

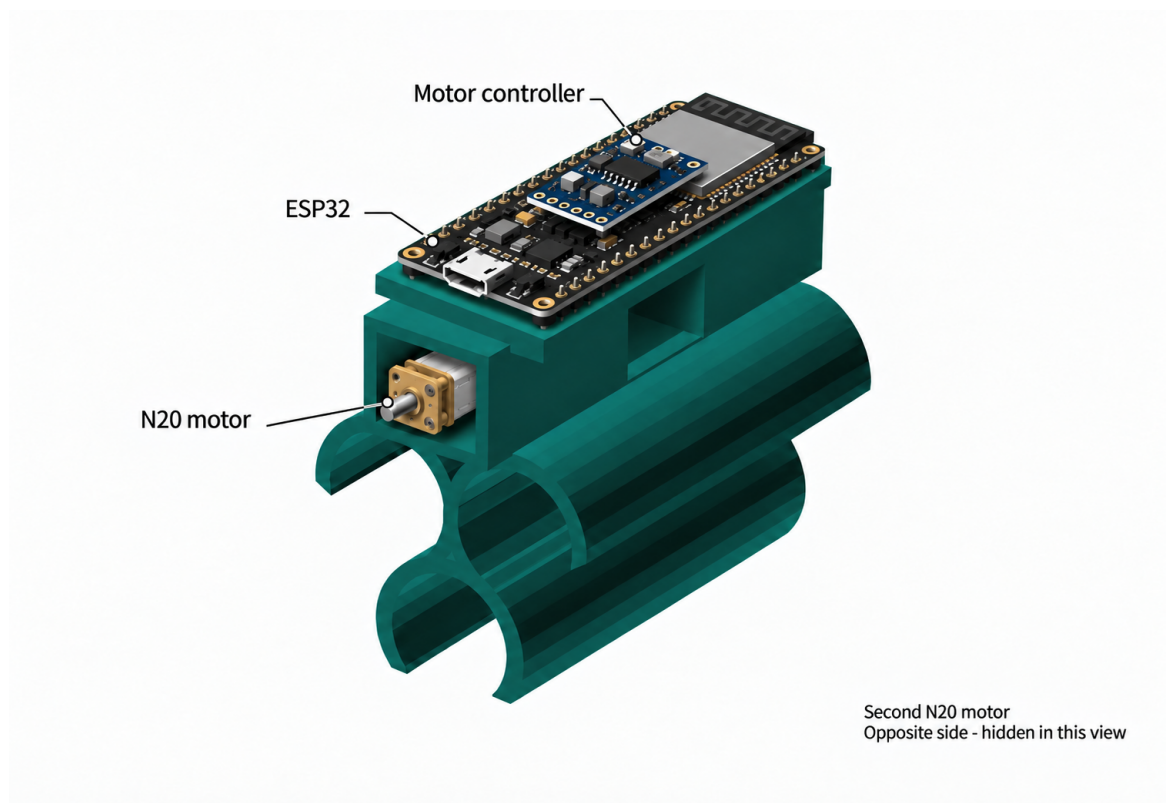
Ball Robot Workshop

Relational arrangement, sessions and resources

A ball-shaped rolling robot built from scratch. The used skeleton had a flat top covered by the ESP32 board, with the motor controller stacked above it, pockets for N20 motors, and three bottom slots holding two batteries and a central screw used as added weight.

Four four-hour lessons retained as a documentation framework.

Prepared by Philip Michalowski



AI-assisted layout illustration based on the candidate STL render: ESP32 covers the top platform; motor controller stacked above it. N20 motors occupy opposite sides, one hidden here. Component sizes are not verified to scale. Exact skeleton file, board models and fit remain unconfirmed. Batteries and weighting screw are omitted.

Working documentation updated 05 October 2026. Compiled from project notes, design records and instructional materials. The reader guide distinguishes reported work, inspected artifacts, planned activities and interpretations.

Contents

01 Reader guide and evidence	3
02 Overview	4
03 Relational arrangement	7
04 Session plan	12
05 Models and materials	15
06 AI and instruction changes	21
07 Research framing	23
08 Documentation guide	27
09 Sources and image provenance	29
10 Session record template	32
11 Instruction change template	34

How to use this PDF

Start with the reader guide, overview and relational arrangement. The session plan and materials sections support teaching; the research framing and documentation guide support interpretation and recording. The final two sections are reusable templates.

Internal references link within this PDF. To use links to the companion workshop PDF, keep both files in the same folder. Links marked local source or source note require the original project files and are not public repository links. The source chapter records their provenance; the PDF itself can be read without Obsidian.

01 Reader guide and evidence

The project in plain language

The Ball Drone workshop concerns a ball-shaped rolling robot built from scratch, rather than an aircraft. The documented designs include a carrier that holds components, wheels, electronics and a browser-based controller. N20 motors are small geared motors used in the build. The organizer confirms that the used skeleton had a flat top covered by the ESP32 board, with the motor controller stacked above it, pockets for the N20 motors, and three bottom slots holding two batteries and a central screw used as added weight. The exact model file, electrical wiring, final enclosure, and chronology of contributions remain to be recorded.

Contribution to the first research question

RQ1 asks how generative AI reshapes the relations through which action, knowledge, responsibility and authority are distributed in instruction-mediated making. The implemented coding room provides an individual question-and-answer route through shared computing infrastructure. This makes the organization of access to AI advice inspectable. Whether and how participants used that advice to change instructions or making decisions requires dated accounts linking suggestions, decisions and physical outcomes. The existing designs and software establish resources and configurations, not participant outcomes by themselves.

Terms used in this record

- **Arrangement** means the people, parts, tools, instructions, time and support through which making becomes possible. It can involve one maker or several people.
- **Configuration** means how those relations are organized, including who acts, checks, decides and can propose a change.
- **ESP32** is a family of programmable microcontroller chips and boards used to control the robots. Specific board versions are distinguished in the source records.
- **PCB** means printed circuit board. **CAD** means computer-aided design. **STL** is a file format for a model's surface geometry; a rendering is not evidence of a printed or tested part.
- **Obsidian** is the note-taking application used to organize this documentation. It is not needed to read this PDF.

Reading the evidence

This is a working documentation record of completed and ongoing workshop work, associated preparation and forthcoming activities. Organizer reports, inspected project files, proposed activities and analytical interpretations are identified separately. Blank templates are recording tools, not completed session observations. The later chapters retain technical detail so the basis of each claim remains inspectable.

02 Overview

Working documentation of completed and ongoing workshop work, preparation and forthcoming activities. The four-session plan is a documentation framework, not an attendance record.

The **ESP ball was built from scratch**, as confirmed by the organizer. This makes its starting point different from the flying drone, which uses a kit. Its **relational arrangement** was configured through bringing components, body and wheel designs, ESP programming, collaborators, tools, and support into a working relationship.

The files in Research/BallDroneWorkshop document body/carrier and wheel designs, motor-control software, a browser controller, and an AI coding room. They show what exists in the project now. They do not by themselves establish which resources were supplied at the start, developed during the workshop, or revised afterwards. The complete controller program must therefore not be described as something participants necessarily received before building.

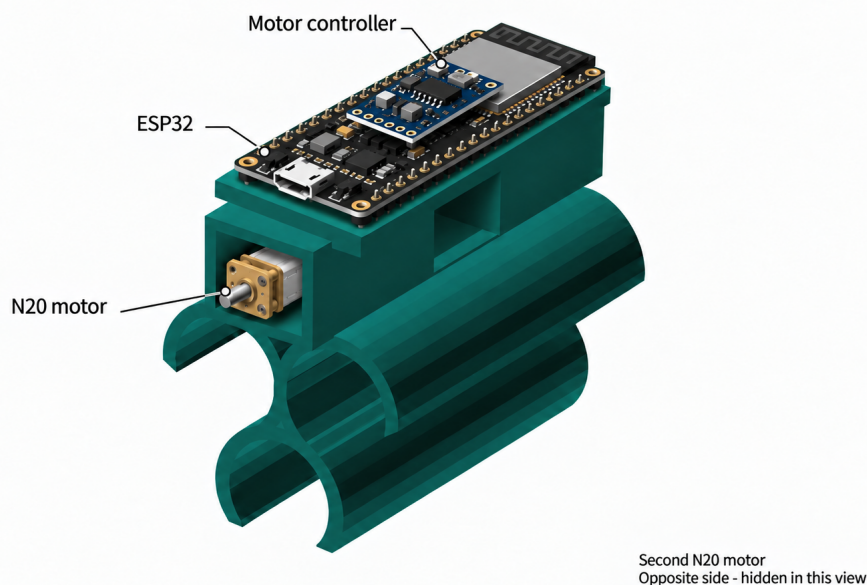
The project is treated as a rolling robot. The older tutorial supplies the spherical-robot concept and four-session outline, while the Research folder supplies the concrete design and software evidence. The organizer has now described the physical skeleton used: a flat top covered by the ESP32 board, with the motor controller stacked above it, pockets holding N20 motors, and three lower slots holding two batteries and a central screw for added weight. The exact matching CAD file and any outer shell remain to be identified.

Related: [Both workshops \(source note\)](#) · [Flying Drone Workshop](#)

The skeleton actually used

The organizer confirms this physical arrangement:

- **Flat top:** ESP32 covered the whole platform; motor controller stacked above it.
- **Motor pockets:** two N20 motors occupied pockets on opposite sides.
- **Three bottom slots:** two held batteries; the middle slot held a screw used to add weight.



Illustrative layout: ESP32 spans the top; motor controller stacked above it. Second N20 motor hidden on the opposite side. Dimensions and exact mesh unverified. Batteries and weighting screw omitted.

This was a different skeleton from the `droneBody.stl` and `droneBodyNew.stl` variants previously illustrated. The account of the actual build takes precedence over those earlier example images. See the [component layout](#) and the [source record](#) for the distinction between the physical description and the exact model file.

Read the documentation

- [Research source files](#) records the inspected CAD, controller code, and AI implementation, including differences from the older tutorial.
- [Relational arrangement](#) explains how these resources configure relations and which alternative routes are evidenced.
- [Session plan](#) retains four four-hour lessons and proposes a 30-minute exploration period in each.
- [Materials and interaction](#) separates existing artifacts, code targets, and unverified physical equipment.
- [Research framing](#) explains relations, configuration, and RQ1-RQ3.
- [AI instruction rewriting](#) relates the existing coding room to the proposed negotiation process.
- [Documentation guide](#) links the recording templates.

What is established and what remains open

Element	Evidence and status
Starting point	The ESP ball was built from scratch; it was not a supplied drone kit. The authorship and timing of individual design and code contributions remain to be documented.
Duration	Four-hour lessons, as specified by the organizer.
Number of lessons	Four, retained from the earlier tutorial; 16 hours in total.
Electronic exploration	A 30-minute period in each ball lesson is proposed by extending the flying-workshop approach.
Skeleton used	ESP32 covers the flat top, with the motor controller stacked above it; N20 motors occupy opposite-side pockets; and three bottom slots: two batteries and a central screw for added weight. Confirmed by the organizer. The exact matching CAD file remains to be identified.
Design and wheel files	Three body CAD sources, one wheel CAD source, 20 body STL files, and 11 wheel STL files exist. The used skeleton differs from the two variants previously pictured. The selected wheel file and fit checks remain unrecorded.
Motors	Regular N20 motors, confirmed by the organizer. Exact variant and mechanical fit remain to be recorded.
Controller and driver	ESP32-WROOM board covering the top platform, with the motor controller stacked above it, is confirmed. The code names ESP32-WROOM and TB6612FNG; the exact installed board and driver models remain unconfirmed. The XIAO ESP32 Sense/DRV8833 combination belongs to the older tutorial.
Batteries and added weight	Two batteries in the lower slots and a screw in the middle slot for added weight. Battery type, wiring, screw size, and measured effects remain unrecorded.

Element	Evidence and status
Manual control	Touch and mouse controls, speed adjustment, and stopping logic are implemented in the browser-controller source. Hardware behaviour has not been tested for this documentation.
AI interface	A coding-room implementation exists with eight profiles and saved conversations. This establishes software preparation, not confirmed attendance or an operating workshop service.
Soldering and board showcase	Established for the flying workshop; their inclusion here remains open.

Two distinct AI activities

Generative AI interface in the project. The Local AI folder contains a coding tutor with a local website and conversation storage, using an external model service. Its default prompt addresses programming, electronics, and technical learning. It supplies a concrete interface through which instructions might be discussed, but the relational revision and negotiation procedure remains a proposed use of that interface.

Planned recognition activity. Teachable Machine gesture or object recognition appears in the earlier tutorial. No trained recognition model or connection to robot commands was found in the inspected project files. It remains an optional extension requiring preparation.

Research opportunities grounded in the files

The body and wheel variants provide a basis for examining how design choices change component fit, assembly work, and repair access. The browser controller configures how a participant holds, releases, and combines movement commands. The coding room configures a route from an individual question through a shared host and request queue to an AI answer. Each offers specific relations to trace: who can modify a design, who can operate a control, whose knowledge is consulted, and who decides whether a suggestion will be used.

Use the episode records to distinguish resources prepared beforehand, designs or code developed during the build, and later revisions. Record when an alternative became available and who could choose it; a file found today is not proof that it was offered at the start.

Details to settle

- ☐ Record who developed the body, wheels, electronics, and code, and when; identify the versions used and their print and fit checks.
- ☐ Record the exact ESP32 and motor-controller models, the two batteries' type and wiring, and the central weighting screw; their placement is now confirmed.
- ☐ Confirm participants, available devices, facilitation, and room arrangements.
- ☐ Establish whether the coding room is deployed for this workshop and how AI suggestions will be evaluated.
- ☐ Decide whether the recognition extension, soldering, or board showcase is included.
- ☐ Confirm arrangements for any participant records.

The original [tutorial \(source note\)](#) is preserved as a historical plan. Its painting-through-gesture idea remains undeveloped in the inspected sources.

03 Relational arrangement

[Back to the workshop](#) · [Research framing](#)

The **ESP ball was built from scratch**. Its workshop is a **relational arrangement** in which components, body and wheel geometry, ESP programming, tools, people, and support were brought into relation to make the robot possible. The organizer confirms this starting point; the attribution and sequence of particular contributions still need to be recorded.

Building from scratch does not mean working without inherited resources or doing every operation alone. Components, software libraries, fabrication tools, examples, and collaborators all shape what can be designed. The research question is how decisions and work were distributed while the system was being developed.

This account uses the CAD, firmware, and AI application in Research/BallDroneWorkshop, identified in [Ball Workshop Source Files](#). The older tutorial supplies historical curriculum intentions; current instructions retain four-hour lessons. The files are evidence of the project's designs and implementations, not proof that a complete solution was provided before the workshop. Their creation, revision, and use need a chronology alongside physical tests and participation records.

What relations and configuration mean here

Relations describe how participants, collaborators, things, and conditions depend on, support, constrain, or authorize one another. **Configuration** describes the particular form and organization of those relations: who acts, what assistance is available, whose explanation counts, who checks, and who can change the method. **The arrangement** includes the wider situation that sustains this activity, including preparation, time, space, and institutional conditions.

A list of equipment cannot establish the configuration by itself. The same people and tools can support different ways of working when their roles, access, sequence, or decision rights change. The sections below trace the starting choices, prepared alternatives, and concrete relations to investigate. [The research framing](#) develops these working definitions and explains configuring as ongoing work.

Building from scratch and assembling a kit

In the ball project, the design work included bringing the body, wheels, components, control behavior, and software into a compatible system. The current files make some resulting decisions inspectable. They do not establish who made each decision or whether it was made before, during, or after a particular lesson.

The [flying drone \(source note\)](#) starts with a kit. Its supplied parts and assembly route carry decisions made upstream into the workshop, while the existing complete program provides a further starting resource. Its confirmed progression is soldering the kit board, learning to program the ESP with that program, then developing participant-chosen features with AI.

These starting points distribute design and integration work differently. Building from scratch can make more integration decisions part of the activity while increasing dependence on technical knowledge, tools, and help. A kit can make some operations easier to enter while stabilizing component and assembly choices. Neither starting point establishes equitable participation by itself: examine whose preferences shape decisions, who can reopen them, and who provides the work that makes alternatives possible.

How the ball arrangement was configured

N20 motors establish a concrete material relation. The organizer confirms their use in the ball. The selected holders and wheel openings must relate to the actual motor body and shaft. Record who established those interfaces, measured or checked them, and revised the geometry when needed. The motor designation alone does not establish the particular shaft, gearing, or fit. [The STL images](#) make the carrier and wheel geometry visible for this discussion.

The used skeleton configured a specific relation among electronics, movement, power, and added weight. The organizer confirms a flat top covered by the ESP32 board, with the motor controller stacked above it, pockets holding the N20 motors, and three bottom slots holding two batteries and a screw in the middle to add weight. The skeleton's geometry carried and located these components. The electronics formed a stack: the ESP32 occupied the top platform and the motor controller sat above it, making access to the lower board dependent on how the stack was mounted. This allocation of space made some placements available and established dependencies among the holders, actual components, wiring, and assembly work.

The screw's role is especially useful for the relational account: it participated as added weight in this configuration. Its contribution cannot be inferred from the object name alone. The batteries likewise contributed physical mass as well as serving as power sources. Their placement below the electronics makes weight distribution part of the arrangement to examine; no measured improvement in balance or movement has been supplied.

The human contributions include choosing that layout, placing and retaining components, making connections, and assessing the result. Record who performed and could revise those tasks, and what prompted any change in weight or placement. This is how the material configuration connects to the distribution of action, knowledge, responsibility, and authority. The organizer's description establishes the used layout; the exact CAD revision and particular tests remain to be linked.

The other CAD variants form a separate design record. Three body SCAD sources and multiple STLs expose alternative holders and mounting arrangements. The previously illustrated `droneBody.stl` and `droneBodyNew.stl` are different from the skeleton described as used. Preserve them as comparisons, with their own preparation and use status. Their presence does not establish a complete spherical outer shell or a sequence of classroom revisions.

The controller code records a hardware and interaction configuration. `doneESP.ino` names ESP32-WROOM and TB6612FNG and supplies browser-based touch and mouse control. It embodies decisions about direction combinations, speed adjustment, sustained presses, release, and stopping. The resulting interface embodies these decisions even when its user does not read the code. The current complete file does not establish that the implementation existed before participants began programming. The older tutorial's XIAO/DRV8833 pairing is a different plan, not the code target verified here.

The coding room configured access to technical assistance. The Local AI implementation organizes questions through eight selectable profiles, profile-specific conversations, a host computer, an external model service, and a request queue. Two simultaneous answers are allowed by default. The code's tutor prompt and supplied conversation history shape what knowledge is available to the model; the model does not automatically inspect the build. As an interpretation, these choices establish an individual question-and-answer route and shared infrastructure on which access and waiting depend. Whether that route supports collaborative negotiation must be observed.

The timetable and facilitation configure how those resources can be used. Four four-hour lessons retain the original curriculum structure. The proposed 30-minute exploration period and instruction-negotiation process add possible openings for participant-selected goals and methods. They are teaching proposals, not behaviours implemented automatically by the controller or coding room. The organizer's provision of computers, assistance, and discussion time remains necessary.

Some possibilities are still curriculum ideas. Teachable Machine recognition, alternative shell closures, and painting through gesture appear in the earlier tutorial, but their implementations were not established in the inspected Research files. They remain distinct from the CAD variants, manual controller, and coding interface that are present in source.

Documented alternatives and their preparation history

The Research folder establishes existing designs and implementations that could support alternatives. Which were prepared in advance, developed during the build, or produced later remains to be recorded. A digital artifact's existence today is separate from its availability at a particular lesson, physical readiness, and participants' authority to choose it.

Alternative route	Existing evidence and timing to establish	What the alternative changes
Compare and adapt body/carrier designs.	Three editable body sources and 20 body STL files exist. The organizer confirms the used flat-top layout with motor pockets, two batteries and a central weighting screw. Matching its exact file and recording fit checks remain open.	Makes component placement and mounting available for comparison; changes who must measure, modify, print, and evaluate the design.
Select or revise a wheel.	One editable wheel source and 11 wheel STL files exist. BOSL2 is referenced by the source; material and shaft fit are unverified.	Offers existing geometry for possible reuse while retaining dependencies on software, printing, and physical checking.
Operate the robot with touch or mouse controls.	Both input routes, speed adjustment, and stopping logic are implemented in the controller source. Operation on the actual robot is unverified.	Changes the device and input action through which movement is initiated; neither input is assumed to suit every participant.
Ask a technical question through the coding room.	The website, tutor prompt, queue, and dialogue-storage implementation exist. Deployment and workshop use are unconfirmed.	Adds a source of explanation and proposals, mediated by the organizer's host, account, network, and queue.
Provide model access through different network arrangements.	The coding-room README describes shared-internet and separate host/participant network arrangements. Installed equipment and live connectivity have not been checked.	Changes who maintains connectivity and whether access to the model depends on a shared network or a separate host connection.
Use recognition, another shell closure, or a painting interaction.	These are possibilities in the earlier tutorial. Matching models, integration, or tested mechanisms were not established by this inspection.	Would introduce additional material and software dependencies; they are not counted as ready alternatives.

An existing CAD variant is not automatically evidence of an option prepared or offered to a child. Record who selects the file, whether participants can inspect or modify it, and what support makes it usable. Likewise, the coding room is an implemented interface; the relational instruction-rewriting process is a proposed way to use it, not an implemented decision protocol.

Relations within this configuration

These relations follow from the inspected sources and identify what to investigate in practice.

Relation	How it is configured in the sources	What to investigate
Participant, component, and body model	Holder and mounting geometry positions parts and establishes fit dependencies.	Who chooses the variant, measures the actual part, and can authorize a design change?
Participant, wheel source, and printer	An editable wheel and exported variants provide starting geometry; the source requires a modelling library.	Who can edit or print it, and what happens when software or physical fit prevents use?
Participant, browser control, and firmware author	Touch/mouse events, sustained commands, speed controls, and stopping logic define an implemented interaction.	Can the participant use, question, or revise the input method while retaining workable control?
Participant, AI tutor, and conversation record	The server supplies one profile's recent dialogue to a coding-tutor prompt and stores messages locally.	Which knowledge is included, whose contributions are left outside the dialogue, and who evaluates the suggestion?
Group, host, network, and queue	Model access depends on the shared host/account and network; the default queue serves two answers concurrently.	Who maintains access, who waits, and how does waiting affect opportunities to ask, test, or discuss?

The same movement task in two configurations

The implemented browser interface supplies a concrete example for analysis: press and hold a direction, then release or use the stop control. In one possible workshop configuration, each participant is expected to operate that interface unaided, while the facilitator retains authority to change the code. The activity therefore depends on the participant using the provided input and on the facilitator accepting any proposed revision.

In another possible configuration, participants discuss how control will be shared, try the implemented touch or mouse route on available devices, and ask the coding tutor to explain the relevant control logic. A participant can direct an agreed collaborator or propose a software change, while the group and facilitator evaluate whether it preserves the intended behaviour. The AI contributes an explanation or proposal; it does not itself authorize a change or operate the robot.

This comparison is illustrative. It distinguishes implemented input routes from unimplemented interface changes and negotiated roles; it does not reconstruct the original workshop sequence. Whether the second arrangement expands participation depends on participant influence, the resources actually supplied, and who takes on the additional work.

What remains open and what requires wider changes

Possible negotiations include carrier geometry, wheel selection, control input, the division of work, and the explanation of a step. Some require more than editing instructions: a different holder needs a suitable design and print, a new control needs implementation and testing, and an AI-supported discussion needs working model access. Recognition and outer-shell changes remain extensions requiring their own preparation.

The actual room, support staff, institutional conditions, and participant requirements are not yet documented. Record these as dependencies when confirmed. The four-hour duration alone does not show that assistance, access, or alternative equipment will be available.

Distinguish preparation from what participants can do

Before each lesson, record **which option has moved from a written possibility to an available resource, who prepared it, and what remains necessary to use it**. During the lesson, record whether it was offered, who could choose or refuse it, and what happened. Include alternatives participants invented that were not anticipated by the organizer.

Manual robot control, the coding tutor, and the proposed recognition activity are distinct. A record should identify which system was actually used and which relation changed, rather than attributing every change to generative instruction rewriting. The exact evidence and remaining uncertainties are listed in [Ball Workshop Source Files](#).

04 Session plan

[Back to the ball workshop](#) · [Both workshops \(source note\)](#)

The **ESP ball was built from scratch**. This outline retains four **240-minute** lessons as the documentation framework and connects the project's design, electronics, programming, and assembly work. It is not a kit-assembly sequence, and the files that exist now are not assumed to have been supplied as completed solutions at the start.

The detailed lesson organization below remains a proposal based on the earlier tutorial, not a verified chronology of the original build. [Ball Workshop Source Files](#) identifies the existing CAD, controller code, and AI coding room. Record when each became available and who developed it. Recognition remains an optional extension, and **30 minutes of self-directed electronics exploration** is proposed in each lesson.

Configure the session before participants arrive

Use the [workshop arrangement account](#) to establish the starting conditions for each lesson. Identify which instruction and design versions will be used, which people and tools support each operation, who checks the result, and what decisions participants can change. The timetable allocates access to activities; it also needs to leave room for participants to question how those activities are organized.

For each alternative, distinguish a written idea from something ready to use. Check that the required materials, time, assistance, and authority are available; explain the available choices to participants and invite other proposals. Where a route remains unavailable, record the missing dependency. This is proposed facilitation practice, not a claim that every listed alternative has already been prepared.

A repeatable four-hour rhythm

Time	Minutes	Activity
0-20	20	Welcome, revisit prior work, introduce the aim, and check available support.
20-45	25	Demonstration, equipment orientation, and discussion of starting instructions.
45-100	55	First making activity, including time to question and revise a step.
100-115	15	Break.
115-145	30	Self-directed electronics exploration with the available kits.
145-200	55	Second making activity, testing, or revision.
200-210	10	Break.
210-230	20	Share the result, review contributions, and record changes.
230-240	10	Pack away and agree the next step.
Total	240	Four hours, including the proposed 30-minute exploration period.

Participants choose their experiment during the exploration period. Help, shared work, and unfinished investigations remain legitimate parts of that time. Choose available low-voltage kit activities appropriate to the group's agreed equipment arrangements; do not make progress on the main robot a condition of joining in.

Lesson 1 Modelling and printing the robot

Introduce the ball-shaped robot and explain how shell, wheels, electronics, and control will come together. Treat the body/carrier and wheel sources as evidence or references for the design work. Develop and compare geometry against the intended component arrangement, recording which starting examples, if any, were available. Check source completeness and modelling dependencies before selecting a file; the inspected designs do not establish a complete outer shell.

Discuss what the design assumes about assembly: who can reach a connector, hold the shell, align a part, or reopen the object for repair. If a step does not fit, consider changes to the object, tools, division of work, or instruction. Use the proposed AI workflow only if AI use has been agreed.

Keep the 30-minute electronics exploration period. Submit models for printing; printer time may extend beyond the session. Prepared parts can support fit checks while participants' parts are queued.

Record: a model version, one design decision and its contributors, and what needs to be printed or changed before assembly.

Lesson 2 Electronics and motor control

Introduce the controller, motor driver, the N20 motors used in the ball, and the power arrangement using the actual components. The existing project file `doneESP.ino` targets ESP32-WROOM and TB6612FNG, whereas the older tutorial names XIAO ESP32 Sense and DRV8833. Establish which parts are present and which code applies before using a wiring procedure.

Develop and explain ESP motor-control behavior as part of building the ball from scratch. Connect intended movement and stopping to code, wiring, and observed results. The current `doneESP.ino` is a complete implementation that can document or support this work; it is not evidence that the original lesson began by uploading a supplied complete program. Record actual starting code, changes, authorship, and assistance. Participants can identify parts, interpret diagrams, make suitable connections, predict behaviour, operate controls, and check results. Agree how those contributions are shared. Where an instruction is revised, retain its original and negotiated versions.

During the exploration period, participants pursue their own question with the available electronics. End with a controlled demonstration of whichever motor functions have been verified.

Record: the tested configuration, how the group detected and resolved a problem, and any required support that remained unavailable.

Lesson 3 Assembly and remote control

Bring the printed parts and electronics together. Investigate component placement, access to the power switch and connectors, shell closure, and the effect of placement on movement. Select and record the actual body/carrier and wheel files used. If an outer shell is included, establish its geometry and closure separately; the earlier closure options are not verified prepared parts.

Use the browser-controller interface developed for the project, after checking it against the actual hardware and network, to investigate touch or mouse commands, movement, speed, and stopping. Record what the mechanism actually does; do not assume that every intended direction is achieved merely because a command exists.

Protect the electronics exploration period. If an assembly task becomes inaccessible or impractical, discuss an alternative arrangement without automatically taking all decisions away from the participant.

Record: an assembled state, observed motion, a repair or access issue, and who could influence the next change.

Lesson 4 AI supported revision and interaction

This placement comes from the proposed teaching outline. It does not establish that AI was first introduced after the original ball build. Record whether AI supported initial design or programming, later revision, or both.

Use the project's coding room, if deployed and checked for the session, to examine a concrete making instruction, body-model question, or control-code question. Keep the original instruction and AI response, discuss dependencies and alternatives together, and record the version participants agree to try. The tutor's reply contributes to the decision; it does not establish that a proposed physical or software change works.

The earlier tutorial's Teachable Machine activity can be included as an extension only when its model and command integration have been prepared. No such implementation was found in the inspected Research files. If used, record recognition and generative instruction rewriting as separate activities.

Keep the final electronics exploration period. Close with a demonstration of the tested result and a participant account of which design or making decisions they could influence.

Record: the actual interface and files used, suggestions accepted or rejected, any changed distribution of work, and what remained untested. If the AI service is unavailable, record that dependency and continue the discussion with the available instructions and collaborators.

After each lesson

Use a [session record](#) for actual activities and timing. Use an [instruction-change record](#) for relevant episodes. The four lessons are a teaching proposal; the notes do not imply that the hardware integration, AI interface, or research procedure has already been validated.

05 Models and materials

[Back to the workshop](#) · [Research source files](#)

The ESP ball was built from scratch. This record draws on `/Users/philipn1/Desktop/Research/BallDroneWorkshop` to describe the designs and implementations now present. It separates those artifacts from evidence of what was available at the beginning, who developed it, and what was used or tested physically.

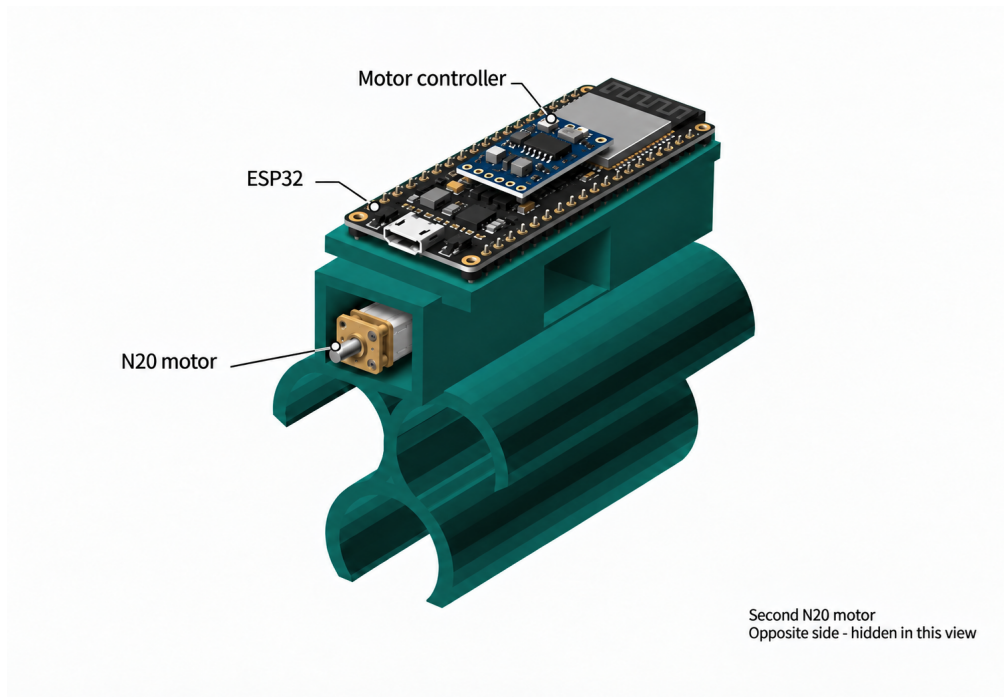
The skeleton used in the workshop

The organizer describes a **different skeleton from the two carrier variants previously pictured**. It had a flat top, motor pockets, and three slots underneath. This is an account of the configuration actually used, rather than an inference from a filename.

Position in the skeleton	What was installed	Role in the arrangement
Flat top and electronics stack	ESP32 board covers the whole top; motor controller sits above the ESP32	The platform carries the electronics stack. This placement shapes mounting and access to the lower board and its connections.
Motor pockets on opposite sides	Two N20 motors	The pockets located the motors in relation to the skeleton and wheels. Exact shaft fit and retention remain to be recorded.
Two of the three bottom slots	Two batteries	The lower holders accommodated the power sources. Battery count and placement are confirmed; battery type and electrical wiring are not.
Middle bottom slot	A screw used as added weight	The screw's documented role was to add mass in the middle. Its size, mass, retention, and effect on motion have not been recorded.

The flat top, pockets, and lower holders allocate different work to the printed geometry: carrying the electronics, locating motors, accommodating batteries, and positioning added weight. Assembly still depends on people placing, connecting, securing, and checking those components. The screw should be documented by its role here as a weight, rather than assumed to be only a fastener.

Illustration of the component layout



AI-assisted illustration of the organizer's described layout: ESP32 covers the top platform; motor controller stacked above it. N20 motors occupy opposite sides, with the second hidden. Exact sizes, board models, and skeleton file remain unverified. Batteries, weighting screw, and wiring are omitted.

The platform carries the electronics stack; the pockets locate the motors. Access to the lower board depends on the stack's mounting, which remains to be recorded. Connecting, securing, testing, and deciding how to change it are work within the relational arrangement. This illustration explains placement; it is not a wiring guide. See the [image provenance and full prompt \(local source\)](#).

Dimensions and scale verification

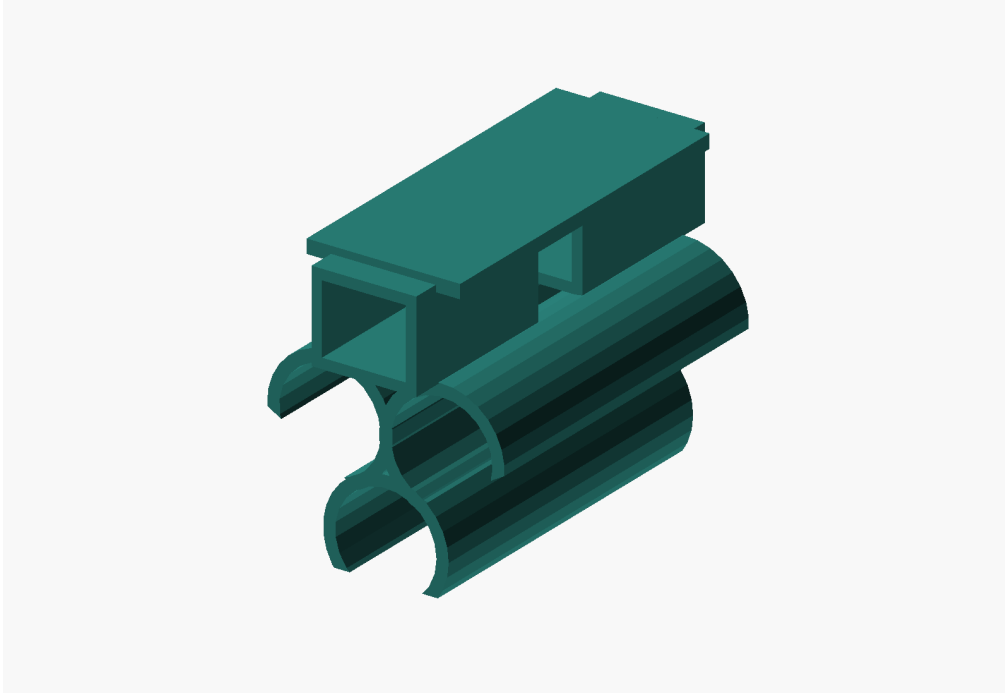
The selected candidate mesh, `droneBodyNewtopESPTriple.stl`, has overall bounding dimensions of **33.962 × 59.000 × 41.800 model units (X × Y × Z)**. Its topmost surface spans **22 × 51 model units** in X and Y; this is a bounding extent, not a guarantee of usable mounting area. These values were measured from the STL vertices, not from the generated illustration. STL files do not specify units: the values correspond to millimetres only if that is the intended import and print scale.

The organizer identifies the ESP32-WROOM family and regular N20 motors, and confirms that the ESP32 covers the whole top with the motor controller stacked above it. These relative placements are reflected in the image; exact sizes are **not verified to scale**. Accurate component proportions require the actual ESP32 board, motor-controller board, and N20 motor dimensions, confirmation of the skeleton file used, and the print scale. The current SCAD source is not assumed to reproduce this STL: its plate dimensions differ. Until those details are supplied, use the image to discuss placement and relations, not to check clearance or fit.

STL views of the ball components

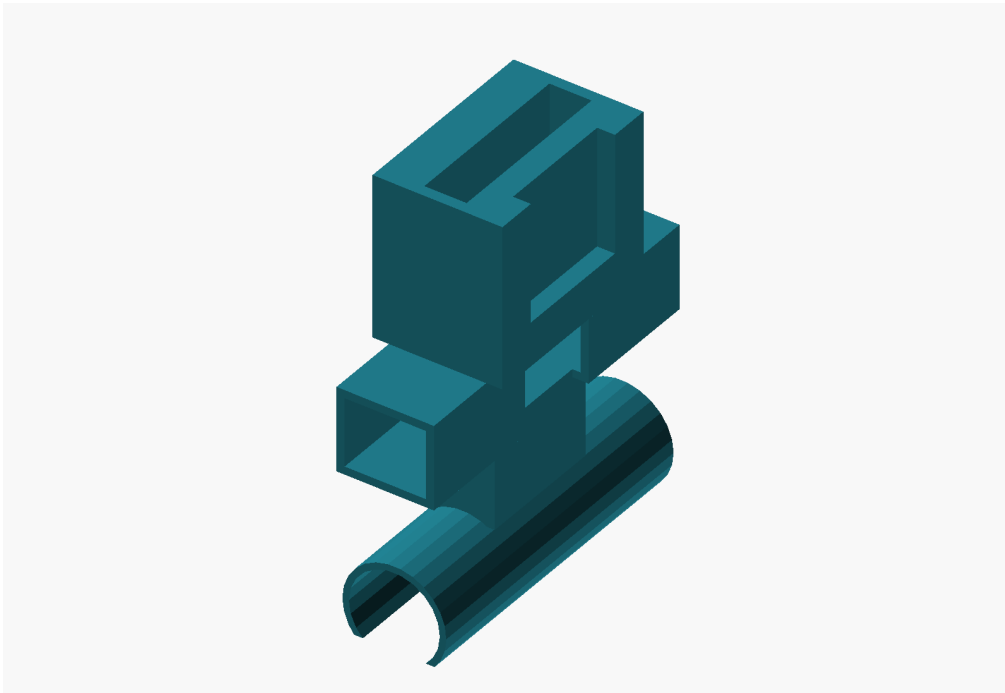
These images are direct renders from the Research folder. The first image is a **flat-top candidate** matching the broad description; the exact file used remains unconfirmed. The `droneBody` and `droneBodyNew` images show **other project variants**, not the skeleton the organizer says was used. They remain comparative source material and must not be labelled as the physical workshop build. The wheel image also remains a project variant; its exact use and fit have not been established. Color is for readability, and each image is framed independently.

Flat-top skeleton candidate



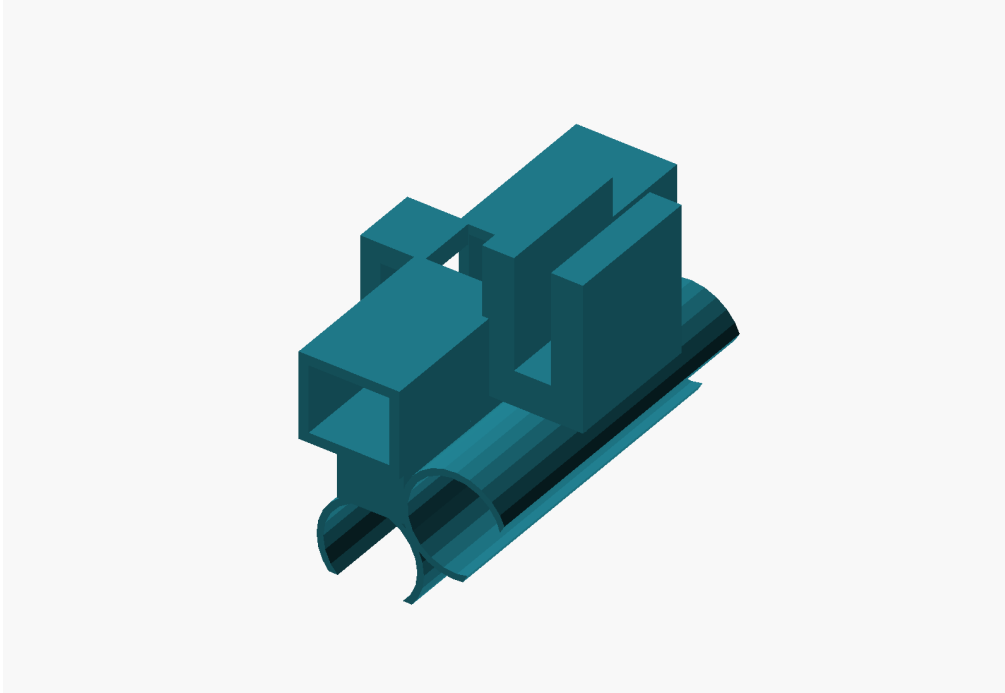
Rendered from `droneBodyNewtopESPTriple.stl`. This candidate matches the described flat-top, motor-pocket, and three-slot layout; the exact mesh used in the workshop is still awaiting identification. Electronics, batteries, and the weighting screw are not part of this STL rendering.

Carrier from droneBody



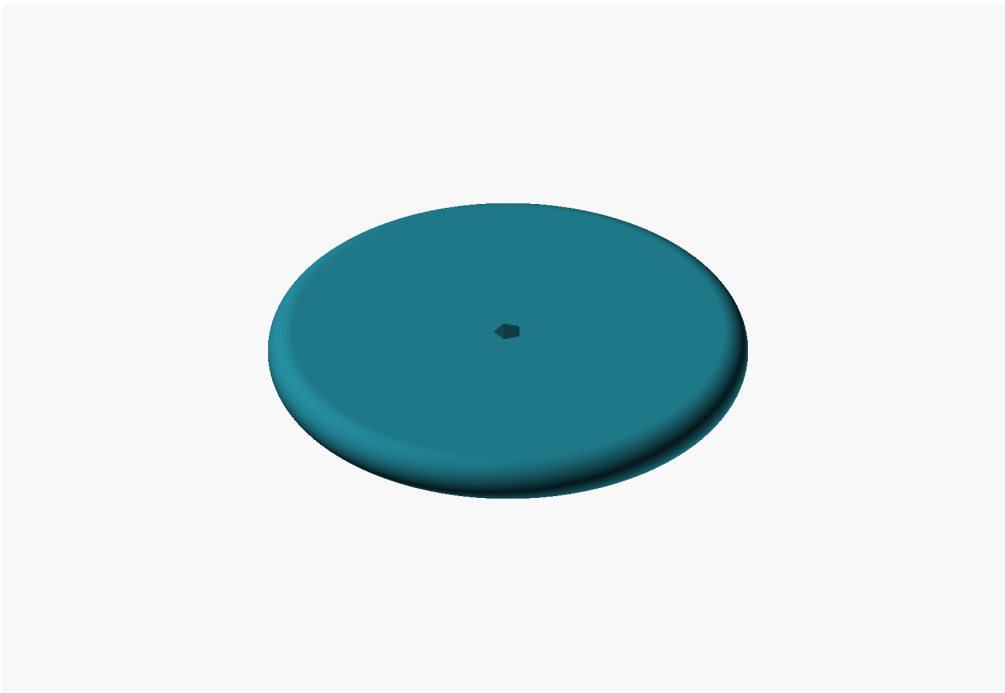
Source: `droneBody.stl` (local source). The visible holders and openings make component placement and assembly access available for discussion.

Carrier from droneBodyNew



Source: *droneBodyNew.stl* (local source). Compare the arrangement of holders and open space with the other variant. The filename alone does not establish when it was used or whether it was the final design.

Wheel



Source: *Wheels.stl* (local source). The shaft opening makes the wheel-to-N20-motor connection a concrete fit question. The specific motor shaft and printed fit still need to be recorded.

Project materials and remaining dependencies

Area	Evidence in the Research folder	What remains to establish
Printed skeleton	Organizer confirms the flat-top skeleton, motor pockets, and three lower slots. Three body SCAD sources and 20 STL files document project variants.	Exact mesh/source corresponding to that physical layout, fit checks, and any outer shell. The two earlier carrier illustrations depict other variants.
Wheels	Wheels.scad and 11 wheel STL files. The source uses BOSL2.	Available modelling library, selected mesh, material, shaft fit, and contact with the rolling mechanism.
Controller	Organizer confirms an ESP32-WROOM board covering the whole flat top; the complete development-board version is not yet identified. doneESP/doneESP.ino is labelled for ESP32-WROOM.	Exact ESP32 board model and matching tested build.
Motors	The organizer confirms that the ball used N20 motors.	Exact motor variant, voltage, gear ratio, shaft dimensions, and the fit of the selected holders and wheels.
Motor driver	Organizer confirms a motor controller stacked on top of the ESP32. The code specifies TB6612FNG and two independently driven motor channels.	Exact installed driver, compatible power, and observed movement.
Power	Organizer confirms two batteries occupying two of the three bottom slots.	Battery type, voltage, connections, protection, and power distribution. The count and placement do not establish series or parallel wiring.
Added weight	A screw occupied the middle bottom slot to add weight, as confirmed by the organizer.	Screw dimensions, mass, retention, and any observed effect or adjustment.
Browser controller	The firmware embeds touch and mouse buttons, a speed slider, and stopping logic.	Tested devices, network access, and whether participants can operate or revise the interface.
AI coding room	Website, server, and conversation-storage code exist in Local AI.	Host readiness, model access, network arrangement, participant devices, and the facilitation process.
Exploration kits	Proposed in the ball session plan by extending the flying-workshop approach.	Contents and availability for this programme.

The older XIAO ESP32 Sense/DRV8833 pairing remains in [TUTORIAL PLAN \(source note\)](#) as historical context. It is not the target named by the inspected controller code. Soldering and the circuit-board showcase are not assumed to be included merely because those resources are planned for the flying workshop.

How the developed manual control is configured

The code describes this route: **participant presses a browser control → the browser sends a command over Wi-Fi → the ESP32 drives the two motor channels**. The controller joins an existing Wi-Fi network. Its interface supports touch and mouse presses, direction combinations, a speed adjustment, and a stop button. Release events send a stopped state, and the firmware includes stopping logic for lost connectivity or expired commands.

These are implementation facts from source inspection. They do not establish that a particular assembled robot responds correctly. They also define a participation question: the interface expects a person or agreed collaborator to sustain and release a press. Changing the physical division of work or the input interface changes that relation and must preserve a workable stopping arrangement.

How the project AI interface is configured

A participant selects a profile and submits a question through the coding-room website. The host assembles context from that profile's conversation, requests an answer from an external model service, and stores the dialogue locally. A queue limits simultaneous answers to two by default. The source prompt positions the AI as a technical tutor and restricts it to supplied conversation content.

These choices distribute knowledge and responsibility. Participants supply the question and decide what to do with the answer; the organizer supplies the host, account, network, and access conditions. The AI does not automatically inspect the physical build or edit the project. A fluent answer is therefore not evidence that the proposed change matches the actual parts.

The README describes several network arrangements, including a shared network with internet access and arrangements separating the participant network from the host's internet connection. Those are documented infrastructure alternatives; the deployed arrangement has not been checked.

Recognition remains an extension

The earlier tutorial proposes gestures, objects, or poses in Teachable Machine, followed by a mapping from recognized labels to robot commands. This implementation was not found in the inspected project files. Retain it as a proposed interaction activity with its own preparation and testing, distinct from the coding room and from manual browser control.

Preserve the configuration used

Record the exact body and wheel files, controller-code version, actual components, control device, and AI interface configuration used in a session. Link these in [the session record](#). The [arrangement account](#) explains how these choices distribute action and authority; it does not substitute for a physical build record.

06 AI and instruction changes

[Back to the workshop](#) · [Both workshops \(source note\)](#)

The ESP ball was built from scratch. AI could support establishing a design or program as well as revising it; the current files alone do not establish when AI was used or who used it. Record its actual role in the build rather than importing the flying kit's confirmed sequence of complete-program teaching followed by AI extensions.

This proposed workflow uses AI to help make the assumptions in instructions discussable. The aim is to support participants in identifying, evaluating, and negotiating alternatives. It is not yet confirmed whether children and facilitators will use AI together or facilitators will use it during preparation.

Existing interface and proposed research use

The Local AI folder in Research/BallDroneWorkshop contains an implemented coding-room interface. Its server uses a coding-tutor prompt, profile-specific dialogue, a request queue, and saved conversations. See [the inspected source files](#). The site and storage are local, while model requests require an external service.

This provides a concrete interface for technical questions; the negotiation process below is a proposed research use. The default tutor prompt does not ask the model to document distributed authority, and ordinary chat records do not automatically establish which instruction people agreed to use or what happened physically. Record those decisions and outcomes in the workshop templates.

The model works from supplied conversation content rather than automatically inspecting the circuit, design files, or workshop. Include the relevant verified context in the prompt and keep the participant's evaluation and the facilitator's technical checks visible. Source existence does not establish deployment or participant use.

Relate the step to the workshop arrangement

Start from [the documented workshop configuration](#), including alternatives already prepared. Record what resources, roles, and decisions the original step assumes. Ask whether AI proposes a new route, retrieves one already supplied, or changes who can choose among existing routes. An instruction rewrite may also reveal a need to change the timetable, material provision, or support arrangement; save that dependency even when it cannot be resolved during the session.

Start with a specific moment

Use a step that someone wants to understand, question, or change. A mismatch might concern reach, visibility, wording, dexterity, confidence, available time, missing equipment, or a preference about how to work. Ask what would make the task workable; a participant need not disclose a diagnosis to propose a different arrangement.

Retain the original instruction before asking AI to rewrite it. Otherwise, the changes to its assumptions can become difficult to trace.

A proposed revision process

1. **Describe the intended result.** Establish what the step is trying to accomplish and which technical constraints have been verified.
2. **Make dependencies visible.** Name required tools, materials, support, information, time, and permission. Identify what is unavailable or unknown.

3. **Collect alternatives.** Invite participant suggestions and, where AI is being used, ask it for a small set of possible arrangements.
4. **Evaluate together.** Discuss who would do and check each part, what resources it needs, and whether the participant wants to work that way. A technically unsuitable option is recorded with the reason it was rejected.
5. **Agree a version to try.** The participant can reject or revise suggestions. The facilitator checks equipment constraints and explains any limits. The AI does not authorize a procedure.
6. **Try and document.** Record what actually happened, including unplanned help, failed attempts, and further changes. Save the agreed instruction as a new version.

Example to discuss

Starting instruction: “Fit the wheel to the motor shaft.”

The wheel source and exported variants make this a concrete task to discuss. The instruction leaves open which variant is intended, how fit is established, who positions the parts, who checks the result, and who can propose a change.

Illustrative negotiated version: “Identify the intended wheel and motor together. Agree who will position the parts and who will check their fit. If the available wheel does not fit the actual shaft or the operation does not suit the participant, pause and consider an inspected alternative, assistance, or a design revision. Record which option was chosen and what still needs testing.”

An AI explanation can help identify dependencies or explain the model. It cannot establish physical fit from a filename. This example is a proposed discussion structure, not evidence that a variant was tested or used.

Reusable prompt for the facilitator

Help us reconsider one making instruction. Our aim is to make its dependencies visible and identify feasible ways to share or change the work. Original instruction: [paste the exact step] Intended result: [describe the result] Available tools, materials, time, and support: [list what is actually available] What does not fit, or what the participant wants to change: [describe the task or preference without identifying the child] Verified technical constraints and equipment rules: [list them; mark unknowns] Identify the instruction's assumptions. Offer up to three possible arrangements. For each, explain who would act, what knowledge or assistance is needed, who would check the result, and which decisions remain open to the participant. State extra resources and unresolved technical questions. Do not assume that independence is the goal, invent available support, or treat an unverified connection method as safe. Include retaining the original method if it fits. End with a short editable instruction for the option we choose.

What to save

Retain the original step, the prompt and AI response if used, human edits, rejected options, the agreed instruction, and the observed result. Record the tool or model and date when known. AI output is a proposal; distinguish it from what participants agreed and what they actually did.

Use the [instruction change template](#). If AI is used only before the workshop, document that timing accurately: the workshop then examines how participants encounter AI-prepared instructions, rather than claiming live AI negotiation.

07 Research framing

[Back to the workshop](#) · [Both workshops \(source note\)](#)

The ball workshop is itself an arrangement through which action, knowledge, responsibility, and authority are distributed. The selection of its object, people, tools, materials, instructions, design files, schedule, and institutional conditions configures participation before the first making task begins. The research therefore examines both how the workshop was configured and how its relations are retained or reconfigured in practice.

The theoretical account and author-year citations below are drawn from the supplied research background. The publications have not been independently checked for this documentation; full bibliographic entries remain to be added.

Relations and configuration

For this study, **relations** describe how people, things, and conditions depend on, support, constrain, inform, or authorize one another in accomplishing an activity. A participant consulting a diagram, a holder stabilizing a board, a facilitator accepting a test result, and a timetable limiting access to a printer are different relations. Naming the people and objects present is a starting inventory; the relational account explains what their connections make possible and on what terms.

A relation has a particular character. Help can be requested or imposed; an instruction can be treated as a suggestion or a rule; a measurement can be available to everyone while only one person's interpretation is accepted. The analysis therefore asks both what is connected and how that connection works. Relations need not be equal or reciprocal, and some important dependencies involve people absent from the room, such as the author of a design file or the supplier of a component.

Configuration describes the particular organization of those relations at a given moment. It includes who can do what, with which resources, in what sequence, whose knowledge is accepted, who must check or repair a result, and who can change the method. The same participants and equipment can form different configurations when these conditions change. Providing an additional tool changes the inventory; making it available at a particular station, assigning access to it, and deciding who may choose it configures its relation to the activity.

Arrangement names the wider, provisional situation through which making becomes possible. A configuration is a way that arrangement is organized. These are working analytic distinctions for the workshop documentation: they allow the account to move between a specific relation, its organization with other relations, and the wider dependencies sustaining the workshop.

Configuring as ongoing work

Configuring begins before participants arrive. Selecting components for a build from scratch, writing an instruction, preparing an example, arranging access to a computer, reserving time, and deciding who checks work establish an initial distribution of possibilities. These choices are partly deliberate and partly inherited from component interfaces, software, existing teaching practices, budgets, and institutional conditions. The organizer has influence over the arrangement without having complete control over it.

The configuration is also sustained or changed during use. Participants may share a task differently from the plan, reinterpret a diagram, wait for help, or decline an activity. A preparation document records the intended configuration; observation is needed to establish the configuration enacted in practice. Repeated routines can stabilize relations even when no written rule fixes them - for example, if all technical questions consistently go to the same person.

Material and social conditions work together. An editable file enables a possible revision, but access to the software, knowledge of which dimensions matter, available printing time, and permission to use the changed design determine whether that revision can become part of the

activity. “Open to change” therefore needs a specific account of what can change, who can initiate it, and what it requires.

What counts as reconfiguration

Reconfiguration changes how the activity is accomplished or governed. A change may redistribute physical work, access to knowledge, responsibility for checking, authority to decide, or several of these together. These dimensions should be examined separately: asking a facilitator to perform an operation can preserve a child's control over the design, or it can transfer the whole decision to the facilitator. The resulting object alone does not distinguish those situations.

An instruction rewrite is evidence of a proposed change. Whether the arrangement changed depends on how the new instruction was used. Clearer wording may alter access to knowledge while leaving other relations stable; an AI-generated alternative may change nothing if its resources are unavailable or participants cannot choose it. Record the particular relation that changed rather than treating every rewrite as a reconfiguration of equal significance.

Distributed action does not imply equal decision-making power or remove human accountability. A tool can stabilize or measure; an AI can propose a route; people still establish how these contributions are used and who is responsible for consequential decisions. Equitable participation is investigated through whose contributions count, whose preferences affect the configuration, and how the work and costs of making alternatives available are shared.

The workshop as a configured arrangement

The ESP ball was built from scratch, as confirmed by the organizer. [Ball Workshop Source Files](#) identifies the actual Research project artifacts and their evidential limits. These files do not establish which resources preceded a lesson and which arose through development. Trace the configuration being built as well as subsequent reconfiguration; the relational interpretations below remain analysis, and the facilitation practices remain proposals.

The workshop arrangement account identifies the documented starting choices and the preparation status of alternative routes. It treats selecting a material, supplying an example, reserving exploration time, or assigning checking to a facilitator as decisions that shape participation. The organizer and researcher are therefore part of the arrangement they examine.

Preparing alternatives also configures the field of possible action. Record whose anticipated needs informed those alternatives when known, which assumptions they retain, and whether participants can propose routes beyond them. A prepared option, an option actually offered, and an option used are different forms of evidence.

Establish the starting configuration before analysing an instruction-revision episode. Distinguish a route supplied in advance from one proposed by a participant, facilitator, or AI during the activity. For RQ1, this matters because AI may repeat an existing alternative rather than originate the change. For RQ2 and RQ3, examine whether the resources and authority needed to use that alternative were available.

Making as a relational accomplishment

A relational view treats making as a sociomaterial arrangement rather than an individual action (Oosterlaken, 2009). Action, knowledge, responsibility, and authority are distributed across that arrangement. Drilling a hole, for example, can involve one person positioning the material, another operating the drill, a clamp stabilizing the work, and someone else checking the result.

In this workshop, a functioning circuit can similarly depend on a child interpreting a diagram, a peer identifying a component, a holder keeping the board steady, a facilitator assisting with a joint, and a meter making a connection testable. The documentation should make those contributions visible.

Configurations and mismatch

Relations are configured rather than given. Instructions and workshop routines may presume a particular bodymind (Price, 2015), prior knowledge, access to equipment, and availability of support. When those assumptions do not fit, a difficulty may be attributed to the participant even though changing the arrangement could change what is possible.

The supplied background connects this problem to makerspaces that privilege independent access to and operation of fabrication technologies (Vossoughi et al., 2016; Steele et al., 2018; Andrews & Boklage, 2024; Allen et al., 2023). The workshop therefore examines how help, shared work, and different ways of contributing are recognized, rather than treating independence as the sole measure of competence.

Instructions within the arrangement

An instruction set is a partial, durable account of a particular way of making: an edited record of its author's build (Tseng & Resnick, 2014). It names some steps, tools, and relationships while leaving others unstated. Fixed documents can also make revision difficult (Oehlberg et al., 2015; Tseng & Resnick, 2014).

Instructions are a constitutive part of the arrangement, not a sufficient explanation of exclusion on their own. Through the actions and tools they specify, they help distribute competences between people and things (Akrich, 1992). For example, “solder the component in place” may leave unspoken who holds it, how orientation is checked, who can request help, and who decides the joint is ready.

AI and the possibility of revision

AI rewriting may preserve these assumptions or make alternatives discussable. The supplied background identifies work on adaptation to individual disabilities or skills (Jerry et al., 2026; Leder et al., 2026; Padma Reddy et al., 2026). Individual adaptation alone does not establish who identifies alternatives, evaluates them, or has authority to change the activity. Automated guidance also raises questions about control and excessive automation (Zhao et al., 2025; Shneiderman, 2020).

Changing how a task is done, by whom, or with what assistance can be understood as design after design (Ehn, 2008). A rewritten instruction cannot supply missing tools, money, time, or institutional permission (Hook et al., 2014; Meissner et al., 2017). It can, however, make those dependencies explicit so participants can judge what is feasible and negotiate support.

Research questions

RQ1. How does generative AI reshape the relations through which action, knowledge, responsibility, and authority are distributed in instruction-mediated making?

Trace what changes between the original instruction, the AI suggestion, the negotiated version, and the activity as carried out. Include cases where AI preserves a restrictive assumption or moves decisions to the facilitator.

RQ2. How can AI-mediated making instructions be designed to make dependencies visible and support makers in identifying, evaluating, and negotiating alternative ways of accomplishing making tasks?

Examine whether the instructions identify required resources and support, offer feasible alternatives, explain tradeoffs, and leave room to reject or revise suggestions.

RQ3. What new possibilities for equitable participation in making emerge when previously stabilized relations can be reconfigured?

Look for changes in whose contributions are recognized and who has a say. Also record unavailable alternatives, increased burdens on helpers, and occasions when an apparent choice does not give the participant meaningful influence.

What to look for in an activity

Dimension	Observation prompt
Action	Who positions, connects, operates, observes, or checks? What does the tool or fixture make possible?
Knowledge	Whose experience, explanation, diagram, measurement, or AI suggestion informs the next step?
Responsibility	Who is expected to prepare resources, verify a result, identify an error, or resolve a problem?
Authority	Who can propose a change, reject a suggestion, choose an alternative, or stop the activity?

What would support a claim about participation

The proposed unit of analysis is an **instruction-revision episode**: a moment when a step is questioned, retained, or changed. Record the initial arrangement, the alternatives considered, the decision process, and what followed.

A completed circuit or successful robot movement alone does not establish equitable participation. More informative evidence would show, for example, that a child's proposal changed the method, that assistance preserved their influence over the work, or that a previously invisible contribution became recognized. These are possibilities to investigate, not anticipated findings.

Use the [instruction change template](#) to connect each episode to the research questions. Retain episodes without AI as useful context, without treating them as a controlled comparison unless a study design establishes one.

08 Documentation guide

[Back to the workshop](#) · [Both workshops \(source note\)](#)

Keep the record readable enough to use between lessons. Write one short session account, then create a separate instruction-change note for episodes that illuminate a research question. A small number of well-described episodes is more useful than a long list of unexplained observations.

Record the starting workshop arrangement

Use the [arrangement account](#) as a starting record, and update it when preparation is confirmed. Record who selected the tasks, tools, files, schedule, and support arrangements; what reasons or constraints informed those choices; and whose needs were anticipated or left unaddressed. Separate documented reasons from later interpretation.

For each alternative, distinguish **anticipated**, **resourced**, **offered**, and **used**. Name the person who prepared it, its supporting artifact or resource, and any remaining dependency. Record which choices participants could influence and whether they proposed something outside the prepared options. These distinctions help trace reconfiguration without assuming that an available file or an AI suggestion created an available activity.

Trace the build from scratch

The organizer confirms that the ESP ball was built from scratch. For each body design, circuit, program, and interface, record who developed it, with what resources or assistance, and when it became available. Distinguish preparation before the workshop from work during sessions and later revisions. Do not infer a complete starting solution from the current `doneESP.ino` or the number of CAD variants.

For past sessions, reconstruct this chronology only from available records and attributed participant or organizer accounts; leave gaps explicit. Record whether AI helped establish a design, implement code, or revise an existing arrangement. The flying drone's kit-based sequence must not stand in for the ball's history.

Before a lesson

Duplicate the [session record template](#) and give the copy a date and lesson title. Link the instruction version, relevant design files, and planned materials. Note resources or support that are still unavailable.

If the workshop produces research data involving children, use the agreed institutional process for consent, assent, storage, and access. The current materials do not establish that these arrangements are in place. This template does not authorize recording. Avoid including names, identifying images, or personal information in AI prompts.

During a lesson

Write brief factual notes without requiring participants to narrate everything they do. Capture a moment when someone questions a step, proposes a change, accepts help, refuses a suggestion, or takes a different role. Record when no workable alternative is available as well as when a change succeeds.

Separate **observation** from **interpretation**. "A participant asked a peer to hold the board and chose the component position" is an observation. "The shared arrangement preserved their control over placement" is an interpretation to examine against their account and the rest of the episode.

After a lesson

Record actual timing, including whether the planned 30-minute exploration period occurred. Summarize what was made or tested, what remained unfinished, and what needs to change before the next session. Save an instruction-change note when the original step, alternatives, negotiation, and outcome can be described.

Invite participants to correct the account of their contribution in an appropriate way. Preserve disagreement rather than forcing one interpretation. Do not label a session equitable simply because everyone finished an object.

Reusable notes

- [Session record template](#) for timing, activities, exploration, and follow-up.
- [Instruction change template](#) for a specific revision episode and its connection to RQ1-RQ3.

The templates are ordinary Obsidian notes. Duplicate them when needed; no additional plugin is required. Leave the originals blank.

Source materials and their limits

The primary artifact evidence comes from `/Users/philipn1/Desktop/Research/BallDroneWorkshop:` body and wheel CAD, exported meshes, `doneESP/doneESP.ino`, and the coding-room README and application source. [Ball Workshop Source Files](#) links each inspected source and identifies the claims it supports.

The preserved [tutorial \(source note\)](#) supplies the historical four-session outline and recognition ideas. Its XIAO/DRV8833 hardware and single-battery intention must not be substituted for the inspected ESP32-WROOM/TB6612FNG code target or treated as a verified physical build. The current conversation confirms four-hour lessons and the actual skeleton layout: ESP32 covering the whole flat top with the motor controller stacked above it, N20 motors in opposite-side pockets, and two batteries plus a central weighting screw in three bottom slots. Retain that physical account separately from uncertain model filenames and wiring details.

The research background supplies the theoretical framing and RQ1-RQ3. The 30-minute exploration period and relational instruction-negotiation process are proposed teaching arrangements. Soldering and the circuit-board comparison remain unconfirmed for this programme.

Source inspection establishes existing files and implemented code paths, not physical readiness, attendance, deployment, or participant outcomes. Existing chat contents and participant names were not inspected for this documentation. If dialogue is later used as research evidence, apply the agreed process and distinguish the profile record from verified authorship and negotiated action.

Literature record

The research framing preserves the supplied author-year citations. Add complete references and check the claims against the original publications before using that note in a paper, ethics submission, or formal proposal. No titles, publication details, or DOIs have been invented.

09 Sources and image provenance

[Back to the workshop](#) · [Relational arrangement account](#)

The primary project source is `/Users/philipn1/Desktop/Research/BallDroneWorkshop`. This record identifies files inspected for the workshop documentation and distinguishes their contents from the earlier Obsidian tutorial and proposed teaching activities.

Starting point and evidence chronology

The organizer confirms that the **ESP ball was built from scratch**, while the flying drone uses a kit. This is the source for that distinction. The organizer also confirms regular N20 motors and the ESP32-WROOM family, and describes the skeleton actually used: a flat top covered by the ESP32 board, with the motor controller stacked above it, pockets containing two motors on opposite sides, and three bottom slots containing two batteries and a central screw used to add weight. That skeleton differs from the `droneBody.stl` and `droneBodyNew.stl` variants previously illustrated. The exact motor variant, electronics models, battery wiring, and matching CAD filename are not inferred from this physical description. The ball's CAD and controller files should be treated as project artifacts whose authorship and timing need to be recorded, not automatically as a completed kit supplied at the outset.

For each relevant file, distinguish **prepared before a session**, **developed during the build**, and **revised later**. An existing complete program or set of exported variants does not settle this chronology. Building from scratch also does not imply that each participant wrote every line or worked without examples, libraries, or assistance.

What the project files establish

Source inspected	Evidence used in the documentation
<code>droneBody.scad</code> (local source)	Editable body geometry with named electronics, motor, battery, and controller holders. This establishes an editable project design, not a completed spherical enclosure.
<code>droneBodyNew.scad</code> (local source)	A second editable carrier design with different motor-case geometry and electronics positioning. This supports a record of design alternatives without establishing their print or fit results.
<code>droneBodyNewtopESP.scad</code> (local source)	Another body source with three battery-holder instances and a mounting plate. It calls <code>chipCube()</code> without defining it in the file; the source needs checking before being treated as a complete renderable design.
<code>Wheels.scad</code> (local source)	Editable wheel geometry using BOSL2. The source specifies a wheel and hub with a central hole. Library availability and printed fit are not established by this inspection.
<code>doneESP/doneESP.ino</code> (local source)	Controller implementation labelled for ESP32-WROOM and TB6612FNG. Includes a browser interface, touch and mouse handling, speed adjustment, direction combinations, release-to-stop behaviour, and a connection or command-timeout stop. These are code behaviours, not demonstrated hardware results.
Local AI/README.md (local source)	Implemented Codex Coding Room: eight profiles, separate conversation histories, a local website and database, remote model requests, and several network arrangements. The document describes setup; it is not a session observation.

Source inspected	Evidence used in the documentation
Local AI/server.js (local source)	Implementation of a coding-tutor prompt, participant-specific conversation context, a request queue with two concurrent answers by default, and restricted answer generation. The prompt does not itself implement the relational instruction-rewriting procedure.
Local AI/db.js (local source)	Database code for users, conversations, and messages. It establishes a mechanism for retaining dialogue, not a dedicated record of instruction versions, negotiated decisions, or physical outcomes.
Local AI/public/app.js (local source)	Participant-facing conversation interface with copyable code and queued-answer feedback. Interface code does not establish how it was used in a workshop.

The directory also contains **20 body STL files and 11 wheel STL files**. These are existing digital artifacts; their role in preparation or development requires a dated account. Names such as `droneBody2Batteries.stl`, `droneBodyBiggerBalancer.stl`, and `Wheels80mm.stl` help locate variants but do not establish their chronology, intended final status, dimensions, or tested performance. No final build should be selected solely because its filename includes “New”, “Good”, or “Corrected”.

Differences from the earlier tutorial

The preserved tutorial (source note) names XIAO ESP32 Sense and DRV8833. The inspected controller code instead names **ESP32-WROOM and TB6612FNG**. The documentation uses the latter as the code target and retains the former as an earlier plan. The organizer confirms an ESP32-WROOM board covering the whole flat top, with the motor controller stacked above it; the exact board versions remain to be recorded.

The earlier tutorial specifies an intended single shared battery. The organizer now confirms **two batteries in lower slots, with a screw used as weight in the middle slot**. This physical account takes precedence for battery count and placement. It does not establish battery type, series or parallel wiring, voltage, or power distribution. A CAD module called `batteryHolder()` does not establish that every matching slot held a battery; the central slot’s actual role was to hold the weighting screw.

The project contains body/carrier and wheel designs. A complete spherical shell, its closure, and tested shell-to-carrier integration are not established by the inspected sources. The earlier tutorial's closure options therefore remain proposals.

The earlier statement that ball-specific models and an AI interface had not been established is replaced by this file evidence. Physical prints, operating deployments, classroom use, and participant outcomes remain unconfirmed. Eight software profiles do not prove that eight children attended.

Separate the two AI activities

The **coding room is implemented in source** and provides an existing interface for technical questions; when it became available to participants remains to be recorded. The proposed relational instruction-rewriting process still requires a facilitation method and an account of how its suggestions are evaluated and negotiated.

Teachable Machine gesture or object recognition is described in the older tutorial. A trained model and connection from its outputs to the robot were not found in the inspected project files. It remains a proposed extension, not an implemented feature of `doneESP.ino`.

The coding-room website and storage are local; its model requests require an external service and internet access from the host. Separate profiles organize conversations, but the README states that selecting a name is not password authentication. This matters when describing who can access and attribute a record. Existing chat contents and participant names were not used to infer workshop activity.

Images rendered from STL files

The workshop notes include direct OpenSCAD renders of these exact STL files. The `droneBody.stl` and `droneBodyNew.stl` images depict other project variants. A new direct render of `droneBodyNewtopESPTriple.stl` supplies a flat-top candidate with motor pockets and three lower slots. Its visible geometry matches the broad description; it is not yet identified as the exact mesh used. No mesh geometry was changed or parts added. Color and camera framing are presentation choices; these are model views, not photographs or evidence of physical fit. Framing is adjusted separately for each file, so the views are not a shared-scale comparison.

STL source	Image in this workshop folder
<code>droneBodyNewtopESPTriple.stl</code> (local source)	<code>ball-flat-top-candidate-stl.png</code> (local source) - candidate matching the described layout; exact use unconfirmed.
<code>droneBody.stl</code> (local source)	<code>ball-carrier-original-stl.png</code> (local source)
<code>droneBodyNew.stl</code> (local source)	<code>ball-carrier-new-stl.png</code> (local source)
<code>Wheels.stl</code> (local source)	<code>ball-wheel-stl.png</code> (local source)

Component layout illustration

`ball-component-layout-illustration.png` (local source) was produced with the built-in image generation tool by editing the direct `droneBodyNewtopESPTriple.stl` candidate render. It illustrates the ESP32 covering the whole top platform, with the motor controller stacked above it and N20 motors in pockets on opposite sides, following the organizer's corrections. Only the near motor is visible; the opposite motor is hidden by the body in this view. Earlier images incorrectly exposed the second motor on the adjacent visible side and placed the boards side by side; both placements have been corrected. Component sizes are not verified to scale. It is distinct from the unchanged direct STL renders above. The exact skeleton file and installed module models remain unconfirmed; the illustration does not validate dimensions, physical fit, or wiring. The reported batteries and weighting screw are omitted. The [provenance record](#) (local source) preserves the initial and correction prompts. The dimension check in [Ball Workshop Materials and Interaction](#) uses the candidate STL vertices; it does not establish the physical size of installed components.

How to use this evidence

Use project files for claims about existing artifacts and implemented behaviour; use the current conversation for the from-scratch starting point and four-hour lessons; use the earlier tutorial for the historical four-session outline. Mark the lesson adaptations and relational interpretations as proposals or analysis. Source inspection does not replace compiling, printing, hardware testing, or observing participation.

10 Session record template

[Workshop overview](#) · [Documentation guide](#)

Duplicate this note for a session and replace the title. Record actual events; leave unknowns explicit.

Session details

- Date and lesson:
- Facilitators and participant count:
- Starting instruction, design, and program versions:
- Which resources existed before this session and which were developed during it, with contributors:
- Intended activity and available materials:
- Missing resources or support:

How this session was configured

- Starting arrangement and changes since the previous session:
- Who chose the tasks, materials, instruction versions, timing, and room setup:
- Recorded reasons or constraints behind those choices, and whose needs were anticipated:
- Planned distribution of doing, explaining, checking, and deciding:
- Which decisions participants could change, and which required further resources or permission:

Alternatives prepared and made available

- Alternative route and who prepared it:
- Preparation evidence and remaining dependencies:
- Status before the session: anticipated or resourced:
- Whether and how it was offered, and who could choose or decline it:
- Whether it was used, with what support and result:
- Alternatives first proposed by participants, facilitators, or AI during the session:

What happened

Brief account of the activity, what was made or tested, and what remained unfinished:

Electronics exploration

- Actual start, finish, and duration:
- What participants chose to investigate:
- Tools, help, or collaboration used:
- Discoveries, difficulties, and participant suggestions:

Participation and decisions

- Who contributed, and how:

- Who proposed or decided changes:
- Support requested, offered, accepted, or declined:
- Choices that were unavailable and why:

Instruction changes

Link any completed instruction-change records. Note whether AI was used, and when:

Evidence and interpretation

- Observed actions and results:
- Participant account, if collected through the agreed process:
- Interpretation and alternative explanations:
- Linked artifacts permitted by the documentation arrangements:

Before the next session

- ☐ Change needed:
- ☐ Person responsible and resource required:
- ☐ Participant suggestion to revisit:

11 Instruction change template

Workshop overview · Research questions

Duplicate this note for one episode. An episode can end with the original instruction retained or with no workable alternative.

Context and original step

- Date, lesson, and linked session record:
- Task and intended result:
- Exact original instruction and source version:
- Related design or program version and its origin: prepared beforehand, developed during the build, or revised later:
- Who questioned the step, and what did not fit:

Initial arrangement

- People, tools, materials, space, and available support:
- Assumed knowledge or abilities:
- Who acted, checked, took responsibility, and made decisions:

Preparation before this episode

- Who configured the initial step and its supporting resources:
- Reasons for the starting configuration, if documented:
- Alternatives already anticipated or resourced before the episode:
- What was needed to make each usable, including support and decision authority:
- Which options had actually been offered to participants:

Alternatives considered

Option, proposer, and origin	Required resources or help	Who would act and check	Accepted or rejected and why

For each option, state whether it was prepared in advance, introduced during the episode, or adapted from an existing route. Record who could choose it and whether its required support was actually available.

AI contribution if used

- Tool or model, date, and whether used before or during the session:
- Prompt and response or link to the permitted record:
- Assumptions carried forward, introduced, or questioned:
- Human edits and rejected AI suggestions:

Negotiation and agreed instruction

- Who could propose, choose, reject, or stop:

- Participant preferences and any disagreement:
- Technical constraints, resource limits, or permissions affecting the choice:
- Exact agreed instruction and new version, or reason for no revision:

What happened when tried

- Observed actions and result:
- Unplanned support or further changes:
- Participant account, if collected through the agreed process:
- Contributions recognized or overlooked:
- Remaining dependencies or burdens on others:

Connection to the research questions

- **RQ1:** What changed or remained stable in action, knowledge, responsibility, and authority?
What evidence links this to AI rather than other changes?
- **RQ2:** Which dependencies became visible? What supported or obstructed identifying, evaluating, and negotiating alternatives?
- **RQ3:** What participation became possible or remained closed? Whose account supports this interpretation?
- Alternative explanations and limits of the evidence: