ezrobot

EDUCATE • ENGAGE • EMPOWER

Educators Guide

JD Humanoid

Welcome to **EZ-Robot Inc**.

EZ-Robot is an edtech solution that supports teaching and inspires learning. Empowering lifelong learning through real world applications and problem-solving, EZ-Robot provides new opportunities for reinforcing the power of science, technology, engineering, arts, and mathematics (STEAM).

The comprehensive and modular EZ-Robot technology platform grows with students as they learn. EZ-Robot software and hardware innovations provide approachable methods for learning complex robotics skills. From the basics of coding to machine learning, students graduate from beginner to advanced users as they gain the skills and knowledge needed to thrive in future STEAM related careers.

This guide provides important information for getting started with your Revolution robot.

| Mobile Play +
Programming | 3D Design
+ Printing | Modular
Hardware | Graphical
Programming | Easy-to-learn
Syntax | Add-Ons +
Sensors |
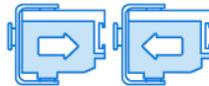
The EZ-Robot pedagogical approach builds competency in 21st century learning skills. Through the exploration of technology and robotics, students will build their confidence and capabilities in creativity and innovation, computational thinking, and critical analysis.

Creativity & Innovation	✓ Design thinking ✓ Algorithmic design ✓ Engineering invention
Computational Thinking	✓ Syntax & conditional logic ✓ Sequential planning ✓ Pattern recognition
Critical Analysis	✓ Problem solving ✓ Scientific inquiry ✓ Collaboration

EZ-Robot in the Classroom

How do I get started?

1. Download the software from www.ez-robot.com/ez-builder
2. Review the Getting Started Guide for battery and safety information.
3. Visit http://www.ez-robot.com/Tutorials for resources, lessons, and tips.

How many students can work with a robot at one time?

Based on educator experience, it is recommended to limit two students per robot.

How can I keep a class set of robots organized when connecting all at once?

Each robot has a unique Wi-Fi identification name. This name can be customized as desired. Designate a number or letter for each robot in a classroom set and mark the robots accordingly.

http://www.ez-robot.com/Tutorials/Lesson/116

Where can I find learning resources and content?

Visit http://www.ez-robot.com/Tutorials/ to learn more about each robot.

EZ-Robot's **The Robot Program** is a freely available educational series that teaches technology using robotics. Various aspects of technological literacy are addressed in follow-along lessons, from building robots to coding programs, from using artificial intelligence to implementing computer vision, and more.

Each lesson is available as both a **written tutorial** and as an educational **video**. Episodes include a step-by-step lesson with identified learning concepts, as well as a quiz for testing knowledge.

More information and resources can be found at www.TheRobotProgram.com or embedded throughout the **Learn** section of the website at http://www.ez-robot.com/Tutorials.

What should I do if something isn't working correctly?

Contact your local reseller for resources and recommendations. If you still need further assistance, please contact us at support@ez-robot.com.

Suggested Lesson Plan for JD Humanoid Basics

Introduction to Teaching and Learning with Robotics

Recommended Lessons

The Robot Program Episode 038: Using The Robot Program

The Robot Program Episode 040: Practical Applications of Robotics

The Robot Program Episode 071: History of Programming

The Robot Program Episode 056: Safe Robot-ing

The Robot Program Episode 005: Battery Charging and Care

Introduction to JD Humanoid

Recommended Lessons (also included in this guide)

The Robot Program Episode 001: JD Humanoid Box Contents

The Robot Program Episode 002: Building JD

The Robot Program Episode 003: Getting JD to Move
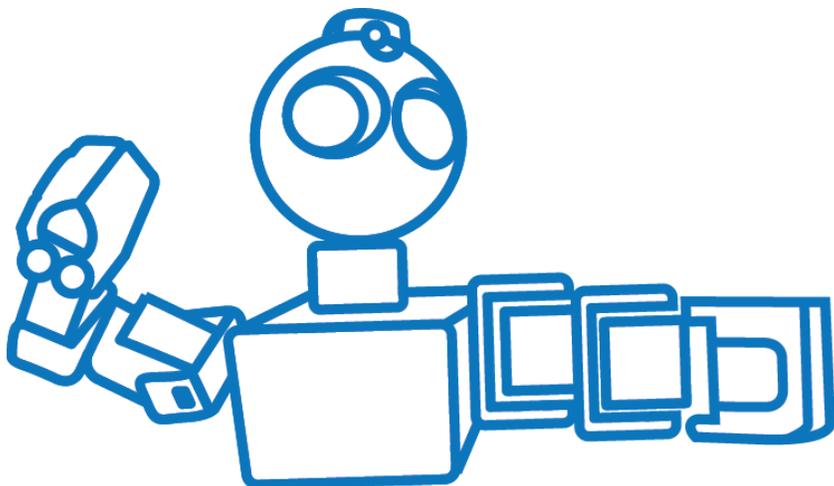
The Robot Program Challenge

Encourage, engage, and deepen student learning through creative design challenges. The Robot Program Challenge scaffolds and connects the lessons found in The Robot Program with practical engineering design problems. Each design challenge incorporates technical learning outcomes while developing creativity and innovation, computational thinking, and critical analysis. Five initial challenges are included in this guide.

More challenges can be found at www.ez-robot.com/challenge. Challenge episodes feature teaching and learning experiences from classrooms that have created, adapted, or programmed robots in innovative and fascinating ways. Educators are encouraged to share student achievements related to an active challenge by entering a submission on the website.

Challenges are designed to grow skills and knowledge while promoting interdisciplinary connections between science, technology, engineering, arts, and mathematics (STEAM). As students move through the design and implementation process, they will incorporate core learning outcomes that provide confidence and competency for all stages of learning.

Lifelong STEAM Learning Outcomes

Science	Technology	Engineering	Arts	Mathematics
Formulate hypotheses	Abstract complex systems	Examine problem constraints	Create original works	Predict outcomes
Explore theories & ideas	Develop a sequential process	Generate multiple ideas	Interpret & express theories	Determine & describe patterns
Record observations & data	Design comprehensive tests	Analyze & compare ideas	Connect interdisciplinary ideas	Measure angles & movement
Identify & analyze trends	Model & visualize information	Develop & manage a plan	Engage responsible citizenship	Analyze position & motion
Collaborate on solutions	Verify outcomes	Design prototypes & testing	Communicate data & results	Apply spatial transformations
Discover through inquiry	Apply knowledge & skills	Reflect & refine solutions	Innovate authentic applications	Represent expressions & data

Challenges for JD Humanoid	Learning Goals	Recommended Lessons
Challenge – Recipes for Robots	*Create a program in RoboScratch using actions, movements and sounds.* Technical Skills: • Code linear, sequential programs using RoboScratch • Differentiate between immediate and delayed actions • Control robot movement • Activate and use peripheral information	Episode 071 Episode 020
Challenge – Obstacle Mission	*Navigate a course within a specified time limit using speech recognition and then by creating a Blockly program.* Technical Skills: • Calculate, plan and analyze robot movements • Code sequential programs using Blockly • Adjust speech recognition confidence intervals • Trigger and navigate movement	Episode 045 Episode 059
Challenge – Secret Message Delivery	*Generate and decode secret messages using QR codes and the camera.* Technical Skills: • Encode QR information • Analyze variable data • Append text to a string • Output audio	Episode 052
Challenge – I Spy (With My RGB Eyes…)	*Create custom RGB animations that are executed when recognizing particular objects.* Technical Skills: • Design custom RGB animations • Train and recognize objects • Implement if-else conditional logic • Execute custom scripts	Episode 039 Episode 067 Episode 046
Challenge – So You Think Your Robot Can Dance	*Design and share a custom choreography timed to a favorite piece of music.* Technical Skills: • Synchronize servo movements to audio • Calculate servo angles and positioning • Create a mobile app interface • Share projects across robots	Episode 034 Episode 037 Episode 053

Implementing a STEAM Classroom Challenge

Understanding the Problem

Provide the goal of the challenge to the students. Have them identify the requirements as they breakdown the overall outcome into smaller objectives. At this stage of the process, students should be focused on defining and examining the problem, rather than focusing on how they will solve the problem.

An engineering log book is a great way to record student thoughts, ideas, and collaborations. Encourage both individuals and teams to express their design process through written, visual, and oral communication. This may include presentations, reports, tables, charts, discussions, predictions, or models.

Generating and Comparing Solutions

Encourage students to ideate individually before coming together as a team or large group. Each student should have the opportunity to share their potential solutions without judgement or criticism. Empower student collaboration by connecting a strong element or thought from each student to grow a group solution. Students can assess and critically analyze multiple potential solutions by weighing criteria such as time constraints, material restrictions, feasibility, and adherence to the requirements. Once a solution has been determined, students can then begin to develop a design plan.

Developing a Plan

Whether students are working as a team or individually, they will benefit from creating a design plan. This allows them to identify and schedule specific software or hardware tasks that need to be completed. Students should also predict anticipated areas of difficulty and identify any gaps in their knowledge that will require extra research or learning.

As they begin development, students should rotate roles to practice the various skills involved, from coding to connecting the hardware to recording design progress.

Prototyping a Solution

As students begin to develop their solution, they may benefit from first mapping out their proposed program on paper or by talking through the logic with another person or team. This will help them to develop their computational thinking skills as they gradually increase the complexity of their

implementation. Rather than implement multiple controls or commands at once, students should build their project modularly, iteratively developing and testing each component. Remind students to focus achieving the challenge outcomes as they innovate.

Testing and Refining the Result

Testing provides students with the opportunity to break and improve their designs. Verifying the outcomes of their work allows students to take responsibility for their own learning while considering practical scenarios that could impact their original plan. As students debug their designs, they will learn to recognize patterns in their code and the corresponding implications for robot movement or response.

Sharing the Final Solution

Challenges encourage creativity and innovation, leading to multiple solutions and varied results for every design problem. Students should have the opportunity to share their work while learning about the solutions of others. Whether this is done through a formal showcase or an informal discussion, students should be encouraged to consider how their final solution could be applied ethically and responsibly to a practical, real-world scenario.

Reflecting on the Challenge

Before ending the challenge, encourage students to reflect on the design process and the implications of their work. As they develop their technical skills, students will also benefit from growing their ability to self-assess and analyze. Identifying both positive and negative aspects of the challenge experience will help students to connect ideas in the future while building their confidence and competencies.

Challenge Descriptions

Challenges for JD Humanoid

Challenge – Recipes for Robots

A program allows humans to give specific instructions to robots. Create a program in RoboScratch that uses the following elements:

- At least ten actions or movements
- A mix of "wait" and regular actions or sounds
- The robot must not remain in the same starting location
- The robot must take two different photos

Students will learn to…

- Program linear sequences
- Implement immediate commands
- Implement delayed commands
- Move the robot
- Reposition the robot
- Output audio
- Activate the camera

Recommended Lessons

The Robot Program Episode 071: History of Programming

The Robot Program Episode 020: Detect Face and Wave – RoboScratch

Challenge – Obstacle Mission

Robots can enter situations that might be too dangerous or difficult for humans. They can navigate over rough terrain and around obstacles.

Travel through an obstacle course or through a maze within a specified time limit first using speech recognition to control movement, and then by creating a Blockly program.

Tip: Remember to calculate how many seconds it takes the robot to complete a 90-degree turn to the right or left.

Tip: Make a note of how far the robot moves forward in a single second.

Students will learn to…

- Adjust speech recognition confidence intervals
- Trigger movement through audio input
- Calculate and analyze robot movements
- Build a series of movement commands
- Navigate turns and obstacles

Recommended Lessons

The Robot Program Episode 045: Control Robot with Speech Recognition

The Robot Program Episode 059: Navigating Using Blockly

Challenge – Secret Message Delivery

Robots can process complex information such as QR codes and react to the provided information.

Generate a secret message in a QR code and swap with someone else. Program your robot to decode the new message and then read it out loud, starting with the phrase "The secret code is…"

Students will learn to…

- Store information in a QR code
- Detect a QR code
- Read information from a variable
- Append text to a string
- Output audio

Recommended Lessons

The Robot Program Episode 052: Using QR Codes

Challenge – I Spy (With My RGB Eyes…)

Robots can be used to interact with humans. Some components, such as LED "eyes", make them seem more approachable. "Eyes" can also be used to convey information to a human.

The objective of this challenge is to play a custom RGB animation when the robot recognizes a particular object. At least two different objects and two different RGB animations should be used.

Tip: Use logic to program a single tracking script for both objects.

Students will learn to…

- Create a custom RGB animation
- Train a new object
- Track an object
- Write a tracking-enabled script
- Implement if-else conditional logic

Recommended Lessons

The Robot Program Episode 039: Programming (Variables, If/Else, Logic)

The Robot Program Episode 067: Changing RGB Animator for JD

The Robot Program Episode 046: Visual Object Recognition

Challenge – So You Think Your Robot Can Dance

Robotics are used in fine arts as a part of performance, animation, and theatrics.

Design a custom robot dance routine to a favorite piece of music. The choreography must be at least one minute in length and use a minimum of five different positions. For an advanced challenge, create one custom action that can be incorporated into the choreography.

One your dance is complete, create and share a mobile interface that allows anyone to showcase the dance routine.

Tip: Search for free music using www.ninety9lives.com.

Tip: Start with an existing mobile interface to see how buttons can be used to execute routines.

Students will learn to…

- Synchronize servo movements to music
- Save a new routine in the Soundboard control
- Transition between servo positions
- Advanced: build actions through still frames
- Advanced: calculate servo angles and positioning
- Modify or build a mobile app interface
- Save and export a mobile app to the EZ-Cloud
- Access apps from the EZ-Cloud community

Recommended Lessons

The Robot Program Episode 034: Teaching JD Humanoid to Dance

The Robot Program Episode 037: Teaching JD Humanoid to Dab!

The Robot Program Episode 053: Creating a Mobile App

Getting Started

with

JD Humanoid

Getting Started with JD Humanoid

What does this section cover?

This section of the guide covers the content of **The Robot Program Episodes 001 – 003**. You will learn to build and connect the **JD Humanoid robot**.

What type of robot is JD?

JD is a humanoid (human-like features) robot that uses two legs for bipedal movement.

The **EZ-Bit** components of **JD** include:

- A **Humanoid Head** and **Humanoid Body**
- The **EZ-B Robot Controller**
- **Six Lever Servos** that act as motorized robot joints
- Two **Humanoid** Feet for bipedal movement
- Two **Grippers** that allow the robot to hold items

Wire Wraps will be used to organize the cables of the robot. The kit also contains the battery charging components and the **Getting Started Guide**.

What is the Getting Started Guide?

The **Getting Started Guide** gives an overview of important information. This guide also includes links to online resources, and contains a detailed diagram of the **EZ-B v4 Robot Controller**, which will be helpful for building the robot.

What do I need to know about charging JD?

Always charge the battery before using **JD**.

Use the appropriate country adapter and connect the transformer to the charger. The rechargeable robot battery is contained in the body, with only the charging cable visible. The **JD Humanoid** battery connection is under the left arm. Use your thumb and finger to securely connect the cable to the charger. To disconnect, gently rock the cable back and forth. Do not pull on the wires directly.

Each robot has a **Battery Saving Monitor** that will alert when the battery is low. Power off the robot and allow the battery to charge for a full cycle. The length of a full charge cycle varies, but may take a few hours. The charger has lights to indicate the battery's charging status.

Failing to charge may cause the battery to "puff up" and could potentially be dangerous. Dead batteries should be discarded according to local recycling and disposal rules.
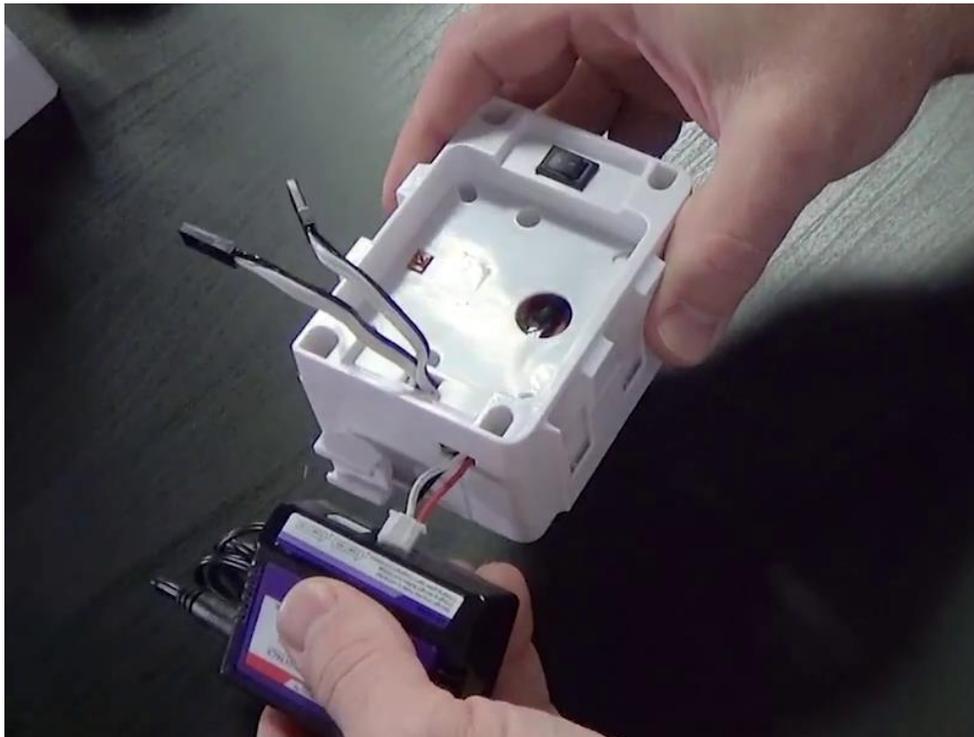
What are EZ-Bits and how do I connect them?

The robot components are called **EZ-Bits**. Each **EZ-Bit** connects to the **EZ-B Robot Controller** using male-to-female connections at the back of the robot. The port layout can be viewed in the **Getting Started Guide**. Be sure to match the wire colors to the corresponding port colors. Adjust the cables so that they are coming out of the back of each **EZ-Bit**, which will make the connections easier to organize.

EZ-Robot Clip'n'Play Technology allows **EZ-Bits** to be connected using a simple pressure fit clip. Insert the male clip into the female side of the **EZ-Bit** and press together. To remove, reverse the process. If the fit is tight, gently wiggle or rock the **EZ-Bits** into place.

Treat your **EZ-Bits** with respect. Electronic components such as servos are vulnerable to heat, moisture, etc. Throughout building and use, ensure that no servos are making grinding noises or pushing against one another. Refer to the lessons for correct usage information.

Where do I find the building instructions?

The **EZ-Builder** software can be downloaded from www.ez-robot.com.

Verify that the software is from **EZ-Robot Inc.** Double-click to begin the installation. Choose **Typical** as the install type.

Read and agree to the **Terms of Use**. The introductory information can also be found in the **Getting Started Guide**.

Login or create an **EZ-Cloud** account. More tutorials about **JD Humanoid** can be found at the **EZ-Robot School**. These tutorials also include related videos and lessons from **The Robot Program**.

EZ-Bits are robot parts. Sync to update the library for the most recent designs and information and select **Yes** to view the build instructions. Follow along with the step-by-step building instructions.

JD Humanoid Building Steps:

First, insert the **EZ-B Robot Controller** into the body. The cables use a male-to-female connection, with pins that fit securely into corresponding holes.

Connect the **Right Shoulder Servo** cable to <u>D2</u> and **Left Shoulder Servo** cable to <u>D3</u>. Match the black wire on the cable to the black side of the **EZ-B** port.
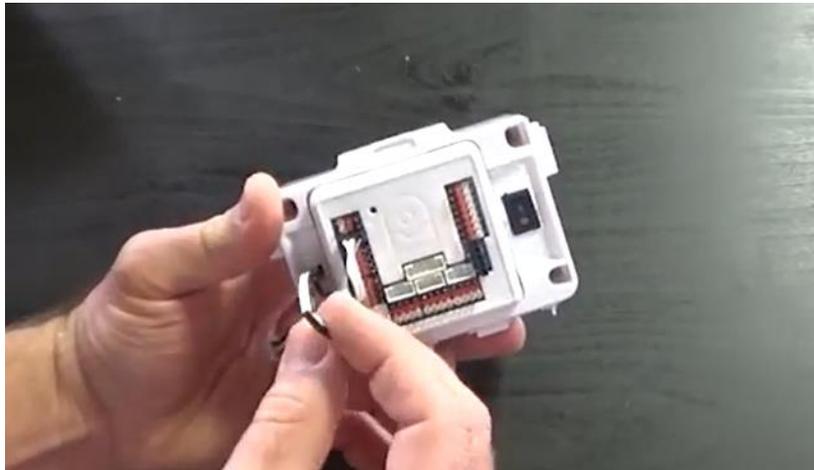
To begin building, **Clip'n'Play JD's Head Assembly** to the slot at the top of the humanoid body.

Connect the horizontal (left-to-right motion) servo to <u>D0</u>. Connect the vertical (up-and-down motion) servo to <u>D1</u>. Connect the **Camera** cable to the <u>camera port</u>. Connect the **RGB Eyes** cable to any of the <u>i2c ports</u>. Review the **Getting Started Guide** for a full diagram of the port configuration.
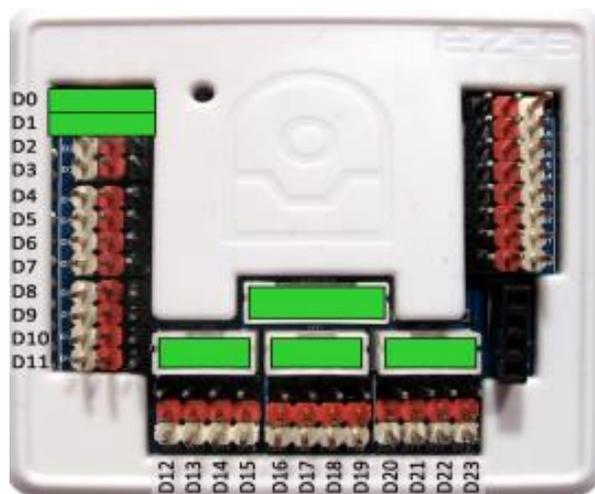
Rotate the left shoulder bracket to be vertically aligned, leaving the female slot open at the top of the bracket. **Clip'n'Play Left Upper Arm Servo** into the shoulder bracket with the white edge of the servo facing downwards. Connect the cable to <u>D4</u>.

Clip'n'Play Left Forearm Servo to the **Left Upper Arm Servo**. Connect the cable to <u>D5</u>.

**Clip'n'Play Left Gripper** to the **Left Forearm Servo**. Connect the cable to <u>D6</u>.
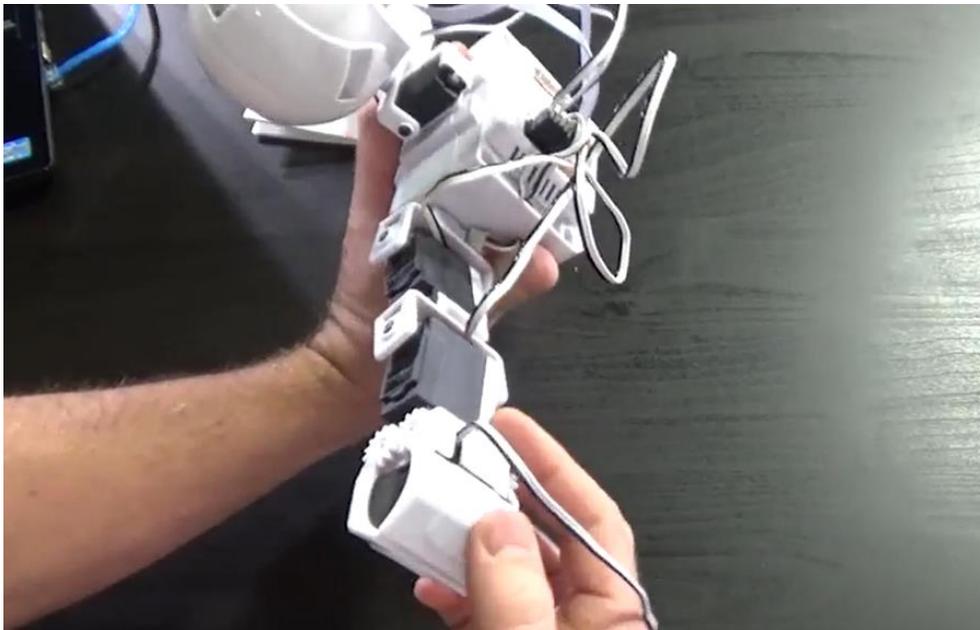
Repeat for the arm arm. Rotate the right shoulder bracket to be vertically aligned. **Clip'n'Play Right Upper Arm Servo** into the shoulder bracket with white servo edge facing downwards. Connect cable to **D7**.

Clip'n'Play Right Forearm Servo to the **Right Upper Arm Servo**. Connect the cable to **D8**.

Clip'n'Play Right Gripper to the **Right Forearm Servo**. Connect the cable to **D9**.
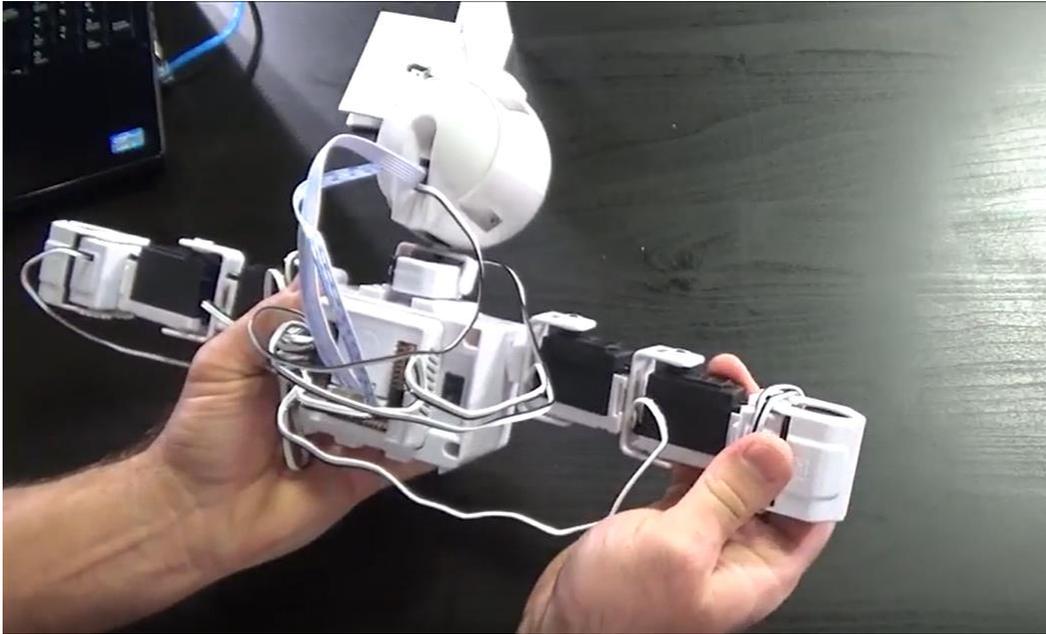
Clip'n'Play Left Upper Leg Servo with the white servo edge facing outwards. Connect the cable to **D12**.

Clip'n'Play Left Foot Assembly to **Left Upper Leg Servo** with the white servo edge facing outwards. Connect the left knee servo cable to **D13**. Connect the left ankle servo cable to **D14**.

Clip'n'Play Right Upper Leg Servo with the white servo edge facing outwards. Connect the cable to **D16**.

Clip'n'Play Right Foot Assembly to the **Right Upper Leg Servo** with the white servo edge facing outwards. Connect the right knee servo cable to **D17**. Connect the right ankle servo cable to **D18**.
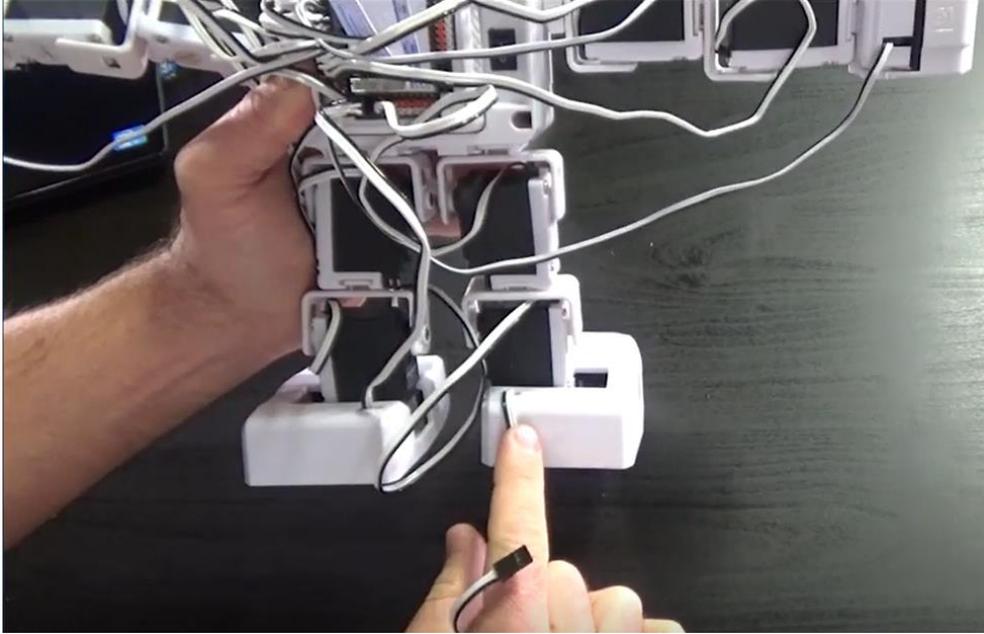
Check that the cable connections are color-matched to the ports and that the servos are correctly positioned. Use the 3D view buttons to check the build from all angles.
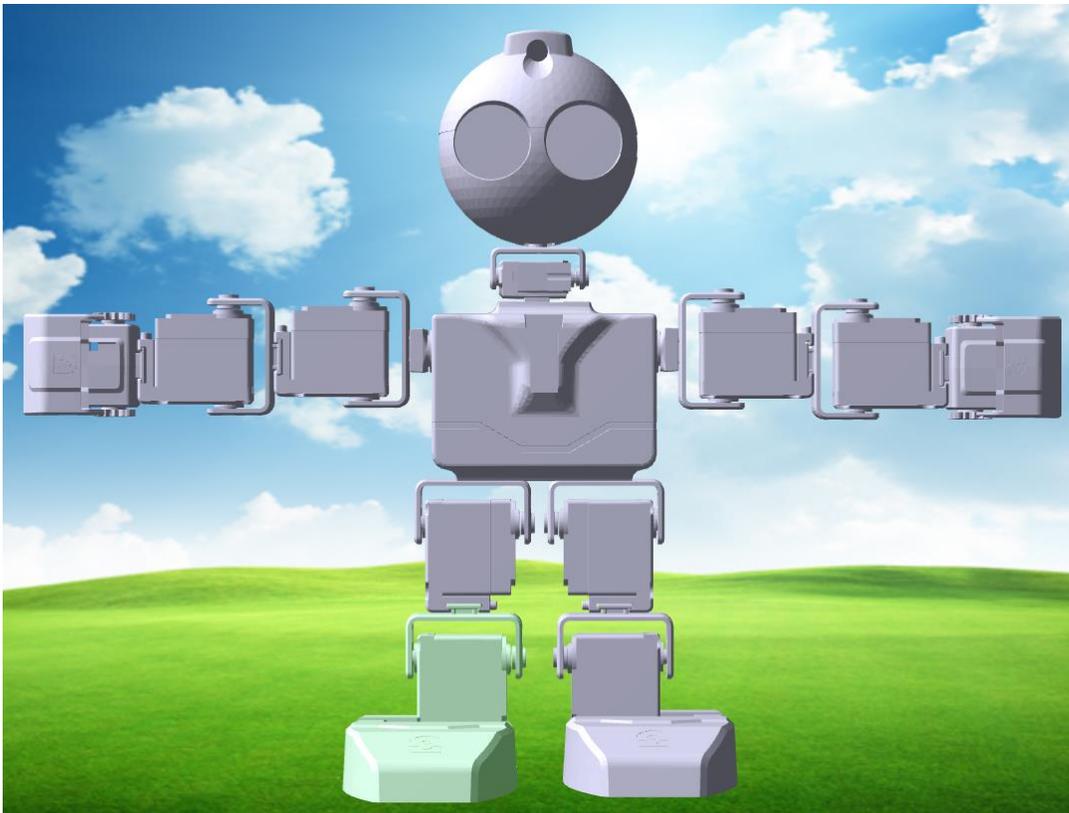
Use **Wire Wraps** to organize the cables. Leave cable slack near the servos for full range of motion. When wrapping the leg cables, begin the wire wrapping near the **EZ-B** and wrap downwards toward the leg servos. Test for full motion.
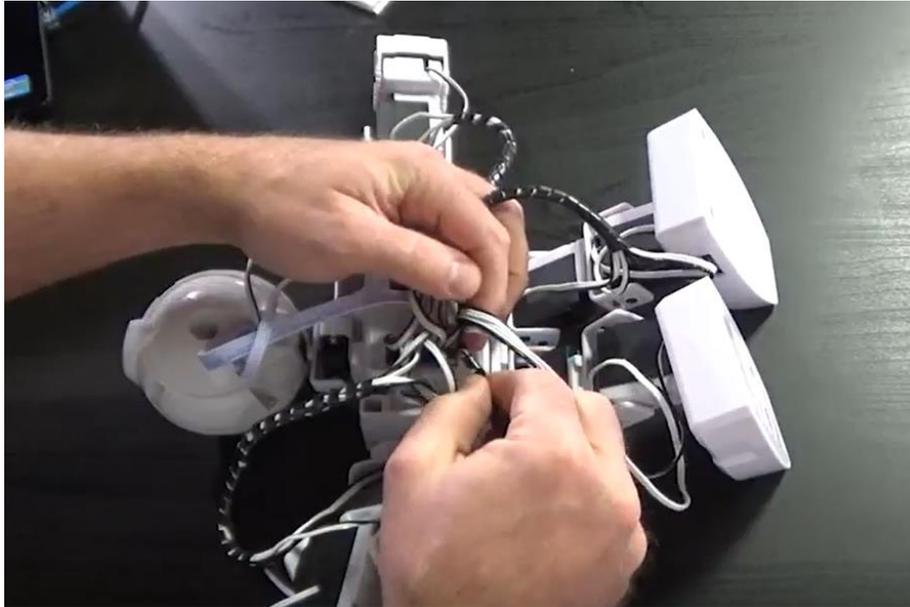
Your **JD Humanoid** is now complete!
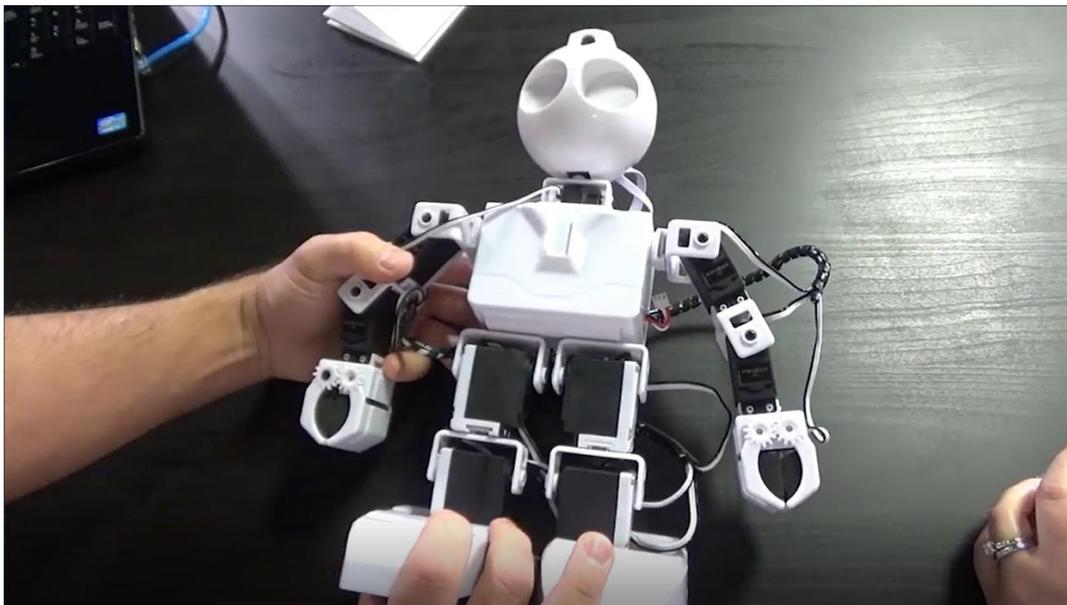
How do I connect the robot to the software?

Always start working with a fully charged robot.

Open **EZ-Builder**. Select **Example Projects** and load the **JD** project. **Skip** the building instructions and servo profile prompts.

Select the **EZ-B v4** Wi-Fi connection.

Select Connect to **EZ-B**. The robot should move into the initialization position.

JD should now be connected to the software and ready for use.

What is a servo profile and do I need to create one?

JD Humanoid should be balanced and the aligned once connected to the software. Do not manually manipulate the servos when the robot is powered on.

If the servos appear to be misaligned, they may need calibration to compensate for any slight hardware discrepancies. From **Project**, select **Configure** to create a new profile or **Load** to use an existing profile.

The servos of each arm and leg should be positioned in a straight line. Adjust the software servo values to line up each servo. Start with the servos closest to the body and move outwards. Adjust the servos of the head if needed. Adjust the leg servos as needed. **JD** should stand evenly on both feet. Once the servos are aligned, there should not be any grinding or vibrating noises. Click on **Save** and save the servo profile for future project use.
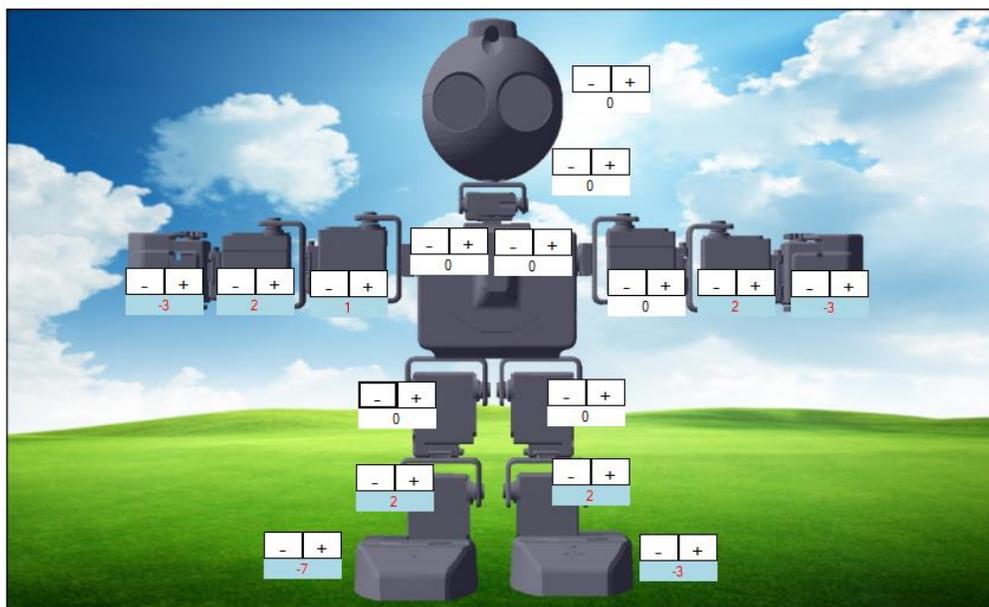
How do I get the robot to move?

There are many available controls in the **Example Project**.
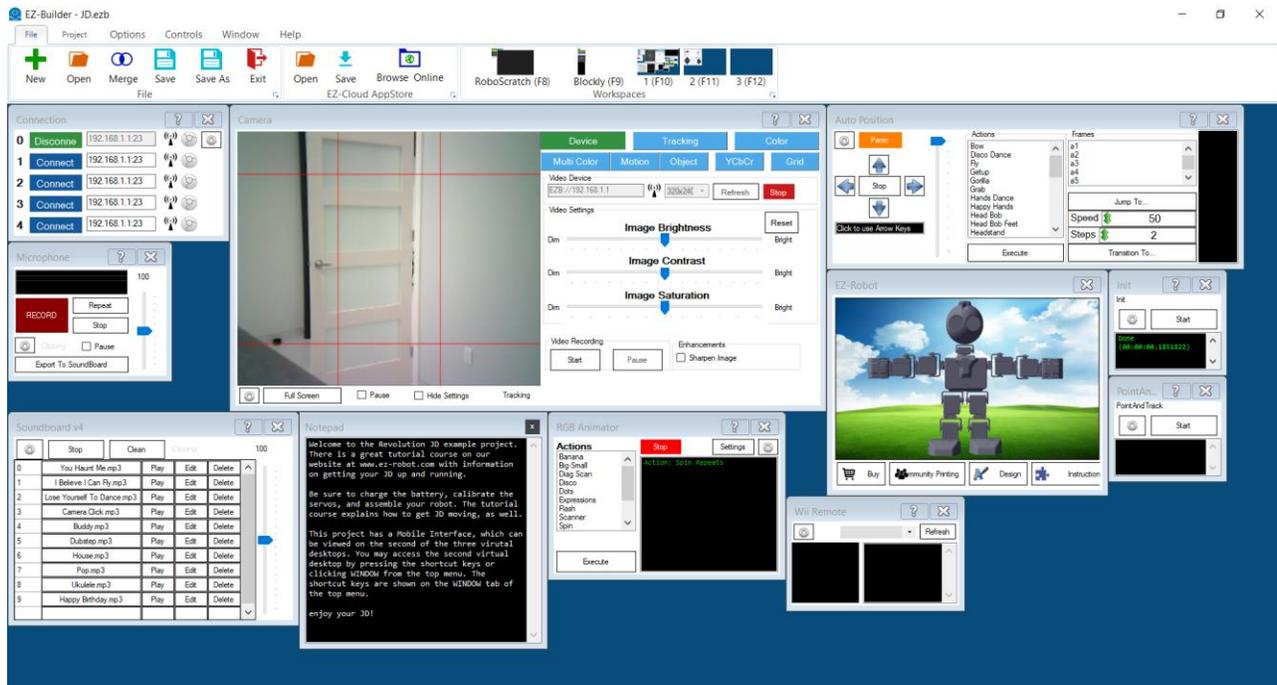
Scroll through the **Auto Position** actions to select the **Wave** command. Click the **Execute** button. **JD** should be waving his left arm. Try another one. Select **Headstand** from the **Auto Position** control and execute. Make sure **JD** has lots of room.

Control walking movement using the arrow keys.
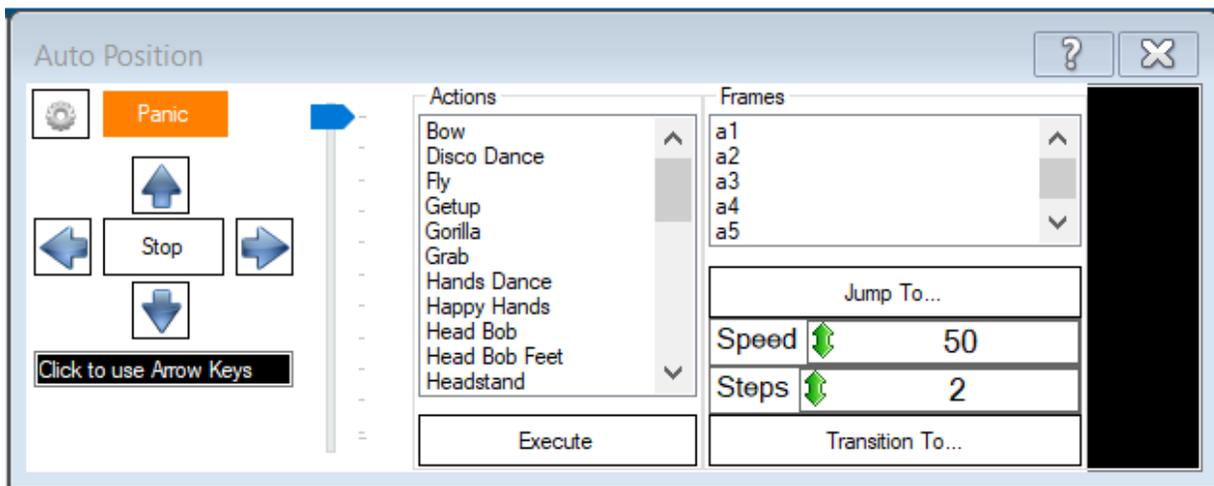
Try executing another pre-built command. In the **Soundboard** control, scroll to **I Believe I Can Fly** and select the **Play** button. Watch **JD** fly!
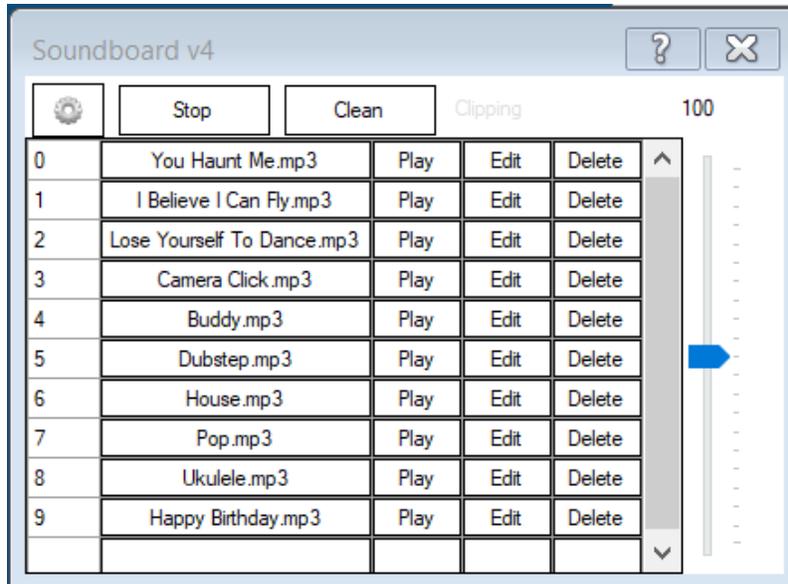
Last one! **Choose You Haunt Me** from the **Soundboard** to see **JD** perform a routine.
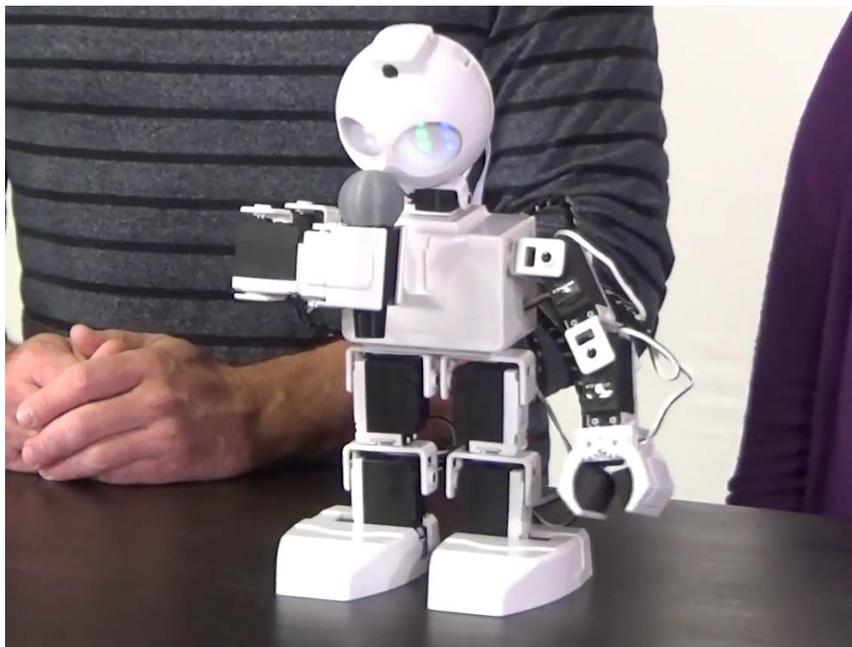
Continue to explore the available actions. Remember to disconnect, power off, and connect to the battery charger when finished.