

The Robot Program Episode 006: Introducing EZ-Builder

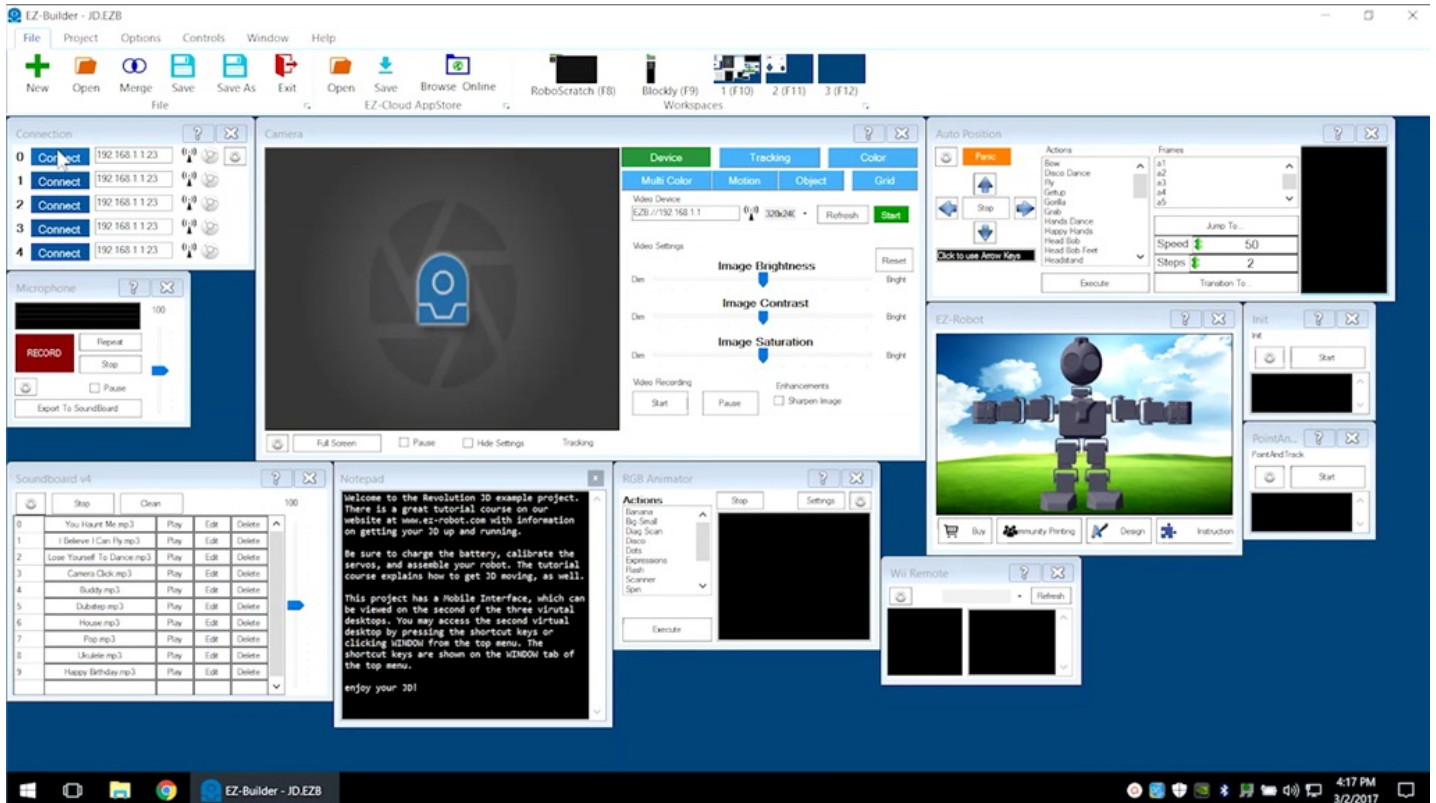
This lesson introduces the EZ-Builder Robot Software by exploring options and describing features. At the end of this lesson, the reader will be familiar with the overall layout and features of EZ-Builder. Follow along with The Robot Program Episode 006: Introducing EZ-Builder. View the video episode here: <https://www.ez-robot.com/Tutorials/Lesson/20>

Last Updated: 6/1/2018

Professor E's Overview

This lesson introduces and demonstrates the EZ-Builder software. Follow along with The Robot Program Episode 006: Introducing EZ-Builder. At the end of this lesson, readers will be familiar with the layout, workspaces, and available controls of the EZ-Builder software.

The RoboScratch and Blockly workspaces will also be introduced, as well as EZ-Script and third-party plugins. Blue question marks and window question marks can be used to find more information about a specific aspect or control.



Download **EZ-Builder** from www.ez-robot.com. Find more tutorials at **EZ-Robot School**.



The screenshot shows a web browser window displaying the EZ-Builder for Windows download page. The browser's address bar shows the URL www.ez-robot.com/EZ-Builder/. The website's navigation menu includes links for Explore, Products, Software, Learn, and Community, along with a Login / Signup option. A secondary menu lists various software categories: Windows, Plugins, Mobile, UniversalBot, Windows SDK, Mono SDK, Open IoT Wifi, and 3rd Party.

EZ-Builder For Windows

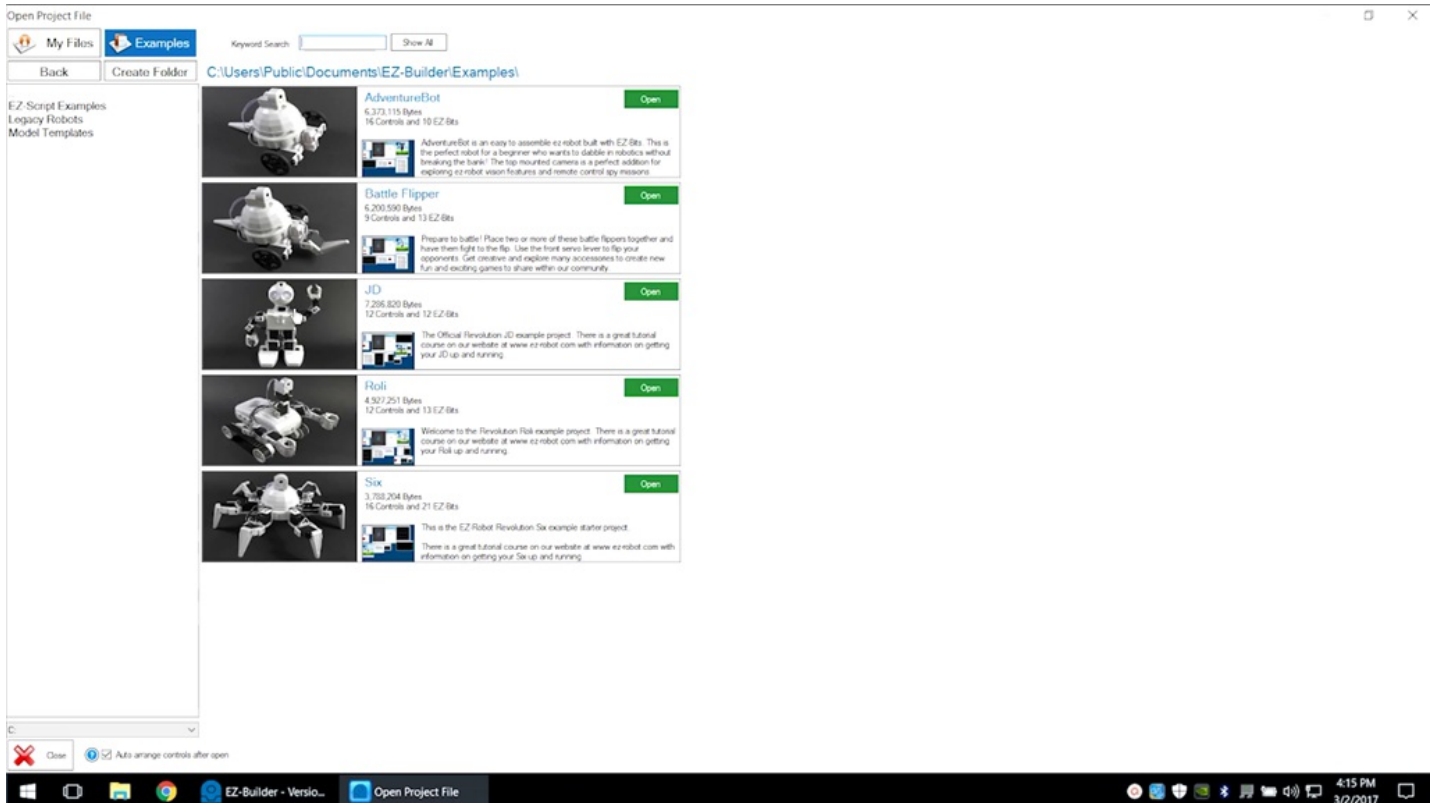
The main content area features a video player with the title "The EZ-Life... All The Robots!". Below the video are two buttons: "Download EZ-Builder Installer.msi" and "Manual". To the right of the video, there is a "Release notes" button and a section titled "EZ-Builder Version 2017.02.20.00".

The software for robots! World's easiest and most powerful robot software designed for **EZ-Robots** and **more**. Scales between beginner and advanced users, this software introduces amazing features that will bring your robot to life by combining engineering and creativity.

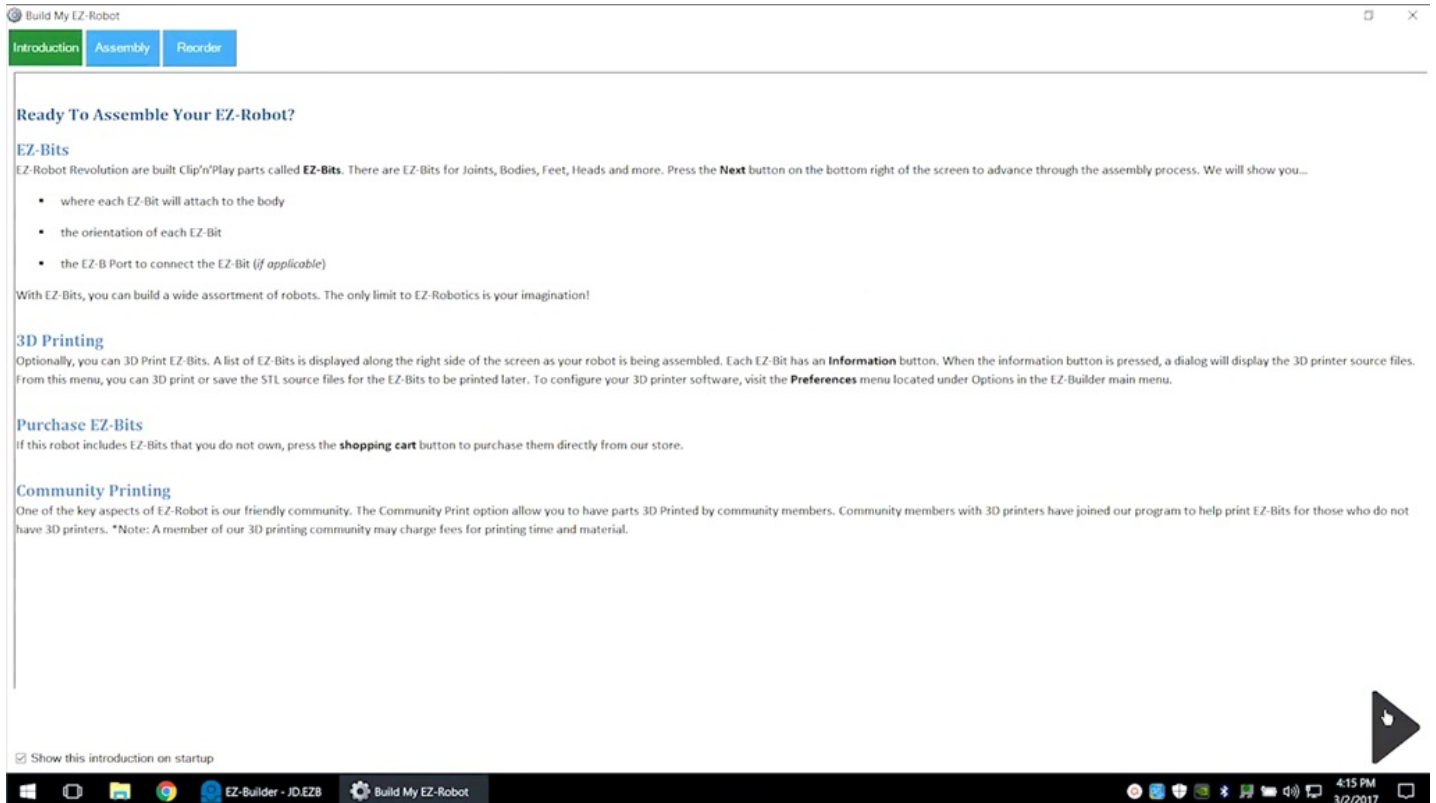
The Windows taskbar at the bottom shows the system tray with the time 4:14 PM and date 3/2/2017. The active window title is "EZ-Builder for Win...".

Loading an Example Project

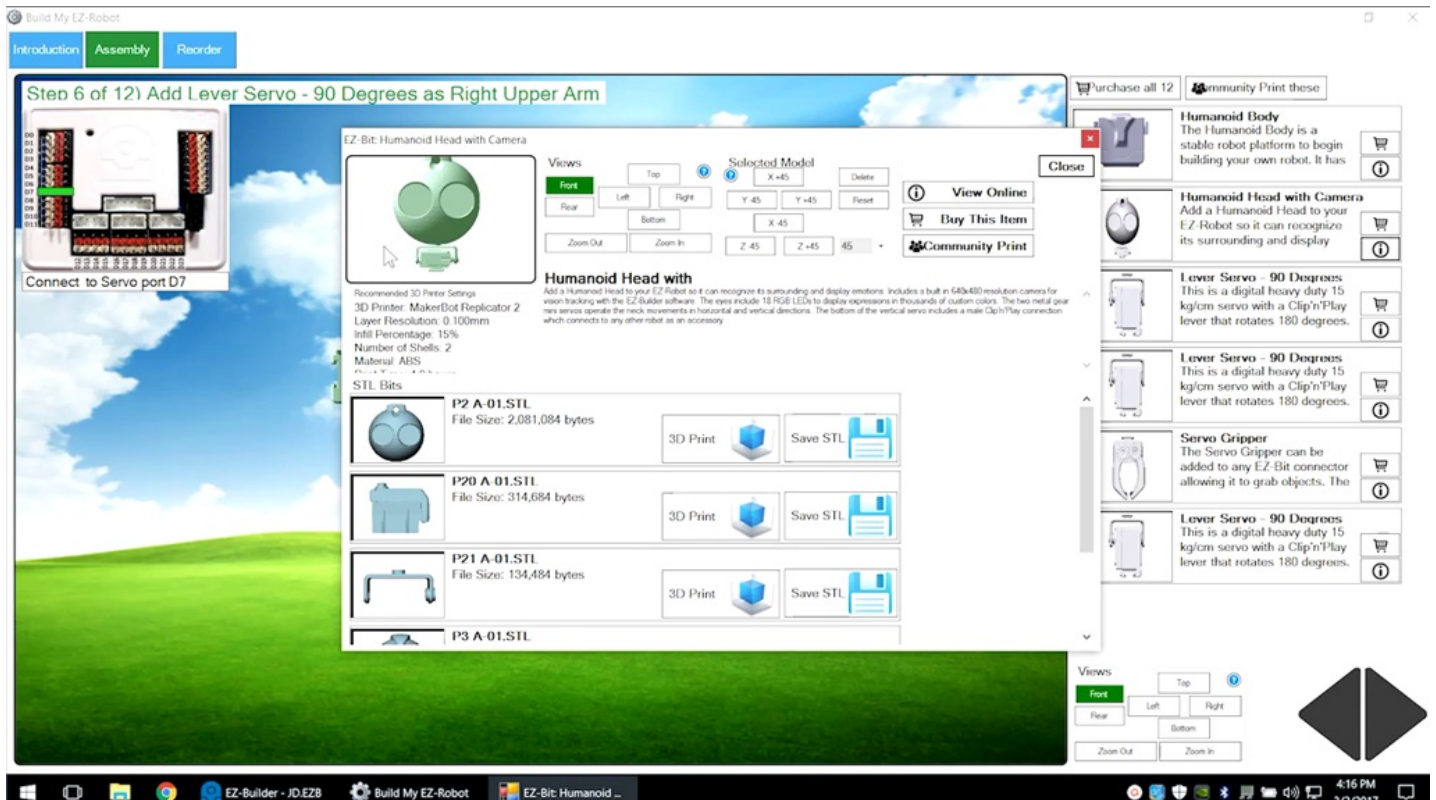
Power on the robot. This example uses **Revolution JD**. Load an example project for the robot.



View building instructions by selecting **Instructions** from the **Project** tab.



Click on the **i** icon for more information about a part, including 3D printing designs.



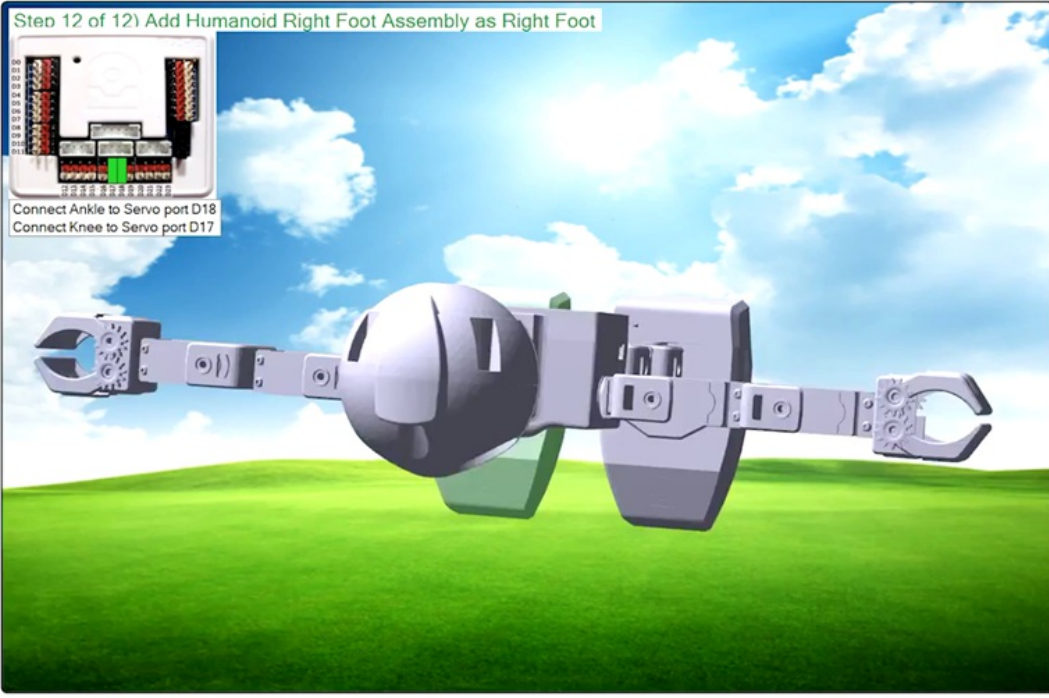
Use the arrow buttons to view the robot from different angles.

Build My EZ-Robot

Introduction Assembly **Reorder**

Step 12 of 12) Add Humanoid Right Foot Assembly as Right Foot

Connect Ankle to Servo port D18
Connect Knee to Servo port D17




Purchase all 12 Community Print these

- Humanoid Body**
The Humanoid Body is a stable robot platform to begin building your own robot. It has
- Humanoid Head with Camera**
Add a Humanoid Head to your EZ-Robot so it can recognize its surrounding and display
- Lever Servo - 90 Degrees**
This is a digital heavy duty 15 kg/cm servo with a Clip'n'Play lever that rotates 180 degrees.
- Lever Servo - 90 Degrees**
This is a digital heavy duty 15 kg/cm servo with a Clip'n'Play lever that rotates 180 degrees.
- Servo Gripper**
The Servo Gripper can be added to any EZ-Bit connector allowing it to grab objects. The
- Lever Servo - 90 Degrees**
This is a digital heavy duty 15 kg/cm servo with a Clip'n'Play lever that rotates 180 degrees.
- Lever Servo - 90 Degrees**
This is a digital heavy duty 15 kg/cm servo with a Clip'n'Play lever that rotates 180 degrees.

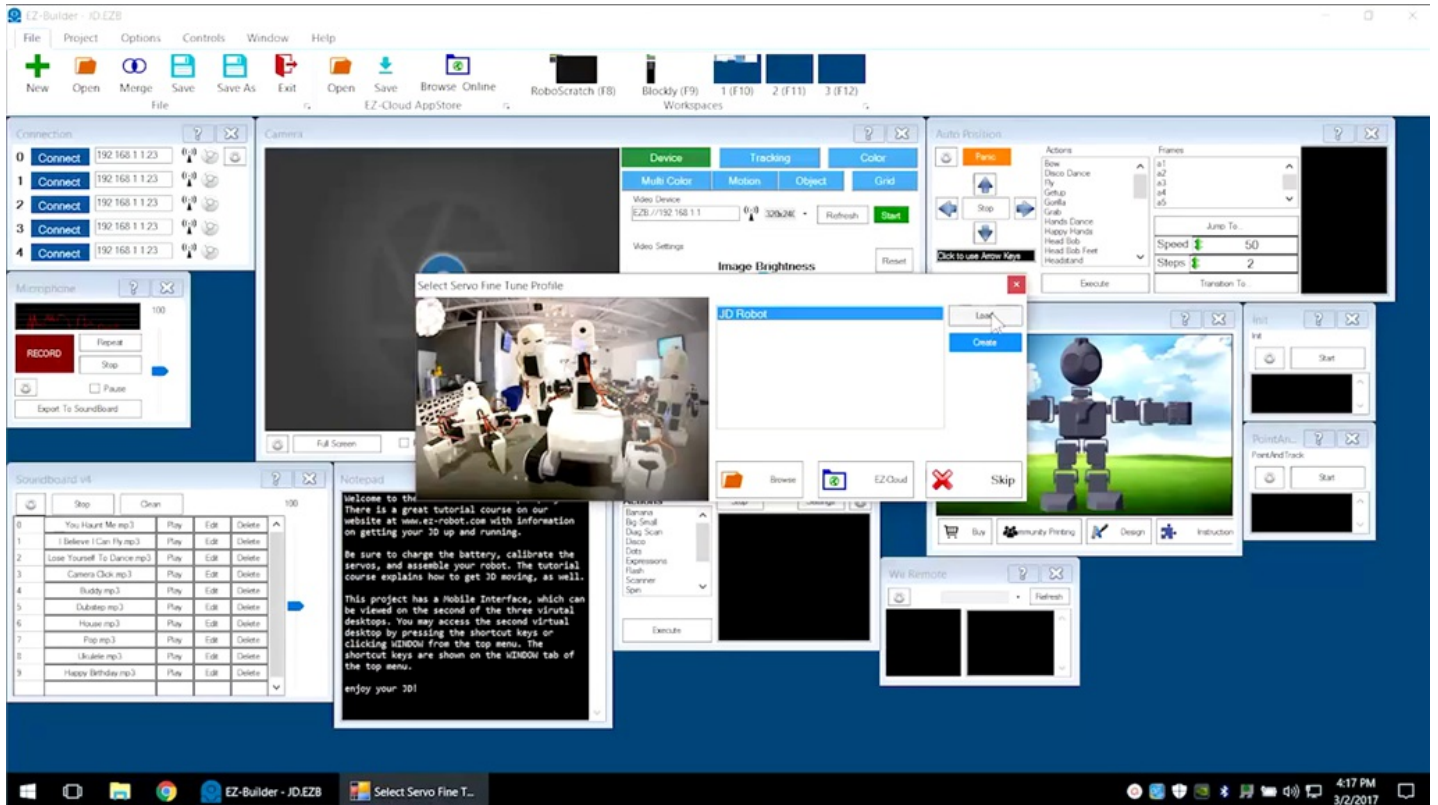
Views

Front Top Right
Rear Bottom
Zoom Out Zoom In

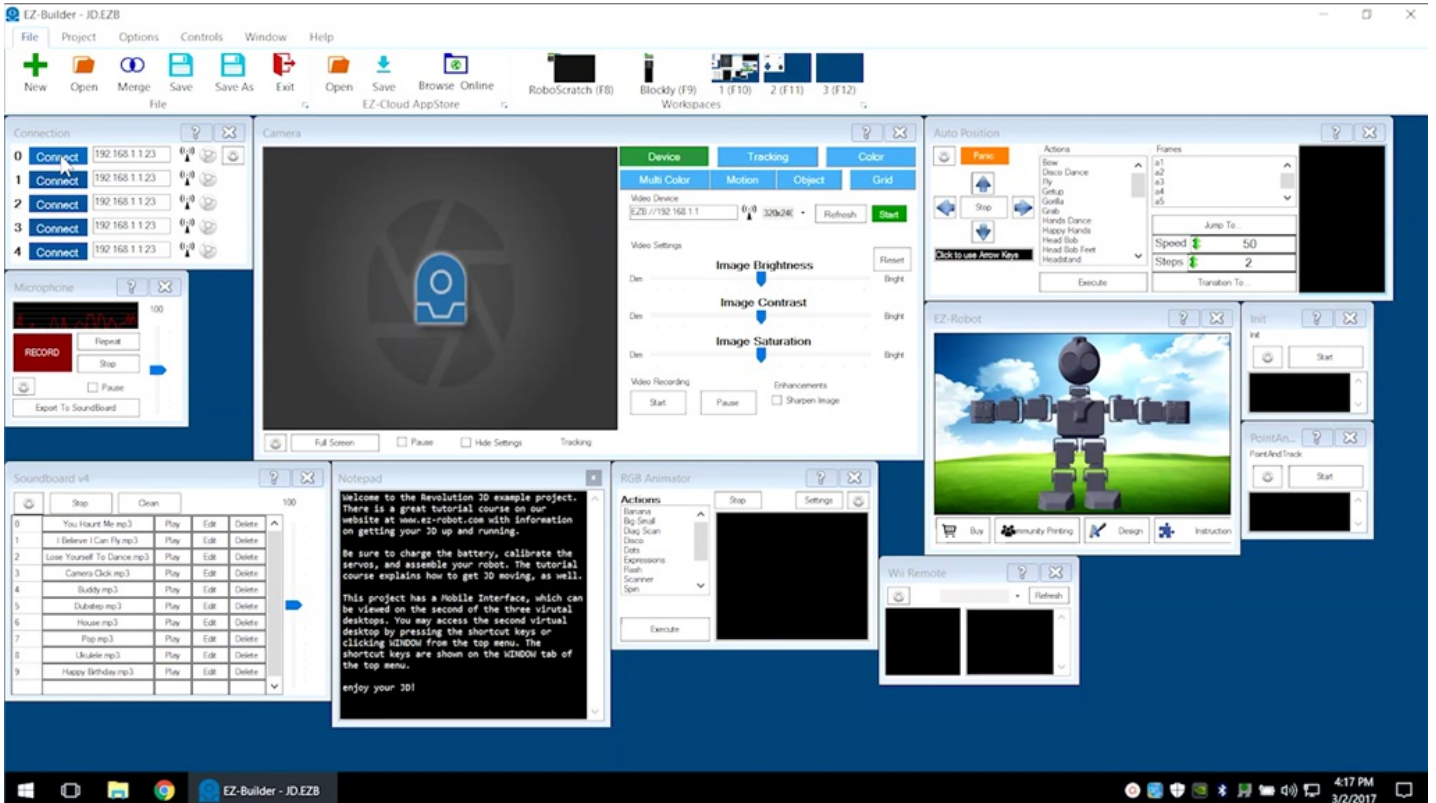


Windows Taskbar: EZ-Builder - JD.EZ8 Build My EZ-Robot 4:16 PM 3/2/2017

Load a servo profile if needed (humanoid robots only).



Connect to the robot using the **EZ-B v4** Wi-Fi connection. Click on the blue **Connect** button. Once connected, each robot has an initialization pose.



Microphone Control

There are many different control windows. Use the **Microphone** control to record and playback a sound.

The screenshot displays the EZ-Builder software interface with several control windows open:

- Connection:** A list of connection attempts with IP addresses (192.168.1.123) and status (Disconnect, Connect).
- Microphone:** A window with a volume slider at 100, a red 'RECORD' button, and options for 'Repair', 'Stop', 'Pause', and 'Export To Soundboard'.
- Camera:** A central window showing a live video feed of a person at a desk with a camera. It includes 'Image Brightness', 'Image Contrast', and 'Image Saturation' sliders, and 'Video Recording' controls.
- Auto Position:** A window with 'Actions' (Dance, Fly, Grab, etc.) and 'Frames' (a1-a5) for programming movements.
- Soundboard v4:** A table with columns for song name, play button, edit, and delete. Songs include 'You Heart Me.mp3', 'I Believe I Can Fly.mp3', etc.
- Notepad:** A text editor window containing a welcome message and instructions for the 'Revolution 3D' project.
- RGB Animator:** A window for setting actions like 'Spin' and 'Repeat'.
- EZ-Robot:** A window showing a 3D model of a robot on a green field.
- Wii Remote:** A window with a 'Refresh' button.

The Windows taskbar at the bottom shows the time as 4:18 PM on 3/2/2017.

Use the **Soundboard** control to play and edit audio files. Code can also be added in sync with the audio waveform.

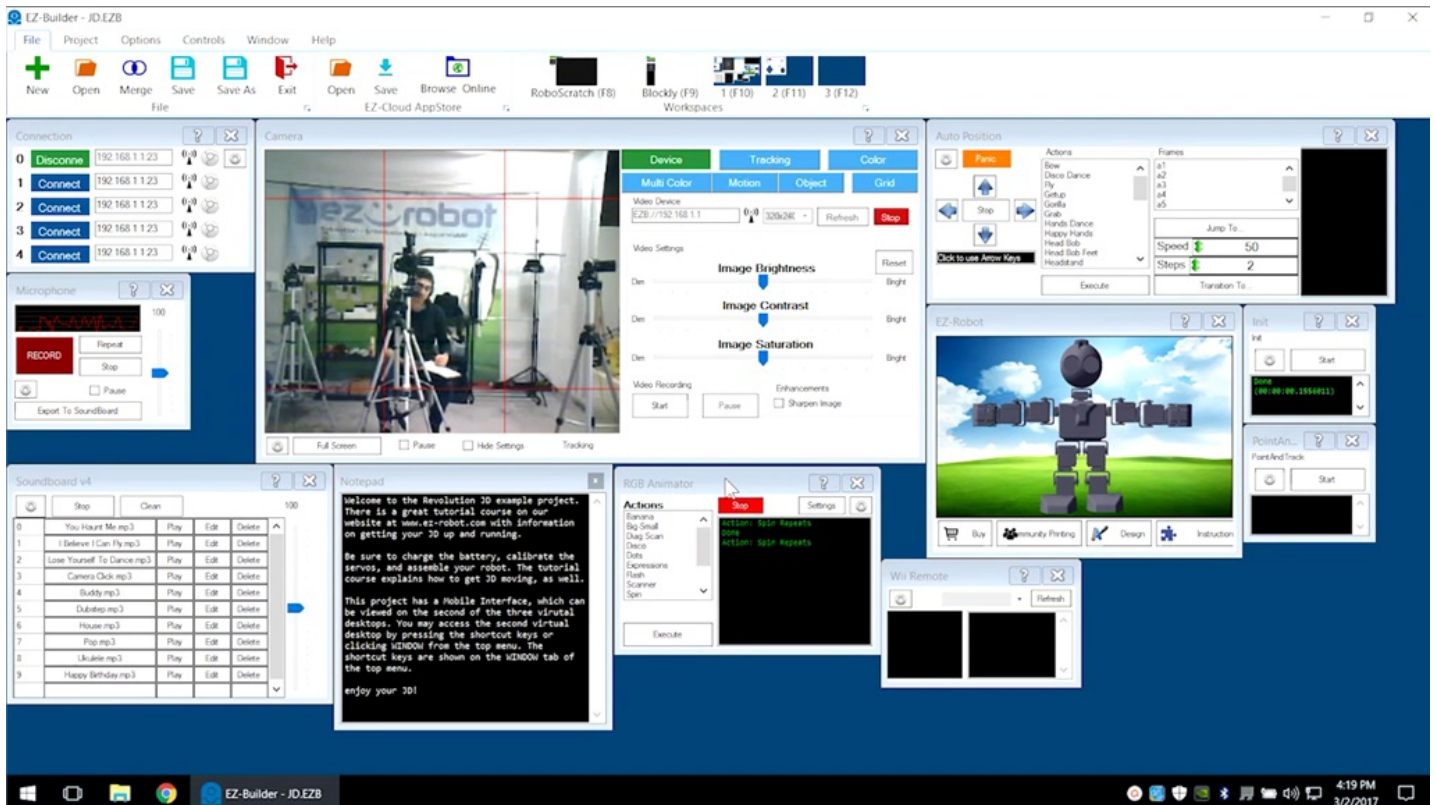
The screenshot displays the EZ-Builder software interface for a robot project. The main window is titled "EZ-Builder - JD.EZB" and features a menu bar (File, Project, Options, Controls, Window, Help) and a toolbar with icons for New, Open, Merge, Save, Save As, Exit, Open, Save, Browse Online, and Workspaces. The interface is divided into several panels:

- Connection:** A list of connection points with IP addresses (192.168.1.123) and status indicators (Disconnect, Connect).
- Microphone:** A control panel with a volume slider, a "RECORD" button, and options for Repair, Stop, Pause, and Export To Soundboard.
- Camera:** A central video feed showing a robot in a room, overlaid with a red tracking grid. Below the feed are controls for Full Screen, Pause, Hide Settings, and Tracking.
- Device Tracking:** A panel with tabs for Multi Color, Motion, Object, and Grid. It includes a video device selection dropdown (EZB/192.168.1.1), a refresh button, and a stop button. Below are sliders for Image Brightness, Image Contrast, and Image Saturation, each with a "Bright" label and a "Reset" button. There are also "Start" and "Pause" buttons for Video Recording and a "Sharpen Image" checkbox.
- Auto Position:** A panel with "Panic" and "Stop" buttons, and a "Click to use Arrow Keys" button. It includes an "Actions" list (Slow, Disco Dance, Fly, GetUp, GoFis, Grab, Hands Dance, Happy Hands, Head Bob, Headband) and a "Frames" list (a1, a2, a3, a4, a5). It also has "Jump To" and "Steps" controls.
- EZ-Robot:** A 3D model of a robot on a green field under a blue sky. It includes buttons for Buy, Community Posting, Design, and Instruction.
- Soundboard v4:** A table with columns for audio files and actions. The files listed are: You Hurt Me.mp3, I Believe I Can Fly.mp3, Lose Yourself To Dance.mp3, Camera Click.mp3, Buddy.mp3, Dubstep.mp3, House.mp3, Pop.mp3, Ukulele.mp3, and Happy Birthday.mp3. Each file has "Play", "Edit", and "Delete" buttons.
- Notepad:** A text editor window containing a welcome message and instructions for the robot project.
- RGB Animator:** A panel with a "Stop" button and a "Settings" button. It includes a list of "Activities" (Timers, Big Small, Diag Scan, Disco, Data, Expressions, Flash, Scanner, Spin) and an "Execute" button.
- Wii Remote:** A panel with a "Refresh" button.

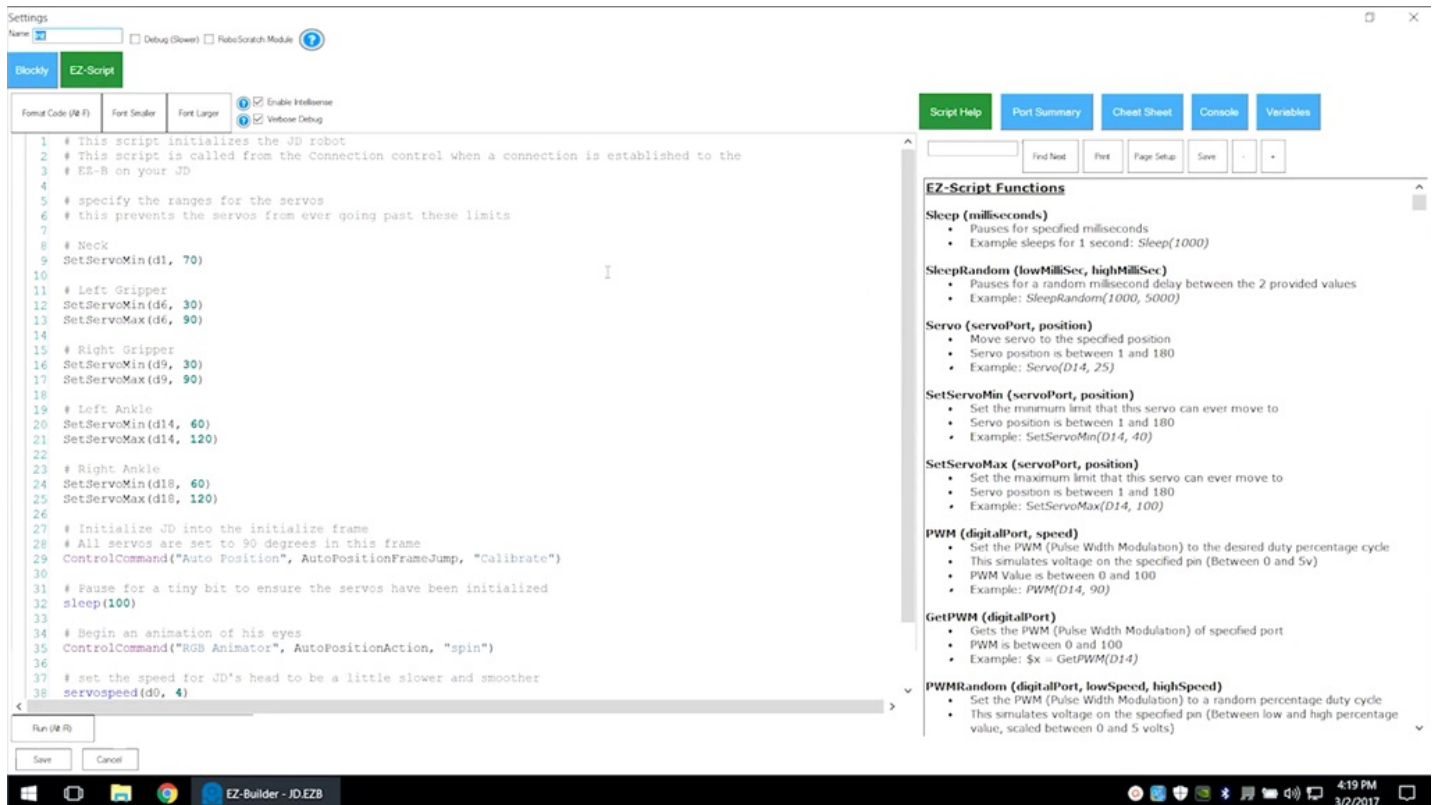
The Windows taskbar at the bottom shows the system tray with the time 4:18 PM and date 3/2/2017.

Commonly Used Controls

Other controls include **Camera**, **RGB Animator**, and **PointAndTrack**.



Click on the **Gear Icon** to see the control configuration code.



The screenshot displays the EZ-Builder software interface. The main window is titled "Settings" and contains a script editor on the left and a help panel on the right.

Script Editor:

```
1 # This script initializes the JD robot
2 # This script is called from the Connection control when a connection is established to the
3 # EZ-B on your JD
4
5 # specify the ranges for the servos
6 # this prevents the servos from ever going past these limits
7
8 # Neck
9 SetServoMin(d1, 70)
10
11 # Left Gripper
12 SetServoMin(d6, 30)
13 SetServoMax(d6, 90)
14
15 # Right Gripper
16 SetServoMin(d9, 30)
17 SetServoMax(d9, 90)
18
19 # Left Ankle
20 SetServoMin(d14, 60)
21 SetServoMax(d14, 120)
22
23 # Right Ankle
24 SetServoMin(d18, 60)
25 SetServoMax(d18, 120)
26
27 # Initialize JD into the initialize frame
28 # All servos are set to 90 degrees in this frame
29 ControlCommand("Auto Position", AutoPositionFrameJump, "Calibrate")
30
31 # Pause for a tiny bit to ensure the servos have been initialized
32 sleep(100)
33
34 # Begin an animation of his eyes
35 ControlCommand("RGB Animator", AutoPositionAction, "spin")
36
37 # set the speed for JD's head to be a little slower and smoother
38 servospeed(d0, 4)
```

Help Panel (EZ-Script Functions):

- Sleep (milliseconds)**
 - Pauses for specified milliseconds
 - Example sleeps for 1 second: `Sleep(1000)`
- SleepRandom (lowMilliSec, highMilliSec)**
 - Pauses for a random millisecond delay between the 2 provided values
 - Example: `SleepRandom(1000, 5000)`
- Servo (servoPort, position)**
 - Move servo to the specified position
 - Servo position is between 1 and 180
 - Example: `Servo(D14, 25)`
- SetServoMin (servoPort, position)**
 - Set the minimum limit that this servo can ever move to
 - Servo position is between 1 and 180
 - Example: `SetServoMin(D14, 40)`
- SetServoMax (servoPort, position)**
 - Set the maximum limit that this servo can ever move to
 - Servo position is between 1 and 180
 - Example: `SetServoMax(D14, 100)`
- PWM (digitalPort, speed)**
 - Set the PWM (Pulse Width Modulation) to the desired duty percentage cycle
 - This simulates voltage on the specified pin (Between 0 and 5v)
 - PWM Value is between 0 and 100
 - Example: `PWM(D14, 90)`
- GetPWM (digitalPort)**
 - Gets the PWM (Pulse Width Modulation) of specified port
 - PWM is between 0 and 100
 - Example: `$x = GetPWM(D14)`
- PWMRandom (digitalPort, lowSpeed, highSpeed)**
 - Set the PWM (Pulse Width Modulation) to a random percentage duty cycle
 - This simulates voltage on the specified pin (Between low and high percentage value, scaled between 0 and 5 volts)

The interface also includes a "Run (R/R)" button, "Save", and "Cancel" buttons at the bottom. The taskbar at the bottom shows the application name "EZ-Builder - JD.EZB" and the system clock "4:19 PM 3/2/2017".

Camera control can be used to change the robot camera settings and to track objects.

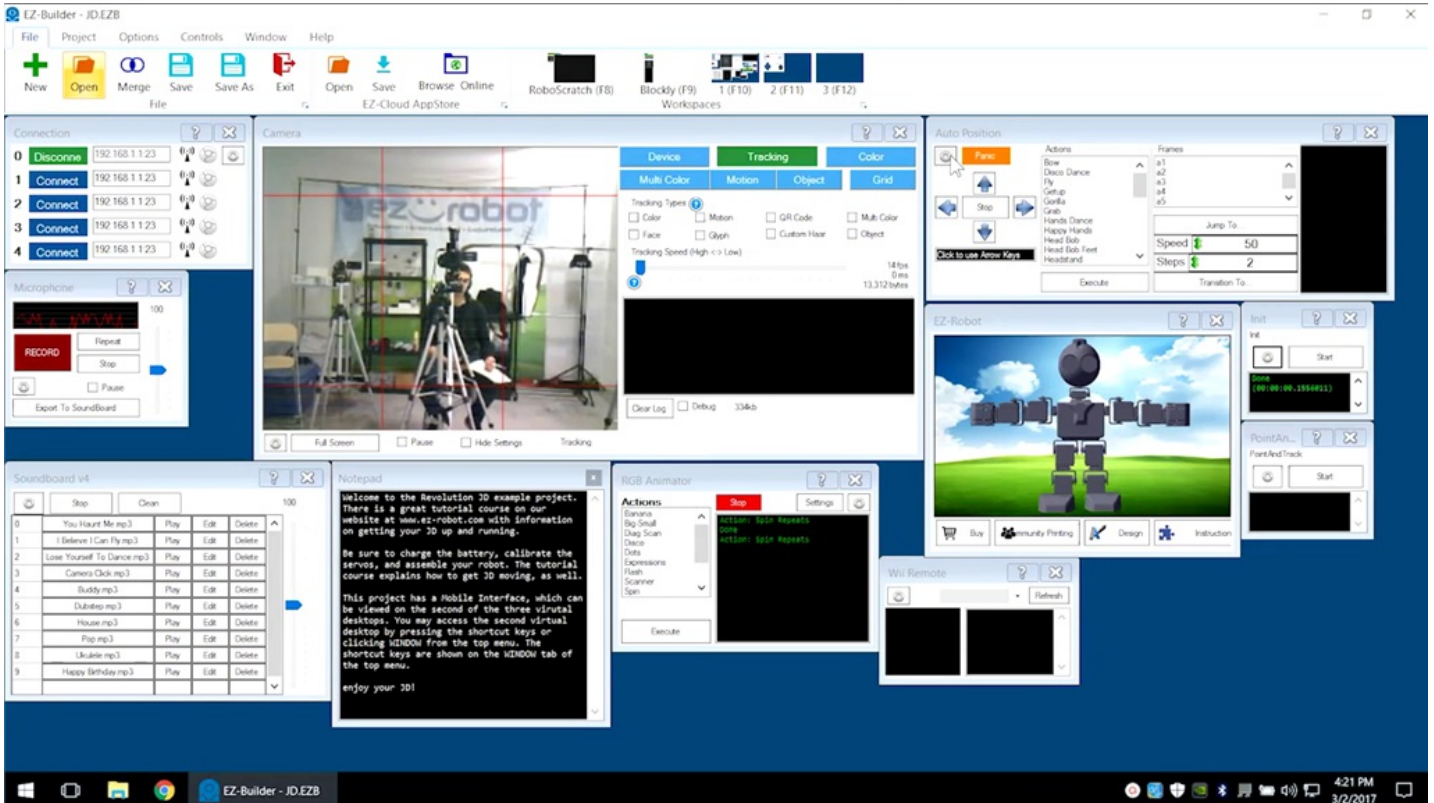
The screenshot displays the EZ-Builder software interface, which is used for controlling a robot. The interface is divided into several panels:

- Connection:** A list of connection attempts with IP addresses (192.168.1.123) and status (Connect/Disconnect).
- Microphone:** A panel with a 'RECORD' button and volume control.
- Soundboard v4:** A table with columns for 'Step', 'Clean', and '100'. It lists various audio files like 'You Heart Me.mp3', 'I Believe I Can Fly.mp3', etc., with 'Play', 'Edit', and 'Delete' buttons for each.
- Notepad:** A text editor containing a welcome message and instructions for the 'Revolution 3D' project.
- RGB Animator:** A panel with 'Actions' and 'Settings' tabs, showing a list of actions like 'Spin Repeats'.
- Wii Remote:** A panel with a 'Refresh' button and two black rectangular areas.
- Camera:** A central panel showing a live video feed of a robot in a room. It includes a 'Tracking' section with tabs for 'Device', 'Motion', 'Object', and 'Color'. The 'Object' tab is selected, showing tracking types like 'Color', 'Motion', 'QR Code', 'Multi Color', 'Face', 'Graph', 'Custom Haar', and 'Object'. There are also checkboxes for 'Color', 'Motion', 'QR Code', 'Multi Color', 'Face', 'Graph', 'Custom Haar', and 'Object'. A 'Tracking Speed' slider is set to 'High (> Low)'. Below the video feed are buttons for 'Full Screen', 'Pause', 'Hide Settings', and 'Tracking'.
- Auto Position:** A panel with 'Actions' and 'Frames' tabs. The 'Actions' tab is selected, showing a list of actions like 'Bow', 'Deco Dance', 'Fly', 'GetUp', 'GoFora', 'Grab', 'Hands Dance', 'Happy Hands', 'Head Bob', 'Head Bob Feet', and 'Headstand'. There are 'Stop' and 'Click to use Arrow Keys' buttons. The 'Frames' tab shows a list of frames (a1 to a5) and a 'Jump To' dropdown. A 'Speed' slider is set to 50, and a 'Steps' dropdown is set to 2. There is an 'Execute' button.
- EZ-Robot:** A panel showing a 3D model of a robot on a green field under a blue sky. It has buttons for 'Buy', 'Community Pinning', 'Design', and 'Instruction'.
- Init:** A panel with a 'Start' button and a list of items.
- PointAn...:** A panel with a 'Start' button and a list of items.

The Windows taskbar at the bottom shows the time as 4:20 PM on 3/2/2017.

Auto Position Control

Auto Position is a movement panel. Each robot has its own type of movement panel for controlling motion.



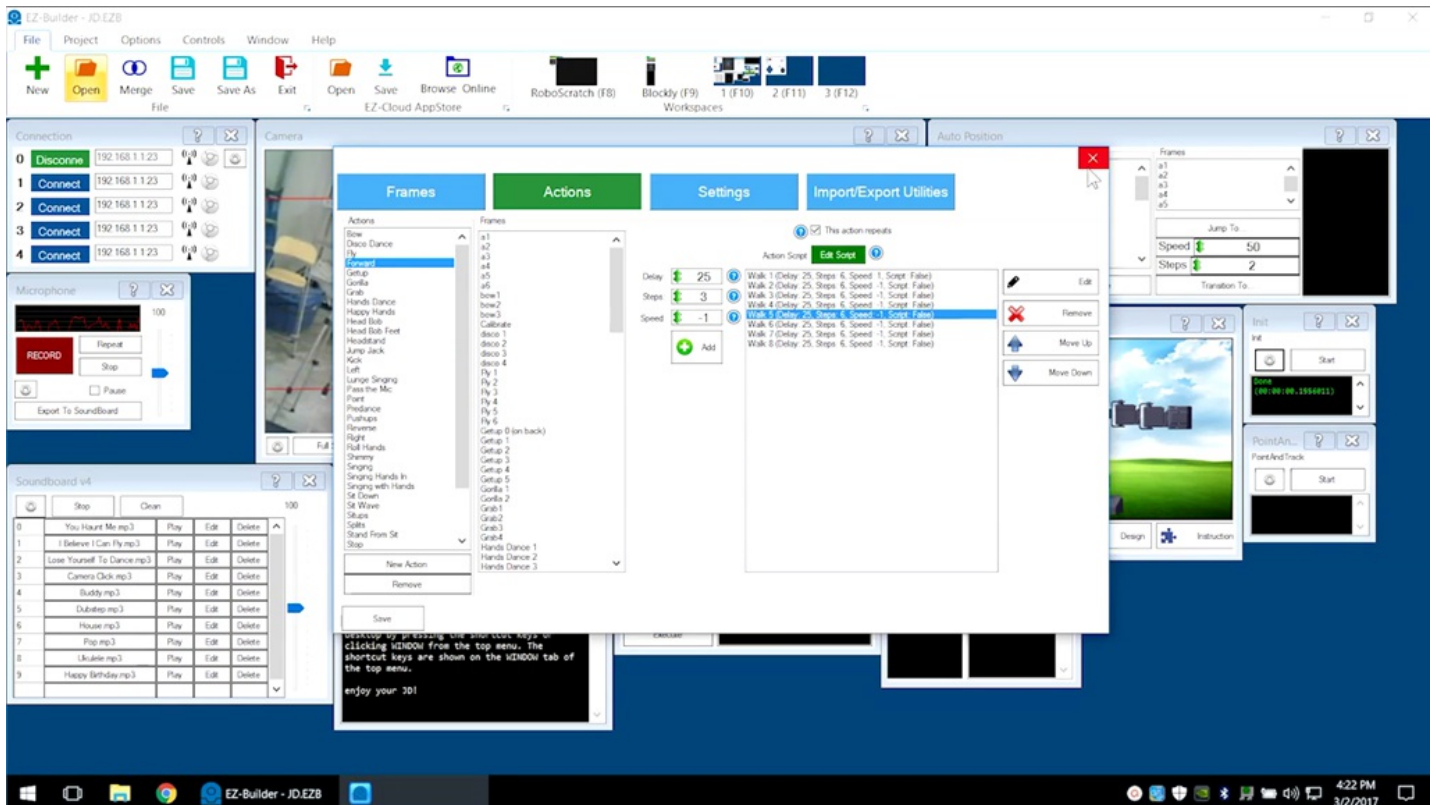
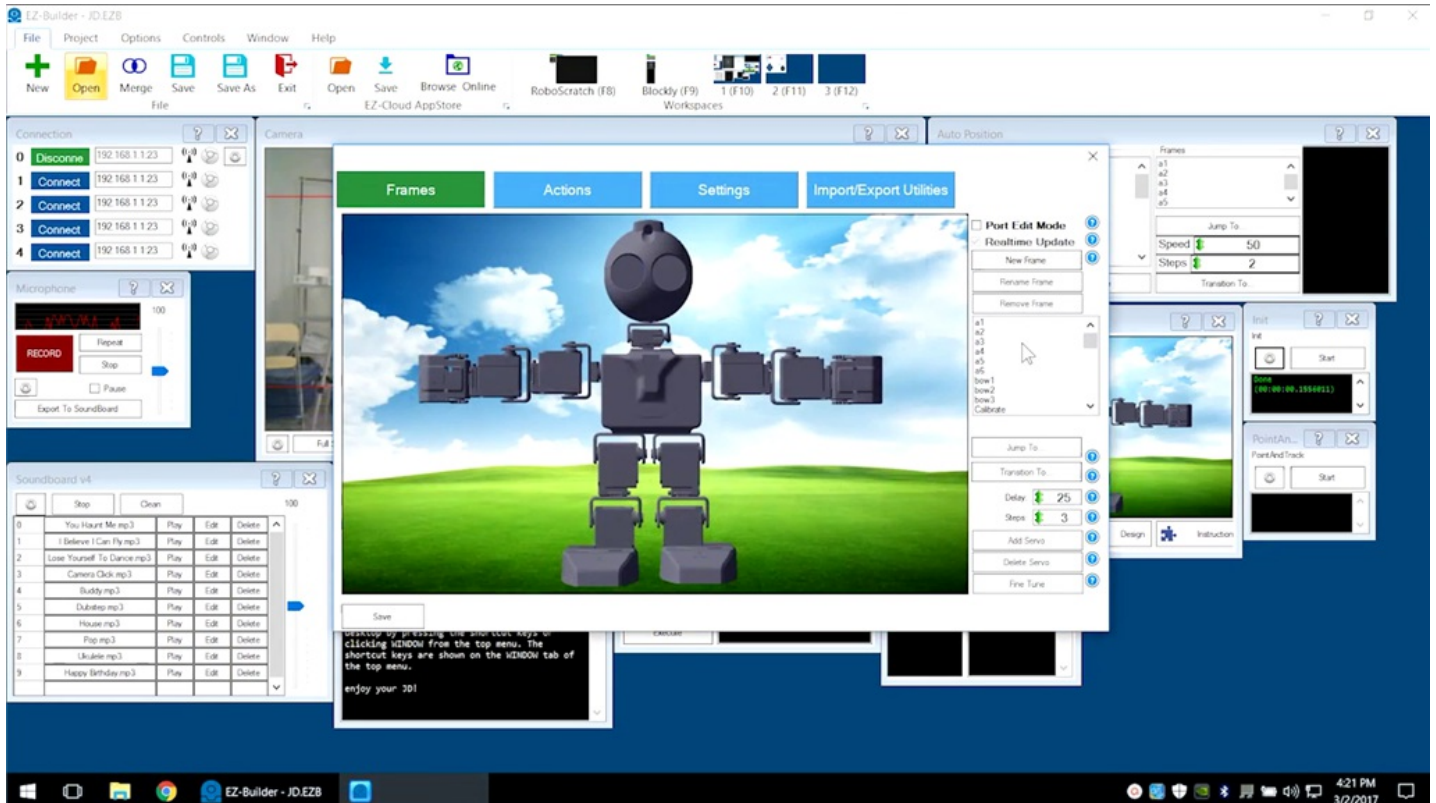
The screenshot displays the EZ-Builder software interface, which is used for controlling an EZ-Robot. The interface is divided into several panels:

- Connection:** A list of connection attempts with IP addresses (192.168.1.123) and status (Disconnect, Connect).
- Microphone:** A panel with a volume slider and buttons for Record, Repeat, Stop, and Export To Soundboard.
- Soundboard v4:** A table of audio files with columns for file name, play button, edit, and delete.
- Notepad:** A text editor containing a welcome message and instructions for the robot project.
- Camera:** A live video feed of the robot in a room, with a red bounding box around it. Below the feed are tracking settings for Device, Tracking, and Color.
- Auto Position:** A panel for configuring movement actions, including a list of actions (Bow, Disco Dance, Fly, etc.), a speed slider, and a steps counter.
- EZ-Robot:** A 3D model of the robot on a green field, with buttons for Buy, Community Posting, Design, and Instruction.
- RGB Animator:** A panel for configuring RGB lighting effects, including a list of actions and a stop button.
- Wii Remote:** A panel for configuring Wii Remote controls, including a refresh button.

The Windows taskbar at the bottom shows the time as 4:21 PM on 3/2/2017.

Frame Creation

Click on the **Auto Position** gear icon to create frame-by-frame movement control.



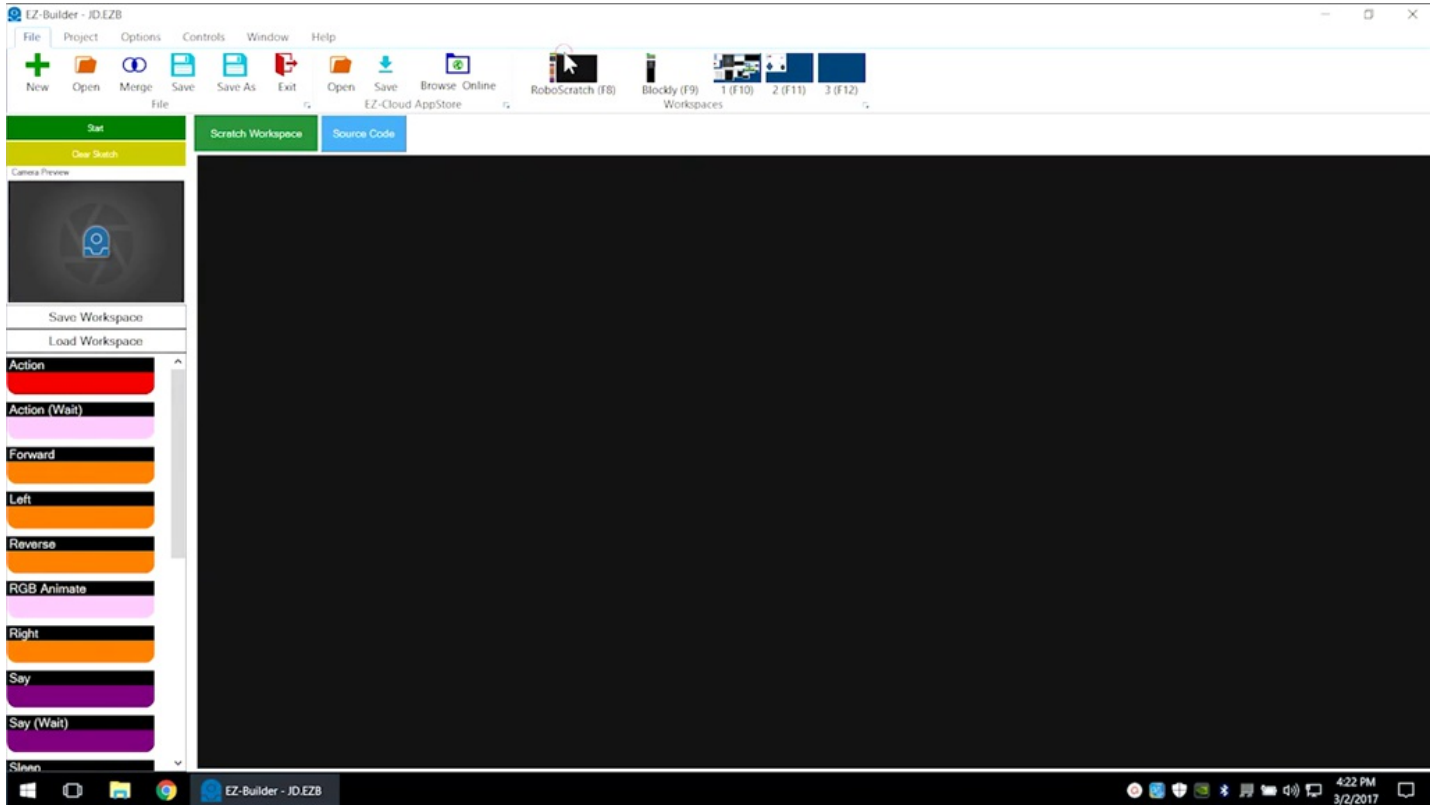
Custom controls can also be created through coding. View available coding **Workspaces** using the **File** tab.

The screenshot displays the EZ-Builder software interface, which is used for programming and controlling a robot. The interface is divided into several panels:

- Connection:** A list of connection attempts with IP addresses (192.168.1.123) and status (Disconnect, Connect).
- Microphone:** A control panel with a volume slider and buttons for Record, Repair, Stop, and Export To Soundboard.
- Soundboard v4:** A table of audio files with columns for file name, play button, edit, and delete.
- Notepad:** A text editor containing a welcome message and instructions for the Revolution 3D example project.
- Camera:** A live video feed of a person operating a robot in a room, with a red tracking box overlaid on the robot.
- Tracking:** A control panel for tracking the robot, with tabs for Multi Color, Motion, Object, and Grid. It includes checkboxes for various tracking types and a tracking speed slider.
- RGB Animator:** A control panel for animating the robot's RGB lights, with a list of actions and a stop button.
- Wii Remote:** A control panel for using a Wii Remote to control the robot, with a refresh button.
- Auto Position:** A control panel for setting the robot's auto position, with a list of actions and a speed slider.
- EZ-Robot:** A 3D model of the robot in a virtual environment.

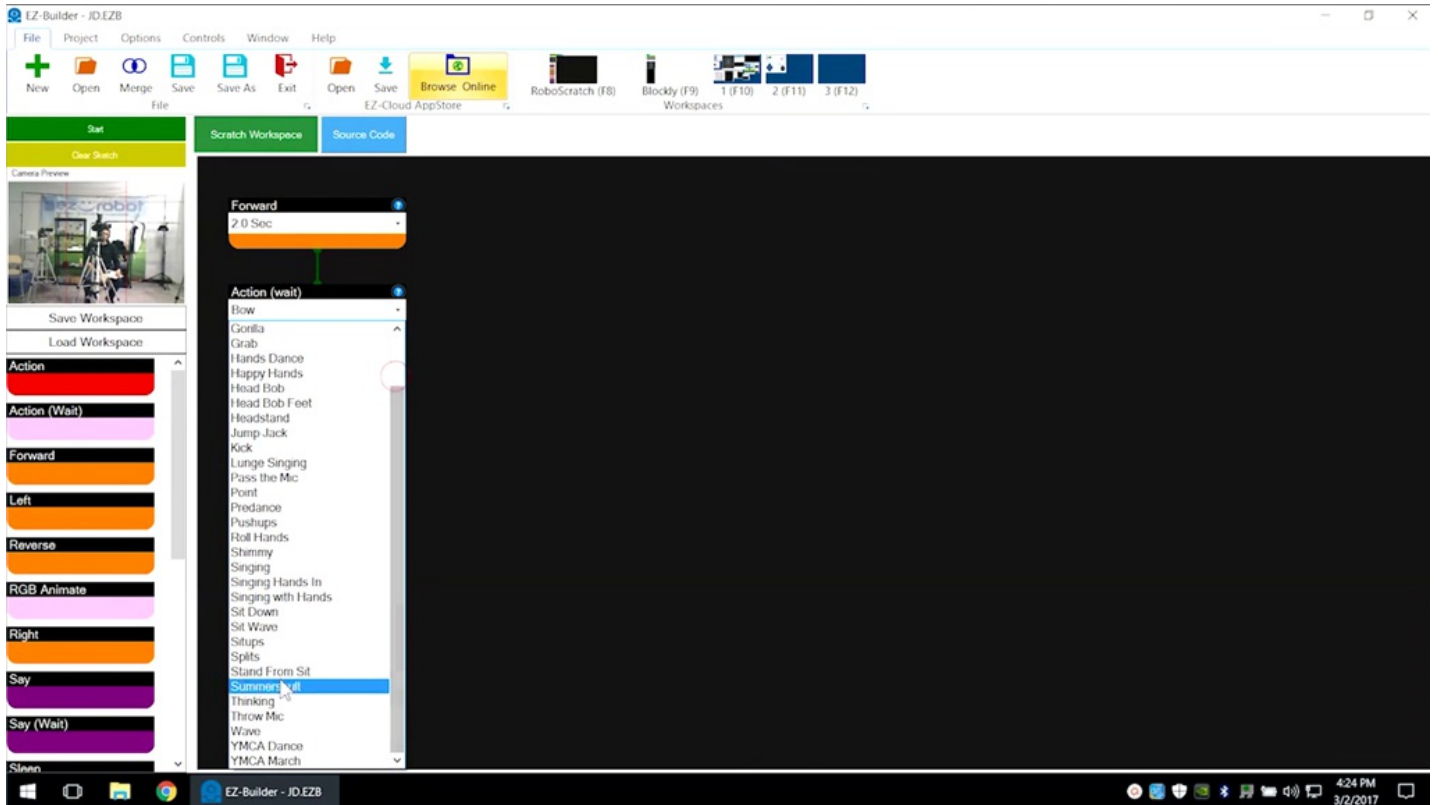
The bottom of the screen shows the Windows taskbar with the EZ-Builder application icon and the system clock showing 4:22 PM on 3/2/2017.

Select **RoboScratch** from the **Workspaces** to create a linear program that runs step-by-step from start to finish.

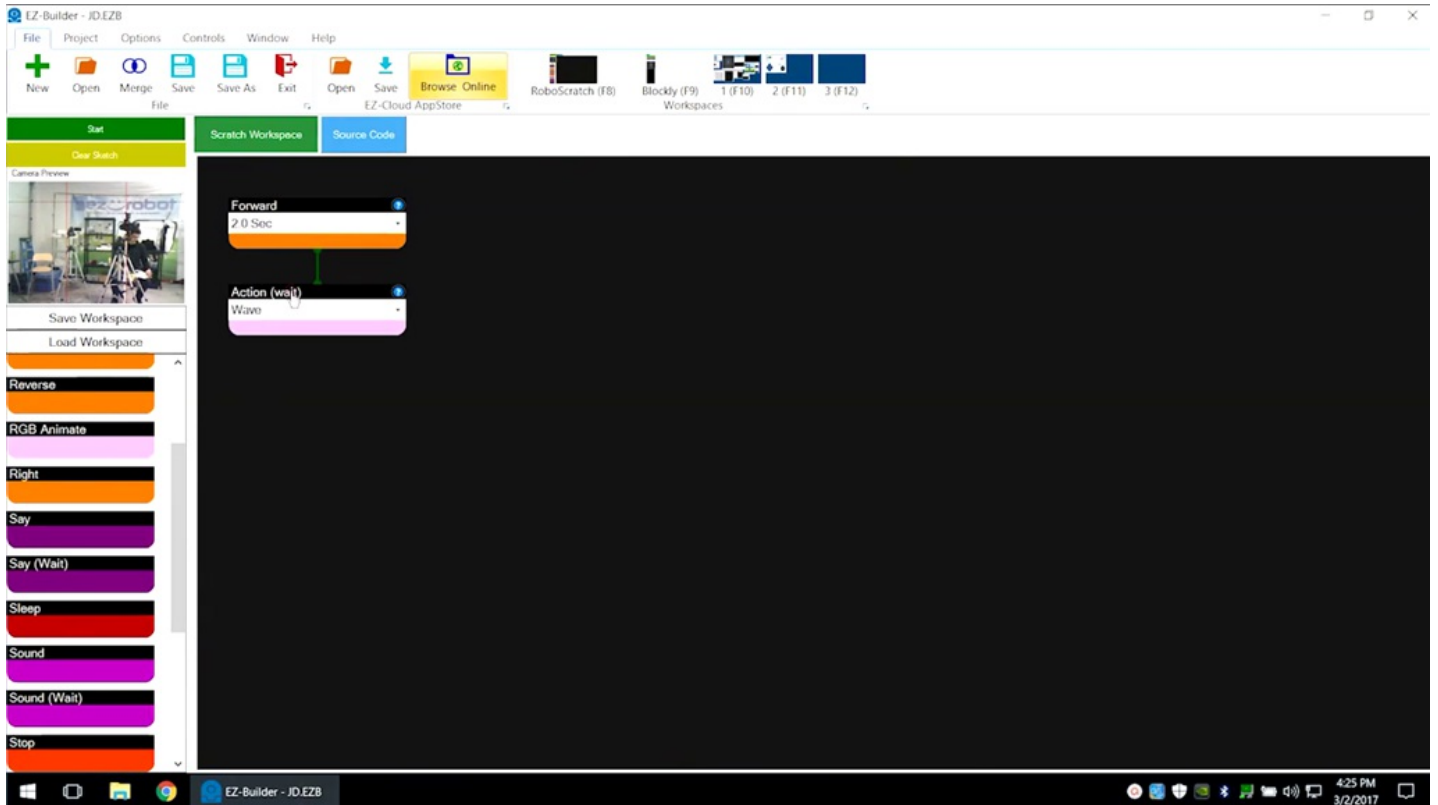


RoboScratch Commands

Click on commands, drag into position, and edit the parameters as desired.

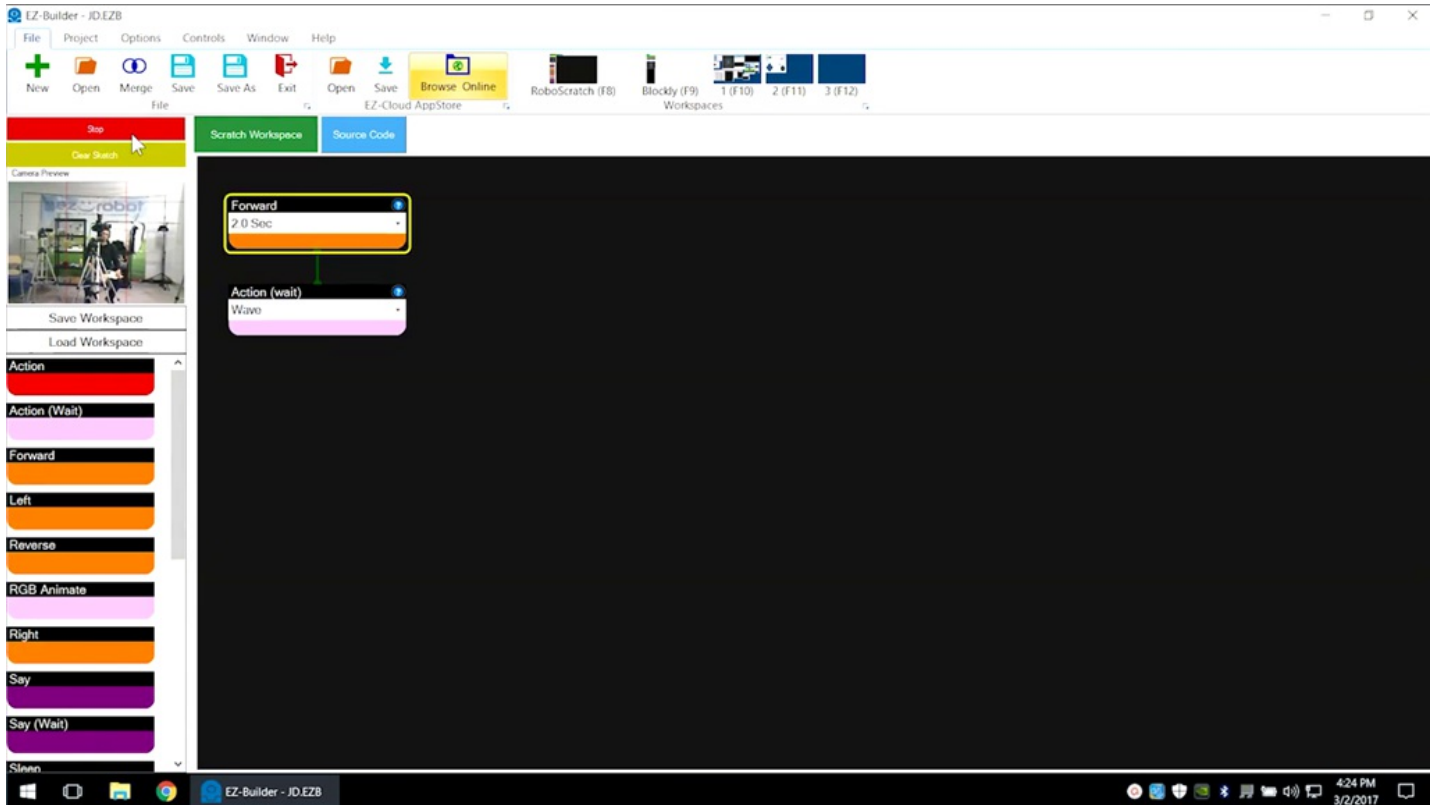


Wait will allow an **Action** to complete before moving to the next command.



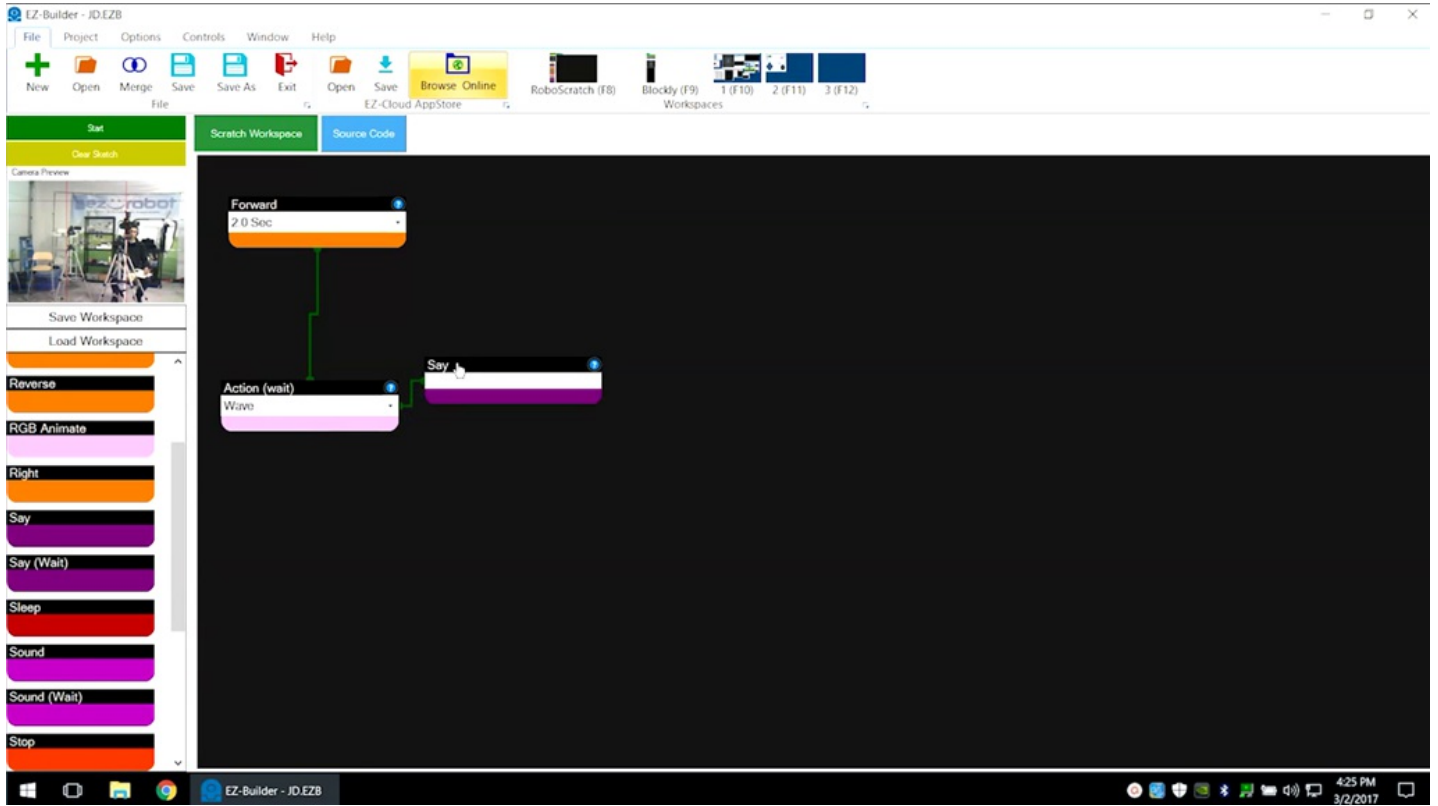
RoboScratch Program Execution

Click on **Start** to run the program. Each command is highlighted in yellow as it executes.

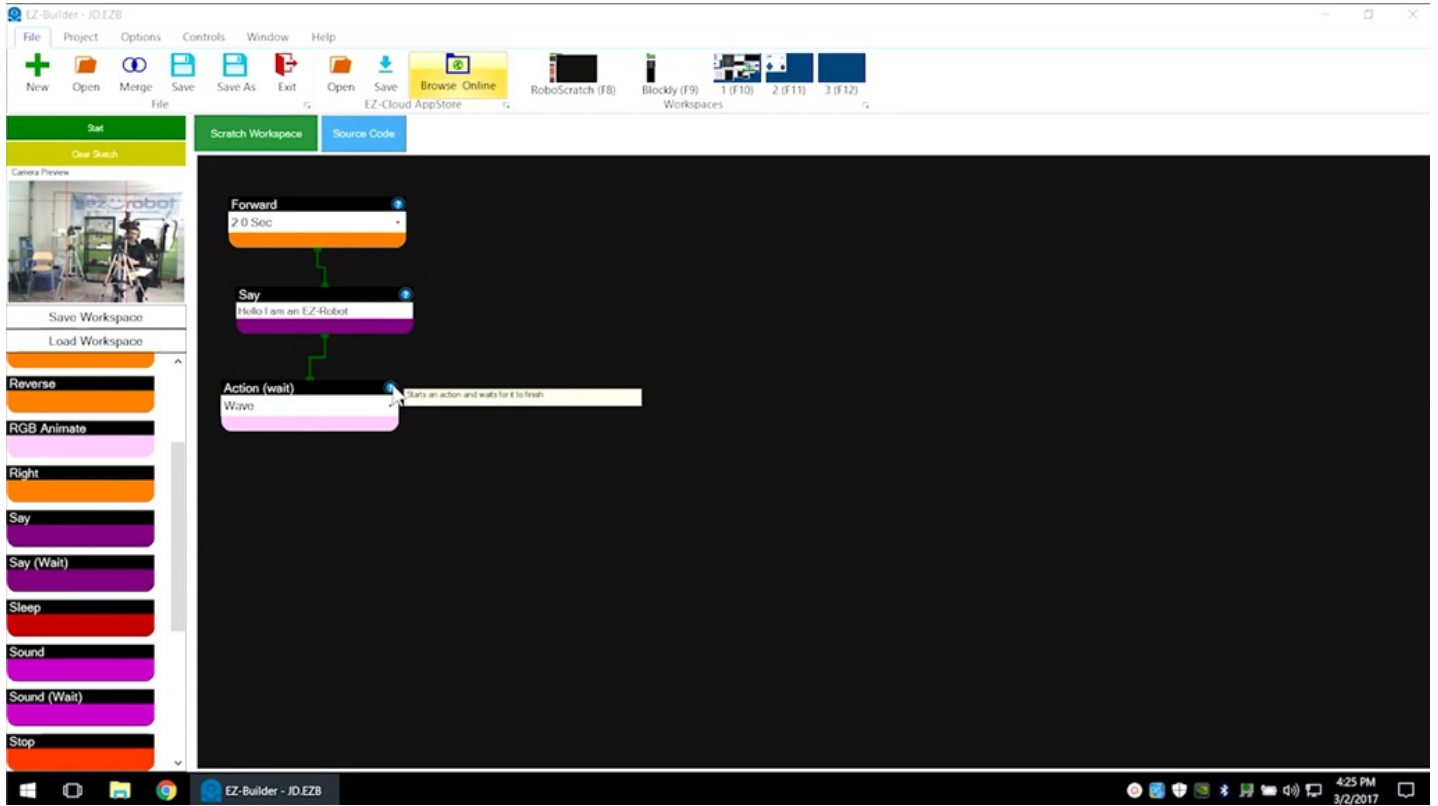


RoboScratch Program Flow

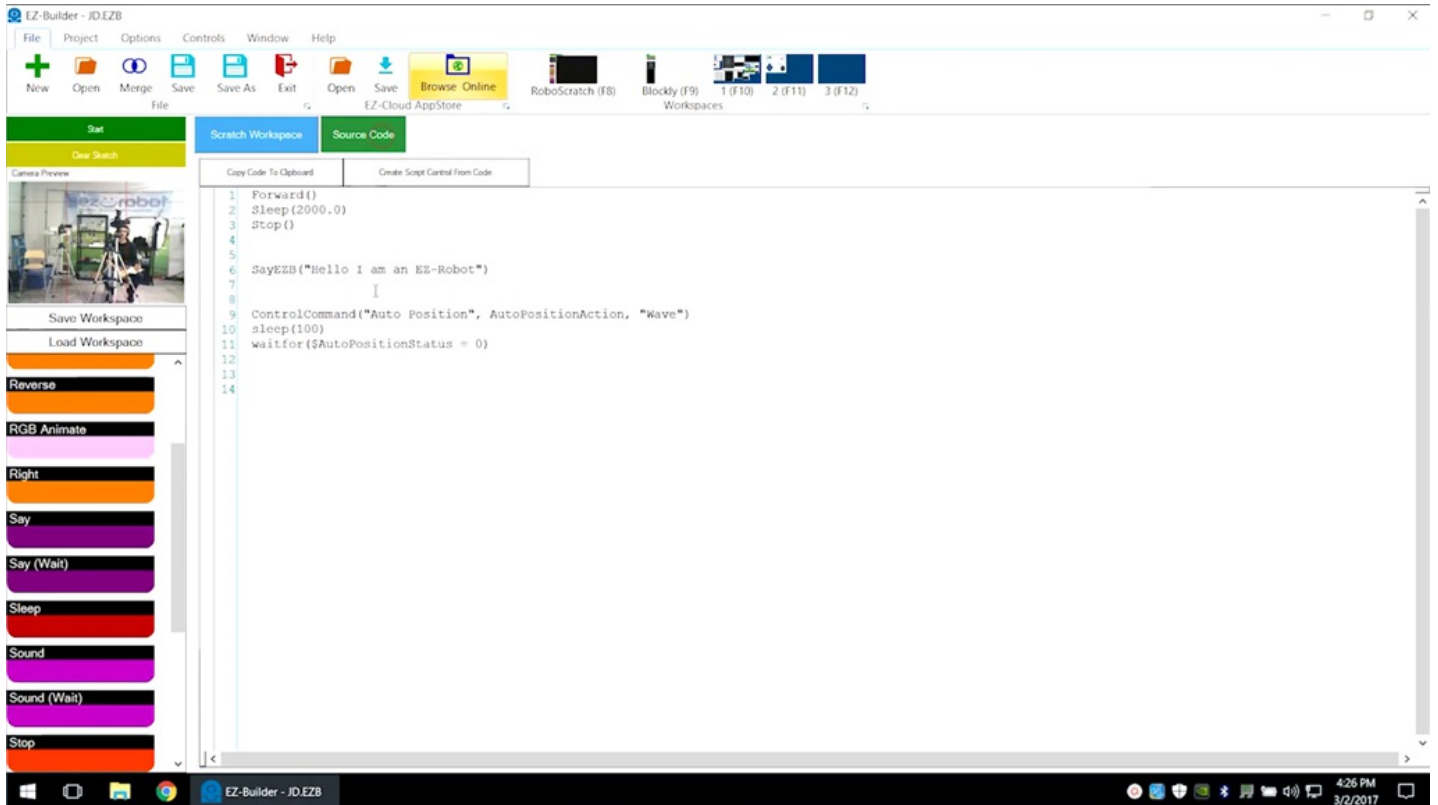
Follow the green line for program flow. Commands can be reordered by dragging into a new position.



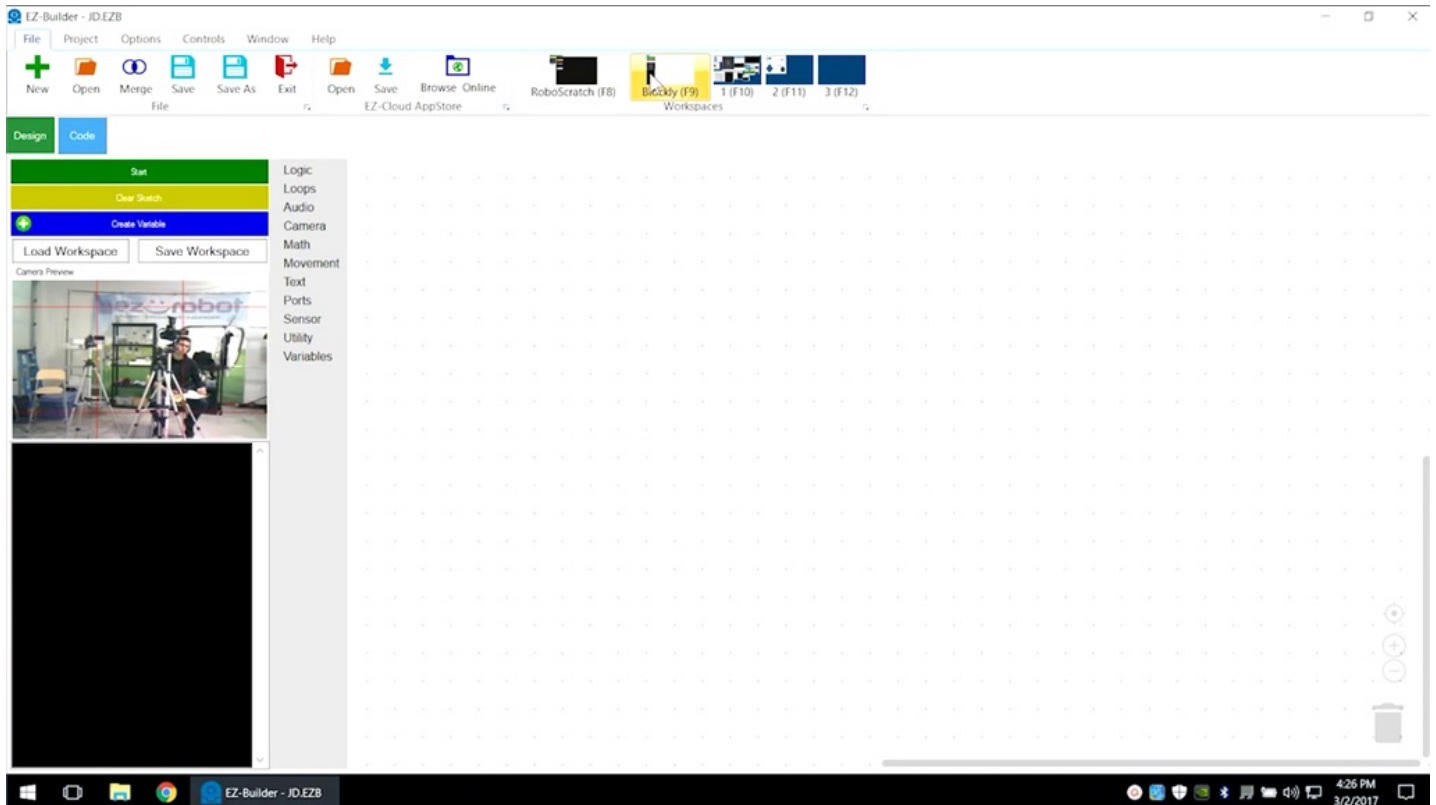
Hover over any blue question mark for more information. Click on window question marks for additional details.



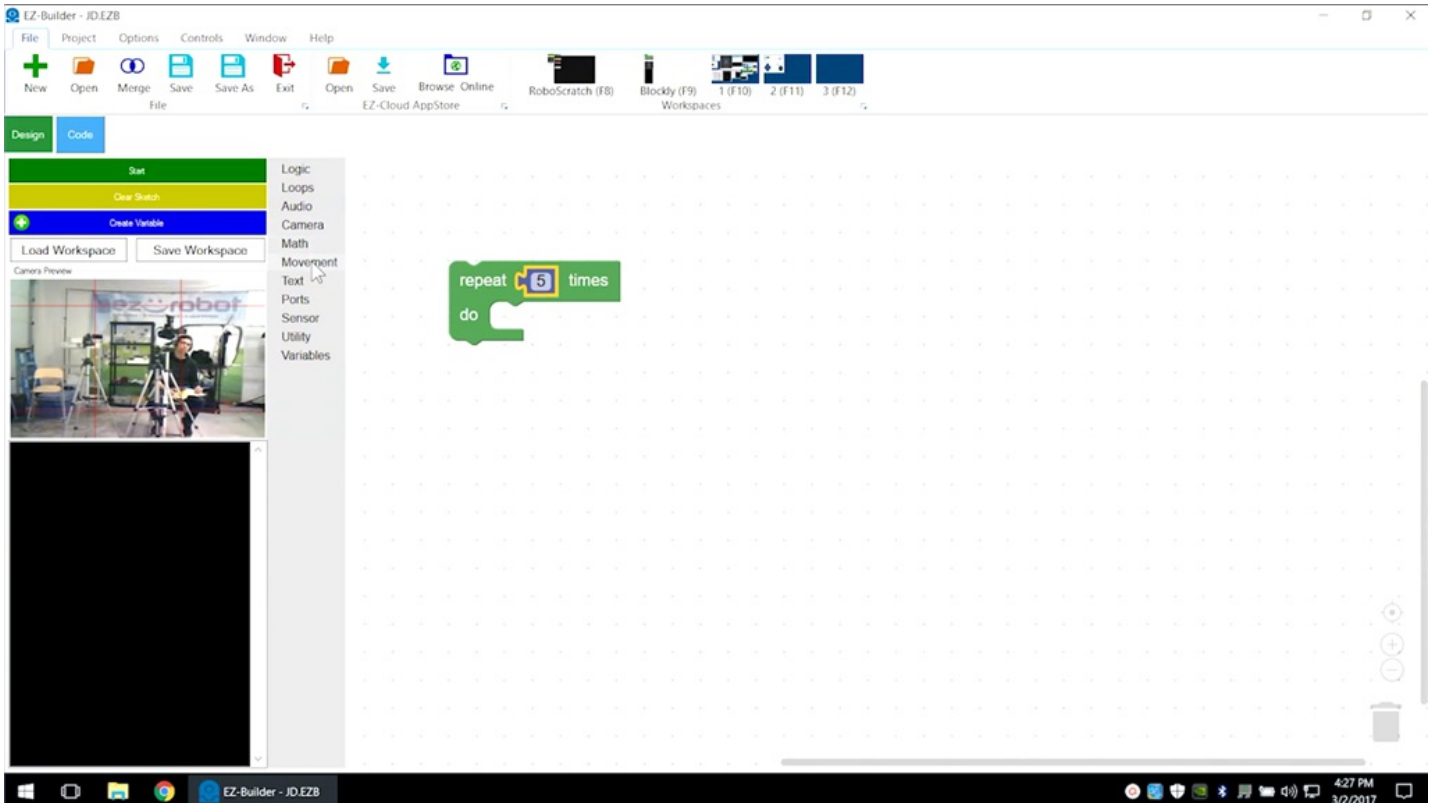
Click on the **Source Code** button to view the generated **EZ-Script** code.



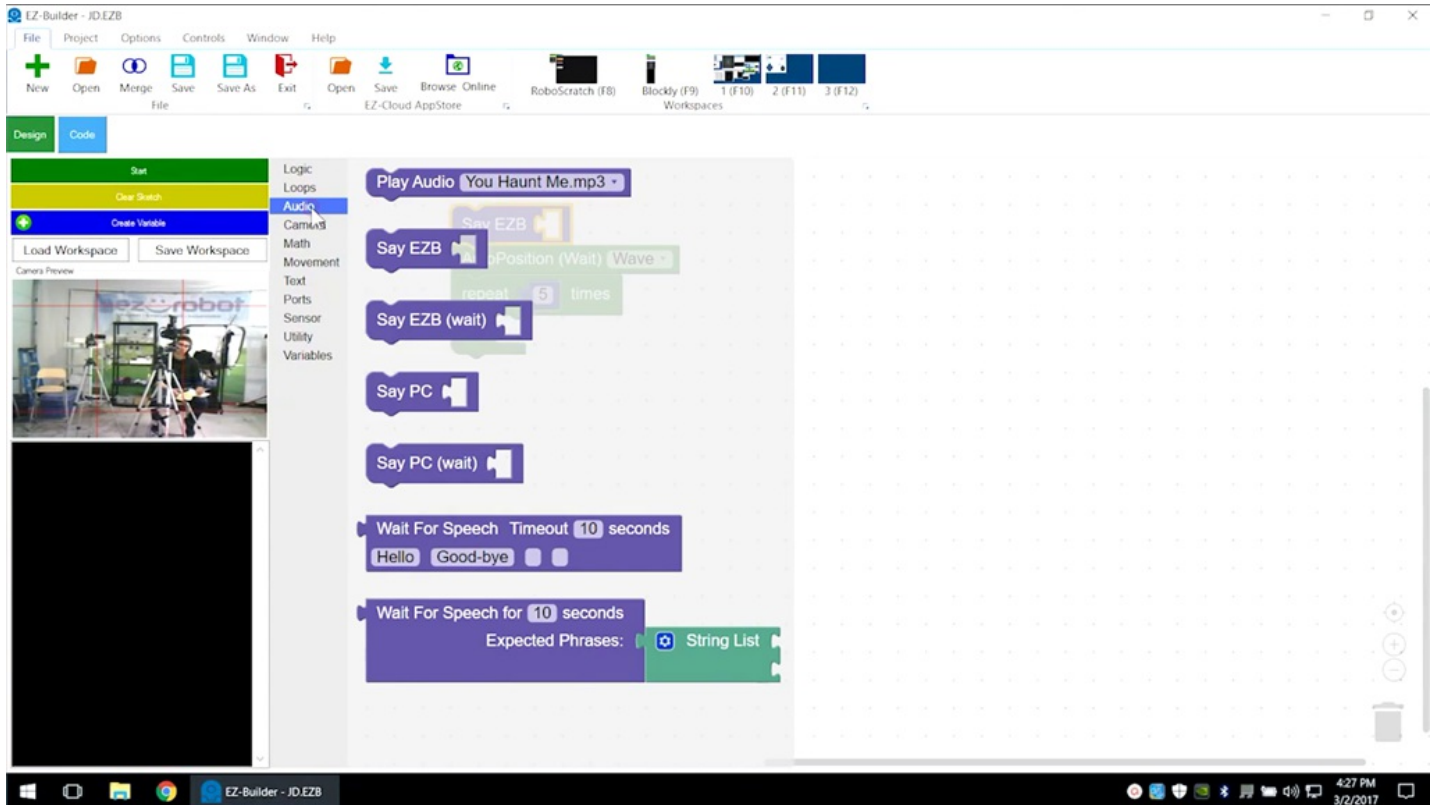
Select **Blockly** from the **Workspaces** to create a more complicated program that uses logic, loops, and branches.



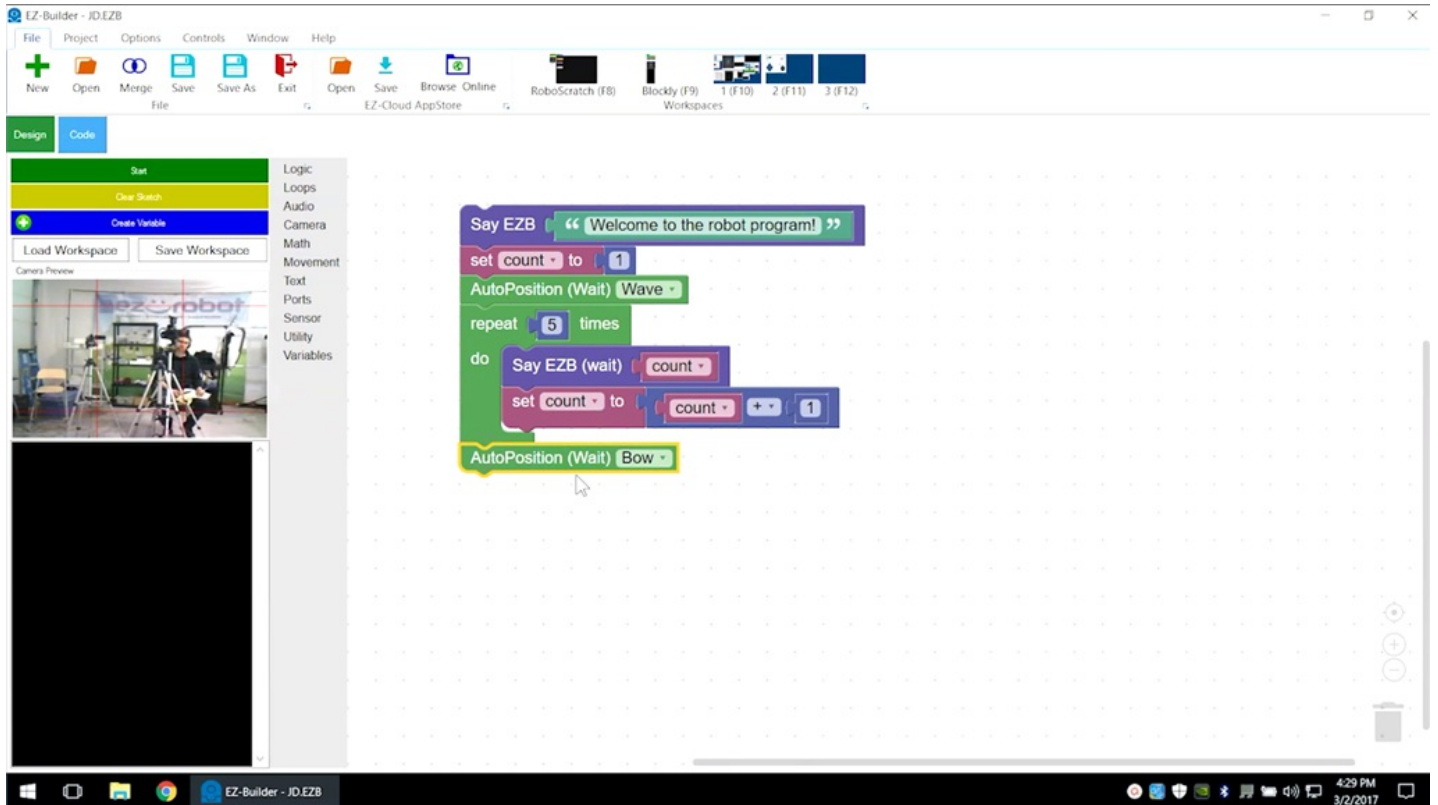
Click on commands, drag into position, and edit the parameters as desired.



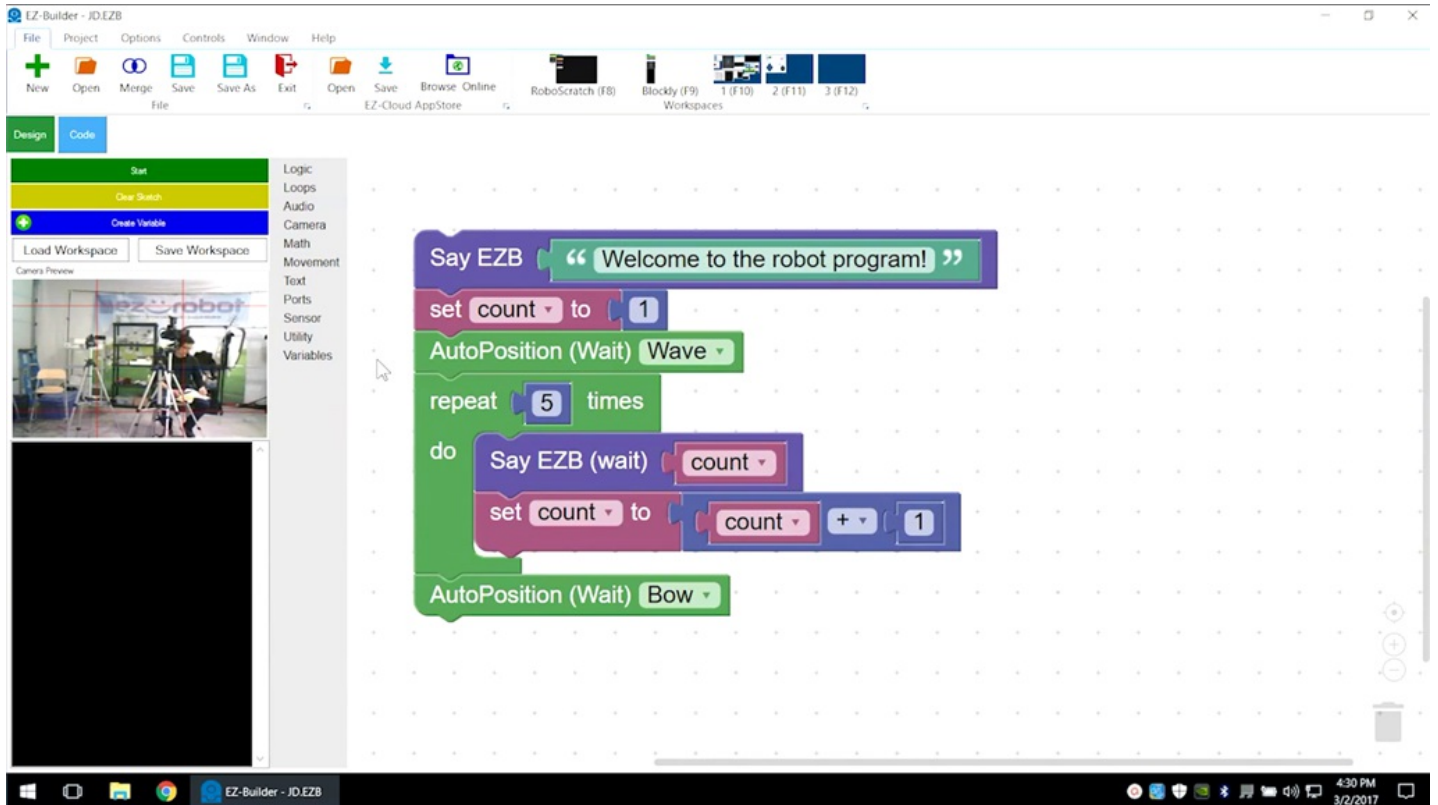
Audio can be output through either the **EZ-B** controller or the **PC** itself.



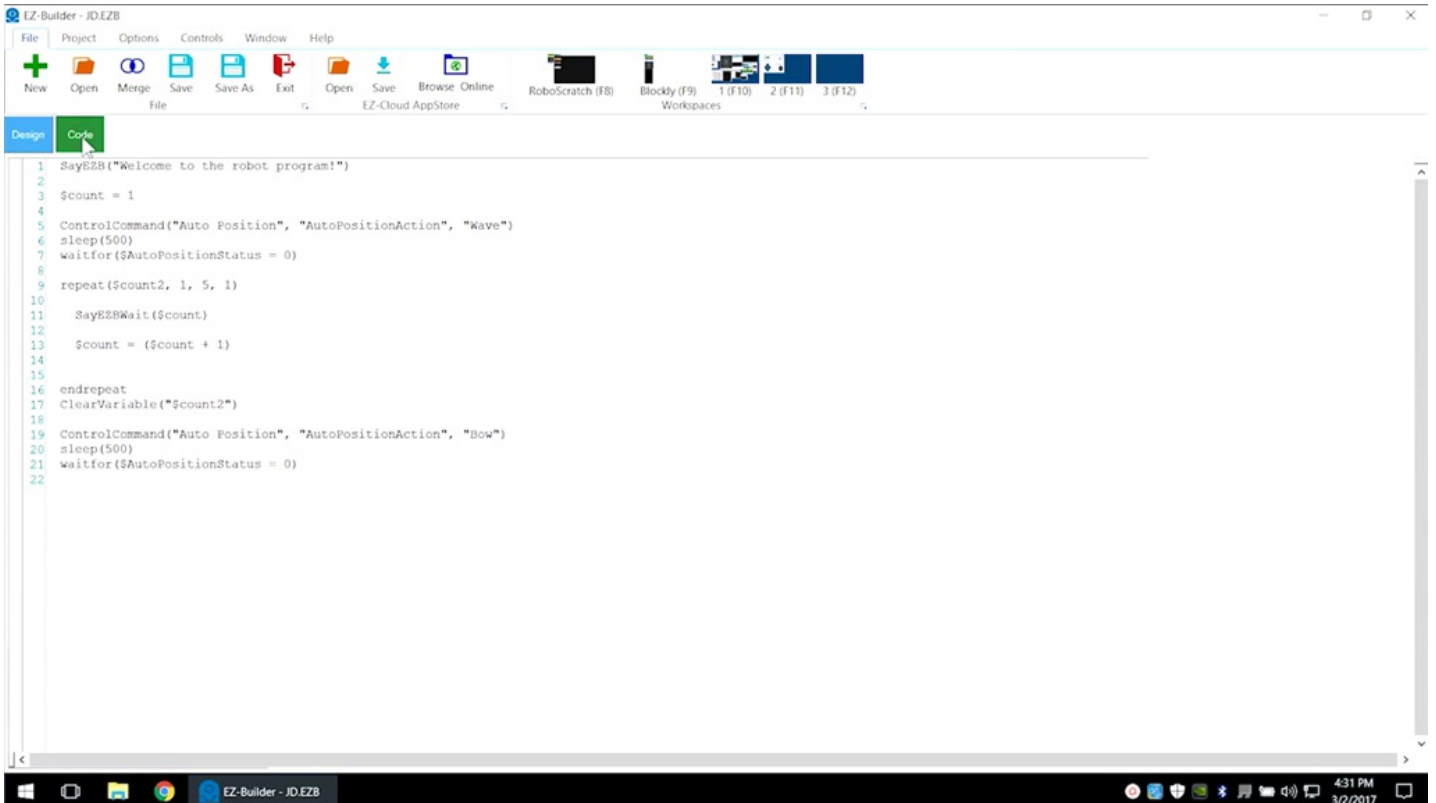
Blockly uses coding elements such as variables to count, track, and repeat.



Click on the green **Start** button to execute the program.



Click on the **Code** button to view the generated **EZ-Script** code.

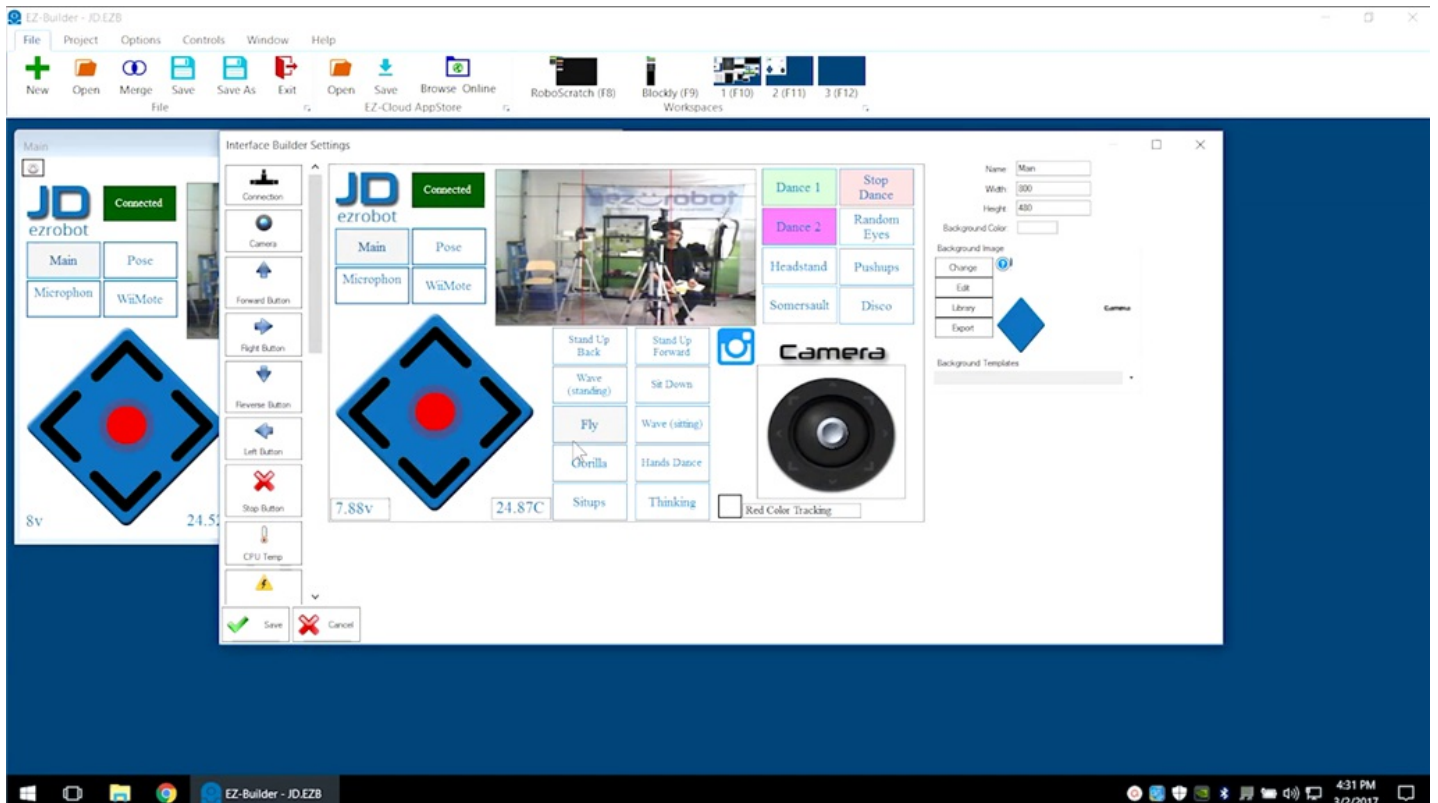
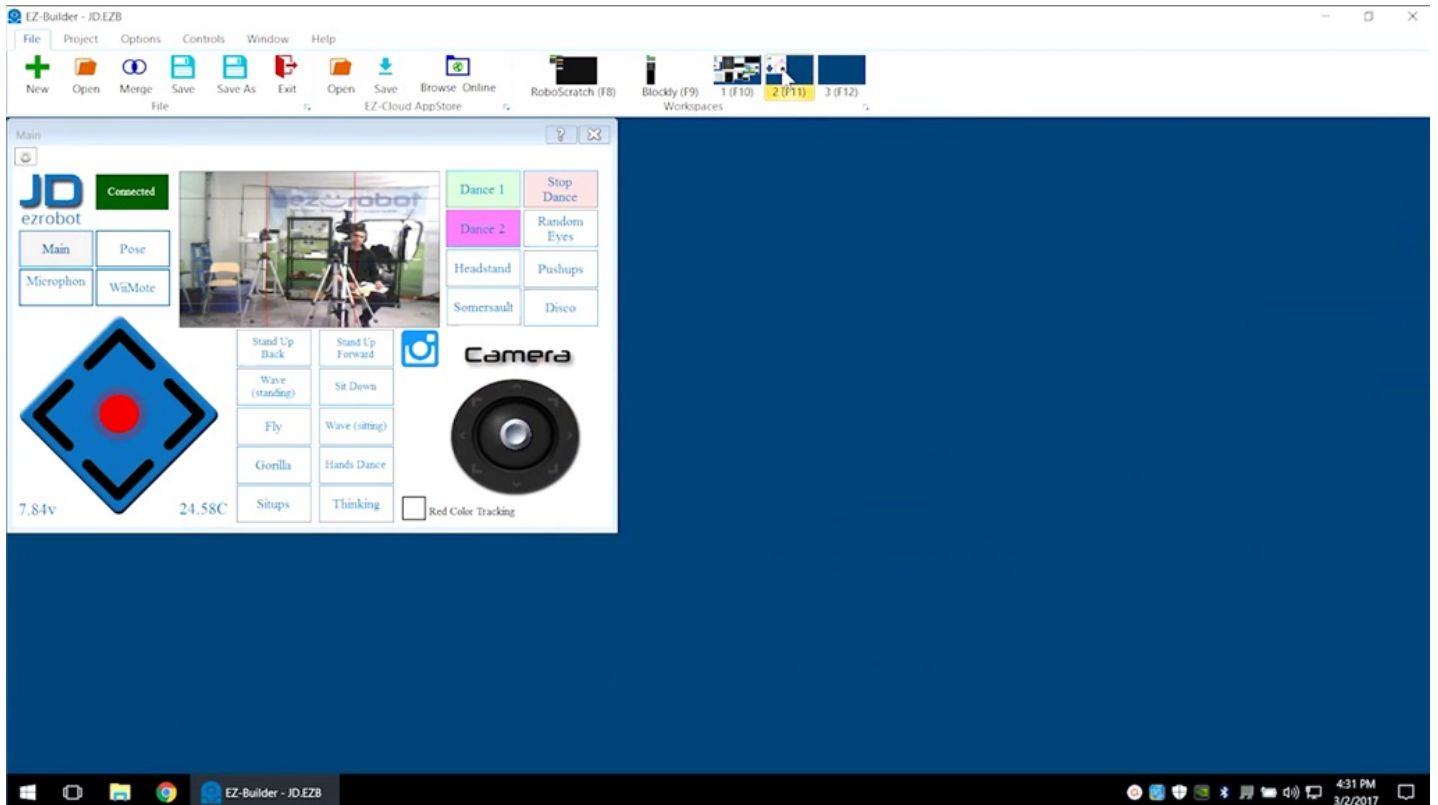


The screenshot shows the EZ-Builder software interface. The 'Code' tab is selected, displaying the following generated EZ-Script code:

```
1 SayEZB("Welcome to the robot program!")
2
3 $count = 1
4
5 ControlCommand("Auto Position", "AutoPositionAction", "Wave")
6 sleep(500)
7 waitfor($AutoPositionStatus = 0)
8
9 repeat($count2, 1, 5, 1)
10
11   SayEZBWait($count)
12
13   $count = ($count + 1)
14
15
16 endrepeat
17 ClearVariable("$count2")
18
19 ControlCommand("Auto Position", "AutoPositionAction", "Bow")
20 sleep(500)
21 waitfor($AutoPositionStatus = 0)
22
```

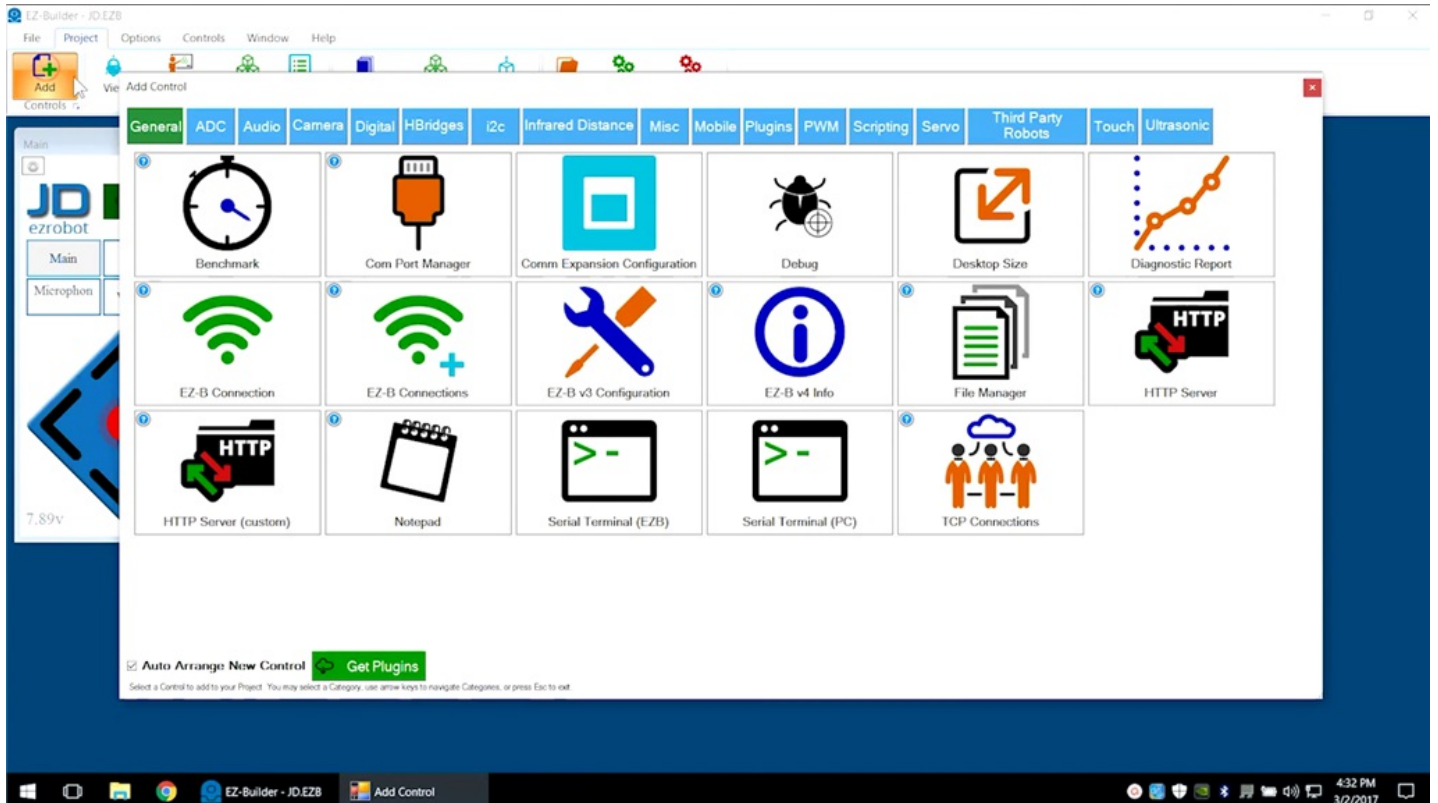
Additional Workspaces

Select **Workspace 2** or **3** for more space to add controls. **Main** control shows the interface for creating an **EZ-Robot** mobile app. Click on the gear to customize the app interface.



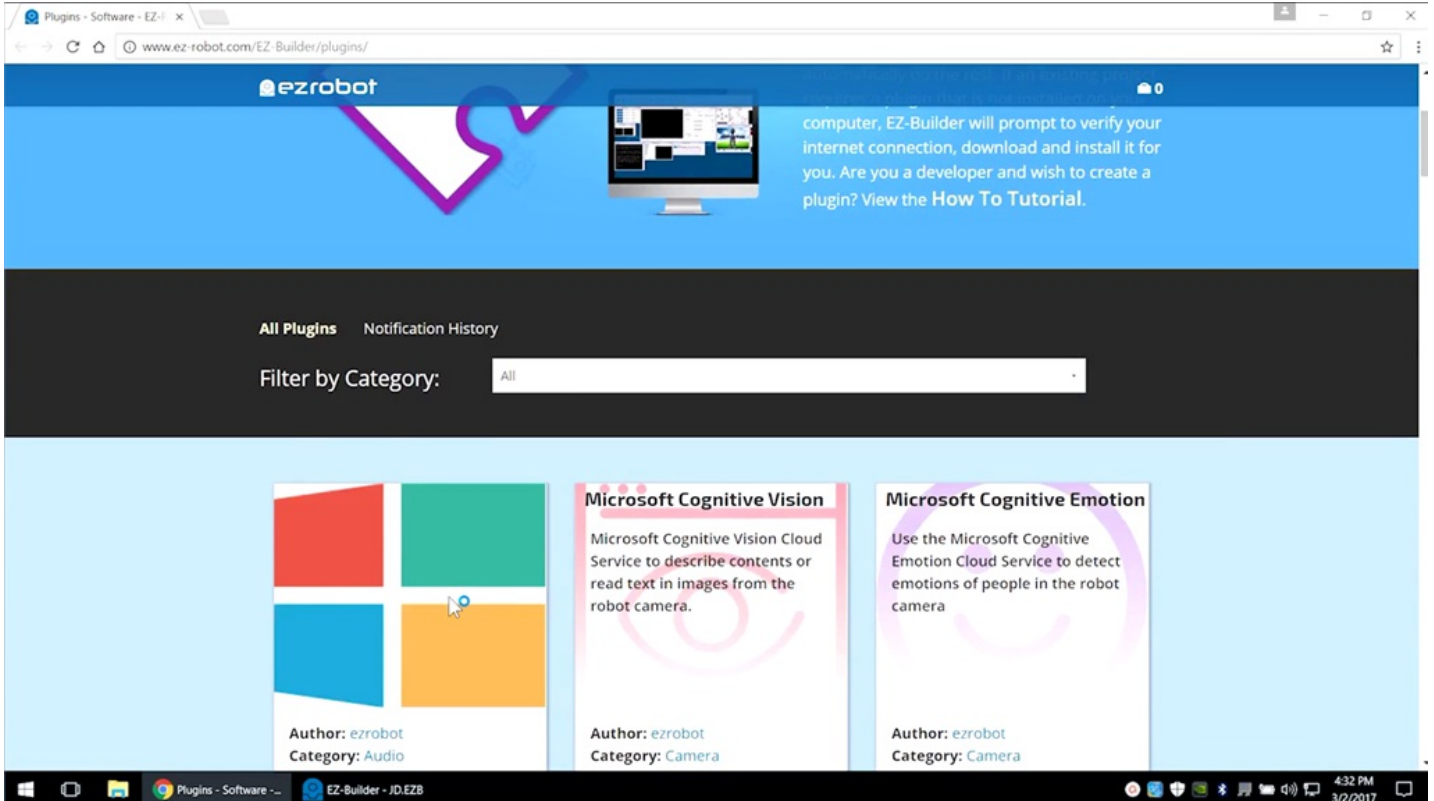
Adding Controls

Find more controls through the **Project -> Add** menu.



Third Party Plugins

Third party plugins can also be downloaded. These are added under **Plugins**.

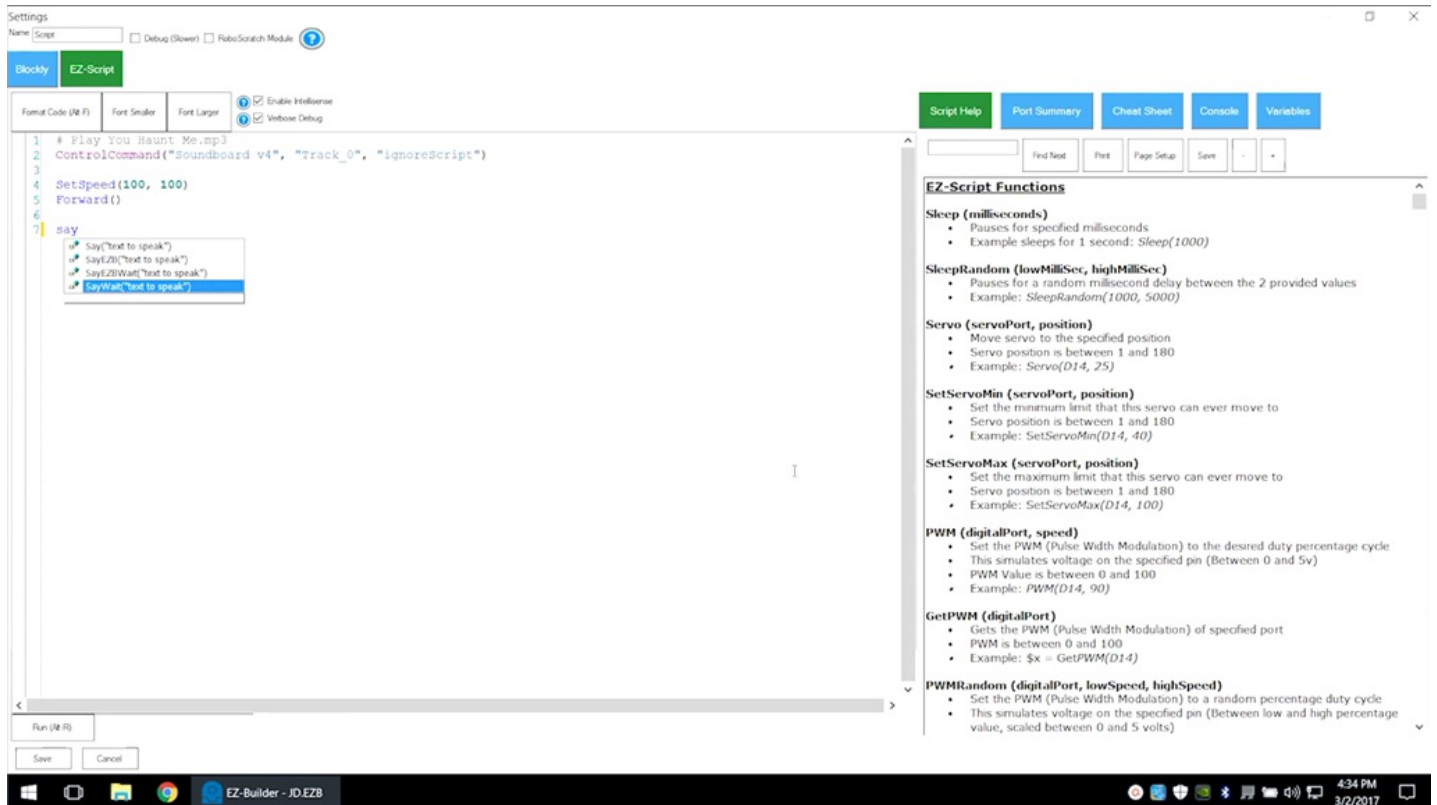


The screenshot shows a web browser window displaying the EZ-robot Plugins page. The browser's address bar shows the URL `www.ez-robot.com/EZ-Builder/plugins/`. The page header features the EZ-robot logo and a navigation menu with "All Plugins" and "Notification History". Below the header, there is a "Filter by Category:" dropdown menu currently set to "All". The main content area displays three plugin cards:

- Microsoft Cognitive Vision**: A card with a red, green, blue, and orange grid. Description: "Microsoft Cognitive Vision Cloud Service to describe contents or read text in images from the robot camera." Author: ezrobot, Category: Camera.
- Microsoft Cognitive Emotion**: A card with a purple and pink circular graphic. Description: "Use the Microsoft Cognitive Emotion Cloud Service to detect emotions of people in the robot camera." Author: ezrobot, Category: Camera.
- Audio**: A card with a red, green, blue, and orange grid. Description: "Audio". Author: ezrobot, Category: Audio.

The Windows taskbar at the bottom shows the time as 4:32 PM on 3/2/2017. The taskbar includes icons for the Start menu, File Explorer, Google Chrome (with a tab for "Plugins - Software - ..."), and EZ-Builder - JD.EZB.

Scripting can be used to create custom controls with the **Blockly** editor. **IntelliSense** will automatically show the available syntax options.



The screenshot displays the EZ-Script editor interface. The main code editor contains the following script:

```
1 # Play You Haunt Me.mp3
2 ControlCommand("Soundboard v4", "Track_0", "ignoreScript")
3
4 SetSpeed(100, 100)
5 Forward()
6
7 say
```

An IntelliSense dropdown menu is visible below the `say` command, listing the following options:

- say("text to speak")
- sayEZ("text to speak")
- sayEZWait("text to speak")
- sayWait("text to speak")

The right-hand side of the interface features a **EZ-Script Functions** help panel with the following content:

EZ-Script Functions

- Sleep (milliseconds)**
 - Pauses for specified milliseconds
 - Example sleeps for 1 second: `Sleep(1000)`
- SleepRandom (lowMilliSec, highMilliSec)**
 - Pauses for a random millisecond delay between the 2 provided values
 - Example: `SleepRandom(1000, 5000)`
- Servo (servoPort, position)**
 - Move servo to the specified position
 - Servo position is between 1 and 180
 - Example: `Servo(D14, 25)`
- SetServoMin (servoPort, position)**
 - Set the minimum limit that this servo can ever move to
 - Servo position is between 1 and 180
 - Example: `SetServoMin(D14, 40)`
- SetServoMax (servoPort, position)**
 - Set the maximum limit that this servo can ever move to
 - Servo position is between 1 and 180
 - Example: `SetServoMax(D14, 100)`
- PWM (digitalPort, speed)**
 - Set the PWM (Pulse Width Modulation) to the desired duty percentage cycle
 - This simulates voltage on the specified pin (Between 0 and 5v)
 - PWM Value is between 0 and 100
 - Example: `PWM(D14, 90)`
- GetPWM (digitalPort)**
 - Gets the PWM (Pulse Width Modulation) of specified port
 - PWM is between 0 and 100
 - Example: `$x = GetPWM(D14)`
- PWMRandom (digitalPort, lowSpeed, highSpeed)**
 - Set the PWM (Pulse Width Modulation) to a random percentage duty cycle
 - This simulates voltage on the specified pin (Between low and high percentage value, scaled between 0 and 5 volts)

The interface also includes a top settings bar with options for `Debug (Dlower)` and `Auto-Scratch Module`, and a right-hand navigation bar with buttons for `Script Help`, `Port Summary`, `Cheat Sheet`, `Console`, and `Variables`. The bottom status bar shows the application name `EZ-Builder - JD.EZB` and the system time `4:34 PM 3/2/2017`.

Scripting Control Options

See all of the control options by right-clicking in the editor, or by selecting the **Cheat Sheet**.

The screenshot shows the EZ-Script editor interface. On the left, a script is being edited with the following code:

```
1 # Play You Haunt Me.mp3
2 ControlCommand("Soundboard v4", "Track_0", "ignoreScript")
3
4 SetSpeed(100, 100)
5 Forward()
6
7 SayEZBWait("I am an EZ-Robot!")
8
9
```

A context menu is open over the script, listing various control options. The 'Auto Position' option is selected, which has opened a list of all available `ControlCommand` functions. The list includes:

- `ControlCommand("Auto Position", AutoPositionAction, "Bow")`
- `ControlCommand("Auto Position", AutoPositionAction, "Disco Dance")`
- `ControlCommand("Auto Position", AutoPositionAction, "Fly")`
- `ControlCommand("Auto Position", AutoPositionAction, "Forward")`
- `ControlCommand("Auto Position", AutoPositionAction, "Getup")`
- `ControlCommand("Auto Position", AutoPositionAction, "Gorilla")`
- `ControlCommand("Auto Position", AutoPositionAction, "Grab")`
- `ControlCommand("Auto Position", AutoPositionAction, "Hands Dance")`
- `ControlCommand("Auto Position", AutoPositionAction, "Happy Hands")`
- `ControlCommand("Auto Position", AutoPositionAction, "Head Bob")`
- `ControlCommand("Auto Position", AutoPositionAction, "Head Bob Feet")`**
- `ControlCommand("Auto Position", AutoPositionAction, "Headstand")`
- `ControlCommand("Auto Position", AutoPositionAction, "Jump Jack")`
- `ControlCommand("Auto Position", AutoPositionAction, "Kick")`
- `ControlCommand("Auto Position", AutoPositionAction, "Lunge Singing")`
- `ControlCommand("Auto Position", AutoPositionAction, "Pass the Mic")`
- `ControlCommand("Auto Position", AutoPositionAction, "Pushups")`
- `ControlCommand("Auto Position", AutoPositionAction, "Point")`
- `ControlCommand("Auto Position", AutoPositionAction, "Predance")`
- `ControlCommand("Auto Position", AutoPositionAction, "Reverse")`
- `ControlCommand("Auto Position", AutoPositionAction, "Right")`
- `ControlCommand("Auto Position", AutoPositionAction, "Roll Hands")`
- `ControlCommand("Auto Position", AutoPositionAction, "Shimmy")`
- `ControlCommand("Auto Position", AutoPositionAction, "Singing")`
- `ControlCommand("Auto Position", AutoPositionAction, "Singing Hands In")`
- `ControlCommand("Auto Position", AutoPositionAction, "Singing with Hands")`
- `ControlCommand("Auto Position", AutoPositionAction, "Sit Down")`
- `ControlCommand("Auto Position", AutoPositionAction, "Sit Wave")`
- `ControlCommand("Auto Position", AutoPositionAction, "Situptions")`
- `ControlCommand("Auto Position", AutoPositionAction, "Splits")`
- `ControlCommand("Auto Position", AutoPositionAction, "Stand From Sit")`
- `ControlCommand("Auto Position", AutoPositionAction, "Stop")`
- `ControlCommand("Auto Position", AutoPositionAction, "Summersault")`

On the right, a 'Script Help' panel is open, displaying a 'Cheat Sheet' of various functions and their usage examples, including `GetDigital`, `ASin`, `ACos`, `Sqrt`, `Map`, `Power`, `Sin`, `Cos`, and `Abs`.

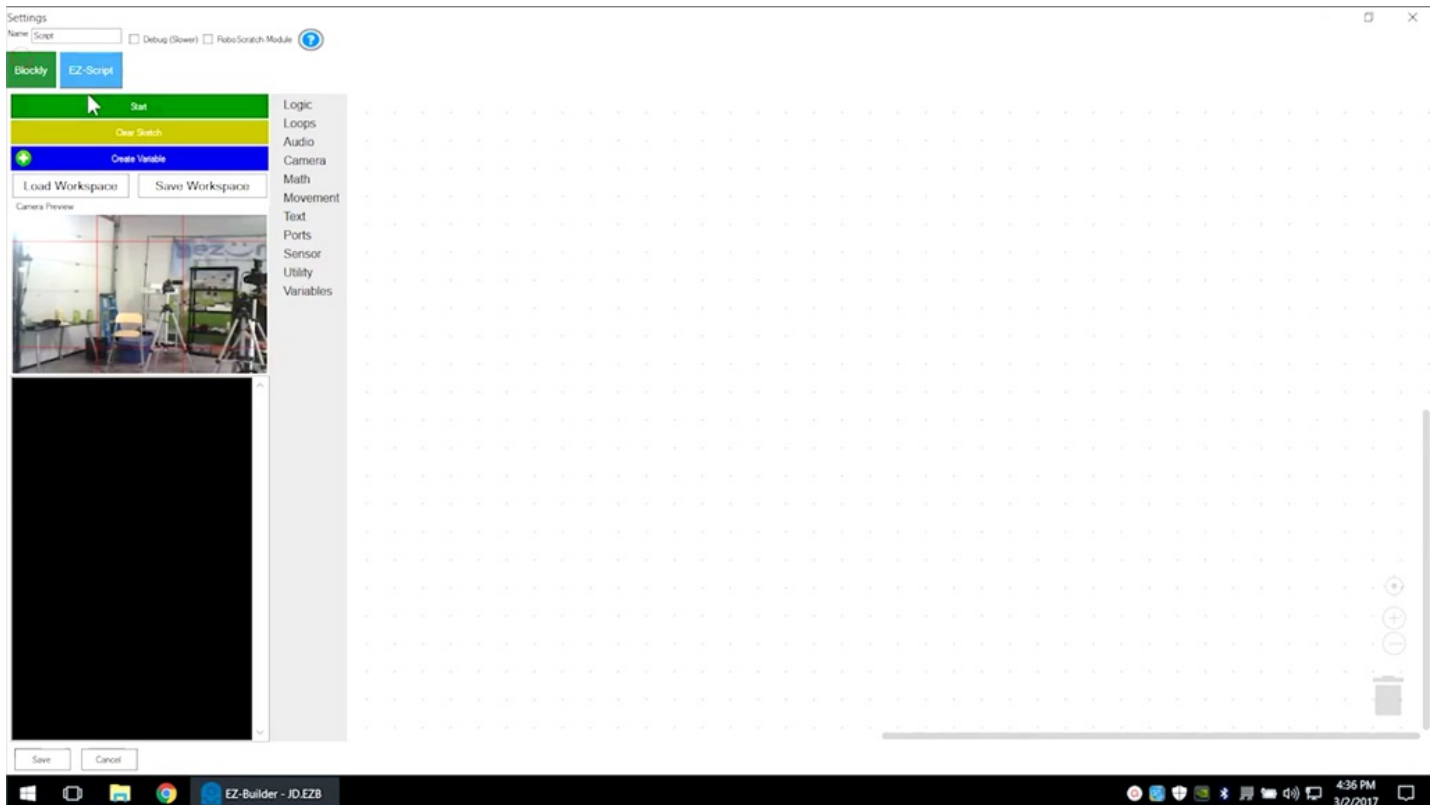
The screenshot shows the EZ-Script editor with a different script:

```
1 # Play You Haunt Me.mp3
2 ControlCommand("Soundboard v4", "Track_0", "ignoreScript")
3
4 SetSpeed(100, 100)
5 Forward()
6
7 SayEZBWait("I am an EZ-Robot!")
8
9 ControlCommand("Camera", CameraFaceTrackingEnable)
10
11
```

The 'Cheat Sheet' panel on the right is open, providing a comprehensive list of available `ControlCommand` functions. It is organized into two main sections:

- Show Controls**: A list of controls that can be displayed, including `ShowDesktop` (1, 2, 3), `ShowControl` for various components like `RoboScratch`, `Auto Position`, `Camera`, `Connection`, `EZ-Robot`, `Init`, `Main`, `Microphone`, `NotePad`, `PointAndTrack`, `RGB Animator`, `Script`, `Soundboard v4`, and `Wii Remote`.
- Auto Position**: A list of `ControlCommand` functions for the 'Auto Position' action, including `Bow`, `Disco Dance`, `Fly`, `Forward`, `Getup`, `Gorilla`, `Grab`, `Hands Dance`, and `Happy Hands`.

Editing in **EZ-Script** will clear the **Blockly** editor.



Connecting Custom Components

Choose which ports will be used for connecting custom components.

The screenshot displays the EZ-Robot Design software interface. At the top, there are view controls for 'Views' (Front, Top, Left, Right, Bottom, Center) and 'Camera' (Up, Down, Left, Right, Zoom In, Zoom Out). Below these are two 'Selected Model' sections with 'X+45', 'Y+45', and 'Z+45' settings and 'Delete' and 'Reset' buttons. The main workspace shows a 3D model of a humanoid robot head with a camera, set against a background of a green field and blue sky. A 'Select Port' dialog box is open in the center, showing a list of ports (D0-D11) and a list of components to be connected. The 'EZ-Robot' tab is selected in the dialog, and 'V0' is chosen as the port. To the right, a 'Female Humanoid Head with Camera' component is shown with its properties: Horizontal Neck (D0), Vertical Neck (NA), RGB Eyes (NA), and Camera (NA). The Windows taskbar at the bottom shows the application 'EZ-Robot Design' and the system clock indicating 4:38 PM on 3/2/2017.

Views: Front, Top, Left, Right, Bottom, Center, Zoom Out, Zoom In

Camera: Up, Down, Left, Right, Zoom In, Zoom Out

Selected Model: X+45, Y+45, Z+45, Delete, Reset

Selected Model: X+45, Y+45, Z+45, Delete, Reset

Female Humanoid Head with Camera
Description:
Girls! I hey you have a girl playing with the robots but non of the Revolution bots are geared towards girls. So i got to thinking, why not make a female version of JD. Meet JD's lil'Sis KD!
Author: EZ-B Connectiv

Horizontal Neck: D0 (Servo)
Vertical Neck: NA (Servo)
RGB Eyes: NA (i2c)
Camera: NA (Camera)

Select Port

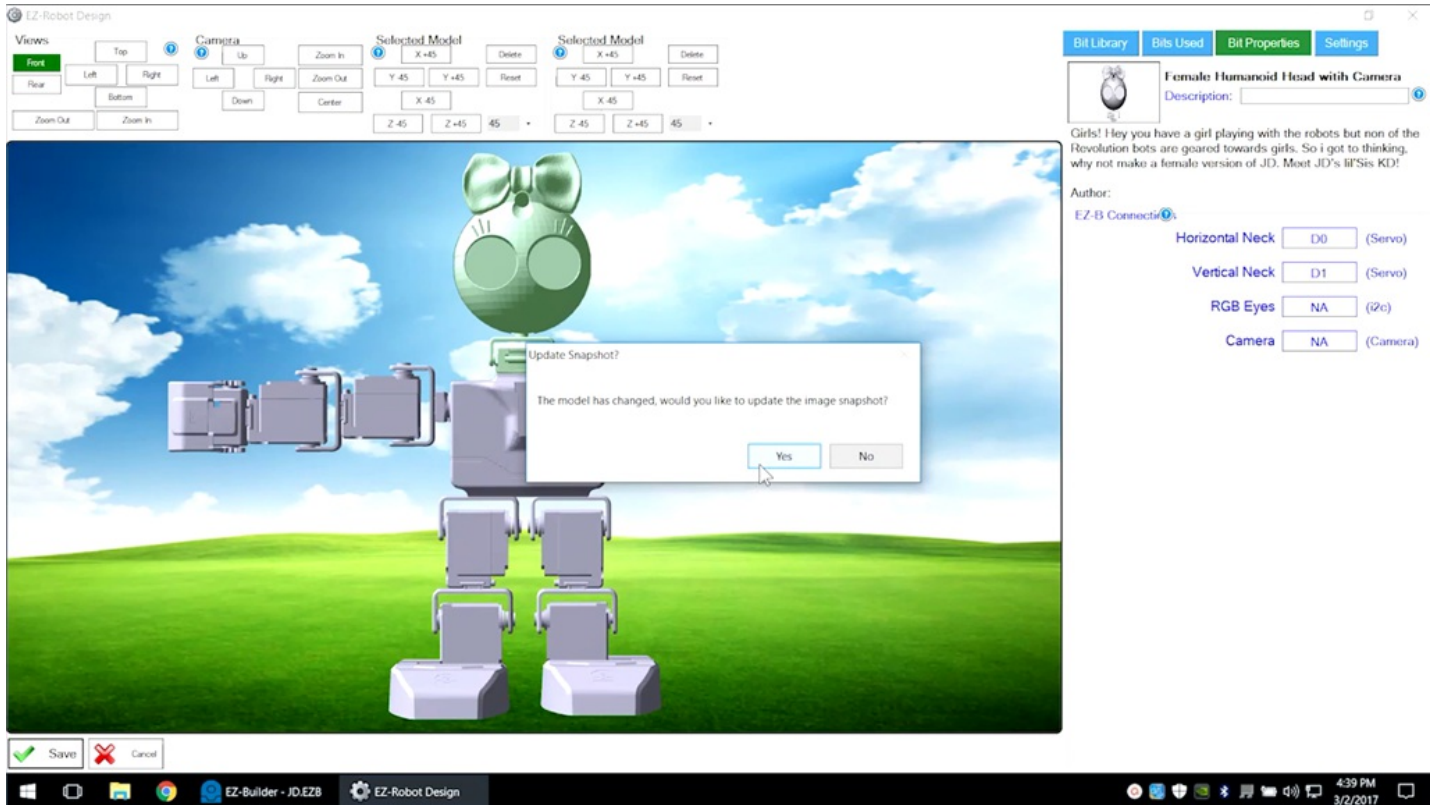
Summary EZ-Robot Image

Port: V0

Component List:
Chest and Shoulder Servos Left Shoulder: D3
Chest and Shoulder Servos Right Shoulder: D2
Head RGB Eyes:
Head Camera:
Head Horizontal: D0
Head Vertical: D1
Left Upper Arm : D4
Left Forearm Servo: D5
Left Gripper Servo: D6
Right Upper Arm : D7
Right forearm Servo: D8
Right Gripper Servo: D9
Left Upper Leg Servo: D12

Updating Build Instructions

Select **Save**. Updating the robot image will change the build instructions.



Reorder Build Steps

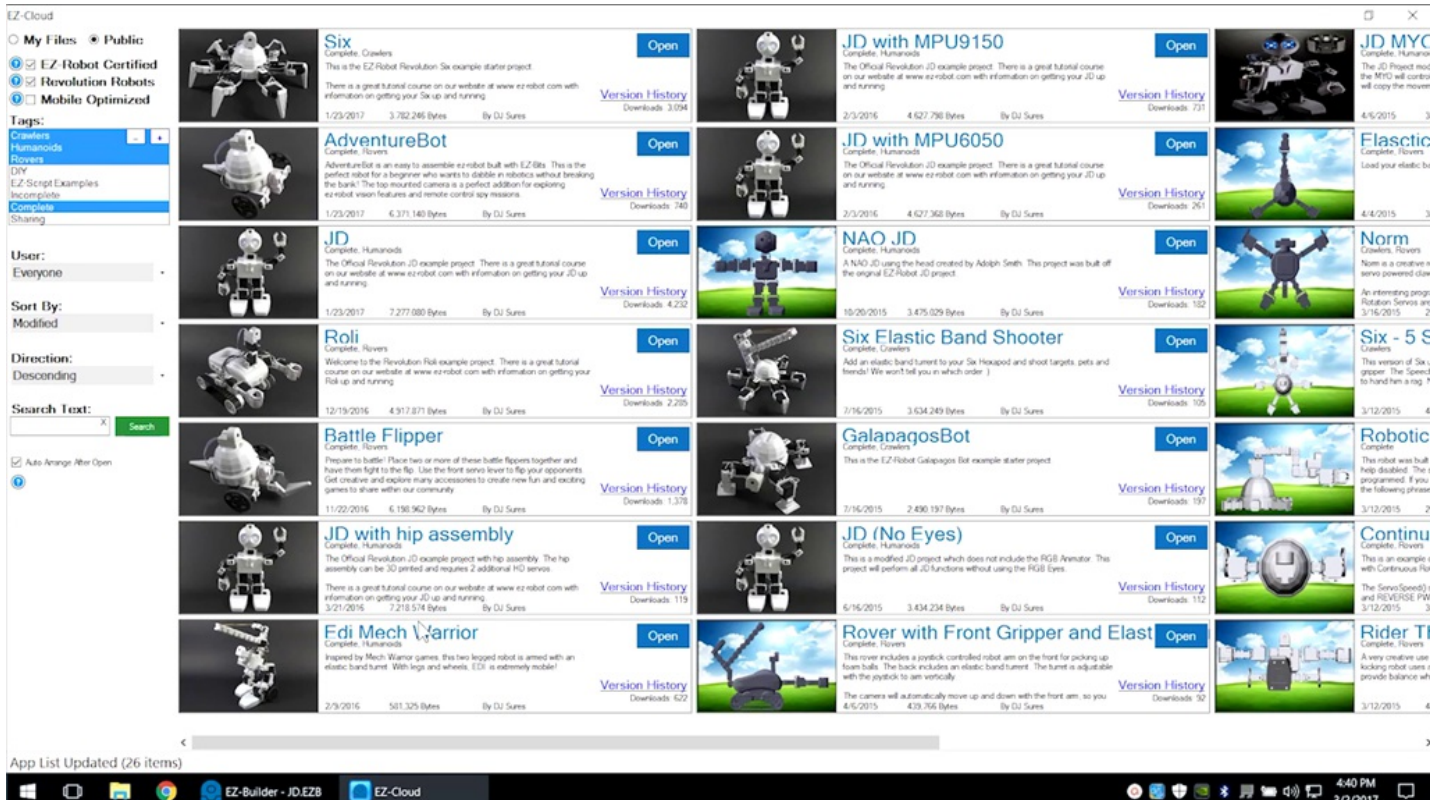
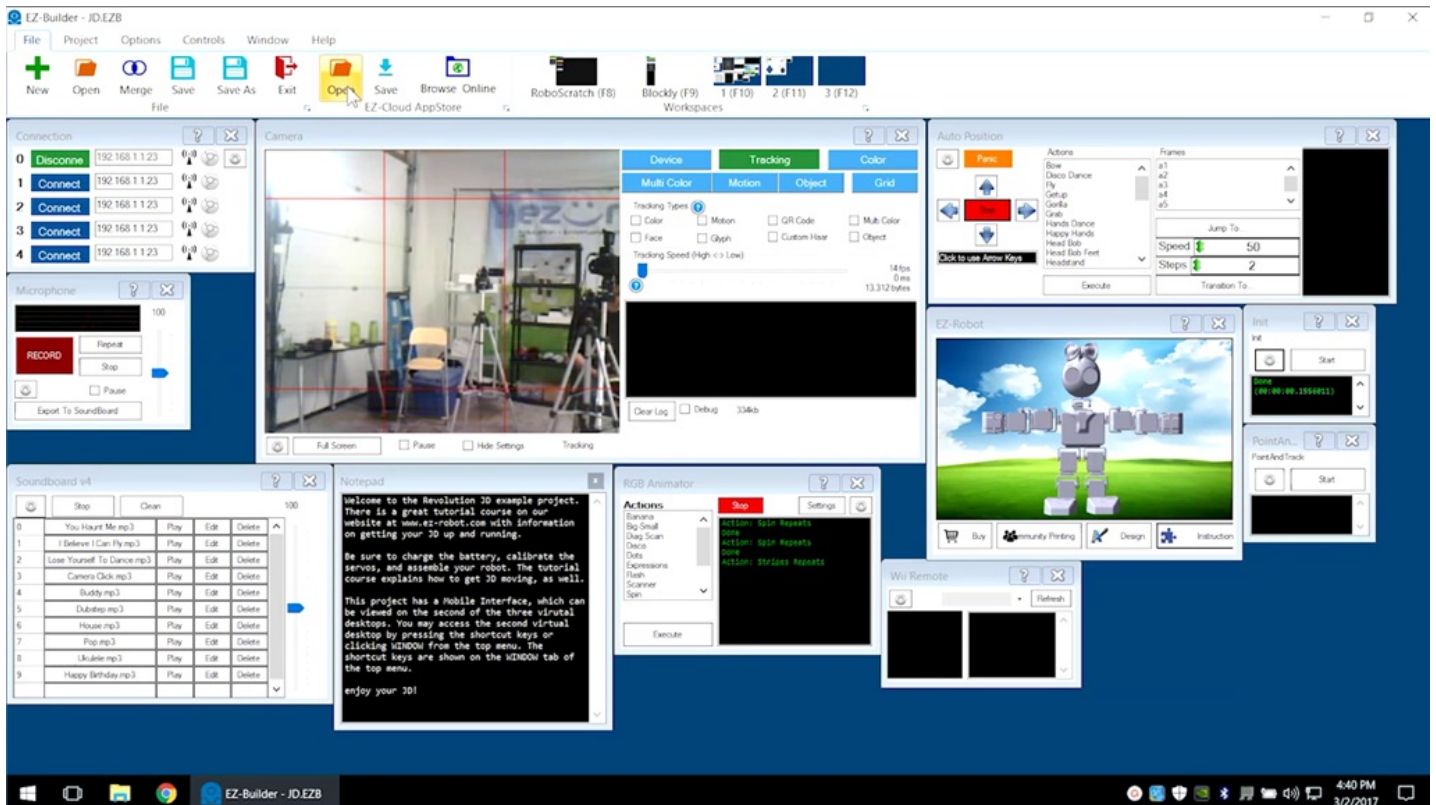
Select the **Reorder** tab to change the order of build instructions.

The screenshot shows the 'Build My EZ-Robot' application window. At the top, there are three tabs: 'Introduction', 'Assembly', and 'Reorder', with 'Reorder' being the active tab. Below the tabs is a list of 12 build steps, each with a small icon and a text label:

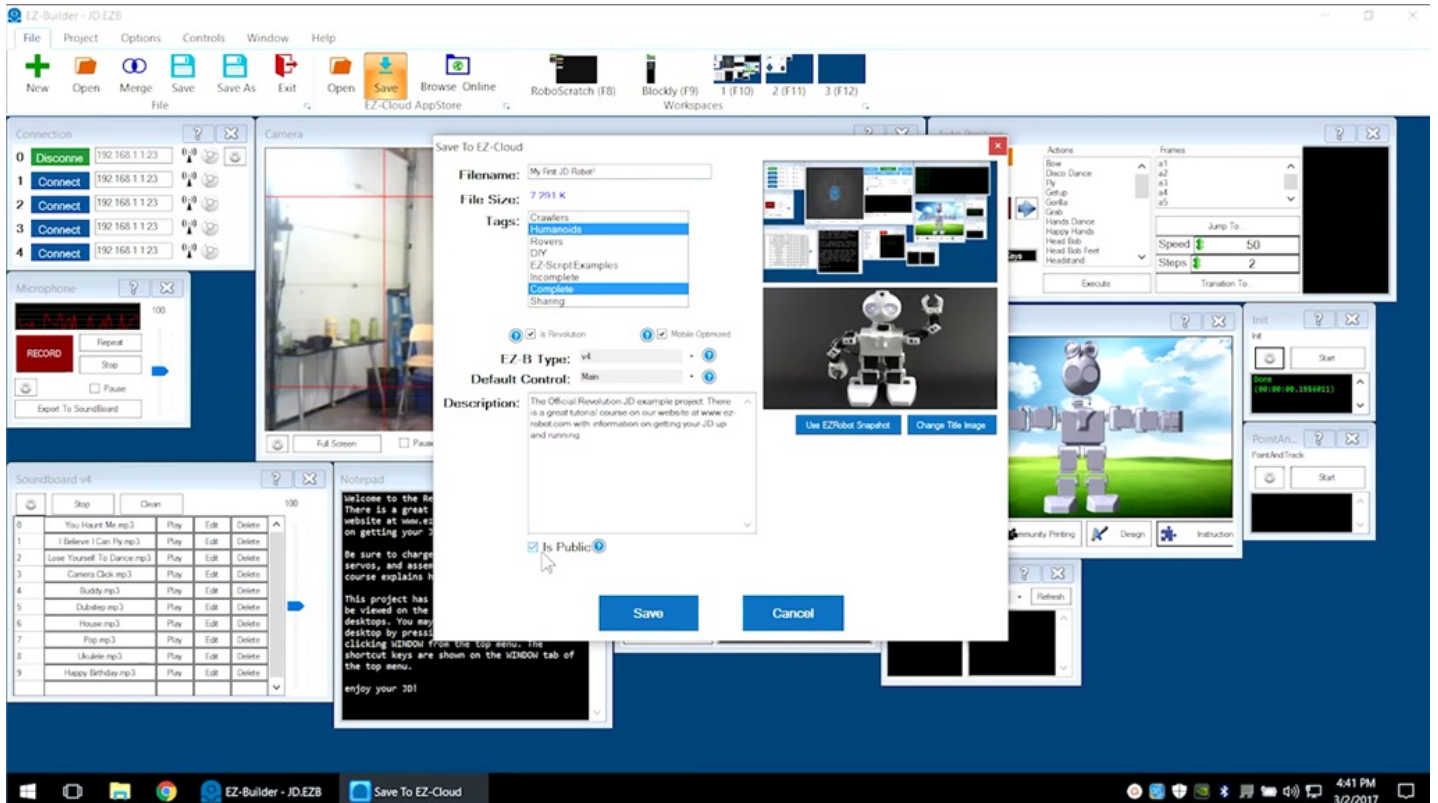
- Chest and Shoulder Servos Humanoid Body
- Left Upper Arm Lever Servo - 90 Degrees
- Left Forearm Lever Servo - 90 Degrees
- Left Gripper Servo Gripper
- Right Upper Arm Lever Servo - 90 Degrees
- Right Forearm Lever Servo - 90 Degrees
- Right Gripper Servo Gripper
- Left Upper Leg Lever Servo - 90 Degrees
- Left Foot Humanoid Left Foot Assembly
- Right Upper Leg Lever Servo - 90 Degrees
- Right Foot Humanoid Right Foot Assembly
- Female Humanoid Head with Camera

On the right side of the window, there is a control panel with four buttons: 'Move Up' (up arrow), 'Move Down' (down arrow), 'Save Changes' (green checkmark), and 'Cancel Changes' (red X). The Windows taskbar at the bottom shows the application is running, with the system clock indicating 4:39 PM on 3/2/2017.

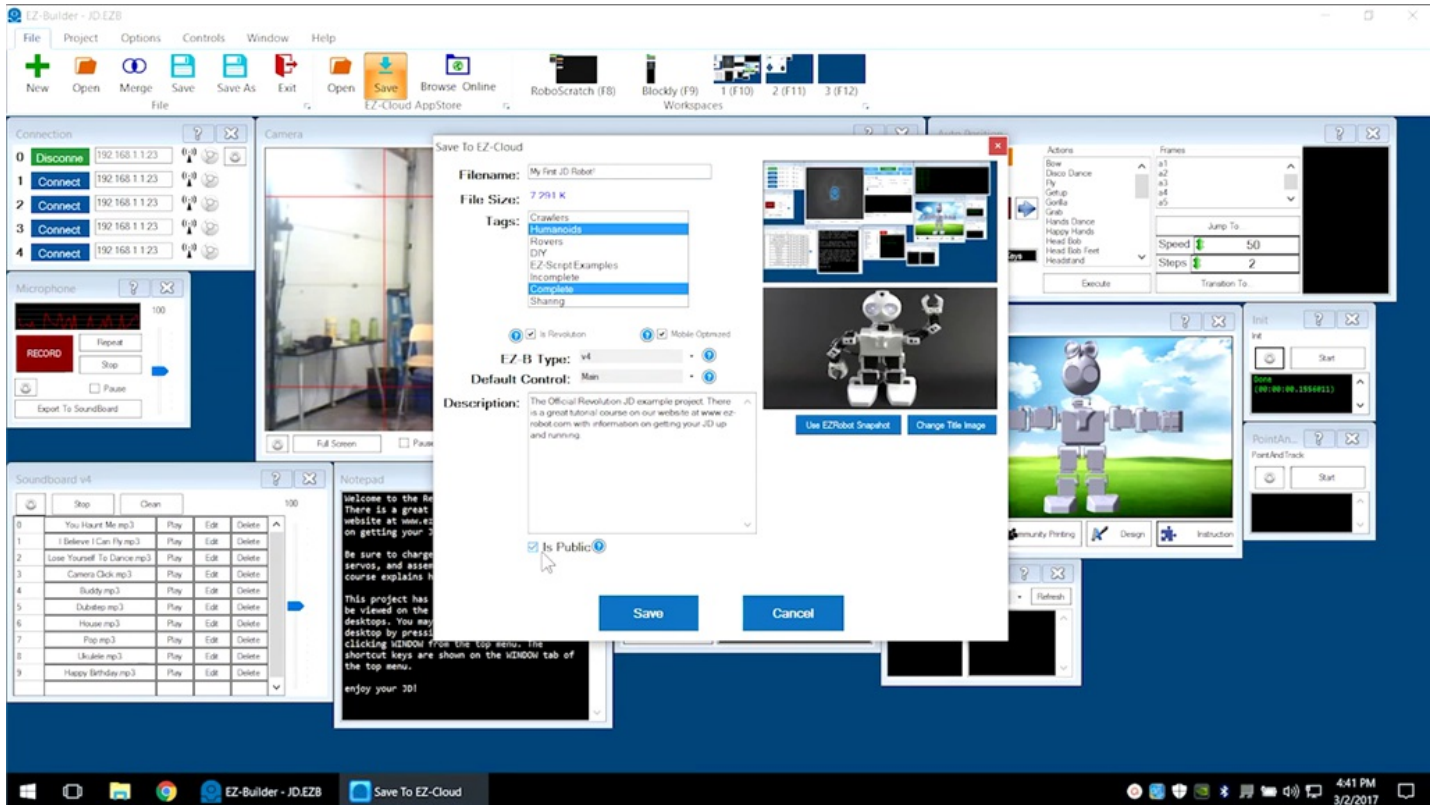
Access the **EZ-Cloud** through **File -> Open** to save and share programs.



Check **Is Public** to share programs with others.



The **EZ-Cloud AppStore** saves all revisions and logs change notes.



Reload a previous version by selecting **File** -> **Open**. Click on the desired project's **Version History** to see listed revisions.

JD

The Official Revision: JD example project. There is a great tutorial course on our website at www.ez-robot.com with information on getting your JD up and running.

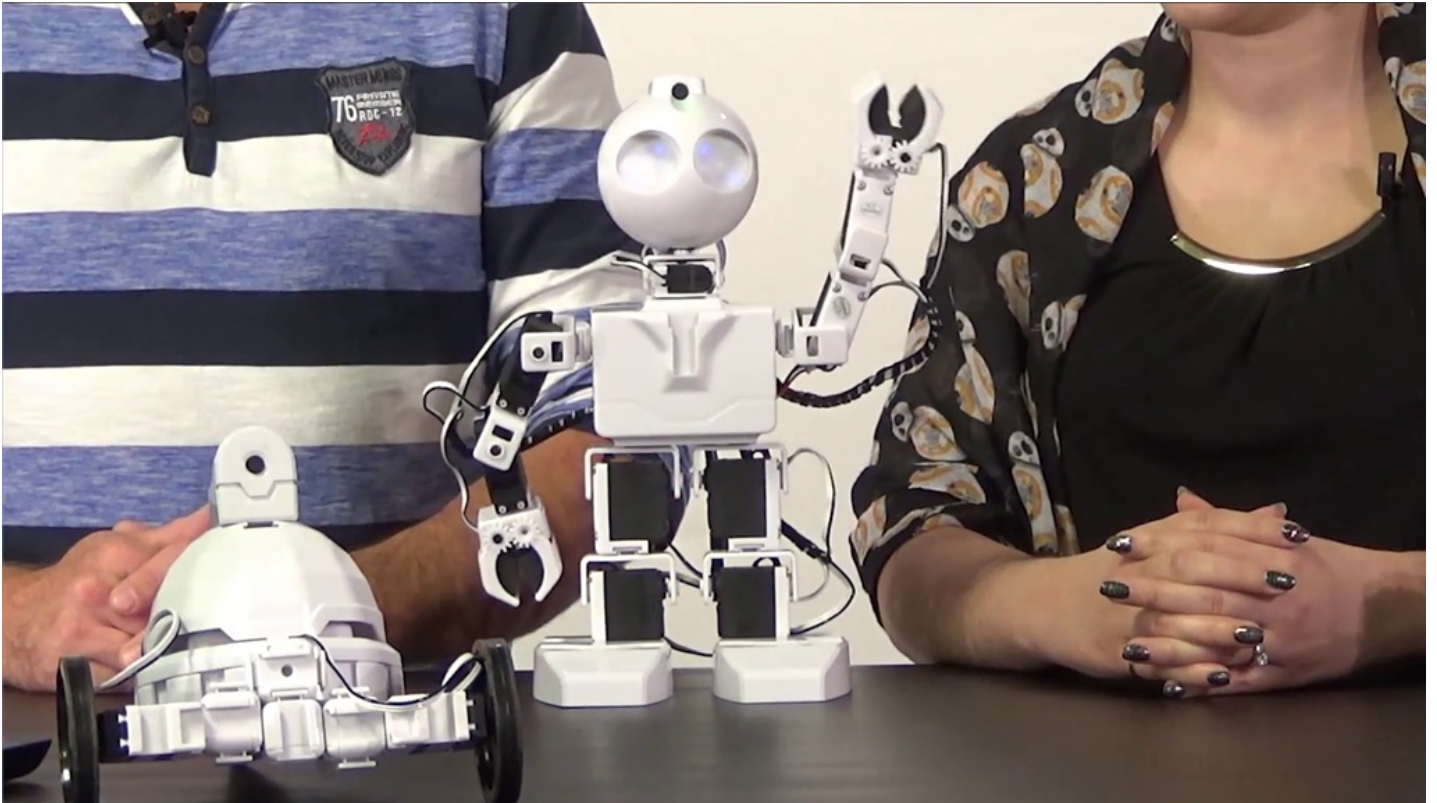
— □ ×

<div style="background-color: #007bff; color: white; padding: 2px; text-align: center; font-weight: bold; margin-bottom: 5px;">Load</div> <p style="font-size: 0.8em; margin: 0;">1/23/2017 2:44:38 PM 7,277,080 Bytes</p> <p style="font-size: 0.7em; margin: 0;">Change Log Updated with new walk, headstand, motions and smoother animations. The camera uses the Grid for tracking. The Point Track is commented in the camera Tracking Script. The camera joystick control in mobile interface does not reset to center when released.</p>	<div style="background-color: #007bff; color: white; padding: 2px; text-align: center; font-weight: bold; margin-bottom: 5px;">Load</div> <p style="font-size: 0.8em; margin: 0;">10/31/2016 6:00:42 PM 7,209,549 Bytes</p> <p style="font-size: 0.7em; margin: 0;">Change Log Disable all tracking types when the checkbox is disabled</p>	<div style="background-color: #007bff; color: white; padding: 2px; text-align: center; font-weight: bold; margin-bottom: 5px;">Load</div> <p style="font-size: 0.8em; margin: 0;">2/7/2016 1:44:53 AM 7,198,023 Bytes</p> <p style="font-size: 0.7em; margin: 0;">Change Log added rgb eye animation to happy birthday dance</p>	<div style="background-color: #007bff; color: white; padding: 2px; text-align: center; font-weight: bold; margin-bottom: 5px;">Load</div> <p style="font-size: 0.8em; margin: 0;">2/3/2016 11:09:22 AM 4,621,855 Bytes</p> <p style="font-size: 0.7em; margin: 0;">Change Log increased i2c speed to 300,000 from 100,000</p>
<div style="background-color: #007bff; color: white; padding: 2px; text-align: center; font-weight: bold; margin-bottom: 5px;">Load</div> <p style="font-size: 0.8em; margin: 0;">11/7/2016 12:42:07 PM 7,246,241 Bytes</p> <p style="font-size: 0.7em; margin: 0;">Change Log camera control uses latest analog joystick mobile control</p>	<div style="background-color: #007bff; color: white; padding: 2px; text-align: center; font-weight: bold; margin-bottom: 5px;">Load</div> <p style="font-size: 0.8em; margin: 0;">10/30/2016 11:44:07 PM 7,209,554 Bytes</p> <p style="font-size: 0.7em; margin: 0;">Change Log made color tracking checkbox larger</p>	<div style="background-color: #007bff; color: white; padding: 2px; text-align: center; font-weight: bold; margin-bottom: 5px;">Load</div> <p style="font-size: 0.8em; margin: 0;">2/6/2016 10:41:22 PM 7,200,541 Bytes</p> <p style="font-size: 0.7em; margin: 0;">Change Log updated default ip address</p>	<div style="background-color: #007bff; color: white; padding: 2px; text-align: center; font-weight: bold; margin-bottom: 5px;">Load</div> <p style="font-size: 0.8em; margin: 0;">2/3/2016 11:03:13 AM 4,618,284 Bytes</p> <p style="font-size: 0.7em; margin: 0;">Change Log Changed capitalization on rgb animator actions</p>
<div style="background-color: #007bff; color: white; padding: 2px; text-align: center; font-weight: bold; margin-bottom: 5px;">Load</div> <p style="font-size: 0.8em; margin: 0;">11/2/2016 3:45:30 PM 7,147,727 Bytes</p> <p style="font-size: 0.7em; margin: 0;">Change Log Vertical camera up/down control inverted</p>	<div style="background-color: #007bff; color: white; padding: 2px; text-align: center; font-weight: bold; margin-bottom: 5px;">Load</div> <p style="font-size: 0.8em; margin: 0;">10/30/2016 11:40:01 PM 7,211,259 Bytes</p> <p style="font-size: 0.7em; margin: 0;">Change Log Added color tracking checkbox</p>	<div style="background-color: #007bff; color: white; padding: 2px; text-align: center; font-weight: bold; margin-bottom: 5px;">Load</div> <p style="font-size: 0.8em; margin: 0;">2/6/2016 10:34:01 PM 7,200,477 Bytes</p> <p style="font-size: 0.7em; margin: 0;">Change Log smaller font size to fit words on mobile interface</p>	<div style="background-color: #007bff; color: white; padding: 2px; text-align: center; font-weight: bold; margin-bottom: 5px;">Load</div> <p style="font-size: 0.8em; margin: 0;">2/3/2016 10:39:37 AM 4,617,391 Bytes</p> <p style="font-size: 0.7em; margin: 0;">Change Log capitalized each first letter of Actions in actions</p>
<div style="background-color: #007bff; color: white; padding: 2px; text-align: center; font-weight: bold; margin-bottom: 5px;">Load</div> <p style="font-size: 0.8em; margin: 0;">11/2/2016 3:42:00 PM 7,147,726 Bytes</p> <p style="font-size: 0.7em; margin: 0;">Change Log servo pad UI for JD's head</p>	<div style="background-color: #007bff; color: white; padding: 2px; text-align: center; font-weight: bold; margin-bottom: 5px;">Load</div> <p style="font-size: 0.8em; margin: 0;">7/29/2016 11:27:39 PM 7,206,354 Bytes</p> <p style="font-size: 0.7em; margin: 0;">Change Log Adjusted size of controls for Windows 10 125% default DPI setting and latest ez-builder</p>	<div style="background-color: #007bff; color: white; padding: 2px; text-align: center; font-weight: bold; margin-bottom: 5px;">Load</div> <p style="font-size: 0.8em; margin: 0;">2/6/2016 10:31:25 PM 7,200,493 Bytes</p> <p style="font-size: 0.7em; margin: 0;">Change Log New music additions and rgb animations dubstep, house music, pop, happy birthday and more...</p>	<div style="background-color: #007bff; color: white; padding: 2px; text-align: center; font-weight: bold; margin-bottom: 5px;">Load</div> <p style="font-size: 0.8em; margin: 0;">2/1/2016 12:00:40 PM 4,617,947 Bytes</p> <p style="font-size: 0.7em; margin: 0;">Change Log JD's hands no longer close automatically will keep their last position during walking have him holding something.</p>

Close

4:42 PM
3/2/2017

Follow [The Robot Program](#) episodes to see all the features **EZ-Builder** has to offer.



Question #1

Which workspace is designed for linear programming?

Question #2

Which workspace is designed for programming with logic, branches, and loops?

Question #3

What is the EZ-Robot scripting language called?

View the answers to this quiz at www.ez-robot.com/Tutorials/Lesson/20.

Visit www.TheRobotProgram.com for more episodes.