CASE 1

Pull-Ups Robot

Nezha Inventor's Kit V2 — STEAM Lesson Plan

Grade 6–8 | 45–90 Minutes | STEAM + Robotics | MakeCode | 5E Model

LESSON OVERVIEW

Category	Details
Subject	STEAM, Robotics, Computer Science, Math
Grade Level	6–8
Time Required	1–2 periods (45–90 minutes)
Kit	Elecfreaks Nezha Inventor's Kit V2
Software	MakeCode (visual block programming)
Approach	5E Learning Model: Engage → Explore → Explain → Elaborate → Evaluate

LEARNING OBJECTIVES

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

01 Design & Build	Design and build a robot that performs pull-up motions using parallelogram linkages from the Nezha Inventor's Kit V2.
02 Explain Geometry	Explain the geometric properties of parallelograms and how they enable controlled, stable mechanical movement.
03 Understand Motors	Understand how motors convert circular rotation into reciprocal (back-and-forth) motion to simulate pull-up action.

04 Code with MakeCode	Program robot movements using MakeCode visual block programming — including motor control, timing, and loop sequences.

MATERIALS & EQUIPMENT

REQUIRED	OPTIONAL / HELPFUL
• Elecfreaks Nezha Inventor's Kit V2	• Cardboard for custom frame builds
• Computer or tablet with internet access	• Craft sticks & additional fasteners
• MakeCode software (web-based, free)	• Student engineering journal
• Timer	• Observation/assessment checklist
• Pull-up bar or similar test structure	

PHASE 1 — ENGAGE — 10 Minutes

1	Show students videos or images of athletes performing pull-ups — spark curiosity about the motion and mechanics involved.
2	Discuss pull-ups as exercise: which muscles are used, how the body moves, and why it requires strength and coordination.
3	Introduce the challenge: "Build a robot that can do pull-ups using the Nezha Inventor's Kit V2!"
4	Class brainstorm: "What mechanisms could allow a robot to pull itself up and down repeatedly?"

Teacher Tip: Set the stage with excitement — challenge students to think like engineers before touching any kit components. Observe their prior knowledge during the brainstorm.

PHASE 2 — EXPLORE — 20 Minutes

1	Divide students into small groups (2–3 per group recommended for balanced participation).
2	Distribute Nezha Inventor's Kit V2 and open MakeCode on devices — allow free exploration first.
3	Guide groups to explore kit components, focusing on parallelogram blocks and motor — "What do you notice?"

4	Students document explorations and discoveries in engineering journals — sketches and observations encouraged.

Key Focus: Hands-on discovery — let students experiment before direct instruction. Resist over-guiding; observe and note misconceptions to address in the Explain phase.

PHASE 3 — EXPLAIN

15 Minutes

PARALLELOGRAM PROPERTIES	RECIPROCAL MOTION
• Opposite sides are equal and parallel	• Repeating back-and-forth motion
• Opposite angles are always equal	• Converts circular motor rotation
• Maintains shape during movement	• Simulates human pull-up movement
• Enables smooth, constrained arm motion	• Crank-slider mechanism principle
• Used in robotic arms & pantographs	• Speed controlled via MakeCode

Teacher-Led Steps:

1	Facilitate class discussion: properties of parallelograms and their role in controlled mechanical movement.
2	Explain reciprocal motion and how it directly connects to the pull-up action of the robot arms.
3	Demonstrate live: how the motor's circular rotation converts to controlled reciprocal arm motion.
4	Introduce MakeCode environment — show basic motor control blocks: run, pause, reverse, loop.

```
motor A run forward @ speed 50 → pause 1000ms → motor A run backward @ speed 50 → pause 1000ms → repeat (forever loop)
```

PHASE 4 — ELABORATE

30–40 Minutes

BUILD	Groups design and assemble pull-up robots using parallelogram blocks, motors, and structural pieces from the Nezha Inventor's Kit V2. Encourage experimentation with different configurations.

CODE	Students program robots in MakeCode to perform the pull-up motion — adjusting motor speed, direction, and timing for best performance. Key blocks: forever, motor run, pause, reverse.
REFINE	Groups iterate their designs: What works? What breaks? Why? Encourage multiple test-fix cycles. What mechanical changes improve performance? What code changes help?

MakeCode Step-by-Step Guide:

- Step 1: Open MakeCode at the Elecfreaks platform or makecode.microbit.org
- Step 2: Create 'on start' block for initialization
- Step 3: Add motor run block — set Motor A forward at speed 50
- Step 4: Add pause block — 1000ms to hold up position
- Step 5: Add motor run block — Motor A backward at speed 50
- Step 6: Add pause block — 1000ms to hold down position
- Step 7: Wrap steps 3–6 in a 'forever' block for continuous pull-ups

Tip: Test at slow speed (30–40) first — then gradually increase for more pull-ups per minute!

PHASE 5 — EVALUATE — 15 Minutes

Pull-Up Challenge Format:

1	Set up pull-up challenge station — each group demonstrates their robot on a bar or structure.
2	Use a timer to count how many pull-ups each robot completes in 30 seconds.
3	Record results on a class leaderboard (whiteboard or shared document).
4	Groups share results and explain their design and coding choices to the class.

Reflection Questions:

- "What worked well in your design?"
- "What would you change if you rebuilt it from scratch?"
- "How does your robot's motion compare to a real human pull-up?"
- "What engineering decisions had the biggest impact on performance?"

Assessment Focus: Observe teamwork, problem-solving approach, and code debugging skills during both the build and challenge phases.

DIFFERENTIATION STRATEGIES

SUPPORT — Struggling Students	CHALLENGE — Advanced Students
• Provide pre-built robot arm structures	• Incorporate distance or touch sensors
• Use simplified MakeCode interface	• Create multi-phase movement patterns
• Pair with a stronger engineering partner	• Program a pull-up counter with variables
• Provide a step-by-step instruction card	• Optimize for max pull-ups in 30 seconds
• Allow extra time on the build phase	• Explore JavaScript mode in MakeCode

EXTENSION ACTIVITIES

1. Research	Explore real-world robots that perform physical tasks (warehouse robots, surgical arms, exoskeletons). Compare mechanisms to the pull-up robot. Present findings to the class.
2. Redesign	Modify the robot to perform a different exercise or motion (push-ups, squats, rowing). Document the mechanical changes needed and present the redesigned robot.
3. Advanced Code	Use Python or JavaScript in MakeCode to program a pull-up counter, sensor-triggered starts, or a timed interval training routine with LED display output.

TEACHER NOTES & SAFETY

PREPARATION	Familiarize yourself with the Nezha Inventor's Kit V2 and MakeCode before the lesson. Build a sample pull-up robot to demonstrate at the start. Pre-load MakeCode on classroom devices.
SAFETY	Handle motors and electronic components carefully. Avoid water near electronics. Secure all loose wires before testing. Supervise motor testing at all times to prevent injury.
COLLABORATION	Ensure equitable participation in groups. Rotate roles each phase: Builder, Coder, Tester, and Recorder — all students should have hands-on time with the kit and programming.

ASSESSMENT	Use an observation checklist during the build phase. Collect engineering journals for the reflection section. Grade on design process and effort, not just final robot performance.

STANDARDS ALIGNMENT

Standard	Codes	Application in This Lesson
NGSS	MS-ETS1-1, MS-ETS1-2, MS-ETS1-3, MS-ETS1-4	Engineering Design — define problems, evaluate solutions, analyze test data to improve design
CSTA	1B-AP-10, 2-AP-11, 2-AP-12	Algorithms & Programming — loops, events, sequences, variables in MakeCode
CCSS Math	7.G.A.1, 8.G.B	Geometry & Spatial Reasoning — properties of parallelograms, shapes in motion
Phys. Ed.	SHAPE America Movement Concepts	Force, leverage, and biomechanics in the context of pull-up motion analysis