
SPACE SCIENCE KIT

Moon Landing Rocket — Lesson 02

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

LESSON OVERVIEW

Lesson Title	Design, Build & Program: Moon Landing Rocket 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 Missions, Trajectory & Orbital Mechanics, Engineering Design, Coding, Teamwork
Framework	5E Instructional Model (Engage, Explore, Explain, Elaborate, Evaluate)
Builds On	Lesson 01 — Rocket Launcher (familiarity with micro:bit, MakeCode, and motor control)

SMART LEARNING OBJECTIVES

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

#	Objective	Domain
1	Design and build a structurally accurate moon landing rocket model using building blocks that visually represents at least three mission stages (launch vehicle, transfer stage, lander).	Design / Engineering
2	Write and upload a MakeCode program that simulates a complete lunar mission sequence: launch, Earth orbit, trans-lunar injection, lunar descent, and landing — using motor speed changes and LED displays.	Programming / Technology
3	Explain the key phases of a lunar mission, the role of each rocket component, and at least two physics principles (orbital mechanics, deceleration thrust) that govern moon landings.	Science / Knowledge
4	Work effectively in a team of 3–4, distributing roles, iterating on designs based on test results, and presenting a coherent mission simulation to the class.	Collaboration / Communication

SUCCESS CRITERIA (I CAN STATEMENTS)

Students will demonstrate success when they can:

- I can sketch and label the key stages and components of my moon landing rocket before building.
- I can build a stable rocket model that represents at least three distinct mission stages.
- I can write a MakeCode program that controls the motor to simulate a launch-to-landing sequence.
- I can describe the five phases of a lunar mission in order and explain what happens at each phase.
- I can use the Sonar:bit sensor to trigger a landing sequence when the rocket approaches the lunar surface.
- I can give specific, constructive feedback to another team using evidence from their demonstration.
- I can reflect on my team's engineering choices and identify one concrete improvement for next time.

KEY VOCABULARY

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

Term	Definition	Real-World Connection
Lunar Module	The spacecraft stage designed to land astronauts on the Moon and return them to lunar orbit.	Apollo 11's Eagle — first crewed Moon landing, 1969
Trans-Lunar Injection (TLI)	A rocket engine burn that propels a spacecraft out of Earth orbit toward the Moon.	Used on every Apollo and Artemis mission
Orbital Insertion	A deceleration burn that slows a spacecraft enough to be captured by a celestial body's gravity.	Lunar Orbit Insertion performed by the Apollo Command Module
Descent Orbit	A lower orbit from which the lander begins its powered descent to the surface.	The final orbit before Apollo touchdown
Deceleration Thrust	Rocket thrust applied opposite to the direction of motion to slow a spacecraft for landing.	Why landers fire engines downward during descent
Trajectory	The curved path of a spacecraft influenced by gravity and thrust at each mission phase.	Mission planners calculate trajectories months in advance
Sonar:bit / Ultrasonic Sensor	A sensor that measures distance using sound waves, used to detect the landing surface proximity.	Similar sensors used in autonomous spacecraft landing systems
Iteration	Repeating the design-test-improve cycle to refine a solution — the core of engineering.	NASA engineers run thousands of simulations before every launch

MATERIALS & RESOURCES

Category	Item	Purpose
Hardware	micro:bit v2 (1 per team)	Main programmable controller for the mission sequence
Hardware	Nezha Breakout Board V2	Connects micro:bit to motors and sensors
Hardware	PlanetX Smart Motor	Simulates thrust at each mission phase
Hardware	PlanetX Sonar:bit (ultrasonic sensor)	Detects distance to trigger the landing sequence
Hardware	USB Cables (1 per team)	Flash code from computer to micro:bit
Construction	ElecFreaks Bricks Pack (LEGO-compatible blocks)	Physical model of the moon landing rocket
Software	MakeCode (makecode.microbit.org)	Block-based / JavaScript coding IDE
Classroom	Projector / Interactive Whiteboard	Teacher demonstrations and team code sharing
Classroom	Mission Stage Diagram (printed)	Visual reference for all five mission phases
Classroom	Design Brief & Testing Log Sheets	Structured planning and iteration recording
Optional	AI Tool (e.g., Claude, ChatGPT)	Vocabulary support, debugging questions, mission research
Optional	NASA Eyes on the Solar System	Visualise real mission trajectories for comparison

LESSON STRUCTURE — 5E MODEL

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

DAY 1: DESIGN & BUILD

ENGAGE — Inspire Awe & Activate Knowledge (15 min)

Time	Activity	Teacher Actions	Student Actions
0–5 min	Entry Task	Display prompt: "What do you think is the hardest part of landing on the Moon — and why?" Students respond on sticky notes.	Write or sketch a response; post on class display. No wrong answers at this stage.

5–12 min	Apollo 11 Video Hook	Play a 3–4 min clip of Apollo 11's lunar descent and landing. Pause at key moments: "What is the rocket doing here? What forces are acting on it right now?"	Watch and note 2 observations about the descent. Discuss briefly with a partner before sharing with the class.
12–15 min	Whole-Class Brainstorm	Facilitate discussion: What are the main parts of a moon landing rocket? What are the steps of a lunar mission? Build a class mind-map on the whiteboard.	Contribute to the mind-map; record mission phases in notebooks: Launch → Earth Orbit → TLI → Lunar Orbit → Descent → Landing.

EXPLORE — Rocket Construction (20 min)

Time	Activity	Teacher Actions	Student Actions
0–5 min	Design Brief	Distribute Design Brief sheets. Prompt teams: "Your model must represent at least 3 mission stages. Where is the main engine? Where is the lander? How will the motor be integrated?"	Sketch and label the rocket design; Builder and Coder agree on structure before touching any blocks.
5–20 min	Rocket Construction	Circulate with guiding questions: "How does your design show stage separation?" "Is it stable enough to hold the Neza board securely?" Encourage sketch-first, build-second.	Construct the rocket model using building blocks. Recorder documents design decisions; Presenter prepares to explain the team's rationale.

EXPLAIN — Components & Mission Phases (10 min)

Time	Activity	Description
0–5 min	Component Explanation	Walk through major rocket components: launch vehicle, service module, command module, lunar module (descent + ascent stage). Students annotate a printed rocket diagram and match each part to its mission role.
5–10 min	Trajectory & Mission Phases	Explain the 5 key phases: Launch, Earth Orbit, Trans-Lunar Injection (TLI), Lunar Orbit Insertion (LOI), Powered Descent & Landing. Prompt: "Which phase uses the most fuel — and why?" Students record phases and brief notes.

Classroom Management Tip — Day 1 Transitions

- Assign team roles before Day 1 begins: Builder, Coder, Recorder, Presenter.
- Use a 2-minute warning timer before each phase transition.

- At the end of Day 1, teams photograph their build and upload to a shared class folder before packing away.

DAY 2: PROGRAMMING THE MISSION

ENGAGE — Review & Coding Setup (10 min)

Time	Activity	Teacher Actions	Student Actions
0–5 min	Recap Quiz	Show 3 quick questions: (1) Name the 5 mission phases in order. (2) What does TLI stand for? (3) Which component lands on the Moon? Teams confer and answer on mini whiteboards.	Discuss as a team; hold up whiteboard answers on teacher signal.
5–10 min	MakeCode Setup	Demonstrate connecting the motor and Sonar:bit to the Nezha board. Show how to load the PlanetX extension. Remind teams of the motor control blocks from Lesson 01.	Connect hardware; open MakeCode and load the PlanetX extension. Coder confirms motor responds to a test block before programming begins.

EXPLORE — Code the Mission (25 min)

Time	Activity	Teacher Actions	Student Actions
0–15 min	Code the Mission Sequence	Display scaffold on board: Phase 1 (Launch) → motor high speed 3s → Phase 2 (Orbit) → motor medium speed 2s → Phase 3 (Descent) → motor slows → Phase 4 (Landing) → motor stops + LED "LANDED". Circulate and support.	Coder writes the full mission sequence; Builder monitors physical connections; Recorder logs each programming decision with a reason.
15–25 min	Optimisation & Sensor Integration	Prompt advanced teams: "Can you use the Sonar:bit to detect when the lander is within 10 cm of the surface and trigger the final motor stop automatically?" Guide beginners to focus on reliable motor sequencing first.	Test and refine the sequence. Advanced teams integrate the Sonar:bit conditional: if distance < 10 cm → stop motor + display landing icon. All teams experiment with motor speeds and pause timings.

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 programming

decision: "We slowed the motor at descent because real landers decelerate to avoid crashing."

5–10
min Troubleshooting Circle

Class builds a shared "Bug Board" on the whiteboard: common issues and the solutions teams found. Teacher highlights debugging as a core engineering skill used by every space agency.

Guided Inquiry Prompts for Teacher Use — Day 2

- "Your motor stops too quickly at landing — what does that tell you about your timing value?"
- "Real lunar landers descend at about 1.5 m/s near the surface. How would you represent that with your motor speed?"
- "What should happen in your code if the Sonar:bit reading is unreliable? How do real spacecraft handle sensor errors?"
- "Could you add a micro:bit LED countdown (5-4-3-2-1) before the launch phase fires?"

DAY 3: TESTING, ITERATION & PRESENTATION

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

Time	Activity	Description
0–15 min	Mission Simulation — 3 Test Runs	Teams run their full mission sequence 3 times, completing the Testing Log after each run. Record: Did all phases execute in order? Did motor speeds change correctly? Did the sensor trigger landing? Make targeted improvements between runs.
15– 20 min	Advanced Enhancements (Stretch)	Teams that complete 3 successful runs choose one enhancement: (A) Add a brief motor pause + restart to simulate lunar orbit insertion. (B) Program a Sonar:bit abort: if obstacle detected during descent, pause + display "ABORT." (C) Use micro:bit radio so one device acts as Mission Control, triggering the launch remotely. (D) Design and attach landing legs from blocks that fold down during the descent phase.

Testing Protocol for Teams

- Run 1: Sequence check — does each mission phase execute in correct order? Record: Pass / Partial / Fail.
- Run 2: Timing check — are motor speed changes smooth and realistic? Record specific observations.
- Run 3: Full simulation — complete sequence with sensor integration (if applicable). Evaluate: Meets / Approaching / Not Yet.

- After 3 runs: agree on the single most important improvement and implement it before the final presentation.

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

Phase	Time	Description
Team Presentations	4–5 min / team	Presenter explains: (1) design choices — how the model represents each mission stage, (2) programming logic — what each code block does and why, (3) physics connection — which real principles are demonstrated. Live mission simulation runs during the presentation.
Peer Feedback	2 min / team	Audience completes a "2 Stars & 1 Wish" card per team: 2 specific strengths + 1 targeted 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 difficult engineering decision you made?" "How does your simulation compare to the real Apollo 11 landing?" "What would you change if you had one more class period?"

DIFFERENTIATION STRATEGIES

Learner Group	Strategy	Concrete Example
Beginning / SEN	Scaffolded tasks with visual supports	Pre-built starter rocket model to modify; illustrated mission phase cards; visual step-by-step coding guide with screenshots
Beginning / SEN	Reduced scope	Focus on a 3-phase sequence (Launch, Descent, Land) rather than all 5; use sentence starters in reflection and presentation
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	Use physical demonstrations alongside verbal instructions; printed hardware setup cards with annotated images
Average Learners	Core task completion	Complete the full 5-phase motor sequence; participate in structured peer feedback; explain 2 physics principles during the presentation
Advanced Learners	Open-ended extensions	Integrate Sonar:bit landing trigger; program orbit insertion pause; add

Gifted / High Ability	Challenge by choice	LED countdown; reference a real mission in their presentation
		Design micro:bit radio mission control link; explore JavaScript view in MakeCode; write a brief mission report comparing their simulation to Apollo 11 data

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 knowledge and misconceptions about lunar landing challenges
Mind-Map Contribution	Day 1, 12–15 min	Check understanding of mission phases and rocket components before building
Design Brief Review	Day 1, Build phase	Assess whether sketches reflect mission stages, structural thinking, and motor placement
Mini Whiteboard Recap Quiz	Day 2, 0–5 min	Verify retention of mission phase vocabulary and sequence from Day 1
Circulating Questions	All days	Use guiding questions to check conceptual understanding and redirect misconceptions in real time
Code Sharing Spot-Check	Day 2, Explain phase	Verify programming logic maps to the correct mission phase; identify and address common bugs
3-Run Testing Log	Day 3, Elaborate phase	Assess iterative thinking — did teams diagnose problems and make targeted improvements between runs?
Exit Ticket (3-2-1)	End of Day 3	3 things learned about lunar missions, 2 engineering decisions made, 1 real-world connection identified

Summative Assessment (End of Lesson)

Criterion	Beginning (1)	Developing (2)	Achieving (3)	Exceeding (4)
Design & Build	Model is incomplete or does not represent mission stages	Model is recognisable but missing key components	Model clearly shows 3+ mission stages and is structurally stable	Model is detailed and polished; includes creative representations of stage separation or landing legs

Programming	Code does not run or has major errors	Code runs partially; motor sequence is incomplete or inconsistent	Code reliably simulates all mission phases with appropriate motor speed changes	Code includes Sonar:bit trigger, LED countdown, or radio link; demonstrates advanced conditional logic
Science Knowledge	Cannot name mission phases or explain forces	Names 1–2 phases or forces with prompting	Explains all 5 mission phases and 2 physics principles independently and accurately	Connects physics to real mission data; references Apollo 11 or Artemis with specific accuracy
Collaboration	Does not contribute meaningfully to team tasks	Contributes minimally; needs frequent prompting	Contributes consistently; shares responsibilities across all team members across all 3 days	Leads team effectively; supports and mentors peers; resolves disagreements constructively
Presentation	Presentation is unclear or live demo does not run	Presents with some clarity; limited explanation of design and code choices	Clearly presents design, code, and physics; live simulation runs successfully	Presents with confidence and technical language; responds well to questions; makes strong real-world connections

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 code logic before flashing
micro:bit v2 + Nezha Board + Sonar:bit	Full physical computing stack; sensor integration teaches conditional logic and real-world signal processing
micro:bit Radio (Advanced)	Pairs of micro:bits communicate wirelessly — one acts as Mission Control, one as the rocket — demonstrating real spacecraft telemetry concepts
AI Chatbot (Claude / ChatGPT)	Students can ask: "What speed did Apollo 11 descend at before landing?" or "Why isn't my if-statement triggering?" — teach responsible AI use: always verify facts and test code independently

NASA Eyes on the Solar System	Free NASA web tool to visualise real mission trajectories — students compare their simulated path to an actual lunar trajectory
QR Code Resource Stations	Link to tutorial videos, MakeCode starter project, vocabulary glossary, and mission phase diagram — supports independent and differentiated pacing
Google Slides / Padlet	Digital gallery for team build photos and reflection posts; Padlet allows structured peer comments on each team's mission plan

AI Responsible Use Guidance for Students

- Use AI tools to help you understand vocabulary, debug code, or research real missions — but always verify the information.
 - Test any AI-suggested code yourself and explain what each block does before presenting it to the class.
 - When researching missions (e.g., Apollo 11, Artemis), cross-check AI answers with NASA's official website (nasa.gov).
 - Do not copy AI-written explanations for your presentation — use them as a starting point, then restate in your own words.
-

REAL-WORLD CONNECTIONS

Connection	Discussion Prompt
Apollo 11 — First Moon Landing (1969)	"Neil Armstrong had 30 seconds of fuel left at landing. How does that connect to the precision your code needs to stop the motor at exactly the right moment?"
NASA Artemis Program	"Artemis is planning to return humans to the Moon in the late 2020s. What improvements have engineers made to the lunar lander since Apollo?"
SpaceX Starship Lunar Lander	"SpaceX is designing the Human Landing System for Artemis. How is its design different from the Apollo Lunar Module — and why?"
Autonomous Landing Systems	"Modern lunar landers use sensors and AI to land autonomously. How is your Sonar:bit sensor similar to — and different from — these real systems?"
Careers in Space Engineering	"What jobs exist on a lunar mission team? Orbital mechanics engineer, software developer, systems engineer, astronaut trainer, mission controller — which interests you most?"
Gravity Differences	"The Moon's gravity is 1/6th of Earth's. How would that change the motor power needed for a real lander compared to your model?"

EXTENSION ACTIVITIES

Activity	Description	Suggested For
Lunar Missions Research	Research 3 historical or planned lunar missions (e.g., Apollo 11, Chandrayaan-3, Artemis III). Compare their lander designs, mission objectives, and landing technologies in a 1-page illustrated summary.	Average & Advanced
Lunar Rover Design & Programming	Build a simple lunar rover from building blocks and program it using the micro:bit to navigate a simulated crater-filled surface using the Sonar:bit for obstacle avoidance.	Advanced / Gifted
Future of Lunar Exploration Presentation	Create a 3-minute multimedia presentation on the Artemis Program: mission goals, the Gateway lunar space station, the Human Landing System, and planned timelines for a sustained lunar presence.	All levels
Mission Abort Sequence	Program an emergency abort: if the Sonar:bit detects an unexpected obstacle at > 15 cm during descent, reverse the motor to simulate re-ascent and display "ABORT" on the LED screen.	Advanced / Gifted
Physics Deep Dive	Research and calculate: given the Moon's gravity (1.62 m/s^2) and a descent speed of 2 m/s, how much thrust would a 15,000 kg lunar module need for a soft landing? Present the working and result.	Gifted
Mission Control Simulation	Use two micro:bits with radio communication — one team member is Mission Control (sends go/abort signals) and one operates the rocket — simulating a real launch decision sequence.	Advanced / Gifted

HOMEWORK / FOLLOW-UP TASKS

Task	Instructions	Due
Mission Reflection Journal	Write 150–200 words: What was the most challenging part of programming your mission sequence? How did your team solve it? If you were designing a real lunar lander, what one feature from your model would you keep — and what would you change?	Next class
Real Mission Connection	Watch a 5-minute NASA video on the Artemis program (nasa.gov/artemis). Write 3 sentences identifying one similarity and one difference between Artemis and your team's simulation.	Next class
Extension (Optional)	Sketch a redesigned mission sequence that includes 6 phases instead of 5 — add a rendezvous and docking phase between lunar orbit and descent. Label all phases and write one sentence explaining each.	Next class

TEACHER NOTES & TIPS

Before the Lesson

- Review Lesson 01 student work and note any teams who struggled with motor control — provide extra support at the start of Day 2.
 - Test all Sonar:bit sensors in advance; some may require calibration or extension-specific setup in MakeCode.
 - Pre-install the PlanetX MakeCode extension on classroom computers if internet speed is limited.
 - Print Mission Phase Diagrams, Design Brief sheets, and Testing Logs for each team before Day 1.
 - Bookmark NASA Apollo 11 and Artemis video clips (3–5 minutes each) on a dedicated browser tab for quick access.
 - Prepare a "Bug Board" space on the whiteboard for Day 2's troubleshooting circle.
-
-

During the Lesson

- This lesson builds on Lesson 01 — if students are unfamiliar with MakeCode and motor blocks, add 10 minutes to Day 2 for a brief review.
 - The Sonar:bit integration is optional — ensure all teams complete the basic 5-phase motor sequence before attempting sensor coding.
 - Encourage teams to research actual Apollo descent speeds and compare them to their motor timing choices for real-world grounding.
 - If a micro:bit or motor malfunctions, use it as an authentic engineering moment: every spacecraft has redundant systems for exactly this reason.
 - For SEN students, the Recorder role is excellent for building confidence; they can annotate printed code screenshots instead of writing from scratch.
 - Keep a parking lot on the board for physics questions that arise during building — address them during the Explain phase.
-
-

After the Lesson

- Photograph each team's final rocket build and save their MakeCode project URL for a class digital portfolio.
- Compile the "Bug Board" into a shared digital troubleshooting guide students can reference at home.
- Use reflection journal responses to identify mission phase concepts that need re-teaching before Lesson 03.

- Consider hosting a "Mission Exhibition" where final builds and code are displayed for other classes or parents.

CURRICULUM STANDARDS ALIGNMENT

Standard Framework	Alignment
NGSS (Next Generation Science Standards)	MS-PS2-1: Newton's Laws of Motion; MS-PS2-4: Gravitational forces; MS-ETS1-1 to 1-4: Engineering Design Process
CSTA (Computer Science)	Level 2 (Grades 6–8): Algorithms, conditionals, sensor input, and iterative debugging using block-based coding
ISTE Standards (Students)	Empowered Learner, Innovative Designer, Computational Thinker, Creative Communicator, Global Collaborator
Common Core (Supporting)	Measurement and calculation in design sketches; data recording and comparison across 3 testing runs; structured oral presentation skills
21st Century Skills (4Cs)	Critical Thinking (mission sequence debugging), Creativity (rocket design), Collaboration (team roles across 3 days), Communication (mission presentations)
Lesson Sequence	Extends Lesson 01 (Rocket Launcher); prepares students for Lesson 03 in the Space Science Kit series

Developed by Techtelligence Technologies LLC in partnership with ElecFreaks • Space Science Kit • Lesson 02