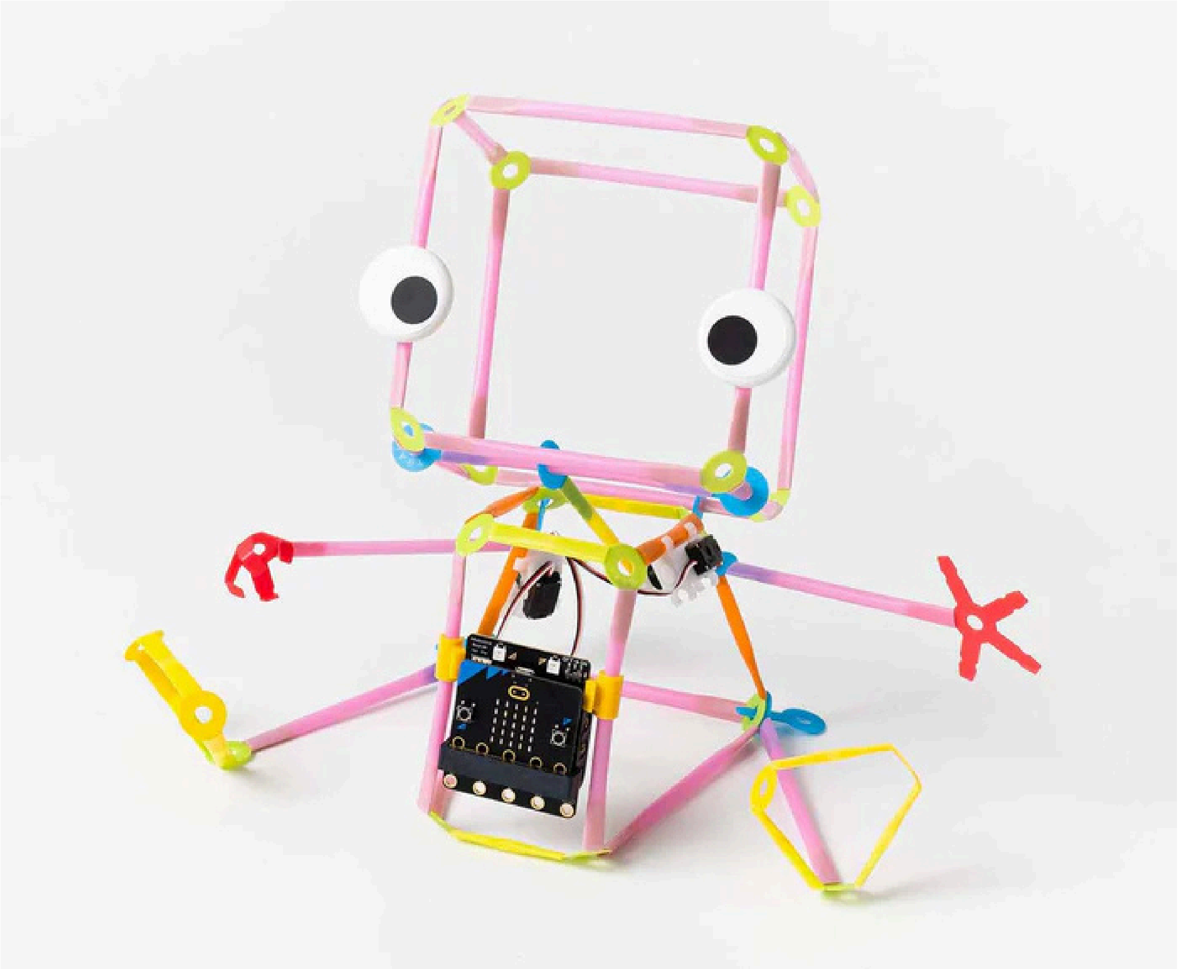
Strawbees Robotics Extensions



Mid-Valley
STEM-CTE HUB

www.midvalleystem.org
midvalleystemctehub@linnbenton.edu
Linn-Benton Community College
Albany Campus - CC-212

Strawbees Robotics Extensions

The Strawbees Robotics Extension Kit with micro:bit combines structural engineering and creative prototyping with simple, block-based programming. Students build moving mechanisms using Strawbees' lightweight connectors and straws, then bring their creations to life using servo motors and the micro:bit controller. With support for MakeCode and physical computing concepts, this kit empowers learners to explore coding, robotics, design, and problem-solving in a hands-on, scalable way—ideal for both guided lessons and open-ended invention challenges.

Grade Level	**5th - 10th**
Group Size	**2 - 3 students per set**
Time Duration	**45 minutes to multi - sessions depending on the complexity of the build**

Content of Kits

Components

- 3 Robot Sensors
- 15 Strawbee Robotics sets
    - 1 Micro:bit
    - 1 Servo motor
    - 1 Micro:bit clip
    - 1 Servo extension cable
    - 1 mini screw driver
    - 3 Servo motor arms
    - 3 screws
    - 2 Servo motor mounts

Consumables

- 3 AAA batteries per set

Pair with Strawbees

- This is an extension kit for Strawbees activities. Must be checked out with a Strawbees kit.

Usage

Getting Started

1. **Introduce the Micro:bit and Servo Motor -** Demonstrate how the Micro:bit connects to the servo motor using the Strawbees Robotics board or servo clips, and explain that code will control motor movement.
2. **Download and Launch MakeCode -** Have students visit makecode.microbit.org and start a new project. Show them how to add the “Strawbees” extension or use servo blocks for control.
3. **Test Basic Movement -** Before building a structure, program a simple motion (like a 90-degree turn) to confirm the motor and Micro:bit are working properly.
4. **Build Simple Mechanisms First -** Guide students through making a waving arm, moving mouth, or rotating sign to understand how motion translates to structural movement.
5. **Assign Roles in Teams -** Designate "builder," "coder," and "tester" roles within groups to support collaboration and maximize participation.

Storage

- **Separate Electronics and Structural Parts -** Store Micro:bits, USB cables, servo motors, and battery packs in labeled pouches or compartments away from building straws and connectors.
- **Bundle Straws by Length**
- **Label Each Kit -** Assign a number or color code to each full kit and its Micro:bit so components don’t get mixed between groups.

Troubleshooting

- **Servo Not Moving -** Check that the micro:bit is powered (via USB or battery pack) and that your code is properly sending signals to the servo pin (usually P0, P1, or P2).
- **Servo Making Noise but Not Rotating -** Ensure nothing is physically blocking its motion, and confirm the programmed angles are within safe range (0–180°).
- **Micro:bit Not Responding -** Try a different USB cable, check for firmware updates, and confirm the device is recognized by your computer when plugged in.
- **Code Upload Fails -** Ensure you're dragging the correct .hex file into the Micro:bit drive or use the "Download" button directly from MakeCode if in paired mode.
- **Structure Falls Apart During Motion -** Reinforce joints with additional connectors or straws and reduce the servo angle range or speed to decrease stress on the model.

Activity Guide

Beginner

Waving Hand
Students build a basic frame with a rotating "hand" attached to a servo motor. Using MakeCode, they write a simple program to make the hand wave back and forth at different angles. This activity introduces servo control, loop blocks, and motion timing.

Intermediate

Animal with Moving Part
Students design and build a simple animal (e.g., a bird with flapping wings or a fish with a swishing tail) using straws and connectors. They program the servo to move in timed intervals or respond to a button press on the micro:bit. This reinforces sequencing and input/output relationships.

Advanced

Obstacle-Detecting Creature
Using a micro:bit with an ultrasonic sensor (if available) or button input as a trigger, students build a robotic creature that reacts to its environment (e.g., a robot that opens its mouth when it “sees” something). This project introduces conditionals, variable control, and basic feedback loops.

Extension Activities:

Invention Design Challenge
In teams, students brainstorm a problem (e.g., a device to alert when a door opens, a robotic greeter, or an expressive puppet) and build a working prototype. They integrate movement, logic, and storytelling, then present their invention and how it works to the class.

Mid-Valley
STEM-CTE HUB

Learning Extensions

STEAM Connections: Engineering - Structure Design - Tech

Learning Objectives:

- Understand the relationship between mechanical structures and programmed motion.
- Use block-based coding to control servo motors through the micro:bit.
- Develop and test physical prototypes that respond to logic-based commands.
- Apply the engineering design process to brainstorm, build, and refine robotic solutions.
- Strengthen collaboration and communication through team-based invention challenges.

Career Connections:

- **Robotics Engineer –** Designs programmable machines and integrates electronics with physical systems.
- **Product Designer –** Creates prototypes that merge mechanical structure with interactive behavior.
- **Mechanical Engineer –** Builds moving systems that depend on levers, motors, and control logic.
- **Software Developer –** Programs inputs and outputs to control hardware in real-world devices.
- **STEM Educator or Fab Lab Instructor –** Teaches coding, construction, and prototyping in makerspaces and classrooms.

Essential Employability Skills:

- Problem-Solving
- Critical Thinking
- Teamwork
- Creativity
- Innovation
- Technical Literacy

Mid-Valley
STEM-CTE HUB

Resources and Accessibility

Safety Guidelines

- **Supervise Motor Use –** Ensure students don’t obstruct or force servo motors while in motion to prevent overheating or breakage.
- **Secure All Electronics –** Keep micro:bits, servo wires, and battery packs firmly connected and off the floor to avoid tripping or damage.
- **Avoid Pinched Fingers –** Remind students to keep fingers away from rotating arms and tight straw connections during testing.
- **Use Proper Power Sources –** Only use approved battery packs or USB cables with the micro:bit to avoid electrical faults.
- **Build on Stable Surfaces –** Use flat, clean tables or trays to prevent projects from collapsing during testing.

Accessibility

- **Offer Pre-Cut Straws or Pre-Built Bases –** Reduce fine motor strain by providing ready-to-use structural parts.
- **Incorporate Screen Reader–Friendly Coding –** For students with visual impairments, use MakeCode’s accessibility features or pair them with a coding buddy.
- **Assign Flexible Roles –** Let students contribute as coders, planners, or presenters if building is physically difficult.
- **Provide Printed and Visual Instructions –** Use large-print diagrams or step-by-step photo guides to support diverse learning styles.

Library Catalog

Library Resources

Feedback

QR to feedback survey