
SPACE SCIENCE KIT

Lunar Landing Rover — Lesson 03

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

LESSON OVERVIEW

Lesson Title	Design, Build & Program: Lunar Landing Rover Navigation 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 Rover Design, Obstacle Avoidance, Sensor-Based Navigation, Autonomous Systems, Teamwork
Framework	5E Instructional Model (Engage, Explore, Explain, Elaborate, Evaluate)
Builds On	Lesson 02 — Moon Landing Rocket (familiarity with micro:bit, MakeCode, motors, and Sonar:bit)

SMART LEARNING OBJECTIVES

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

#	Objective	Domain
1	Design and build a structurally functional lunar landing rover model using building blocks that incorporates wheels, a motor drive system, and a forward-facing Sonar:bit sensor for obstacle detection.	Design / Engineering
2	Write and upload a MakeCode program that controls the rover to move forward, read distance data from the Sonar:bit, and automatically stop or turn when an obstacle is detected within a defined threshold (e.g., 15 cm).	Programming / Technology
3	Explain the key components of a real lunar rover, the environmental challenges of operating on the Moon, and at least two principles from physics or engineering (e.g., autonomous navigation, traction on low-gravity terrain) that apply to rover design.	Science / Knowledge
4	Collaborate in a team of 3–4 across all three days to plan, build, code, test, iterate, and present a working rover navigation simulation with justified design 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 rover design (wheels, motor, sensor, chassis) before building.
- I can build a rover that moves forward using motor drive and holds the Sonar:bit sensor securely at the front.
- I can write a MakeCode program that reads Sonar:bit distance values and uses a conditional to stop or turn the rover.
- I can name and describe the key systems of a real lunar rover and explain how each helps it survive on the Moon.
- I can explain two challenges of the lunar environment (e.g., low gravity, craters, no atmosphere) and how rover engineers address them.
- I can run 3 test cycles, record results on a Testing Log, and make at least one targeted improvement between runs.
- I can present my rover's design, code, and navigation behaviour clearly to the class with a live demonstration.

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 Rover	A wheeled vehicle designed to travel across the Moon's surface to conduct exploration and collect data.	Apollo Lunar Roving Vehicle (1971), China's Yutu-2 (2019)
Autonomous Navigation	The ability of a vehicle to move and make decisions independently using sensors, without human control of each action.	Mars Perseverance rover drives autonomously due to signal delay
Obstacle Avoidance	A system that detects objects in a vehicle's path and adjusts direction or speed to avoid collision.	Used in all planetary rovers and self-driving cars
Sonar:bit / Ultrasonic Sensor	A sensor that emits sound waves and measures the time for them to return, calculating distance to an object.	Used in parking sensors, submarines, and rover navigation
Traction	The grip between a wheel and the surface, allowing the rover to move without slipping.	Rover wheels have special treads for loose lunar regolith
Regolith	The loose layer of rock fragments and dust that covers the Moon's surface, creating a soft, uneven terrain.	Rovers must be designed to navigate regolith without getting stuck

Conditional Logic	An if-then programming structure: "IF sensor reads X, THEN do Y." The foundation of autonomous decision-making.	Core to all autonomous systems from rovers to aircraft
Iteration	The engineering process of testing, identifying failures, and improving the design in repeated cycles.	Every real rover undergoes thousands of test iterations before launch

MATERIALS & RESOURCES

Category	Item	Purpose
Hardware	micro:bit v2 (1 per team)	Main programmable controller for rover navigation
Hardware	Nezha Breakout Board V2	Connects micro:bit to motors and sensors
Hardware	PlanetX Smart Motor (2 recommended)	Drives left and right wheels independently for turning capability
Hardware	PlanetX Sonar:bit (ultrasonic sensor)	Detects obstacles and triggers avoidance behaviour
Hardware	USB Cables (1 per team)	Flash code from computer to micro:bit
Construction	ElecFreaks Bricks Pack (LEGO-compatible blocks)	Physical chassis, wheels, and body of the lunar rover
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 (large tray or floor area)	Obstacles (blocks, paper craters) for the navigation challenge
Classroom	Rover Design Brief & Testing Log Sheets	Structured planning and iteration recording across all 3 days
Optional	AI Tool (e.g., Claude, ChatGPT)	Vocabulary support, debugging guidance, rover mission research
Optional	Tape / Chalk for Surface Boundaries	Define the navigation zone on the floor or desk surface

LESSON STRUCTURE — 5E MODEL

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

DAY 1: DESIGN & BUILD

ENGAGE — Inspire Curiosity & Activate Knowledge (15 min)

Time	Activity	Teacher Actions	Student Actions
------	----------	-----------------	-----------------

0–5 min	Entry Task	Display prompt: "If you had to design a vehicle to drive on the Moon, what 3 features would it absolutely need — and why?" Students respond on sticky notes or mini whiteboards.	Write/sketch 3 features with brief justifications. Post or hold up for the class to see. No wrong answers.
5–12 min	Lunar Rover Video Hook	Play a 3–4 min clip of a lunar rover in action (e.g., Apollo Lunar Roving Vehicle footage or China's Yutu-2 highlights). Pause and ask: "What do you notice about the wheels? The speed? How does it handle uneven ground?"	Watch and note 2 observations about rover design or behaviour. Discuss briefly with a partner.
12–15 min	Whole-Class Brainstorm	Facilitate discussion: What are the essential features of a lunar rover? What challenges does the lunar environment create? Build a class list on the board: Features vs Challenges.	Contribute to the board list; record key features and challenges in notebooks for reference during the build.

EXPLORE — Rover Construction (20 min)

Time	Activity	Teacher Actions	Student Actions
0–5 min	Design Brief	Distribute Design Brief sheets. Prompt teams: "Your rover must have: (1) a stable chassis with wheels, (2) a mounted Sonar:bit at the front, (3) the Nezza board accessible for motor connections. Sketch first — then build."	Sketch and label the rover design; Builder and Coder agree on structure and sensor placement before touching blocks.
5–20 min	Rover Construction	Circulate with guiding questions: "Are your wheels aligned so the rover drives straight?" "Is your Sonar:bit aimed forward and unobstructed?" "Is the chassis rigid enough to hold all components?" Encourage design thinking over speed.	Construct the rover using building blocks. Builder leads; Coder checks that the Sonar:bit and motor connections are accessible. Recorder documents key decisions.

EXPLAIN — Rover Systems & Lunar Environment (10 min)

Time	Activity	Description
0–5 min	Component Explanation	Walk through major lunar rover systems using a projected diagram: chassis & suspension, wheel drive motors, power source (solar panels / batteries),

sensors (cameras, spectrometers, ultrasonic), and communication antenna. Students annotate a printed diagram and label each system's function.

5–10
min

Lunar Environment Challenges & Adaptations

Explain key lunar environment challenges: (1) Low gravity (1/6th of Earth's) — wheels must provide traction without digging in. (2) Extreme temperature swings (–173°C to +127°C) — materials and electronics must be insulated. (3) Craters and boulders — sensors detect and route around obstacles. (4) No atmosphere — no air cooling, no sound, dust clings electrostatically. Ask: "Which of these challenges did you already design for in your rover?"

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 rover and upload to a shared class folder before packing away.
- Prepare the simulated lunar surface (tray with obstacles) before Day 3 — students can help set it up at the start of Day 3.

DAY 2: PROGRAMMING FOR NAVIGATION

ENGAGE — Review & Coding Setup (10 min)

Time	Activity	Teacher Actions	Student Actions
0–5 min	Recap Quiz	Show 3 quick questions: (1) Name 2 challenges of the lunar environment. (2) What does the Sonar:bit measure? (3) What is autonomous navigation? 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 two motors (left and right drive) and the Sonar:bit to the Nezha board. Show how to read a Sonar:bit value in MakeCode and display it on the LED screen. Show a basic forward movement block.	Connect hardware; open MakeCode with PlanetX extension loaded. Coder displays Sonar:bit readings on the LED screen to confirm the sensor works before programming the navigation logic.

EXPLORE — Code the Rover Navigation (25 min)

Time	Activity	Teacher Actions	Student Actions
------	----------	-----------------	-----------------

0–15 min	Core Navigation Program	<p>Display scaffold on board: forever loop → read Sonar:bit distance → IF distance < 15 cm THEN stop motors + turn left (or right) for 1s ELSE drive forward. Circulate and support. Prompt: "What should your rover do when it detects an obstacle? Stop only? Turn? Back up first?"</p> <p>Prompt advanced teams: "Can you make the rover turn right for some obstacles and left for others depending on which side the obstacle is on?" "Can you add an LED arrow that shows which way the rover is turning?" Guide beginners to focus on a reliable stop-and-turn response first.</p>	<p>Coder writes the obstacle avoidance program; Builder monitors that wheel connections remain secure during testing; Recorder logs each programming decision with a reason.</p> <p>Test the rover on the desk or a small trial surface. Adjust distance threshold, motor speed, and turn duration to improve navigation accuracy. Coder and Builder work together; Recorder notes what changed and why.</p>
15–25 min	Code Optimisation & Testing		

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 our threshold to 12 cm because the rover needs time to stop before hitting the obstacle at our motor speed."
5–10 min	Troubleshooting Circle	Class builds a shared "Bug Board": common issues (rover drives in circles, sensor not triggering, one motor faster than the other) and the solutions teams found. Teacher highlights that real rover engineers face identical calibration challenges.

Guided Inquiry Prompts for Teacher Use — Day 2

- "Your rover stops but doesn't turn — what part of your code controls the turn, and what might be missing?"
- "If you increase your distance threshold from 10 cm to 20 cm, what effect will that have on rover behaviour — and why?"
- "Real rovers on Mars take photos of an obstacle and plan a path around it before moving. How could you simulate that decision-making with your code?"
- "What happens if your rover gets stuck in a corner with obstacles on both sides? How would you program an escape behaviour?"

DAY 3: TESTING, ITERATION & PRESENTATION

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

Time	Activity	Description
0–15 min	Simulated Lunar Surface — 3 Test Runs	Teams deploy their rovers on the simulated lunar surface (large tray or floor area with obstacles: blocks, paper craters, small objects). Run 3 timed navigation trials. After each run, complete the Testing Log: Did the rover move forward? Did it detect and avoid all obstacles? Did it get stuck? Make targeted improvements between runs.
15–20 min	Advanced Enhancements (Stretch)	Teams that complete 3 successful runs choose one enhancement: (A) Add a "sample collection" pause — rover stops at a marked location, pauses for 3 seconds (simulating sample collection), then continues. (B) Program a "lunar base return" — after navigating the course, rover turns 180° and returns to start. (C) Add directional LED arrows on the micro:bit display to show the rover's navigation decisions in real time. (D) Design and build a sample collection arm or scoop attachment from blocks.

Testing Protocol for Teams

- Run 1: Basic movement — does the rover move forward and stop when an obstacle is detected? Record: Pass / Partial / Fail.
 - Run 2: Full navigation — does the rover successfully navigate around at least 2 obstacles without getting stuck? Record observations.
 - Run 3: Optimised navigation — improved threshold, speed, or turn logic applied. Evaluate: Meets / Approaching / Not Yet.
 - After 3 runs: agree on the single most impactful 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 — how the rover incorporates features for the lunar environment, (2) programming logic — how the obstacle avoidance code works and why the threshold was set as it was, (3) physics/engineering connection — how their rover design mirrors real lunar rover principles. Live navigation demonstration on the simulated surface.
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 hardest engineering problem your team solved?" "How does your rover compare to a real lunar rover?" "What one feature would you add if you had more time?" "What does this project teach us about autonomous systems in the real world?"

DIFFERENTIATION STRATEGIES

Learner Group	Strategy	Concrete Example
Beginning / SEN	Scaffolded tasks with visual supports	Pre-built starter rover chassis to modify; illustrated step-by-step coding guide with screenshots; visual flowchart of the obstacle avoidance logic
Beginning / SEN	Reduced scope	Focus on forward movement + stop only (no turn); 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 connection step
Average Learners	Core task completion	Complete the full stop-and-turn obstacle avoidance program; navigate the simulated surface successfully; explain 2 rover systems and 2 environment challenges in the presentation
Advanced Learners	Open-ended extensions	Add directional LED feedback; program sample collection pause; optimise turning logic for both left and right avoidance; research and reference a real rover mission
Gifted / High Ability	Challenge by choice	Design a "lunar base return" navigation sequence; explore JavaScript view in MakeCode; build a sample collection arm attachment; write a brief engineering report comparing their rover to Yutu-2 or Perseverance

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 of rover design; identify misconceptions about the lunar environment to address in Explain
Features vs Challenges Board	Day 1, 12–15 min	Check depth of understanding of rover systems and lunar environment before building begins
Design Brief Review	Day 1, Build phase	Assess whether sketches include sensor placement, wheel alignment thinking, and structural planning
Mini Whiteboard Recap Quiz	Day 2, 0–5 min	Verify retention of key vocabulary and concepts from Day 1 before coding begins
Sonar:bit Display Check	Day 2, Setup phase	Confirm all teams can read and display sensor values before attempting navigation code
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 conditional logic maps to correct behaviour; identify and address common threshold and motor bugs
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 rover design features learned, 2 engineering decisions made, 1 connection to real autonomous systems

Summative Assessment (End of Lesson)

Criterion	Beginning (1)	Developing (2)	Achieving (3)	Exceeding (4)
Design & Build	Rover is incomplete, unstable, or sensor is not mounted	Rover moves but has structural issues or sensor placement problems	Rover is stable, wheels are aligned, Sonar:bit is correctly mounted and unobstructed	Rover is polished and robust; includes creative design features such as a sample arm, antenna, or solar panel representation
Programming	Code does not run or sensor is not read	Code runs but obstacle detection is inconsistent or rover does not turn	Code reliably detects obstacles and executes a stop-and-turn response	Code includes directional LED feedback, sample collection

Science Knowledge	Cannot name rover components or lunar challenges	Names 1–2 components or challenges with prompting	Explains key rover systems, 2 lunar environment challenges, and how the rover addresses them	pause, return-to-base sequence, or advanced turn logic Connects design and code decisions to real mission data; references Yutu-2, Apollo LRV, or Perseverance with specific accuracy
Collaboration	Does not contribute meaningfully to team tasks	Contributes minimally; needs frequent prompting across the 3 days	Contributes consistently; all team members participate in building, coding, and testing	Leads team effectively; mentors peers; adapts role as needed; resolves disagreements constructively
Presentation	Presentation is unclear; live demo does not run	Presents with some clarity; limited explanation of design and code logic	Clearly presents rover design, navigation code, and lunar science; live demo runs successfully	Presents with confidence using technical vocabulary; responds well to questions; makes strong connections to real autonomous systems

TECHNOLOGY & AI INTEGRATION

Tool / Platform	How to Use in This Lesson
MakeCode (makecode.microbit.org)	Primary coding environment with PlanetX extension; block view for beginners, JavaScript view for advanced; use the simulator to visualise Sonar:bit readings before flashing
micro:bit v2 + Nezha Board + Sonar:bit + Motors	Full physical computing stack; dual-motor setup enables independent left/right wheel control for turning; Sonar:bit provides real sensor data for conditional navigation logic
Simulated Lunar Surface	Physical obstacle course (tray + blocks + paper craters) provides authentic engineering test conditions; students

	must optimise code to real-world performance, not just simulator results
AI Chatbot (Claude / ChatGPT)	Students can ask: "Why does my rover keep turning in circles?" or "How does the real Yutu-2 rover avoid obstacles?" Teach responsible AI use: verify answers, test code yourself, don't copy explanations verbatim
NASA & ESA Rover Resources	NASA's Mars Rover page and ESA lunar mission pages provide real engineering data students can compare to their own design decisions
QR Code Resource Stations	Link to MakeCode starter project, Sonar:bit wiring diagram, vocabulary glossary, and rover design reference sheet for independent and differentiated pacing
Google Slides / Padlet	Digital gallery for rover build photos and testing log results; Padlet allows structured peer comments on navigation performance across teams

AI Responsible Use Guidance for Students

- Use AI to help you understand vocabulary, troubleshoot code logic, or research real rover missions — 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., Yutu-2, Apollo LRV, Perseverance), 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 Lunar Roving Vehicle (1971–1972)	"The Apollo LRV had no automatic obstacle avoidance — astronauts drove it manually. What problems could that cause on future long-distance Moon missions where astronauts can't be outside for hours?"
China's Yutu-2 Rover (2019–present)	"Yutu-2 is the first rover on the Moon's far side — where Earth is never visible. How does it communicate with Earth without a direct line of sight? How does this affect its autonomous navigation?"
NASA Artemis — Lunar Terrain Vehicle	"NASA is developing an autonomous Lunar Terrain Vehicle for Artemis. What sensor systems would you expect it to have — and how are they similar to your Sonar:bit setup?"
Mars Perseverance Rover	"Perseverance uses cameras and AI to navigate autonomously because signals take up to 20 minutes to travel from Earth. How is your rover's obstacle

avoidance similar to — and different from —
Perseverance's system?"

Self-Driving Cars "The same sensor and conditional logic principles you used today are in every self-driving car. What are the shared engineering challenges between a lunar rover and an autonomous vehicle on Earth?"

Careers in Robotics & Space "What careers involve designing and programming autonomous vehicles? Robotics engineer, AI developer, systems engineer, space mission scientist, software developer — which interests you most?"

EXTENSION ACTIVITIES

Activity	Description	Suggested For
Rover Missions Research	Research 3 lunar or planetary rover missions (e.g., Apollo LRV, Yutu-2, Perseverance). Compare their sensor systems, navigation methods, and mission objectives in a 1-page illustrated summary.	Average & Advanced
Lunar Base Design & Navigation	Design and build a "lunar base" structure from blocks at the edge of the navigation surface. Program the rover to navigate to the base, dock (stop within 5 cm), wait 3 seconds, and return to start.	Advanced / Gifted
Obstacle Course Championship	Design an increasingly complex obstacle course and time each team's rover navigating it. Teams improve their code between runs. Discuss: what code changes made the biggest difference to performance?	All levels
Future Lunar Exploration Presentation	Create a 3-minute multimedia presentation on the NASA Artemis Lunar Terrain Vehicle or ESA's planned lunar surface exploration systems, comparing them to the team's own rover design.	All levels
Sample Collection Arm	Design and attach a physical sample collection arm from blocks. Program a stop-pause sequence at a marked "sample site." Photograph the rover in collection mode as part of the team's portfolio.	Advanced / Gifted
Autonomous Mapping Challenge	Program the rover to navigate a defined square boundary using timed motor runs (no sensor) to map out a grid. Compare this "dead reckoning" approach to sensor-based navigation. Discuss: which is more reliable and why?	Gifted

HOMEWORK / FOLLOW-UP TASKS

Task	Instructions	Due
Rover Reflection	Write 150–200 words: What was the hardest problem your team solved during the 3 days? What	Next class

Journal	engineering decision are you most proud of? If you could add one more sensor to your rover, what would it be and why?	
Real Rover Connection	Find a news article or video about a real lunar or planetary rover published in the last 2 years. Write 3 sentences: (1) what the rover does, (2) one similarity to your rover, (3) one key difference.	Next class
Extension (Optional)	Sketch an upgraded rover design that includes at least 3 sensors (e.g., ultrasonic, light, temperature). Label each sensor, describe what it measures, and explain how it would change your navigation code.	Next class

TEACHER NOTES & TIPS

Before the Lesson

- Review Lesson 02 student work and note any teams who struggled with Sonar:bit integration — provide extra support at the start of Day 2.
- Test all Sonar:bit sensors, both motors, and Nezha board connections at least one day in advance.
- Pre-install the PlanetX MakeCode extension on classroom computers if internet speed is limited.
- Prepare the simulated lunar surface before Day 3: a large tray, section of floor, or desktop area with at least 4–6 obstacles of varying size.
- Print Rover Design Brief sheets, vocabulary cards, and Testing Logs for each team.
- Bookmark rover video clips (Apollo LRV, Yutu-2) and NASA/ESA rover pages on a dedicated browser tab.

During the Lesson

- This lesson builds on Lesson 02 — if students are unfamiliar with the Sonar:bit, add 5–10 extra minutes to the Day 2 setup phase.
- Two motors (left and right) enable turning — if only one motor is available, turning can be simulated with a brief pause and manual redirect, but discuss this limitation with students as an engineering constraint.
- The simulated lunar surface works best with obstacles that are tall enough for the Sonar:bit to detect reliably (at least 5–10 cm tall).
- Encourage teams to document every code change in the Recorder's Testing Log — this develops real engineering discipline.
- For SEN students, the Recorder role allows full participation without physical pressure; they can annotate printed code screenshots.

- If a team's rover drives in circles, the most common cause is unequal motor speeds — use this as a real-world calibration teaching moment.

After the Lesson

- Photograph each team's final rover 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 autonomous navigation concepts that need re-teaching.
- Consider hosting a "Rover Exhibition" or inter-class navigation challenge to celebrate and extend student work.
- Save the simulated lunar surface setup for use as an obstacle course extension or assessment tool in future lessons.

CURRICULUM STANDARDS ALIGNMENT

Standard Framework	Alignment
NGSS (Next Generation Science Standards)	MS-PS2-1: Newton's Laws; MS-ETS1-1 to 1-4: Engineering Design; MS-PS3: Energy in motion; HS-ETS1-2: Design solutions for complex problems
CSTA (Computer Science)	Level 2 (Grades 6–8): Algorithms, conditionals, sensor input, iterative debugging, and event-driven programming using block-based and JavaScript coding
ISTE Standards (Students)	Empowered Learner, Innovative Designer, Computational Thinker, Creative Communicator, Global Collaborator
Common Core (Supporting)	Measurement and proportional reasoning in threshold calibration; data recording and analysis across 3 testing runs; structured oral presentation
21st Century Skills (4Cs)	Critical Thinking (sensor calibration and debugging), Creativity (rover design), Collaboration (team roles across 3 days), Communication (navigation presentations)
Lesson Sequence	Extends Lesson 02 (Moon Landing Rocket); prepares students for Lesson 04 in the Space Science Kit series

Developed by Techtelligence Technologies LLC in partnership with Elecfreaks • Space Science Kit • Lesson 03