to the product. The reference monitor reports classroom conditions and is not a safety alarm or calibrated measurement instrument.



Instructor preparation

Four to six weeks before the course

1. Confirm the delivery mode, dates, timezone, cohort limit, venue or meeting platform, and named instructor. Recruit a lab assistant for hardware support and online chat if possible.
2. Order the selected kit and at least two spare complete electronics sets for eight participants. Confirm the actual breakout revision, connector type, button dimensions, and breadboard size before fabricating mechanical parts.
3. Build the reference unit from the same parts participants will receive. Run every lesson and test. Time the slowest assembly steps and keep photographs of known-good wiring.
4. Print one base and lid from the supplied CAD source. Check fit with the actual boards, cable, mounting pads, fasteners, and access opening. Revise and record the mechanical revision before producing the class set.
5. Select and record the Arduino IDE, UNO R4 board package, Adafruit SHT31 library and dependencies, and OpenSCAD versions that pass rehearsal. Freeze that combination for the cohort. Keep permitted offline installers and a recovery copy.

Two weeks before the course

Send a participant preparation sheet containing the schedule, equipment list, installation instructions, kit inventory, and support contact. Ask about access needs, computer operating system, administrative restrictions, hardware experience, and an optional product idea. Ask only for information needed to support attendance.

For online participants, dispatch kits with enough time for a replacement delivery. Confirm receipt and run a brief connection check. The check must establish that the laptop can upload a sketch and display serial data. An untested installation is not ready just because the download succeeded.

One week before the course

- Complete the cohort release record in the teaching pack. Record hardware revisions, software versions, test date, known limitations, and the instructor who performed the checks.
- Prepare a working unit, a disassembled unit, an enclosure comparison, and the fault cards. Label the reference unit so it is not inadvertently modified.
- Duplicate W01 through W12 for each participant. Keep instructor answers separate until debrief.
- Test projector or screen sharing at the size participants will actually see. Verify captions and legibility of the wiring close-up.
- Prepare a local copy of all course files. Do not depend on a live download during the build.

The day before and each morning

Count kits and spares; run the reference unit; check camera focus; open the day's files; reset demonstration settings; and prepare a visible help queue. Confirm emergency arrangements with the venue. Online, verify that participants know how to disconnect their own kit and rejoin after a connection loss.

Release decision. Deliver the hardware labs only after the selected kit passes the preparation record. If a part or software version changes, repeat the affected build and tests. Record a substitution rather than silently treating it as equivalent.



Managing the classroom and online cohort

In person

Arrange individual work areas with a clear aisle and a separate instructor demonstration bench. Give each participant a laptop, kit, labeled parts tray, and space to spread a drawing beside the hardware. Keep food and drink away from electronics. Place cutting, drilling, and other fabrication demonstrations in a supervised area separate from powered circuits.

Use a document camera for close work. A projector view of the actual component orientation is more useful than asking everyone to crowd around the bench. Have an assistant verify wiring while the lead instructor maintains the lesson flow. If teaching alone, inspect kits in a queue and give waiting participants the documentation task.

Live online

Use a main camera, a close-up bench camera, and screen sharing. Ask participants to place a phone or camera above the kit where practical; provide a photo-upload alternative. Explain how to pin the bench view, request help, and share a single application window. Enable captions and give written steps before opening breakout rooms.

Keep the electrical build solderless for remote delivery. Ship boards with required headers already installed and checked. Every participant needs a known-good data cable and basic multimeter. Confirm whether the computer can install drivers and access USB devices before enrollment is finalized.

Use paired breakout rooms for design reviews, not for unsupervised corrective wiring. An assistant can handle a support room while the lead teaches. For a single instructor, pause at shared checkpoints and use the dataset fallback when one connection problem would otherwise stall the cohort.

Facilitation routines

- Start each module with an observable outcome and end with the associated BenchBook evidence.
- Show a completed example briefly, then demonstrate only the next step. Keep the full answer until the debrief.
- Rotate builder and reviewer roles during paired work. Each participant keeps their own record and explains at least one decision.
- When helping, ask what was expected, what was observed, and what evidence supports the next check. Avoid taking over the keyboard unless the participant requests it.
- Allow five minutes of focused recovery before switching to a prepared fallback. Return to the unresolved issue during a break or supported practice period.

Access and pacing

Provide editable documents, selectable text, high-contrast diagrams, and text labels alongside LED colors. Offer larger printed diagrams, adapted grips, or an assistant for fine motor tasks when needed. Accept an oral explanation or dictated record where handwriting is a barrier. Preserve the learning objective when adapting the method.

The minimum pathway is one measurable requirement, one CAD change, one inspected circuit, one firmware change, one retest, and one production decision. Advanced learners can examine measurement bias, design a test fixture, or compare manufacturing methods after completing that pathway. Never add stretch tasks to the assessed core without adding time.



Practical lab boundaries and incident response

Scope of classroom work

Use USB power through the board's USB connector only. Keep mains wiring, batteries, motor drives, heaters, and hazardous loads outside this reference build. These boundaries keep the labs focused on product development and make online supervision practical. Instructor-led fabrication demonstrations require the venue's normal training and equipment controls.

Before anyone powers a circuit, demonstrate unplugging the USB cable, identifying the supply and ground rails, checking LED polarity and series resistors, and inspecting for stray conductors. Use the multimeter in voltage mode across the intended points. Explain that current mode requires a different connection and can short a supply if placed directly across it; current measurement is instructor-led with a suitable USB meter in this course.

Before power checklist

- USB disconnected while making or changing connections.
- Correct board and breakout identified from their printed labels.
- Common ground present; supply polarity checked against the connection table.
- No bare wire bridging adjacent contacts; LED resistors installed.
- Breadboard rail continuity understood; a printed line is not proof of continuity.
- Circuit rests on an insulating surface and cable movement cannot sweep loose metal across it.
- Instructor or assistant initials W05 after inspection.

If a fault looks hazardous

If a participant reports unusual heat, smoke, odor, a damaged cable, or repeated USB protection trips, have them disconnect power if safe, leave the assembly unpowered, and alert the instructor. Do not repeatedly reconnect to see whether it recovers. At the venue, follow its incident procedures and use the designated first-aid or emergency response. Remotely, the participant controls their own workspace and follows local emergency arrangements.

Quarantine the affected kit, record what happened, and use a spare or dataset for the lesson. Do not ask a remote learner to troubleshoot damaged powered equipment. Record equipment failures separately from assessment; a failed kit is not a failed learner.

Approved fault exercises

Use software commands to simulate alert and invalid-data states. Hardware disconnection exercises are performed with power removed, followed by an inspected restart. Never deliberately short a supply, reverse power, overheat a sensor, or hot-plug bare wires as a teaching demonstration. Do not use a hot-air gun, flame, steam, or liquid to cross a temperature threshold.

Data and recordings

Share only the course folder when screen sharing. Use the supplied fictional user scenario for recorded discussions. Obtain an explicit recording agreement before recording participant voices, images, or project discussions. Provide a non-recorded participation route. Store participant feedback and contact details only for the agreed course purpose.



Equipment and teaching materials

Per participant kit

Item	Quantity	Selection and preparation
Arduino UNO R4 Minima	1	Use the same board model for the whole cohort
Adafruit SHT31 D breakout	1	Product 2857 or its identified revision; headers fitted and checked
USB data cable	1	Correct laptop connector to USB C; verify data transfer
Half size breadboard	1	Measure actual body; inspect rail continuity
Green and amber LEDs	1 each	Low-current visible indicators; label NORMAL and STATUS
1 kilohm resistors	2 plus 2 spares	One series resistor for each LED
Normally open pushbutton	1	Identify terminal pairs before use
Jumper wires	About 20	Correct ends for the supplied headers and breadboard
Enclosure base and lid	1 each	Preprinted and checked against the chosen kit
M3 hardware	4 sets	Nominal M3 by 55 mm screws, washers and nuts; verify the fitted stack
Insulating mounting material	1 set	Removable pads or mounting plate that protect exposed solder joints
Multimeter and small tools	1 set	Voltage and continuity functions; screwdriver and ruler or caliper

Class resources. For eight learners, prepare ten complete electronics kits, ten sets of cables, at least one spare enclosure, two instructor meters, a suitable USB power meter, an independent temperature reference for comparison, a document camera, labels, tape, paper, and a stopwatch. A 3D printer is optional during class because all required parts are prepared beforehand.

Software and files

Install Arduino IDE and its UNO R4 board support, the Adafruit SHT31 library with required dependencies, OpenSCAD, and a spreadsheet or text editor for CSV files. Version control can use Git if already installed; otherwise use the release record and clearly named revision folders during the core lesson. Demonstrate Git without making an account or cloud repository a course prerequisite.

The teaching pack contains `workshop_monitor.ino` and `monitor_logic.h` in the same sketch folder; `workshop_enclosure.scad`; the sensor comparison and state sequence datasets; a worked cost CSV; a blank cohort release record; and a README. The manual contains the participant forms and instructor answers.

Substitutions and purchasing

The kit is a teaching baseline, not a vendor endorsement or a confirmed purchase list. Price the actual cart, shipping, taxes, fabrication, spare allocation, and instructor preparation before setting tuition. A substitute sensor may require changes to voltage, wiring, library, geometry, and tests. A mechanically similar board is not automatically electrically interchangeable.



Reference product and measurable requirements

Product brief

The workshop monitor helps a person at an indoor workbench notice a selected temperature condition and inspect temperature and relative-humidity readings on a connected computer. It uses two labeled LEDs, a button event counter, and USB serial output. Temperature thresholds are classroom examples, not health, storage, or equipment protection limits.

The baseline has no wireless connection, cloud account, battery, actuator, buzzer, or independent screen. The serial monitor is its numeric interface. Those features can become later design exercises after the core product is working.

Reference requirements

ID	Baseline requirement	Evidence
R01	Operate from USB through the board connector and produce a valid record within 5 seconds of reset when the sensor is working	Timed cold-start test
R02	Output timestamp in milliseconds, source, temperature C, relative humidity percent, state, button count, and firmware ID; nominal sample interval 1 second	Ten-record log inspection
R03	Enter ALERT at temperature at or above 28 C; remain there until temperature is at or below 27 C	Boundary tests of logic and simulated sequence
R04	Display FAULT for invalid sensor data, omit numeric readings in that record, and leave FAULT after a valid reading	Simulated fault and physical absence tests
R05	Count one event per deliberate button press; ten presses of at least 100 ms separated by at least 200 ms produce ten events	Counter comparison
R06	Provide readable NORMAL, ALERT and FAULT states using labels plus distinct light behavior	Peer use task and state observations
R07	Close the enclosure without pinched wiring; preserve cable access and expose the sensor to room air	Inspection and assembly check
R08	Save a traceable mechanical revision, firmware version, BOM and test record	Handoff review

Treat R01 through R08 as proposed classroom acceptance criteria. The instructor qualifies them on the actual kit before teaching; participants record pass, fail, or not tested. For R02 use a classroom interval band of 0.9 to 1.2 seconds between normal records, measured by device timestamps. Investigate outliers rather than silently deleting them.

Architecture and boundaries

Power travels from USB to the controller and sensor. The sensor communicates over I2C. The controller reads the button and drives each LED through a resistor. Firmware produces serial records for the computer. The enclosure constrains cable paths, component access, and sensor exposure. Ask participants to identify an interface owner whenever they change one of these paths.

The accuracy of the sensor component does not establish the accuracy of the assembled product. The course comparison identifies potential bias and setup problems; it does not certify calibration. [S6]



Electrical connections and power checks

Use the UNO R4 Minima at its documented 5 V logic level and the identified Adafruit SHT31 D breakout. Its default I2C address is 0x44. The board's SDA and SCL labels correspond to A4 and A5 on this reference controller. Consult the exact pinout before substituting parts. [S1 S2 S3]

From	To	Inspection note
UNO 5V	Sensor VIN	This is the regulated board supply; use USB as the power input
UNO GND	Sensor GND and common breadboard ground	All signal circuits share this reference
UNO SDA or A4	Sensor SDA	Do not confuse SDA with SCL
UNO SCL or A5	Sensor SCL	Keep wiring short and tidy
UNO D8	1 kilohm resistor then green LED anode	LED cathode returns to GND
UNO D9	1 kilohm resistor then amber LED anode	LED cathode returns to GND
UNO D2	One side of normally open button	Other side returns to GND; firmware uses INPUT_PULLUP
Sensor ADR RST ALR	Leave at board defaults	Do not connect unused pins for the reference build

Explain the circuit

The resistor and LED are in series; the resistor may be on either side of the LED in that branch. For an illustrative 2 V LED drop, a 5 V output and 1 kilohm resistor imply about 3 mA. Actual voltage and brightness vary. This deliberately modest load is below the UNO R4 Minima's documented 8 mA maximum per pin; also respect aggregate limits. [S2]

The button input is normally pulled high. Pressing connects it to ground, producing a low reading. On common four-leg tactile switches, two pins may already be connected internally. Use an unpowered continuity check to identify the contact pairs; do not infer the switching direction from appearance alone.

Safe inspection sequence

1. Lay out the unpowered controller, sensor, breadboard, LEDs, resistors, and button. Identify every printed pin label.
2. Make the supply and ground connections first, then the sensor signal connections, then the LED and button branches.
3. Trace each connection against the table, one row at a time. Check both ends rather than wire color alone.
4. Verify intended continuity with USB disconnected. Capacitors and electronics can affect meter readings; do not treat a momentary beep as a complete short-circuit diagnosis.
5. Obtain the instructor's check, connect USB, and verify the sensor is detected before fitting the enclosure.

Illustration instruction. Draw the same circuit as functional blocks, then as labeled electrical connections. Explain that a schematic describes electrical relationships while a wiring view helps locate physical connections. Neither replaces inspection of the actual assembly.



Commissioning the reference build

Software setup

Install the Arduino board package that includes UNO R4 Minima and select that board and the observed USB port. Install Adafruit SHT31 and its requested dependencies through the library manager. Open the supplied workshop_monitor sketch with monitor_logic.h beside it. Compile and upload. The library's public interface provides the sensor initialization and reading operations used by the sketch. [S4]

Open Serial Monitor at 115200. Reset the board if the header has already scrolled past. A normal record resembles 1000,LIVE,24.00,45.00,NORMAL,0,gsg-bootcamp-1.0; the numeric values are illustrative. Confirm that source is LIVE before treating a record as a measurement.

Commands and expected behavior

Command	Input used on subsequent samples	Expected result
n	Simulated 24 C and 45 percent RH	NORMAL; green on
a	Simulated 29 C and 45 percent RH	ALERT; amber continuously on
h	Simulated 27.5 C and 45 percent RH	ALERT if already alert; otherwise NORMAL
c	Simulated 27 C and 45 percent RH	NORMAL; clear the alert
f	Simulated invalid reading	FAULT; amber flashes; numeric fields are NA
l	Real sensor reading	LIVE source; state follows current reading

Commands are single lower-case letters. A carriage return or newline is ignored. Wait for the next record after each command. A reset returns the firmware to live sensing and clears the button count. The button records an event; it does not acknowledge or suppress an alert.

Baseline rehearsal

Run n, a, h, c, f, n, then l. Save the log. Perform ten deliberate button presses and confirm a ten-count increase. Disconnect USB, remove the sensor connector, inspect, and restart; expect FAULT. Disconnect power, restore the sensor connection, inspect, and restart; expect valid live readings. This verifies absence at startup. Live fault simulation tests the software path without disturbing powered wiring.

Measure cold-start behavior and record intervals with the actual library version. Sensor transactions are synchronous and can briefly occupy the loop; the implementation is suitable for this teaching exercise, not a guarantee of real-time response under every bus fault. If the bus locks or the program becomes unresponsive, disconnect, inspect, restart, and record the limitation for the production discussion.

Mechanical baseline

The supplied model uses a 190 by 120 by 45 mm base, 2.5 mm walls, a 3 mm floor and 3 mm lid. Four corner posts accept through-bolts. The lid includes vents and an access opening. Use insulating mounts; the source does not contain a controller-specific mounting pattern. Qualify hole fit, screw length, cable clearance, control access, and sensor position using actual parts before making class sets.



Day 1 morning product definition and design

Outcome. Participants turn a user problem into a small set of measurable requirements and choose a concept with an explained tradeoff. Use W01, W02 and W03. Prepare the reference unit, two concept sketches, blank paper, and a deliberately vague requirement: “The monitor must be easy to use.”

Time	Instructor action	Learner action
9:00 to 9:20	Introduce the course, outcomes and reference product; collect prior experience	State one goal and inspect the product
9:20 to 9:45	Present the user scenario and model a short interview	Complete W01 in pairs
9:45 to 10:15	Teach measurable requirements and constraints	Draft five requirements in W02
10:15 to 10:30	Connect each requirement to evidence	Add a verification method and acceptance criterion
10:30 to 10:45	Break	Break
10:45 to 11:20	Demonstrate concept sketching and user interaction	Sketch two concepts and a user sequence
11:20 to 11:55	Explain weighted selection and uncertainty	Compare concepts in W03
11:55 to 12:15	Facilitate a short design review	Present the selected concept and biggest assumption

Opening script

“Over these three days you will make decisions about a whole product. Some decisions will improve one part and make another harder. Keep a record of why you chose something and what evidence would make you change your mind. At the end, someone else should be able to understand your build and identify what still needs work.”

Teach these ideas

Start with a person and a task. A feature list does not explain the need. Separate user needs, measurable requirements, constraints, and proposed solutions. “Notice a warm condition while working” is a need; a threshold and response-time criterion are requirements; “use this kit” is a course constraint; a flashing LED is one possible solution.

Explain design as the shaping of usefulness, interaction, appearance, service access, and manufacturing choices. Have participants sketch the user approaching the monitor, interpreting it, connecting a computer, and recovering from a fault. Ask how a person with limited color discrimination knows the state.

Introduce three prototype questions: does the principle work, does it fit together, and can the intended user operate it? A single prototype need not answer all three equally well. Identify which uncertainty each activity is intended to reduce.

Checkpoint and recovery

Accept a brief naming the user, setting, problem, success measure, and excluded functions; five testable requirements; and a justified concept. If a team is behind, supply R01 through R08 and have them rewrite one user-facing requirement in their own words. Preserve the selection discussion rather than spending the entire morning polishing sketches.



Product definition exercise and debrief

Scenario for the interview

Use this fictional customer: “I work at a shared indoor bench. I want to notice when the temperature crosses a limit I have chosen. I can connect a laptop when I need numbers, but I do not want to create an account. I need to understand whether the device is measuring successfully. I would like to build a small batch for other benches.”

One learner plays the user and another asks questions. After six minutes, switch roles. The user may invent details but must mark them as assumptions. The instructor asks: What action follows an alert? Where will the device sit? How far away is the user? What does an unavailable reading look like? What evidence would justify building 25 units?

Activity steps

1. Write a one-sentence problem statement without naming a sensor, microcontroller, fabrication method, or app.
2. List three assumptions requiring evidence. Select one to investigate first and explain why it matters.
3. Write five requirements using a measurable behavior, condition, and pass criterion. Include one user interaction, one interface, one mechanical requirement, and one fault behavior.
4. Sketch a desktop box and an alternative arrangement such as an angled stand or sensor pod. Show cable entry, user viewing direction, button access, and room-air exposure.
5. Select four decision criteria, assign weights totaling 100 percent, score the concepts from 1 to 5, and discuss whether uncertain scores could change the result.
6. Deliver a two-minute design review: problem, selected concept, one rejected alternative, and one unresolved assumption.

Worked selection example

Criterion	Weight	Flat desktop box	Angled stand
Readable state from the bench position	35 percent	3	5
Ease of assembly	25 percent	5	3
Access for repair	20 percent	4	4
Low-volume fabrication effort	20 percent	5	3
Weighted total out of 5	100 percent	4.10	3.90

Scores are illustrative judgments, not test results. The flat box leads under these weights. If readability becomes more important, the angled option may win. Have learners identify which score needs a physical mockup rather than more discussion.

Instructor answers and prompts

An improved usability requirement might be: “Given the labeled reference unit, a new user identifies NORMAL, ALERT and FAULT correctly in three presented scenarios within ten seconds each without verbal coaching.” This supports a classroom check; three trials with a peer do not establish usability for the target population.



Day 1 afternoon mechanical development

Outcome. Participants make a documented enclosure change and evaluate fit, fabrication, assembly and service access. Use W04 and the supplied OpenSCAD source. Prepare actual components, a printed base and lid, a dimensional sketch, and a poorly arranged layout for comparison.

Time	Instructor action	Learner action
1:15 to 1:30	Reconnect the selected concept to interfaces	Identify mechanical constraints from the morning
1:30 to 2:00	Demonstrate component envelopes and access space	Measure parts and sketch a layout
2:00 to 2:30	Show parameters, preview and render in OpenSCAD	Open source and make one controlled change
2:30 to 2:45	Compare printed, cut, machined and stock enclosures	Select a process for the first 25 units
2:45 to 3:00	Break	Break
3:00 to 3:40	Coach the fit and serviceability exercise	Modify model and complete W04
3:40 to 4:10	Run paired mechanical reviews	Inspect cable, fastener, sensor and control access
4:10 to 4:30	Record decisions and close Day 1	Save revision and write the next unresolved question

Demonstration script

“The visible outside shape is only part of mechanical design. We also need room for the cable plug, the screwdriver, the wires, the user's finger, and the path used to remove a part. Show me the assembly sequence before you decide the enclosure is finished.”

Explain the tradeoffs

Teach dimensions, nominal size, tolerance and clearance with physical parts. A clearance is space intentionally left between parts; a tolerance describes allowed variation. A printer's nominal layer height does not establish the accuracy of every feature. Use a small hole coupon or an existing process sample to connect a model dimension to a measured result.

Compare fabrication methods by setup effort, unit effort, geometry, finish, material behavior, and revision cost. A stock enclosure can be economical for 25 units if modifications are simple. Additive manufacturing allows rapid geometric changes but still requires print time, inspection and finishing. Laser cutting suits flat parts and requires suitable material and assembly design. Machining can give controlled features but brings setup and workholding decisions.

Checkpoint and recovery

Require one saved model change, a dimensioned sketch, a chosen process with a reason, and two assembly or service checks. If CAD fails, use the supplied source values and a scale drawing to mark the change, then document the intended revision. A learner can demonstrate sound mechanical judgment without completing an export during class.



Mechanical lab instructions and review

Open and modify the enclosure

1. Open `workshop_enclosure.scad`. Explain millimeters, length, width, height, wall, `floor_t`, `lid_t`, and `hole_d`. The origin is a lower corner of the base. All dimensions in the file are nominal design values.
2. Preview the model. Identify the base, lid, corner posts, cable notch, vent slots and access opening. OpenSCAD constructs geometry from primitives and Boolean operations; use its official documentation for interface details. [S5]
3. Set part to base or lid to examine and export one item. Change one parameter, such as height from 45 to 50 mm, and explain its effect on internal clearance, material use and fastener length.
4. Measure the actual board, breadboard, plugs and cable bend space. Draw their occupied regions on a top view; include corner posts and mounting material.
5. Identify the sensor position relative to the controller and airflow. Keep the sensing opening exposed and avoid covering it with tape or mounting material.
6. Render before exporting STL. Keep the editable SCAD source and record the changed parameter, revision and reason. STL is a manufacturing exchange mesh, not the complete editable design history.

Fit calculation exercise

Assume a board is 68.6 mm long with an allowed variation of plus or minus 0.2 mm. A locating pocket is 69.4 mm long with an allowed variation of plus or minus 0.3 mm. Ask for minimum and maximum total clearance.

Answer. Minimum clearance is 69.1 minus 68.8, or 0.3 mm. Maximum is 69.7 minus 68.4, or 1.3 mm. If centered, nominal clearance is 0.4 mm per end, but assembly may place all the available clearance at one end. These are exercise assumptions, not verified tolerances of the supplied board or printer.

Hands-on assembly review

Use the preprinted baseline enclosure, even if the learner's new revision cannot be fabricated during class. Place components on insulating mounts, route wires, and fit the lid loosely. Check whether the USB cable can be inserted without loading the board, the button can be operated, indicators are visible, and the sensor has an air path. Open the enclosure again and explain the removal order.

The teaching lid has a large access opening and is intentionally easy to inspect. It does not claim ingress protection, impact resistance, flame performance, or production finish. Ask what must change for the intended environment and what test would establish the result.

Common errors and debrief

Watch for measuring only the PCB, forgetting plug overhang, routing wires across fasteners, placing the sensor beside a warm component, and making a lid that cannot be removed with the cable connected. Ask: "Which dimension is critical, and how would the fabricator know?" Conclude with a drawing note that identifies the critical feature and a proposed inspection method.



Day 2 morning electronics development

Outcome. Participants connect a schematic to a physical assembly, perform an unpowered inspection, and obtain live sensor readings. Use W05 and the connection table. Prepare the inspected reference unit and a separate unpowered example with a harmless open connection.

Time	Instructor action	Learner action
9:00 to 9:15	Review Day 1 and introduce electrical interfaces	Explain one enclosure decision that affects wiring
9:15 to 9:40	Teach voltage, current, resistance, power and logic levels	Identify supply and signal paths
9:40 to 10:05	Translate the connection table into a simple schematic	Annotate the architecture and pin map
10:05 to 10:30	Demonstrate meter use and the before-power inspection	Practice unpowered continuity and polarity checks
10:30 to 10:45	Break	Break
10:45 to 11:30	Coach assembly and approve power-up	Build and inspect the circuit
11:30 to 12:00	Demonstrate one controlled diagnostic sequence	Confirm live data and resolve an assigned fault
12:00 to 12:15	Review evidence and parts records	Save wiring photo and preliminary BOM

Teach these ideas

Voltage is a difference between two points; current flows through a path. A component's maximum rating is a boundary to respect, not an operating target. Signal logic voltage and power supply voltage must be compatible across an interface. Ground is the shared reference in this small circuit.

Use the LED branch to connect the model to a measurement. Have learners locate the resistor, identify its nominal value, and predict whether increasing resistance increases or decreases current. Then connect power, signal, mechanical connector and software driver to the sensor interface as four separate considerations.

Introduce the bill of materials as a controlled list of identifiable parts with quantity and revision information. "Temperature sensor" is too vague for repeat purchasing. Explain why a connector footprint, pin assignment, mating part, and orientation matter in addition to a catalog name.

Demonstrate the path toward a custom board

Show how the prototype's electrical relationships would become a schematic, board layout, fabrication files, assembly BOM, placement information, and test access. Discuss protection, decoupling, connector retention and programming access as engineering work to evaluate. The class does not design and fabricate a complete custom PCB during this module.

Checkpoint and recovery

Require a checked connection map, instructor-approved assembly, live log or a clearly identified fallback dataset, and a parts record. If a kit cannot be restored quickly, issue a known-good replacement. Have the participant document the original symptom and the evidence that isolated the failed item.



Electronics lab and fault investigation

Participant procedure

1. Inventory the kit against the materials page. Identify the controller and sensor revision and write them on W05.
2. With USB disconnected, identify button contact pairs and breadboard connections using continuity mode. Return the meter to voltage mode before putting it aside.
3. Assemble each connection from the table. Draw or annotate the schematic as each connection is completed.
4. Ask a peer to trace the wiring, then obtain the instructor's inspection. Do not infer approval from the fact that a neighboring kit is powered.
5. Connect USB. Select the board and port, upload the reference sketch, and open Serial Monitor at 115200.
6. Send I to select live readings. Save ten complete records and note the source field. If the room is warm enough for ALERT, that state can be correct; use the simulated commands for predictable demonstrations.
7. Press and release the button slowly. Observe the count and compare it with the physical action.
8. Save a wiring photograph or annotated drawing and record the exact parts used.

Diagnose before changing

Ask the participant to state the expected result, the actual result, and the last known-good condition. Inspect power and ground, then communication wiring, then software selection. Change one thing at a time and record the result. A new symptom after several simultaneous changes is difficult to interpret.

For a sensor not detected at startup, disconnect USB and check VIN, GND, SDA and SCL against labels. Check that the address configuration matches 0x44. Use the manufacturer's example only as a separate diagnostic if needed, recording which firmware is loaded. Return to the course sketch before the next module.

For a dark LED with otherwise valid data, select n or a to force a known state. Disconnect power, inspect LED polarity, the series resistor, and the correct GPIO branch. The light being off may be correct for the current state; establish that first.

Approved instructor fault

On a separate unpowered demonstration unit, remove the sensor signal connection. Have the class predict the result before powering it. Show FAULT, disconnect, restore the connection, inspect, and restart. Ask why replacing the sensor before checking the wire would have been a weak first step.

Debrief questions

What evidence shows that the sensor is communicating? Which tests depend on real measurements and which can use simulated data? What does the BOM need so another person can buy the same part? How would an intermittent connector appear differently from a consistently incorrect pin assignment?

Expected evidence. A live log supports basic operation at the observed conditions. It does not establish temperature accuracy, connector life, environmental durability, EMC performance, or suitability for production.



Day 2 afternoon software and integration

Outcome. Participants explain the firmware states, change one behavior, integrate the assembly, and identify the version they tested. Use W06 and W07. Prepare a state diagram on paper and the serial command sequence. Keep an untouched copy of the reference sketch available.

Time	Instructor action	Learner action
1:15 to 1:35	Explain states, inputs, outputs and transitions	Draw NORMAL ALERT and FAULT transitions
1:35 to 2:00	Demonstrate the reference code and serial commands	Predict and observe each state
2:00 to 2:30	Coach one controlled behavior change	Edit threshold, version label and corresponding test
2:30 to 2:45	Show change history and reproducible configuration	Save a baseline and changed revision
2:45 to 3:00	Break	Break
3:00 to 3:40	Supervise mechanical and electrical integration	Fit the inspected assembly and repeat basic checks
3:40 to 4:15	Run normal, boundary and fault scenarios	Record observations in W07
4:15 to 4:30	Close Day 2 with an integration review	State the version, remaining issue and next test

Explain the firmware

The loop services serial commands, debounces the button, samples the sensor approximately once per second, selects a state, and updates indicators. Scheduling uses elapsed time rather than a long application delay. The sensor driver still performs synchronous transactions; explain that limitation rather than describing the whole program as nonblocking.

Hysteresis gives ALERT different entry and exit thresholds. For the reference settings, 28 C enters ALERT and 27 C clears it. At 27.5 C the previous valid state matters. A FAULT takes priority over interpreting the temperature. On recovery, the supplied logic evaluates the new reading; it does not retain an alert latch through a fault.

The button uses a pull-up and records a press after a stable low transition. Debouncing addresses rapid contact changes. It does not guarantee detection of an arbitrarily short press or every event during a blocked sensor transaction. The requirement therefore defines a deliberate press duration.

Teach software as part of the product

Separate inputs, decision logic, outputs and configuration. Show how a seemingly small threshold change alters test expectations and documentation. Explain firmware version, source history, dependency version and build environment as distinct pieces of a reproducible release.

Discuss local data handling and future connected features. If networking is later added, identify access control, credential handling, updates, failure behavior and support ownership before adding cloud dependencies. No account or network connection is needed for the reference runtime.



Firmware change and integration lab

Required change exercise

1. Run n, a, h, c, f, n, l and record the state after each new sample. Identify which records are simulated.
2. Save a baseline folder or commit. Record the firmware ID and installed dependencies on W06.
3. Change ALERT_C from 28 to 30 and CLEAR_C from 27 to 29. Change VERSION to a learner-specific revision such as team02-r02. Keep the clear threshold below the alert threshold.
4. Predict the effect of the existing a command, which supplies 29 C. It should no longer enter ALERT from NORMAL. Change the simulated alert value to 31 C, hold value to 29.5 C, and clear value to 29 C if testing the changed thresholds through those commands.
5. Compile, upload, and execute the revised sequence. Record expected and actual states. Restore the original baseline for the shared Day 3 comparison, or clearly identify the changed thresholds in every dependent test.

Instructor explanation

The exercise deliberately couples configuration, test input and expected output. Learners often change the threshold and assume that an unchanged test still exercises it. Ask them to identify every artifact that must change. A firmware label alone does not make a build reproducible; preserve the source and dependency record too.

Integration sequence

Disconnect USB before moving the circuit. Fit the controller and breadboard on insulating supports, secure the sensor without covering its opening, and route cables away from fastener paths. Fit the lid loosely, inspect the wire path and access opening, and reconnect only after approval. Repeat live sensing, indicator selection, and button counting after integration.

Ask a partner who has not watched the build to connect the cable, identify the states, press the button, and explain how to open the enclosure. Observe without coaching first. Record a specific difficulty, then make one improvement and repeat the same task.

Completion evidence

- Saved baseline and modified source with a readable version identifier.
- A state table showing entry, hold, exit, fault and recovery.
- A log identifying firmware version and LIVE or SIM source.
- One observed user or integration problem and a documented response.

Recovery and extension

If compilation fails, compare the changed lines with the baseline and read the first compiler error before later cascading errors. Restore the baseline if the session is running short, then explain the intended change using W06. If upload fails, check board, port and cable before editing application logic.

Advanced learners can add a configurable threshold with range checking, separate the UI from the sensor driver, or design a persistent configuration record. They must define failure behavior and tests before adding the feature. Leave persistent storage and remote updating outside the required exercise.



Day 3 morning testing and improvement

Outcome. Participants execute a traceable test, investigate a fault, make a change and retest. Use W02, W07 and W08. Prepare the test matrix, independent reference instrument, supplied datasets and fault cards. Restore the reference settings before a shared demonstration.

Time	Instructor action	Learner action
9:00 to 9:20	Review requirements and testable claims	Identify the evidence still missing
9:20 to 9:45	Teach verification, validation and measurement limits	Write one complete test procedure
9:45 to 10:30	Supervise functional tests	Execute tests and retain raw observations
10:30 to 10:45	Break	Break
10:45 to 11:20	Issue approved fault scenarios	Investigate and record a fault
11:20 to 11:55	Coach corrective action and regression checks	Make one change and repeat relevant tests
11:55 to 12:15	Facilitate evidence review	Report outcome, limits and remaining work

Teach these distinctions

Verification asks whether a defined requirement was met. Validation asks whether the product helps the intended person perform the intended task in the intended setting. Inspection, analysis, demonstration and measurement can all supply evidence when the method matches the claim.

Define conditions, equipment, configuration, procedure, acceptance criterion and expected evidence before running a test. “It works” is not a record. Preserve failed results and link them to the change and retest. A not-tested item remains not tested even when adjacent tests pass.

Explain repeatability, bias and measurement uncertainty with the comparison exercise. Two nearby sensors can disagree because of the devices, placement, airflow, timing or reference limitations. A manufacturer's accuracy statement for a component does not establish the error of the enclosed assembly. A traceability claim requires an appropriate chain of calibrations and associated uncertainty. [S6]

Demonstration script

“Tell me the exact claim this test supports. If we run a simulated threshold sequence, we learn about the software's decisions. If we compare a real sensor with a reference, we learn about that assembly and setup. Keep those two kinds of evidence labeled.”

Checkpoint and recovery

Require at least one actual functional test, one fault investigation, and one retest or justified corrective plan. Use supplied simulated data when measurement equipment is unavailable, and label the conclusion as data analysis. A participant who analyzes the fallback data still needs an observed functional check or scheduled practical make-up for full practical completion.



Functional test procedures

Test	Method and acceptance	Required record
T01 linked to R01	Reset with working sensor; first valid LIVE record within 5 seconds; repeat three starts	Start method, observed times, firmware ID
T02 linked to R02	Save ten normal records; inspect fields and timestamp differences; interval band 0.9 to 1.2 seconds	Raw log and any interval exceptions
T03 linked to R03	Send n a h c in order; observe NORMAL ALERT ALERT NORMAL after new records	Inputs, outputs and source SIM
T04 linked to R04	Send f then n; expect FAULT with NA numeric fields then NORMAL	Fault and recovery records
T05 linked to R04	Unplug USB, disconnect sensor, inspect and restart; expect FAULT; power off, restore and restart	Physical setup and both outcomes
T06 linked to R05	Note initial count; press ten times using the stated duration and spacing; expect increase of ten	Initial and final count
T07 linked to R06	Present three labeled states to a peer; each identified within ten seconds without coaching	Task order, time and interpretation
T08 linked to R07	Unpowered inspection of lid, wire paths, cable access, controls and sensor opening	Photo or drawing with findings
T09 linked to R08	Another learner identifies the build's model, firmware, BOM and test revisions	Missing items and disposition

Boundary values and limitations

T03 demonstrates state retention at 27.5 C but does not alone test equality at every boundary. Use the host logic checks in the teaching pack or edit simulated input values to exercise exactly 28.0 C and exactly 27.0 C. The expected outcomes are ALERT at entry equality and NORMAL at clear equality. Record how the boundary was supplied.

Use simulated values for exact threshold tests; an uncalibrated room sensor cannot establish exact threshold accuracy. For physical fault tests, remove power before changing wires. Do not claim that a startup absence test proves recovery from every electrical bus fault while running.

Comparison exercise

Place the real sensor and reference close together with similar air exposure, away from hand heat and the controller. Let readings stabilize and record the setup. Collect paired readings at consistent intervals. Calculate device minus reference for each pair, the mean difference and observed range. Document reference resolution, identification and calibration status if known.

In the supplied simulated dataset, the open assembly has a mean temperature difference of plus 0.22 C and the enclosed arrangement plus 0.92 C over five pairs. The difference in mean offsets is 0.70 C. This suggests a placement or enclosure investigation in the example. It does not prove causation or justify a calibration claim; a physical repeat with controlled conditions is needed.



Fault cards and corrective action

Cut or copy one scenario per team. Keep the expected diagnosis private until the group reports its evidence. Assign an observer to record the sequence of checks. Multiple explanations may be plausible; assess the reasoning and evidence rather than a lucky guess.

Card A sensor absent at startup

Setup. On an unpowered demonstration unit, remove the sensor's signal connection. Inspect before power-up. **Observed symptom.** FAULT and no numeric readings. **Expected investigation.** Confirm firmware and source, disconnect power, trace supply and signal connections, restore and retest. **Debrief.** Distinguish the symptom from a conclusion that the sensor itself is defective.

Card B alert never appears after a change

Setup. Use firmware thresholds of 30 and 29 C but leave simulated a at 29 C. **Observed symptom.** The command does not produce ALERT from NORMAL. **Expected investigation.** Compare configuration, test input and requirement. **Corrective action.** Supply an input above the new entry threshold and update the test record. **Debrief.** A test can become invalid even when the implementation is behaving correctly.

Card C button appears permanently pressed

Setup. Present an unpowered four-leg button drawing with the input connected to two terminals that are internally common. Do not require a live miswire. **Expected investigation.** Identify contact pairs with continuity and redraw the wiring across the switching contact. **Debrief.** A component's physical outline does not describe its internal connections.

Card D enclosed readings are higher

Setup. Provide the simulated sensor comparison CSV. **Expected investigation.** Calculate offsets, inspect sensor placement and controller heat sources, check timing and reference conditions, and propose a controlled repeat. **Corrective action.** Move the sensor or revise the air path and repeat the comparison. **Debrief.** Do not erase failed evidence or subtract an arbitrary offset to make the result look correct.

Card E a second assembler cannot finish

Setup. Give a partner assembly instructions that omit LED polarity, firmware version and the test limit. **Expected investigation.** Mark the missing information before assembling. **Corrective action.** Add the orientation, version and acceptance details, then repeat the handoff. **Debrief.** Documentation defects can create physical and functional defects.

Corrective action sequence

Record symptom, configuration, severity, evidence, hypothesis, next check, change, and retest. Separate containment from correction: using a spare kit restores the lesson but does not establish the cause. Retest the original failed behavior and at least one related behavior that could have been affected. Close the issue only when the evidence supports closure; otherwise assign an owner and next action.



Day 3 afternoon preparing the first batch

Outcome. Participants prepare a draft package for 25 units and defend its remaining risks. Use W09, W10, W11 and W12. Prepare the cost example, supplier request checklist and one deliberately incomplete handoff package.

Time	Instructor action	Learner action
1:15 to 1:35	Show what a second builder needs	Identify missing release information
1:35 to 2:05	Work through parts, labor, yield and setup costs	Complete W09 and compare two production routes
2:05 to 2:30	Demonstrate an assembly traveler and test record	Draft W10 with quality checkpoints
2:30 to 2:45	Explain supplier questions and unresolved obligations	Record sourcing and compliance follow-up
2:45 to 3:00	Break	Break
3:00 to 3:30	Facilitate paired handoff reviews	Review W11 and record blockers
3:30 to 4:10	Run eight five-minute final reviews	Present evidence and answer one question
4:10 to 4:30	Complete assessment and course feedback	Submit W12 and identify the next development decision

Teach the transition

A functional prototype shows that something can work under observed conditions. A repeatable build needs defined parts, controlled files, assembly instructions, test methods, acceptance limits and a record of what was built. A release also needs an owner who can decide how to handle failures and changes.

Explain design for manufacture, assembly and test using the actual unit. Could the wire set be replaced with a harness? Can an assembler see polarity? Can firmware be loaded without dismantling the unit? Can a failing unit be isolated before packaging? Would a stock enclosure reduce setup work at this quantity?

Final review script

“Show the requirement you tested, the evidence, and a change you made because of that evidence. Then show what another builder would receive. Finish with the biggest reason you would hold the first batch today and the work needed to resolve it.”

For eight participants, use three minutes of presentation, one minute for a question, and one minute for feedback each. With six participants, use the remaining ten minutes for a shared production problem. Keep the final action-plan period intact.

Completion checkpoint

Accept a documented prototype, a traceable test and change, a first-batch estimate, a draft assembly and test sequence, and an honest list of release blockers. “Hold” can be the correct engineering decision. Do not reward a confident release recommendation that lacks evidence.



Production handoff and readiness review

Minimum draft handoff package

Artifact	Required information
Product specification	Intended use, requirements, exclusions and acceptance criteria
Mechanical package	Editable source, manufacturing exports, units, material, critical dimensions and revision
Electronics package	Schematic or controlled connection map, parts list, connector orientation and revision
Software package	Source, version, dependencies, configuration, build and programming instructions
BOM and sourcing	Exact part identity, quantity, supplier, approved alternatives and unresolved supply risks
Assembly instructions	Sequence, orientation, tools, inspections and rework route
Test package	Equipment, setup, procedure, limits, record format and disposition rules
Unit identification	Serial or batch identification linked to hardware and firmware revisions
Packaging and user information	Contents, setup, limitations, support route and handling needs
Release record	Open issues, decisions, responsible owner and release or hold status

Peer review procedure

Exchange packages with a learner who did not build the unit. Give them ten minutes to locate files and identify the exact configuration. Allow ten minutes to walk through assembly and functional testing without inventing missing instructions. Use the final ten minutes to record blockers, clarify the package and assign follow-up actions.

Require the reviewer to distinguish a missing item, an ambiguous instruction, and an unverified assumption. A polished drawing with unspecified material can still block a quote. A correct program without a known board package can still block reproduction. A passing test without a unit ID can still fail to identify what was tested.

Release decisions for the exercise

Proceed to a limited pilot build when the package can support a controlled learning batch, the open risks are understood, and the build owner has defined stop conditions. **Hold** when a critical requirement is untested, assembly cannot be reproduced, a serious failure remains unexplained, or a sourcing or compliance question changes feasibility. **Revise and review again** when the gap is specific and bounded.

The 25-unit exercise is planning work. Do not classify the classroom unit as ready for sale merely because its lessons are complete. A small quantity does not by itself determine the obligations for a marketed electronic product. Digital electronics can require an equipment-authorization assessment even without a radio; identify the applicable route with a qualified lab or specialist using the current product and market. [S7]

Record market, intended use, power source, radio functions, environment, user group, materials, claims, and proposed sales channels as inputs to that assessment. Assign an owner and obtain a scope or quote before committing a commercial launch date.



Worked example for a first batch of 25 units

All figures on this page are fictional teaching inputs in US dollars, not supplier quotes, market prices, or a recommended selling price. The \$38 electronics allowance is a hypothetical production design allowance; it is not the purchase price of the branded classroom kit. Have participants replace every allowance with identified quotes before using the model for a business decision.

Recurring cost input	Per unit started	Basis
Electronics and interconnect	\$38.00	Hypothetical parts allowance
Enclosure and hardware	\$14.00	Hypothetical fabricated parts allowance
Label and packaging	\$3.00	Simplifying allowance
Assembly labor	\$15.00	20 minutes at \$45 per hour
Test labor	\$7.50	10 minutes at \$45 per hour
Recurring subtotal	\$77.50	Before losses, freight and one-time work

Batch calculation

Use a planning yield of 92 percent. To target 25 accepted units, divide 25 by 0.92 and round up to 28 starts. This is a planning allowance, not a guarantee that the batch will produce 25 good units. Define how to handle a shortfall and do not treat a forecast yield as measured process performance.

The conservative classroom model charges the full recurring cost to every start: 28 times \$77.50 equals \$2,170. Add \$450 for a reusable programming and test fixture, \$600 for documentation and setup work, and \$180 for inbound freight and miscellaneous batch costs. The modeled batch total is \$3,400. Allocated across 25 accepted units, the total is \$136 per accepted unit.

The recurring cost allocated across the planned 25 accepted units is \$86.80 each. One-time fixture and setup costs total \$1,050. Keep those categories visible when comparing a repeat batch. This example omits broader overhead, selling costs, warranty reserves, taxes and outbound delivery; add applicable costs for an actual decision.

Sensitivity questions and answers

Assembly takes 35 minutes. The extra 15 minutes costs \$11.25 per start. Across 28 starts, the batch increases by \$315, or \$12.60 per planned accepted unit. The revised total is \$3,715 and \$148.60 per accepted unit.

Planning yield falls to 80 percent. Round 25 divided by 0.80 up to 32 starts. Recurring cost becomes \$2,480 and total batch cost \$3,710, or \$148.40 per planned accepted unit. The model still does not guarantee the required quantity.

A stock enclosure costs \$6 more but saves ten assembly minutes. At \$45 per hour it saves \$7.50 of labor per start. The modeled net saving is \$1.50 per start before modification tooling, qualification and supply differences. Ask whether the required fit and tests remain satisfied.

Avoid false precision

Show the source, date, quantity break and uncertainty for each estimate. Distinguish purchased quantity, started quantity, first-pass yield, reworked units, final accepted units and scrap. A more detailed model can charge packaging only after acceptance and separate rework labor; the simple example deliberately bundles those costs for teaching.



Assembly flow and supplier preparation

Proposed pilot build sequence

1. Receive and identify parts. Count quantities, check damage, and quarantine substitutions until their equivalence is reviewed.
2. Inspect mechanical critical features using the selected gauges or fit samples. Record the incoming lot and mechanical revision.
3. Assemble and inspect the electronics. Check polarity, connector orientation, retention and workmanship before power.
4. Program the controlled firmware and record its version against the unit or batch ID. Verify configuration rather than relying on a filename alone.
5. Perform functional tests before closing the enclosure so defects remain accessible.
6. Complete mechanical assembly, inspect wiring and access, and repeat the relevant integrated tests.
7. Record pass, rework, hold or scrap. Reworked units repeat the affected tests and any defined regression checks.
8. Apply labels, add user instructions, verify package contents, and record release by the named build owner.

Test fixture discussion

A low-volume fixture may be as simple as a stable holder, protected connections, a known input source and a repeatable test script. Define what it measures, how its reference is checked, how it prevents incorrect orientation, and how it records a result. Compare a verified working unit before and after a batch to detect gross fixture changes; that check does not establish calibration by itself.

Ask learners to estimate fixture setup time, per-unit time, maintenance and the cost of an incorrect pass. A fixture that saves two minutes but hides intermittent connections may be a poor trade. Preserve access to programming and test points when considering a custom PCB.

Supplier request checklist

Give a supplier the current drawing or file revision, quantities to quote, material and finish, critical dimensions and tolerances, inspection needs, delivery location, required date, and packaging expectations. Ask about minimum order quantity, setup charges, lead time, capacity, inspection records, substitution policy, remake terms and ownership of tooling. Identify any information still provisional.

Request comparable quotes at 5, 25 and 100 units to understand setup and quantity effects. Compare the same scope, delivery basis and inspection requirement. Unit price alone can hide substantial assembly work, shipping or tooling. A supplier's unconfirmed lead time should remain a schedule risk.

Change control during a batch

Freeze the approved build files for the pilot. Log each proposed change with reason, affected units, affected tests, approval and effective serial or batch range. Preserve the previous revision. If a substitution is unavoidable, review electrical, mechanical, software and test impacts before using it. Make the unit history clear enough that a later problem can be traced to the relevant configuration.



Assessment and completion rubric

Use the rubric as structured feedback. Assess each learner's explanation and evidence even when work is paired. A successful recovery from a documented problem can demonstrate stronger understanding than an unexplained first-time success.

Area	Weight	Evidence for a competent score of 2
Product definition	15	User and need are clear; requirements have conditions and pass criteria
Mechanical development	15	One controlled change with fit, fabrication and access reasoning
Electronics	15	Correct interface map and inspected build or completed practical make-up
Software and integration	15	Versioned behavior change and correct state explanation
Test and improvement	20	Traceable test, honest results, diagnosis and relevant retest
Production preparation	20	BOM, labor and yield assumptions, assembly and test plan, release blockers

Scoring anchors

Score each area from 0 to 3. **0** means evidence is absent or the explanation is materially incorrect. **1** means partial evidence with unresolved gaps requiring substantial guidance. **2** means the required evidence is present and the learner can explain the decision with limited prompting. **3** means the learner also evaluates a tradeoff or limitation and uses evidence to improve the work.

Calculate weighted points as the score divided by 3, multiplied by the area's weight. A score of 2 in every area totals 66.7 out of 100. Use 65 as the feedback threshold, with no area at 0. This course threshold is a proposed educational rule, not a professional accreditation standard.

Completion conditions

- Attend at least 16 of the 18 instructional hours, with missed core work completed by an agreed make-up date.
- Participate in an inspected hardware activity and demonstrate or explain a controlled firmware change.
- Submit W01 through W12 with the required evidence or an agreed accessible equivalent.
- Achieve the rubric threshold with no missing area and complete the final review.
- Resolve unsafe work practices before continuing any practical activity; provide coaching and reassessment where needed.

A learner who uses only the fallback datasets can complete the conceptual work but should be recorded as pending practical completion until the hardware check is observed. Distinguish attendance, learning evidence and equipment failure in the record.

Feedback and certificate wording

Give one demonstrated strength, one specific gap, and one next action with an owner and date. Example: "Your fault record clearly identifies the changed firmware and the retest. The assembly instructions still omit connector orientation. Add a labeled view and ask a new builder to follow it."

Use a certificate title such as Green Shoe Garage Product Development Boot Camp Certificate of Completion. State dates, instructional hours and instructor. Do not imply engineering licensure, professional certification, accredited continuing education, or approval of the participant's product.



Instructor troubleshooting and fallback guide

Symptom	First checks	Recovery for the lesson
No USB port appears	Known-good data cable, computer connection, selected board, OS permissions	Use spare cable or board; analyze supplied log while support continues
Compile error	First error, missing library, board selection, changed lines	Restore reference folder; reopen complete sketch
Upload fails	Board and port, competing serial application, cable	Close other serial users, reconnect, retry baseline once
Serial data unreadable	Baud setting and correct port	Set 115200; reset and capture a new header
FAULT in live mode	Unpowered supply and signal inspection, address, sensor identity	Restore known-good wiring or issue spare sensor
NORMAL at a warm simulated value	Threshold revision, previous state and actual command value	Compare configuration with test expectation
Button count stays zero	Contact pairs, common ground, D2 and press duration	Unpowered continuity check; use spare button
LED is dark	Expected state, GPIO, resistor, polarity	Force n or a; inspect with power removed
CAD is slow or fails	Source syntax, supported parameter range, available memory	Use baseline file and dimensioned paper revision
Lid will not fit	Mounting height, wire path, fasteners, actual part dimensions	Use open baseline for electronics; log mechanical change
Online participant loses connection	Rejoin route and local written instructions	Pause their powered work; return at a defined checkpoint

Escalation rhythm

Let the learner describe the symptom and show evidence. Spend up to five minutes on a focused check. If the problem remains, move to a spare, baseline file, paired observation or dataset so learning continues. Record the blocked practical item and schedule a supported return. Do not turn the entire class into spectators for one laptop repair.

If the group is behind

Protect the core outcomes. Shorten the number of concept alternatives, provide the baseline requirements, use prepared enclosure parts, and demonstrate optional manufacturing techniques instead of having everyone perform them. Retain before-power inspection, the fault exercise, test evidence, handoff review and final action plan.

If the group is ahead

Assign a second test condition, a usability observation, an assembly-time measurement, an alternate sourcing comparison, or a test-fixtured sketch. Ask for evidence and documentation rather than adding unrelated features. Use the extra time to improve the product package.



Course closeout and continuous improvement

End of each day

Ask each learner to save their files with a clear revision and submit a short checkpoint: what works, what changed, what remains uncertain, and what they need next. Review these before the following session. Keep the reference files unchanged and distribute a new revision only when a correction is required.

For online cohorts spaced a week apart, offer optional practice: refine the brief after Day 1 and repeat the integration sequence after Day 2. Do not make additional unscheduled hours necessary to complete the advertised core course. Provide an equivalent supported path for learners without access to equipment between sessions.

Final learner review

Collect the BenchBook, rubric, final action plan and three feedback answers: Which activity changed how you think about product development? Where did instructions or equipment slow you down? What would help you apply this to your own project? Ask separately about pace, access and technical support so a hardware problem is not mistaken for a content problem.

Follow-up session

Include a group follow-up about two weeks after the course if it is part of the sold package. Suggested duration is 60 minutes: ten minutes for progress, thirty minutes for selected development questions, fifteen minutes for peer feedback, and five minutes for commitments. Ask participants to submit one question and one artifact beforehand. Keep confidential project consulting outside a recorded group session unless the owner explicitly chooses to share it.

Instructor retrospective

Measure	Record after each cohort
Preparation	Actual procurement, fabrication, packing and rehearsal hours
Setup success	Participants ready before class and common setup failures
Module timing	Planned versus actual activity times and skipped optional work
Practical outcomes	Kits operating, incomplete practical checks and make-up needs
Learning	Rubric distribution and recurring misconceptions
Materials	Missing parts, failed parts, replacements and geometry revisions
Delivery cost	Teaching, support, venue, shipping, consumables and follow-up
Improvements	Proposed change, owner, reason and next rehearsal

Course revision control

Keep a course release record containing manual edition, companion pack revision, hardware configuration, software versions and test evidence. Correct a dangerous or materially incorrect instruction immediately and notify affected participants. For ordinary improvements, revise between cohorts, rehearse the changed exercise and preserve the prior edition for traceability.



W01 Product brief

Participant _____ Team _____ Date _____

Use during Day 1 morning. Write a brief that another person can understand without hearing the original discussion. Label assumptions that still need evidence.

User and setting

Who is the primary user, where will the product be used, and what task are they trying to complete?

Problem and current approach

What is difficult today? How does the user handle it now? What consequence makes it worth improving?

Success and limits

What observable result would make the product useful? What are the important cost, time, environmental or access constraints?

Scope for this iteration

Functions included _____

Functions deferred _____

Assumptions to investigate

Assumption	Evidence needed	Owner and next action

Review. Can a partner explain the problem without naming a particular technology? Reviewer

_____ Required revision _____



W02 Requirements and verification

Participant _____ Product revision _____ Date _____

Use one requirement per row. Include conditions, units and a pass criterion. Method may be test, inspection, analysis or demonstration. Add pages as needed. Record a test ID so results remain traceable.

ID	Requirement and condition	Acceptance and method	Test ID

Review one requirement

Which user need does it support? _____

Could two testers disagree about pass or fail? Rewrite any ambiguous language.

Important boundary conditions

What happens at the threshold, when an input is missing, at startup, and after power returns?

Requirement change record

Changed ID _____ Previous criterion _____

New criterion _____ Reason _____

Affected design files and tests _____

Review. Every requirement has observable evidence and a defined condition. Reviewer _____

Date _____



W03 Concept selection

Participant _____ Date _____

Two concepts

Sketch each concept with user orientation, sensor location, cable path and service access. Use the space below or attach a clearly identified drawing.

Concept A _____

Concept B _____

Decision matrix

Choose weights totaling 100 percent. Score each concept from 1 to 5 with 5 most favorable. Multiply each score by its fractional weight and add the products.

Criterion	Weight	A score	B score	Evidence or assumption
Total	100 percent			

Selected concept _____ Reason _____

Which uncertain score could change the choice? What quick test would resolve it?



W04 Mechanical design review

Participant _____ Model filename _____ Revision _____

Dimensions and access

Feature	Nominal dimension and units	Variation or clearance allowance	How checked
Board and supports			
Cable and plug			
Button and indicators			
Sensor air path			
Lid and fastening			

Change made

Parameter or feature _____ Before _____ After _____

Reason and affected interfaces _____

Manufacturing choice

Material _____ Process _____ Quantity _____

Why does this choice suit the quantity, function and revision risk?

Assembly and service check

Assembly order _____

How is a failed board or sensor removed? _____

Critical dimension and proposed inspection _____

Remaining risk and required test _____

Reviewer _____ Outcome Proceed / Revise / Hold



W05 Electronics and interface record

Participant _____ Hardware revision _____ Date _____

Controller identity _____ Sensor identity _____

Power input _____ Logic level _____

Connection record

Signal or supply	From pin	To pin or component	Inspection result
Supply			
Ground			
Sensor data			
Sensor clock			
Normal indicator			
Status indicator			
Button			

Before power

Mark checked or describe the problem: polarity _____ series resistors _____

Shared ground _____ rail continuity _____ bare conductors _____

Insulating support _____ correct supply and board _____

Instructor or assistant approval _____ Time _____

First operation

Firmware loaded _____ Sensor detected _____

Observed source LIVE / SIM _____ Initial state _____

Evidence filename _____ Wiring photo or drawing _____

Preliminary parts record

Part identity	Quantity	Revision or rating to preserve

Problem or unresolved interface _____



W06 Firmware and integration record

Participant _____ Firmware ID _____ Date _____

IDE version _____ Board package and version _____

Library names and versions _____

Source folder or commit _____ Hardware revision _____

State behavior

Current state	Input or event	Expected next state	Output indication
NORMAL	At entry threshold		
ALERT	Between thresholds		
ALERT	At clear threshold		
Any state	Invalid reading		
FAULT	Valid normal reading		

Controlled change

Behavior changed _____

Old setting _____ New setting _____

Test inputs that also need changing _____

Expected result _____

Observed result and evidence file _____

Integration checks

Cable access _____ button access _____ visible indications _____

Wires clear of fasteners _____ sensor opening clear _____

Repeated live test _____ repeated button test _____

Reproduction and recovery

How would another person restore the baseline?

Remaining software or integration issue _____



W07 Test procedure and result

Test ID _____ Requirement ID _____ Date _____

Tester _____ Unit ID _____ Hardware revision _____

Firmware version _____ Source LIVE / SIM / Dataset _____

Claim and acceptance criterion

What exactly will this test establish, and what result counts as a pass?

Setup and equipment

Include configuration, environment, reference instrument identity and relevant limitations.

Procedure

Step	Action or input	Expected observation	Actual observation
1			
2			
3			
4			

Result and evidence

Result Pass / Fail / Not tested _____ Raw evidence filename _____

Exceptions and limitations _____

Linked issue ID _____ Retest required _____

Reviewer _____ Date _____



W08 Issue and corrective action

Issue ID _____ Reported by _____ Date _____

Unit and revision _____ Related requirement or test _____

Observation

Expected behavior _____

Actual behavior and reproduction conditions _____

Consequence if unresolved _____

Immediate containment _____

Investigation

Hypothesis	Evidence or next check	Result

Cause supported by evidence _____

Uncertainty that remains _____

Change and retest

Change made _____ New revision _____

Original failed test repeated _____ Result _____

Related regression check _____ Result _____

Evidence filename _____

Disposition

Status Open / Contained / Corrected pending retest / Closed _____

Owner _____ Next action _____ Due date _____

Closure reviewer _____ Closure date _____



W09 First batch cost and sourcing

Participant _____ Planned accepted quantity _____ Date _____

Use actual quotes when available. Otherwise mark each value as an estimate. Keep one-time and recurring costs separate. Record currency and whether tax and freight are included.

Part or assembly identity	Quantity per start	Unit cost	Supplier and quote date

Labor and yield assumptions

Assembly minutes per start _____ Test minutes per start _____

Labor rate per hour _____ Planned yield _____ Evidence for yield _____

Starts required rounded upward _____ Recurring cost per start _____

Expected rework allowance and method _____

Batch calculation

Recurring total _____ Fixture and tooling _____ Setup and documentation _____

Freight and other batch costs _____ Modeled batch total _____

Cost allocated per planned accepted unit _____ Omitted costs _____

Sensitivity and supply risk

What changes if assembly takes 15 minutes longer or yield is lower?

Longest lead item _____ Alternate and qualification work _____

Minimum order or quantity break _____

Most uncertain input and next action _____



W10 Assembly and acceptance traveler

Unit or batch ID _____ Build date _____

Mechanical revision _____ Electronics revision _____ Firmware ID _____

Assembler _____ Test operator _____

Write instructions that another builder can follow. Identify orientation, tools, inspections and evidence. Add pages for a real build. Do not use an unsigned traveler as a release authorization.

Step	Operation and critical instruction	Inspection or test limit	Initials and result
1	Receive and identify parts		
2	Prepare mechanical parts		
3	Assemble and inspect electronics		
4	Program and verify version		
5	Run functional tests		
6	Close and inspect enclosure		
7	Repeat integrated tests		
8	Label and package		

Rework and disposition

Issue ID _____ Rework performed _____

Tests repeated and evidence _____

Final result Pass / Rework / Hold / Scrap _____

Build owner _____ Release or hold date _____

Record location _____



W11 Production readiness review

Product and revision _____ Reviewer _____ Date _____

Use Ready, Gap or Not applicable with an explanation. Readiness here means sufficient evidence for the proposed next development step. It does not certify a product for sale.

Review item	Status	Evidence or gap
Intended use and requirements defined		
Mechanical files and critical features controlled		
Electrical interfaces and parts identified		
Software reproducible and configuration recorded		
Sourcing and alternatives understood		
Assembly sequence can be followed		
Acceptance tests and record format defined		
Unit identification and change control defined		
Packaging and user information drafted		
Compliance scope assigned for review		
Unresolved failures and risks assigned		

Decision

Proposed next step Limited pilot / Revise and review / Hold _____

Reason _____

Top blocker _____ Owner _____ Due date _____

Evidence needed to resolve it _____

Review participants _____ Decision owner _____



W12 Development action plan

Participant _____ Product or idea _____ Date _____

What the course changed

One assumption revised by evidence _____

One decision from the reference build that applies to my own idea

The next 30 days

Priority and date	Action or experiment	Owner	Evidence or decision produced
Week 1			
Week 2			
Week 3			
Week 4			

Resources and constraints

Skills, equipment or outside help needed _____

Budget or schedule constraint _____

Most important unknown _____

Decision to make before spending on the first batch _____

Follow-up question

What specific artifact or evidence will I bring to the follow-up session?

Participant commitment _____ Instructor feedback _____



Knowledge check and discussion answers

Use these questions at the start as a low-pressure discussion and again at the end to identify learning. Do not count prior knowledge against a participant. One point per sound answer can provide a simple ten-point progress measure, separate from the practical rubric.

1. **What makes a requirement testable?** It identifies a behavior or property, the conditions that apply, and an observable acceptance criterion. “Easy to use” needs a defined task and measure.
2. **How do design and mechanical engineering interact?** User interaction and appearance constrain geometry; fit, materials, assembly and fabrication constrain the usable form. Neither can be completed in isolation from the other interfaces.
3. **Why does a component envelope include more than its body?** The design needs room for plugs, cables, tolerances, tools, fingers, mounts and the path used for assembly or removal.
4. **Why does the LED branch need a series resistor?** It limits current for the chosen supply, LED and output. A component's maximum rating should not be treated as the normal operating target.
5. **Why can 27.5 C produce different states?** In the baseline hysteresis band, a previous ALERT remains active while a previous NORMAL remains normal. Entry and exit thresholds differ.
6. **What does a successful simulated threshold test establish?** It supports the tested decision logic and output behavior for those inputs. It does not establish sensor accuracy or physical thermal performance.
7. **How do verification and validation differ?** Verification checks specified requirements. Validation checks fitness for the intended user and use. Each needs evidence appropriate to its question.
8. **Why is a BOM alone insufficient to estimate unit cost?** Labor, test, losses, setup, tooling, freight, packaging, overhead and other applicable costs also affect the result. Some are recurring and some are one-time.
9. **What should happen after a corrective change?** Preserve the original finding, identify the changed revision, repeat the failed test, and perform relevant regression checks before closing the issue.
10. **What should make an instructor accept a hold decision?** The learner identifies a material unresolved issue, explains its consequence, and defines the evidence and owner needed to resolve it.

Misconceptions to correct explicitly

More features do not automatically make a concept more useful. A fabrication export does not replace editable source. A device reading that looks plausible does not prove measurement accuracy. A component or development board's approvals do not automatically settle the finished product's assessment. A small batch still needs controlled files and a way to identify what was built.

Ask learners to revise an answer using their own course evidence. For example, “I changed the cable opening because the plug did not fit” connects a mechanical principle to an observed result better than reciting a definition.



Worked answers and assessment examples

Expected state sequence

Starting condition	Supplied input	Expected state	Reason
STARTUP	Valid 24 C	NORMAL	Valid value below alert entry
NORMAL	28 C	ALERT	Entry includes equality
ALERT	27.5 C	ALERT	Above the clear threshold
NORMAL	27.5 C	NORMAL	Below the entry threshold
ALERT	27 C	NORMAL	Clear includes equality
Any	Invalid reading	FAULT	Invalid data takes priority
FAULT	Valid 29 C	ALERT	Recovery evaluates current input
FAULT	Valid 24 C	NORMAL	Recovery evaluates current input

Mechanical calculation

The board example gives a minimum pocket of 69.1 mm and maximum board of 68.8 mm, leaving 0.3 mm total clearance. The largest pocket of 69.7 mm and smallest board of 68.4 mm leave 1.3 mm. A correct answer distinguishes total from per-side clearance and identifies the assumed process tolerances.

Dataset interpretation

The five open-assembly differences are 0.2, 0.3, 0.1, 0.2 and 0.3 C; mean 0.22 C, minimum 0.1 C, maximum 0.3 C. Enclosed differences are 0.9, 1.0, 0.8, 0.9 and 1.0 C; mean 0.92 C, minimum 0.8 C, maximum 1.0 C. A strong answer proposes a repeated, controlled comparison and keeps the data labeled simulated.

Cost answer

The baseline requires 28 planned starts for 25 accepted units at a planning yield of 92 percent. Recurring cost is \$2,170, one-time fixture and setup \$1,050, and other batch cost \$180, totaling \$3,400. The allocated cost is \$136 per planned accepted unit. A strong answer identifies omitted costs and distinguishes planning yield from a guarantee.

Examples of rubric judgment

Score 1 for testing. The learner says the indicator worked but records no input, firmware version or acceptance criterion. **Score 2.** The learner records the input sequence, version, expected and actual states, a failure and a relevant retest. **Score 3.** The learner also identifies what the test does not establish and adds a boundary or regression check based on the observed risk.

Score 2 for production preparation. The package identifies exact parts, labor assumptions, build steps, tests and unresolved release questions. **Score 3.** It also examines a sourcing or yield sensitivity and revises a decision using that analysis. A polished presentation does not compensate for missing evidence.



Technical references and teaching terminology

The sources below support the selected hardware interfaces and technical distinctions. Course requirements, schedules, thresholds, case dimensions, cost figures and assessment rules are teaching choices. Recheck manufacturer documentation and the actual parts before each cohort.

S1 Arduino UNO R4 Minima hardware documentation. Controller description and operating logic voltage.
<https://docs.arduino.cc/hardware/uno-r4-minima/>

S2 Arduino UNO R4 Minima full pinout. Pin identities and current limits.
<https://docs.arduino.cc/resources/pinouts/ABX00080-full-pinout.pdf>

S3 Adafruit SHT31 D breakout pinouts. Supply, signal pins and default address for the identified breakout.
<https://learn.adafruit.com/adafruit-sht31-d-temperature-and-humidity-sensor-breakout/pinouts>

S4 Adafruit SHT31 library. Public sensor initialization and reading interface.
https://github.com/adafruit/Adafruit_SHT31

S5 OpenSCAD documentation. Modeling, preview, render and export references.
<https://openscad.org/documentation.html>

S6 NIST policy on metrological traceability. Traceability and the role of measurement uncertainty.
<https://www.nist.gov/calibrations/traceability>

S7 FCC equipment authorization guidance DA 22 312. Background on equipment authorization and device categories. Confirm the current applicable route for an actual commercial product.
<https://docs.fcc.gov/public/attachments/DA-22-312A1.pdf>

Terms used in the course

BOM means bill of materials, the identified parts and quantities required for a build. **CAD** means computer-aided design; **MCAD** refers to mechanical design and **ECAD** to electronics design. **Firmware** is software running on the embedded controller.

DFM means design for manufacture; **DFA** means design for assembly; **DFT** means design for test. **First-pass yield** is the fraction passing the defined process without rework. **Final yield** includes the effect of permitted rework. State which one a calculation uses.

Revision identifies a controlled change to an artifact. **Configuration** identifies the combination of hardware, software and settings used. **Traceability** in the course means linking requirements, artifacts, units and evidence; **metrological traceability** is the more specific measurement concept discussed in S6.

Verification establishes whether requirements were met. **Validation** evaluates whether the product serves its intended use. **Regression testing** checks related behavior after a change. **Acceptance testing** determines whether a built unit meets its specified release criteria.