
SPACE SCIENCE KIT

The Lunar Lander — Lesson 04

Enhanced Lesson Plan | Grades 6–8 | Science, Technology, Engineering

---

LESSON OVERVIEW

Lesson Title	Design, Build & Program: The Lunar Lander Descent Simulation
Subject Area	Science, Technology, Engineering (STEM)
Grade Level	Grades 6–8 (Adaptable for 5 and 9)
Duration	3 Class Periods × 45 Minutes (135 minutes total)
Key Themes	Lunar Lander Design, Powered Descent, Sensor-Triggered Landing, Engineering Precision, Teamwork
Framework	5E Instructional Model (Engage, Explore, Explain, Elaborate, Evaluate)
Builds On	Lesson 03 — Lunar Landing Rover (familiarity with micro:bit, MakeCode, motors, Sonar:bit, and conditional logic)

SMART LEARNING OBJECTIVES

By the end of this lesson, students will be able to:

#	Objective	Domain
1	Design and build a structurally stable lunar lander model using building blocks that incorporates landing legs, a descent motor mechanism, and a downward-facing Sonar:bit to detect the simulated lunar surface.	Design / Engineering
2	Write and upload a MakeCode program that simulates a controlled multi-stage descent: high-speed approach → deceleration phase → slow final descent → Sonar:bit-triggered motor stop at a defined surface distance (e.g., 5 cm).	Programming / Technology
3	Explain the key engineering challenges of a lunar landing (precision deceleration, fuel management, terrain sensing) and connect at least two physics principles (Newton's 3rd Law, gravitational potential energy) to their lander's design and code.	Science / Knowledge
4	Collaborate in a team of 3–4 across all three days to design, build, code, test, iterate, and present a working lander simulation with clearly justified engineering decisions.	Collaboration / Communication

SUCCESS CRITERIA (I CAN STATEMENTS)

Students will demonstrate success when they can:

- I can sketch and label the key components of my lunar lander (landing legs, descent stage, thruster motor, Sonar:bit) before building.
- I can build a stable lander model with landing legs that can support the structure and a motor that simulates downward thrust.
- I can write a MakeCode program with at least two motor speed phases (high approach speed, slow final descent) and a Sonar:bit-triggered stop.
- I can explain the purpose of deceleration thrust in a real lunar landing and connect it to my motor speed changes in code.
- I can run 3 timed test cycles, record results on a Testing Log, and make at least one specific targeted improvement between runs.
- I can present my lander's design choices, coding logic, and physics connections clearly, with a live descent demonstration.
- I can give and receive structured constructive feedback using a "2 Stars & 1 Wish" format.

KEY VOCABULARY

Introduce these terms using a word wall, glossary cards, or a Quizlet Live activity at the start of Day 1:

Term	Definition	Real-World Connection
Lunar Lander	A spacecraft designed to descend from orbit and land softly on the Moon's surface.	Apollo Lunar Module, SpaceX Starship HLS, Blue Origin Blue Moon
Powered Descent	The phase of a lunar landing where the descent engine fires continuously to slow the spacecraft from orbital speed to a near-stop just above the surface.	Apollo 11's final 12-minute powered descent — "the 12 minutes of terror"
Deceleration Thrust	Rocket thrust applied opposite to the direction of motion to reduce speed for a controlled landing.	Landing engines fire downward to slow descent — Newton's 3rd Law in action
Landing Legs	Extendable structural supports that absorb the final impact and keep the lander stable on the surface.	Apollo LM had 4 legs; Blue Moon has 4 folding legs for Artemis
Descent Stage	The lower portion of a lunar lander that contains the descent engine and fuel, left behind on the surface after landing.	Apollo LM descent stage remains on the Moon to this day
Ascent Stage	The upper portion of a lunar lander that carries the crew back to lunar orbit after the mission.	Apollo LM ascent stage docked with the Command Module in orbit

Sonar:bit / Proximity Sensor	A sensor that measures distance using ultrasonic waves, used to detect the surface and trigger the landing motor stop.	Real landers use radar altimeters for the same purpose during descent
Gravitational Potential Energy	The stored energy an object has due to its height above a surface; converted to kinetic energy during descent.	A lander's potential energy must be removed by thrust before touchdown

MATERIALS & RESOURCES

Category	Item	Purpose
Hardware	micro:bit v2 (1 per team)	Main programmable controller for the descent sequence
Hardware	Nezha Breakout Board V2	Connects micro:bit to descent motor and Sonar:bit sensor
Hardware	PlanetX Smart Motor	Simulates the descent engine; speed changes represent deceleration phases
Hardware	PlanetX Sonar:bit (ultrasonic sensor)	Mounted facing downward to detect the simulated lunar surface and trigger touchdown
Hardware	USB Cables (1 per team)	Flash code from computer to micro:bit
Construction	Elecfreaks Bricks Pack (LEGO-compatible blocks)	Physical lander chassis, landing legs, and descent stage structure
Software	MakeCode (makecode.microbit.org)	Block-based / JavaScript coding IDE with PlanetX extension
Classroom	Projector / Interactive Whiteboard	Teacher demonstrations and team code sharing
Classroom	Simulated Lunar Surface (raised platform + soft landing pad)	Foam, felt, or cardboard surface placed below the lander for touchdown testing
Classroom	Lander Design Brief & Testing Log Sheets	Structured planning and iteration recording across all 3 days
Optional	Ruler or Measuring Tape	Measure Sonar:bit trigger distance and log changes in threshold between test runs
Optional	AI Tool (e.g., Claude, ChatGPT)	Vocabulary support, debugging guidance, lander mission research

LESSON STRUCTURE — 5E MODEL

Total time: 3 × 45-minute class periods. Day 1: Design & Build | Day 2: Programming the Descent | Day 3: Testing, Iteration & Presentation.

DAY 1: DESIGN & BUILD

ENGAGE — Inspire Wonder & Activate Knowledge (15 min)

Time	Activity	Teacher Actions	Student Actions
0–5 min	Entry Task	Display prompt: "A lunar lander must slow from 1,800 km/h in orbit to 0 km/h at touchdown — in 12 minutes. What engineering problems does this create?" Students respond on sticky notes.	Write or sketch a response; post on the class board. No wrong answers. Ideas will be revisited in the Explain phase.
5–12 min	Lunar Lander Video Hook	Play a 3–4 min clip of a lunar landing (Apollo 11 descent footage, or India's Chandrayaan-3 Vikram lander touchdown in 2023). Pause and ask: "What do the engines do just before touchdown? Why are the landing legs important?"	Watch and note 2 observations about the descent sequence or lander structure. Discuss with a partner.
12–15 min	Whole-Class Brainstorm	Facilitate discussion: What are the key components of a lunar lander? What must happen for a safe landing? Build a class list on the board: Components vs Landing Challenges.	Contribute to the board list; record key components and challenges in notebooks for reference during the build phase.

EXPLORE — Lander Construction (20 min)

Time	Activity	Teacher Actions	Student Actions
0–5 min	Design Brief	Distribute Design Brief sheets. Prompt teams: "Your lander must have: (1) at least 3 landing legs for stability, (2) a motor mounted to simulate the descent engine, (3) a Sonar:bit facing downward to detect the landing surface. Sketch first — then build."	Sketch and label the lander design; Builder and Coder agree on leg placement and sensor orientation before touching blocks.
5–20 min	Lander Construction	Circulate with guiding questions: "Are your landing legs spread wide enough to keep the lander stable at touchdown?" "Is the Sonar:bit aimed straight down and unobstructed?" "Is the motor securely mounted and will it drive the descent mechanism?" Encourage structural thinking over speed.	Construct the lander using building blocks. Builder leads structure; Coder checks that the motor and Sonar:bit connections will be accessible on the Nezha board. Recorder documents design decisions with reasons.

EXPLAIN — Lander Systems & Landing Challenges (10 min)

Time	Activity	Description
0–5 min	Component Explanation	Walk through major lunar lander systems using a projected diagram: descent stage (engine + fuel tanks), ascent stage (crew cabin + ascent engine), landing legs (shock absorption + stability), thrusters (attitude control), sensors (radar altimeter, cameras). Students annotate a printed diagram and match each component to its function.
5–10 min	Landing Challenges & Maneuvers	Explain the key challenges of lunar landing: (1) Precision deceleration — from 1,800 km/h to 0 in 12 minutes using only engine thrust. (2) Fuel management — Apollo 11 had 30 seconds of fuel left at touchdown. (3) Uneven terrain — landing legs must absorb impact on slopes and rocks. (4) No GPS — sensors and onboard computers must calculate position independently. Ask: "Which of these challenges is hardest to solve — and why?"

Classroom Management Tip — Day 1 Transitions

- Assign team roles before Day 1: Builder, Coder, Recorder, Presenter.
- Use a 2-minute warning timer before each phase transition.
- At the end of Day 1, teams photograph their lander and upload to a shared class folder before packing away.
- Prepare the simulated landing surface (raised platform + soft pad) before Day 3 — a stack of books with foam on top works well.

DAY 2: PROGRAMMING THE DESCENT

ENGAGE — Review & Coding Setup (10 min)

Time	Activity	Teacher Actions	Student Actions
0–5 min	Recap Quiz	Show 3 quick questions: (1) Name 2 components of a lunar lander and their functions. (2) What does "deceleration thrust" mean? (3) What is a radar altimeter used for? Teams confer and answer on mini whiteboards.	Discuss as a team; hold up whiteboard answers on teacher signal. Correct any misconceptions before coding begins.
5–10 min	MakeCode Setup & Demo	Demonstrate: connecting the descent motor and Sonar:bit (facing downward) to the Nezha board. Show how to read a Sonar:bit distance value in MakeCode and display it on the LED.	Connect hardware; open MakeCode with PlanetX extension loaded. Coder confirms: (1) Sonar:bit reads and displays a distance, (2) motor runs at a test speed,

		Demonstrate a basic motor-on block to confirm the motor runs.	before beginning the descent program.

EXPLORE — Code the Descent Sequence (25 min)

Time	Activity	Teacher Actions	Student Actions
0–15 min	Core Descent Program	Display scaffold on board: on button A pressed → Phase 1: motor high speed (simulates approach) for 3s → Phase 2: motor medium speed (deceleration) for 2s → Phase 3: forever loop: read Sonar:bit → IF distance < 5 cm THEN stop motor + display "LANDED" on LED. Circulate and support.	Coder writes the 3-phase descent program; Builder monitors hardware connections during testing; Recorder logs each programming decision with a reason.
15–25 min	Code Optimisation & Enhancement	Prompt advanced teams: "Can you add a 5-second countdown LED display before Phase 1 begins?" "Can you add a second threshold: if distance < 10 cm, switch to ultra-slow speed before the final stop?" "Can you make the LED display the current distance during descent?" Guide beginners to focus on reliable 3-phase execution first.	Test the descent sequence (hold lander above the surface; lower slowly to test sensor). Adjust motor speeds, phase durations, and Sonar:bit threshold. Coder and Builder collaborate; Recorder notes each change and its effect.

EXPLAIN — Code Sharing & Troubleshooting (10 min)

Time	Activity	Description
0–5 min	Code Showcase	Each team displays their MakeCode project for 90 seconds. The Coder explains one key decision: "We set Phase 1 to high speed for 3 seconds because real landers travel fast in the approach phase, and then we decelerate gradually — just like the Apollo powered descent."
5–10 min	Troubleshooting Circle	Class builds a shared "Bug Board": common issues (motor not stopping, Sonar:bit not triggering at correct distance, motor runs but lander tips over) and the solutions teams found. Teacher highlights that Apollo 11's onboard computer showed a 1202 alarm during descent — the crew and ground team debugged it in real time.

Guided Inquiry Prompts for Teacher Use — Day 2

- "Your motor stops before reaching the surface — is your Sonar:bit threshold too large? What would you change it to, and why?"
- "Apollo 11 had 30 seconds of fuel at touchdown. How does your motor timing represent fuel management — and what happens if Phase 1 runs too long?"
- "Could you add a third deceleration phase — ultra-slow speed from 8 cm to 5 cm? How would that make your landing more realistic?"
- "What happens if the Sonar:bit reads an incorrect value mid-descent? How could you add a safety check to your code?"

DAY 3: TESTING, ITERATION & PRESENTATION

ELABORATE — Lander Simulation: Test & Refine (20 min)

Time	Activity	Description
0–15 min	Simulated Lunar Landing — 3 Test Runs	Teams hold their lander above the simulated lunar surface (raised platform with soft pad) and run the descent program 3 times. After each run, complete the Testing Log: Did all 3 phases execute? Did the Sonar:bit trigger the motor stop at the correct height? Did the lander land stably on its legs? Make one targeted improvement between runs.
15–20 min	Advanced Enhancements (Stretch)	Teams that complete 3 successful runs choose one enhancement: (A) Add a 5-4-3-2-1 LED countdown before Phase 1 begins. (B) Program a 3-phase deceleration: high → medium → ultra-slow → Sonar:bit stop, with a second closer threshold. (C) Add obstacle avoidance: if Sonar:bit detects a side obstacle (rotate sensor), trigger a "landing abort" and display "ABORT" on LED. (D) Simulate terrain variation: test landing on surfaces at different heights and verify the sensor stops the motor correctly each time.

Testing Protocol for Teams

- Run 1: Phase check — do all 3 motor speed phases execute in sequence? Record: Pass / Partial / Fail.
- Run 2: Sensor check — does the Sonar:bit trigger the motor stop at the correct distance above the surface? Record the actual stopping height.
- Run 3: Full simulation — smooth 3-phase descent with clean sensor-triggered touchdown. Evaluate: Meets / Approaching / Not Yet.
- After 3 runs: identify the single most important improvement and implement it before the presentation.

EVALUATE — Presentations, Peer Feedback & Reflection (25 min)

Phase	Time	Description
Team Presentations	4–5 min / team	Presenter explains: (1) design choices — why landing legs are placed as they are, how the motor represents the descent engine, (2) programming logic — what each phase does and why the Sonar:bit threshold was set at that distance, (3) physics connection — how deceleration thrust, Newton's 3rd Law, and gravitational potential energy appear in their simulation. Live descent demonstration runs during the presentation.
Peer Feedback	2 min / team	Audience completes a "2 Stars & 1 Wish" card per team: 2 specific strengths observed + 1 targeted and constructive suggestion. Cards are given to presenting teams at the end of the session.
Wrap-Up Reflection	5 min	Whole-class discussion: "What was the most challenging engineering decision your team made — and how did you resolve it?" "Apollo 11 had 30 seconds of fuel left. How does that connect to your code?" "What one feature would you add if you had one more class period?"

DIFFERENTIATION STRATEGIES

Learner Group	Strategy	Concrete Example
Beginning / SEN	Scaffolded tasks with visual supports	Pre-built starter lander chassis to modify; illustrated step-by-step coding guide with screenshots; visual flowchart of the 3-phase descent logic
Beginning / SEN	Reduced scope	Focus on 2-phase descent (approach + stop) without medium speed; use sentence starters for reflection and presentation; annotate printed code instead of writing from scratch
ELL Students	Language support	Bilingual vocabulary card for all key terms; permit labelled diagrams instead of written explanations; pair with a bilingual peer where possible
ELL Students	Comprehensible input	Physical demonstrations alongside verbal instructions; hardware setup cards with annotated images for each motor and sensor connection
Average Learners	Core task completion	Complete the full 3-phase descent program with Sonar:bit trigger; land successfully in 3 test runs;

		explain 2 physics principles and 2 lander components in the presentation
Advanced Learners	Open-ended extensions	Add LED countdown; program 3-phase deceleration with dual Sonar:bit thresholds; test landing on surfaces at different heights; research and reference a real lander in the presentation
Gifted / High Ability	Challenge by choice	Program a landing abort sequence; simulate terrain variation; calculate the fuel-to-time ratio for Apollo 11's final descent and compare it to their motor timing; explore JavaScript view in MakeCode

ASSESSMENT — FORMATIVE & SUMMATIVE

Formative Assessment (Ongoing — During the Lesson)

Method	When	What to Look For
Sticky Note Entry Task	Day 1, 0–5 min	Surface prior understanding of deceleration and landing precision; identify misconceptions to address in Explain
Components vs Challenges Board	Day 1, 12–15 min	Check depth of understanding of lander systems before building begins
Design Brief Review	Day 1, Build phase	Assess whether sketches include landing leg placement, downward sensor orientation, and motor mounting logic
Mini Whiteboard Recap Quiz	Day 2, 0–5 min	Verify retention of key vocabulary and lander concepts from Day 1
Hardware Confirmation Check	Day 2, Setup phase	Confirm all teams have motor running and Sonar:bit reading a distance before the descent program begins
Circulating Questions	All days	Use guiding prompts to check conceptual understanding and redirect in real time
Code Sharing Spot-Check	Day 2, Explain phase	Verify that all 3 phases are present in code and Sonar:bit trigger is logically correct
3-Run Testing Log	Day 3, Elaborate phase	Assess iterative thinking — did teams identify specific problems and make targeted improvements between runs?
Exit Ticket (3-2-1)	End of Day 3	3 lander components/principles learned, 2 engineering decisions made, 1 connection to a real lunar mission

Summative Assessment (End of Lesson)

Criterion	Beginning (1)	Developing (2)	Achieving (3)	Exceeding (4)

Design & Build	Lander is incomplete, unstable, or Sonar:bit is not mounted downward	Lander stands but has leg instability or incorrect sensor orientation	Lander is stable with well-placed legs; Sonar:bit correctly faces downward and is unobstructed	Lander is detailed and polished; includes creative features such as folding legs, thruster details, or an ascent stage representation
Programming	Code does not run or motor does not respond	Code runs but only 1–2 phases execute or Sonar:bit trigger is missing	Code reliably executes all 3 descent phases and Sonar:bit triggers a clean motor stop at the correct height	Code includes LED countdown, dual thresholds, terrain variation testing, or a landing abort sequence; shows advanced conditional logic
Science Knowledge	Cannot name lander components or explain landing challenges	Names 1–2 components or challenges with prompting	Explains key lander systems, 2 landing challenges, and 2 physics principles independently	Connects design and code to real mission data; references Apollo 11 fuel figures, Chandrayaan-3, or Artemis HLS with specific accuracy
Collaboration	Does not contribute meaningfully to team tasks	Contributes minimally; needs frequent prompting across all 3 days	Contributes consistently; all team members participate in building, coding, and testing across all 3 days	Leads team effectively; mentors peers; adapts role as needed; resolves disagreements constructively and keeps the team on track
Presentation	Presentation is unclear; live demo does not run	Presents with some clarity; limited explanation of design, code,	Clearly presents lander design, descent code, and physics; live	Presents with confidence and technical vocabulary; responds well to questions;

		or physics connections	simulation runs successfully with visible sensor-triggered stop	makes strong connections to real lunar landing missions

TECHNOLOGY & AI INTEGRATION

Tool / Platform	How to Use in This Lesson
MakeCode (makecode.microbit.org)	Primary coding environment; block view for beginners, JavaScript view for advanced learners; use the built-in simulator to preview phase logic before flashing to hardware
micro:bit v2 + Nezha Board + Sonar:bit	Full physical computing stack; downward-facing Sonar:bit integration teaches proximity sensing and real-time conditional response — directly analogous to a radar altimeter
Simulated Landing Surface	Raised platform with soft pad provides authentic test conditions; students must calibrate Sonar:bit threshold to the actual surface height, not a simulator value
AI Chatbot (Claude / ChatGPT)	Students can ask: "How much fuel did Apollo 11 have left at landing?" or "Why isn't my Sonar:bit triggering at 5 cm?" Teach responsible AI use: verify facts, test code yourself, rephrase answers in your own words
NASA Apollo & Artemis Resources	NASA's Apollo 11 mission page and Artemis Human Landing System pages provide real engineering data for comparison with student designs
QR Code Resource Stations	Link to MakeCode starter project, Sonar:bit wiring diagram, lander component diagram, and vocabulary glossary for independent and differentiated pacing
Google Slides / Padlet	Digital portfolio for lander photos, testing log screenshots, and team reflections; Padlet allows structured peer comments across teams

---

AI Responsible Use Guidance for Students

- Use AI to help understand vocabulary, troubleshoot code logic, or research real missions — always verify the information from an official source.
- Test any AI-suggested code yourself and explain what each block does before presenting it to the class.
- When researching missions (Apollo 11, Chandrayaan-3, Artemis HLS), cross-check AI answers with NASA or ESA official websites.
- Do not copy AI-written explanations — use them as a starting point, then restate the ideas in your own words.

REAL-WORLD CONNECTIONS

Connection	Discussion Prompt
Apollo 11 — "12 Minutes of Terror" (1969)	"Apollo 11 had only 30 seconds of fuel left at touchdown and the onboard computer flagged a 1202 alarm mid-descent. The crew and mission control debugged it in real time. How does your testing and debugging experience connect to what they faced?"
India’s Chandrayaan-3 — Vikram Lander (2023)	"Chandrayaan-3 became the first mission to land near the lunar south pole in 2023 — a region of extreme terrain. How would uneven terrain affect your lander’s leg design and Sonar:bit trigger height?"
SpaceX Starship Human Landing System	"SpaceX is building a giant Starship variant to land Artemis astronauts on the Moon. It is 50 metres tall and must land on its landing legs. What engineering challenges does its size create compared to your model?"
Blue Origin Blue Moon Lander	"Blue Origin’s Blue Moon lander uses a BE-7 engine burning liquid hydrogen. Why might fuel choice matter for a lunar lander — and how does fuel relate to your motor timing in code?"
Radar Altimeters & Sonar:bit	"Real landers use radar altimeters to measure height above the surface during descent — exactly what your Sonar:bit does. What are the key differences between your sensor and a real radar altimeter?"
Careers in Aerospace Engineering	"What careers are involved in designing a lunar lander? Structural engineer, guidance systems engineer, fuel systems engineer, software developer, test engineer — which connects most to your work in this lesson?"

EXTENSION ACTIVITIES

Activity	Description	Suggested For
Lunar Lander Missions Research	Research 3 historical or planned lunar lander missions (e.g., Apollo LM, Chandrayaan-3 Vikram, Artemis HLS). Compare their descent engine systems, landing leg designs, and surface precision requirements in a 1-page illustrated summary.	Average & Advanced
Terrain Variation Challenge	Set the simulated landing surface at 3 different heights (low, medium, high). Program the lander to detect and land successfully at each height without changing the code threshold. Discuss: how did the sensor reading change, and what does this reveal about calibration?	Advanced / Gifted
Multi-Stage Deceleration Program	Program a 4-phase descent: ultra-high (approach) → high (deceleration burn) → medium (hover check) → ultra-slow (final approach) → Sonar:bit stop. Add a dual-threshold system with two Sonar:bit readings at different heights.	Advanced / Gifted

Future Lunar Exploration Presentation	Create a 3-minute multimedia presentation on the Artemis Human Landing System: mission goals, lander design, landing site selection near the lunar south pole, and the engineering challenges of landing in permanently shadowed craters.	All levels
Landing Abort Sequence	Program an emergency abort: if the Sonar:bit reads an unexpected obstacle or terrain spike during descent (value suddenly much lower than expected), reverse motor + display "ABORT" on LED. Discuss: when would a real lander abort?	Advanced / Gifted
Physics Calculation Challenge	Research: the Apollo LM descended at about 3 m/s in the final approach. Calculate: if the lander starts at 150 m altitude and decelerates from 50 m/s to 3 m/s using constant thrust, how long does the powered descent take? Show working.	Gifted

HOMEWORK / FOLLOW-UP TASKS

Task	Instructions	Due
Lander Reflection Journal	Write 150–200 words: What was the most precise engineering challenge your team solved? How did you decide on your Sonar:bit threshold distance — and what happened when you changed it? If you were designing a real lunar lander, which part of your model would you keep, and what would you completely redesign?	Next class
Real Lander Connection	Find a news article or video about a real lunar lander mission published in the last 2 years. Write 3 sentences: (1) what the mission aims to do, (2) one similarity to your lander model, (3) one key engineering difference.	Next class
Extension (Optional)	Sketch an upgraded descent sequence that uses 4 motor speed phases instead of 3. Label each phase with a name, a motor speed, a duration, and a real-world equivalent from an actual lunar landing mission.	Next class

TEACHER NOTES & TIPS

Before the Lesson

- Review Lesson 03 student work and note any teams who struggled with Sonar:bit conditional logic — provide extra setup support at the start of Day 2.
- Test all motors and Sonar:bit sensors in advance; confirm the sensor reads reliably when pointed downward at the landing surface height you plan to use.

- Pre-install the PlanetX MakeCode extension on classroom computers if internet speed is limited.
- Build the simulated landing surface before Day 3: a raised platform (stack of books or a box) with a foam, felt, or cardboard pad on top. Test that the Sonar:bit can detect it consistently from 10–15 cm.
- Print Lander Design Brief sheets, vocabulary cards, and Testing Logs for each team.
- Bookmark Apollo 11 powered descent footage and Chandrayaan-3 landing clips on a dedicated browser tab before Day 1.

During the Lesson

- This lesson builds directly on Lesson 03 — if students are not confident with Sonar:bit conditional logic, add 10 minutes to Day 2 setup for a review.
- The key distinction from Lesson 03 (rover) is that the Sonar:bit now faces downward to detect the surface below, not forward to avoid obstacles — make this explicit at the start of Day 2.
- Encourage students to measure the actual stopping height with a ruler after each test run and record it on their Testing Log — this develops precision thinking.
- Apollo 11's 1202 alarm story is a powerful classroom moment: real engineers debug under pressure, just as students are doing.
- For SEN students, the Recorder role allows full participation; they can annotate printed code screenshots and measure/log test results.
- The most common bug: Sonar:bit threshold is too small (lander crashes before stopping) or too large (lander stops too high above the surface). Use both as a calibration teaching moment.

After the Lesson

- Photograph each team's final lander and save their MakeCode project URL for a class digital portfolio.
- Compile the "Bug Board" into a shared troubleshooting guide for students to reference at home.
- Use reflection journal responses to identify any descent physics concepts that need re-teaching before Lesson 05.
- Consider hosting a "Mission Critical" class challenge: which team's lander achieves the softest landing (closest to 0 cm above the surface at motor stop)?

CURRICULUM STANDARDS ALIGNMENT

Standard Framework	Alignment
NGSS (Next Generation Science Standards)	MS-PS2-1: Newton's Laws of Motion (deceleration thrust); MS-PS3-2: Gravitational potential energy; MS-ETS1-1 to 1-4: Engineering Design Process
CSTA (Computer Science)	Level 2 (Grades 6–8): Multi-phase algorithms, proximity sensor input, conditional logic, and iterative debugging in block-based and JavaScript coding
ISTE Standards (Students)	Empowered Learner, Innovative Designer, Computational Thinker, Creative Communicator, Global Collaborator
Common Core (Supporting)	Measurement and threshold calibration in testing; data recording across 3 runs; ratio and proportional reasoning in descent phase timing; structured oral presentation
21st Century Skills (4Cs)	Critical Thinking (descent phase calibration and sensor debugging), Creativity (lander design), Collaboration (team roles across 3 days), Communication (landing simulation presentations)
Lesson Sequence	Extends Lesson 03 (Lunar Landing Rover); prepares students for Lesson 05 in the Space Science Kit series

Developed by Techtelligence Technologies LLC in partnership with Elecfreaks • Space Science Kit • Lesson 04