

SYNTHIAM

synthiam.com

The Robot Program Episode 006: Introducing ARC

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.

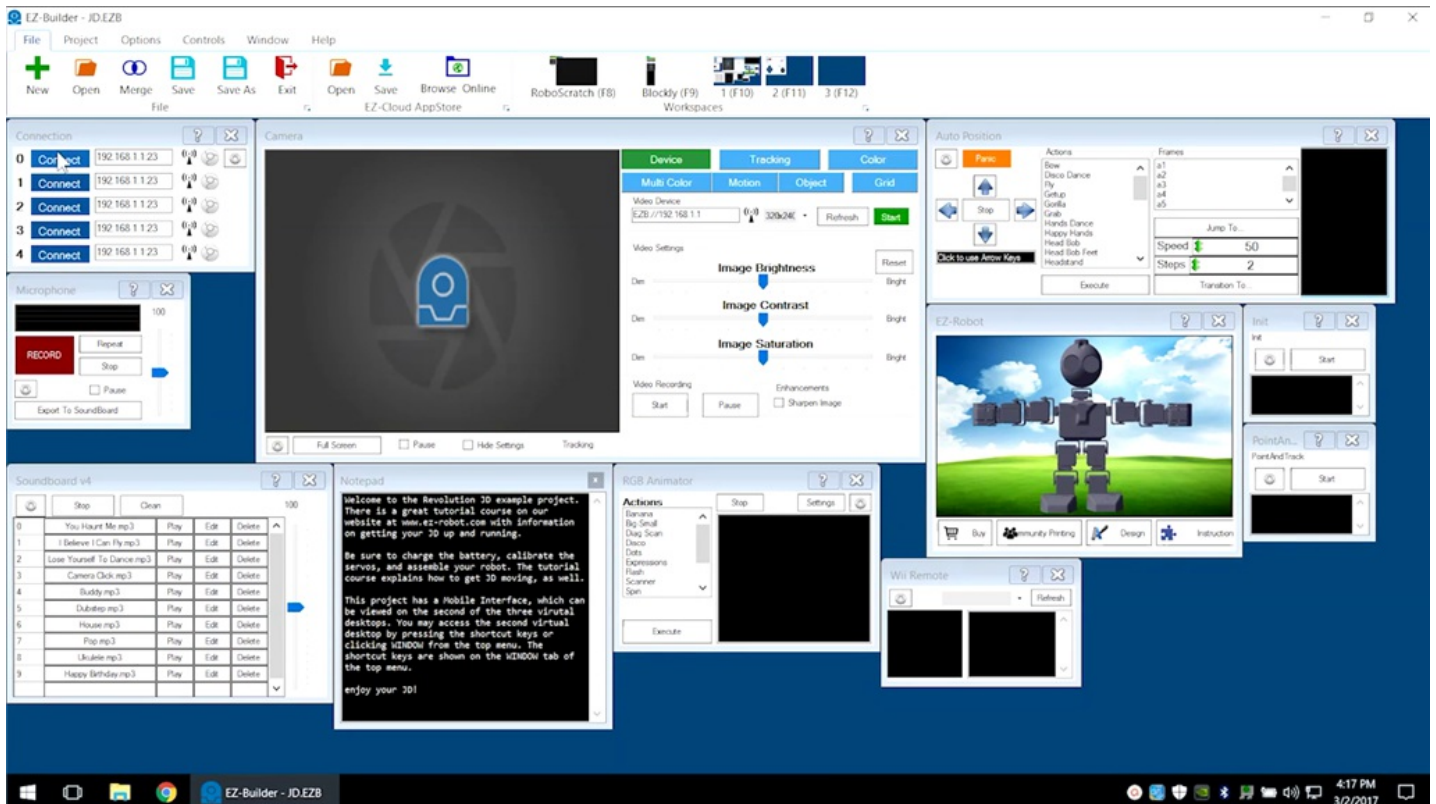
View the video episode here: <https://www.ez-robot.com/Tutorials/Lesson/20>

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 has a blue header with the 'ezrobot' logo and navigation links: Explore, Products, Software (selected), Learn, and Community. A secondary navigation bar lists categories: Windows, Plugins, Mobile, UniversalBot, Windows SDK, Mono SDK, Open IoT Wifi, and 3rd Party. The main content area features a large blue banner with the title 'EZ-Builder For Windows'. On the left, there is a video player with the title 'The EZ-Life... All The Robots!' and a play button. To the right of the video are three buttons: a green 'Download EZ-Builder Installer.msi' button, a blue 'Manual' button, and a blue 'Release notes' button. Below these buttons, the text reads 'EZ-Builder Version 2017.02.20.00' and '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 time as 4:14 PM on 3/2/2017 and an active window for 'EZ-Builder for Win...'.

Ⓢ Loading an Example Project

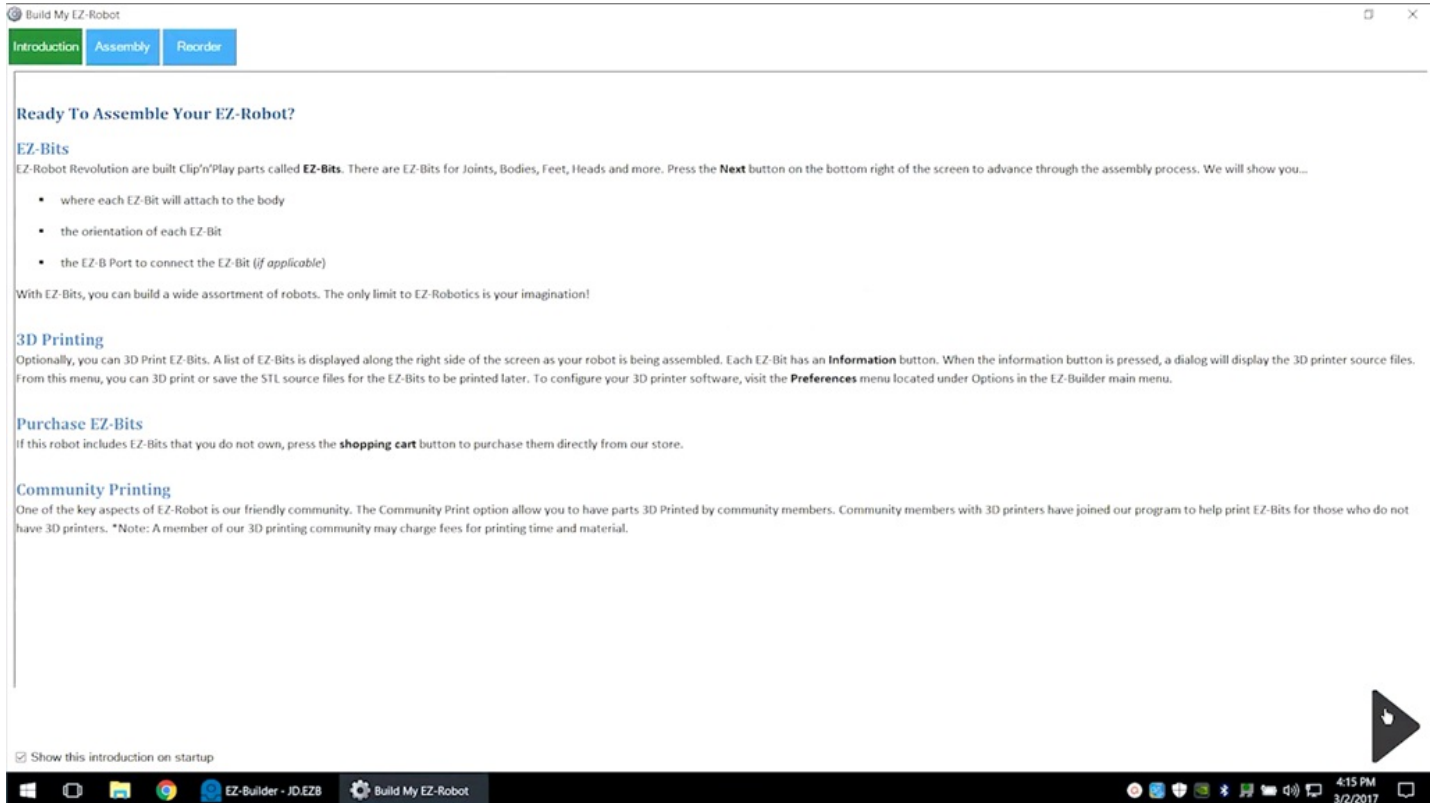
Power on the robot. This example uses **Revolution JD**. Load an example project for the robot.

The screenshot shows the 'Open Project File' dialog box in the EZ-Builder software. The dialog is titled 'Open Project File' and has a search bar and a 'Show All' button. The current directory is 'C:\Users\Public\Documents\EZ-Builder\Examples\'. The left sidebar shows 'My Files' and 'Examples' tabs, with 'Examples' selected. Below the sidebar are 'Back' and 'Create Folder' buttons. The main area displays a list of example projects:

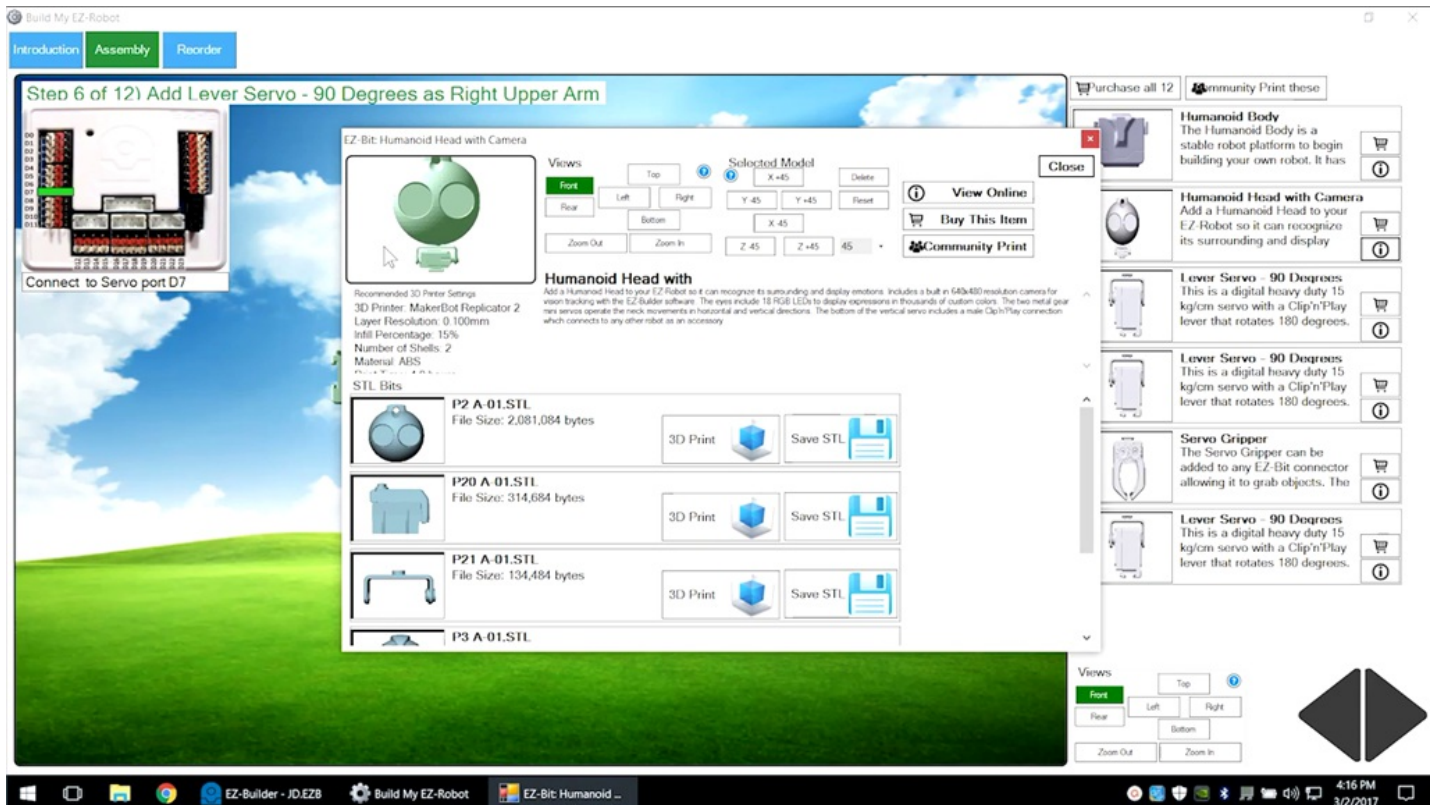
Project Name	File Size	Controls	EZ-Bits	Action
AdventureBot	6,373,115 Bytes	16 Controls	10 EZ-Bits	Open
Battle Flipper	6,200,550 Bytes	9 Controls	13 EZ-Bits	Open
JD	7,205,820 Bytes	12 Controls	12 EZ-Bits	Open
Rolli	4,587,251 Bytes	12 Controls	13 EZ-Bits	Open
Six	3,788,204 Bytes	16 Controls	21 EZ-Bits	Open

At the bottom of the dialog, there is a 'Close' button and a checkbox labeled 'Auto arrange controls after open'. The Windows taskbar at the bottom shows the 'EZ-Builder - Versio...' and 'Open Project File' windows, along with the system tray showing the time as 4:15 PM on 3/2/2017.

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

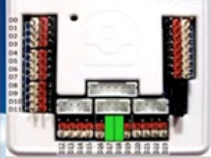


Click on the **i** icon for more information about a part, including 3D printing designs.

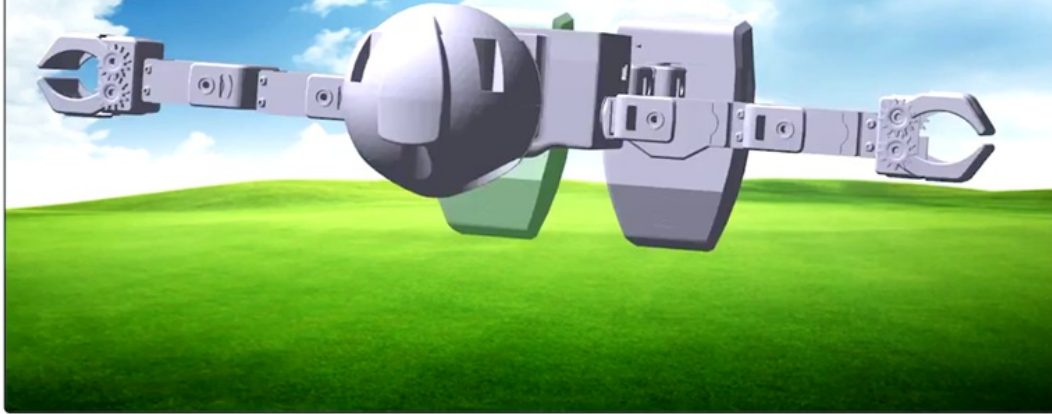


Use the arrow buttons to view the robot from different angles.








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

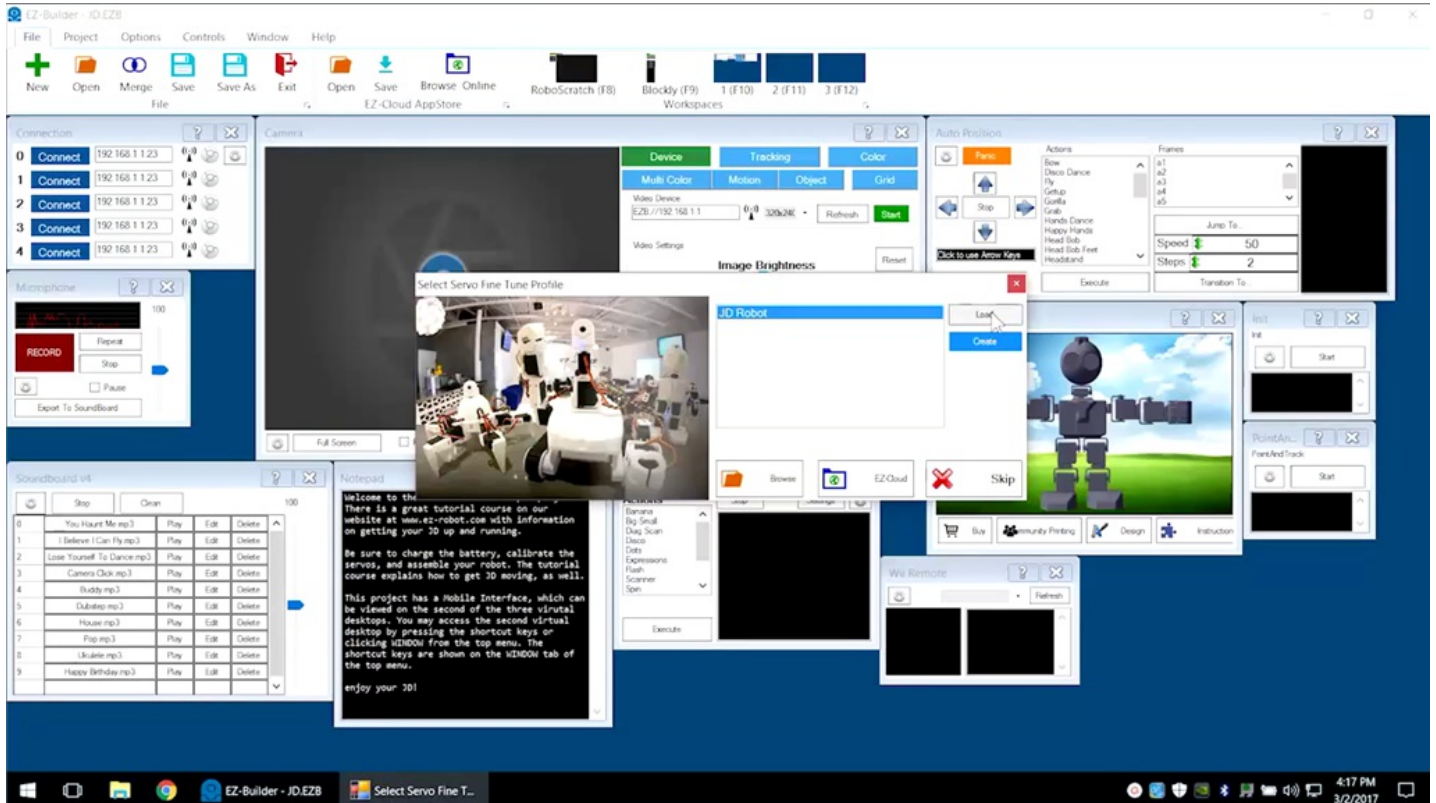
-  **Humanoid Body**
The Humanoid Body is a stable robot platform to begin building your own robot. It has
-  **Humanoid Head with Camera**
Add a Humanoid Head to your EZ-Robot so it can recognize its surrounding and display
-  **Lever Servo - 90 Degrees**
This is a digital heavy duty 15 kg/cm servo 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

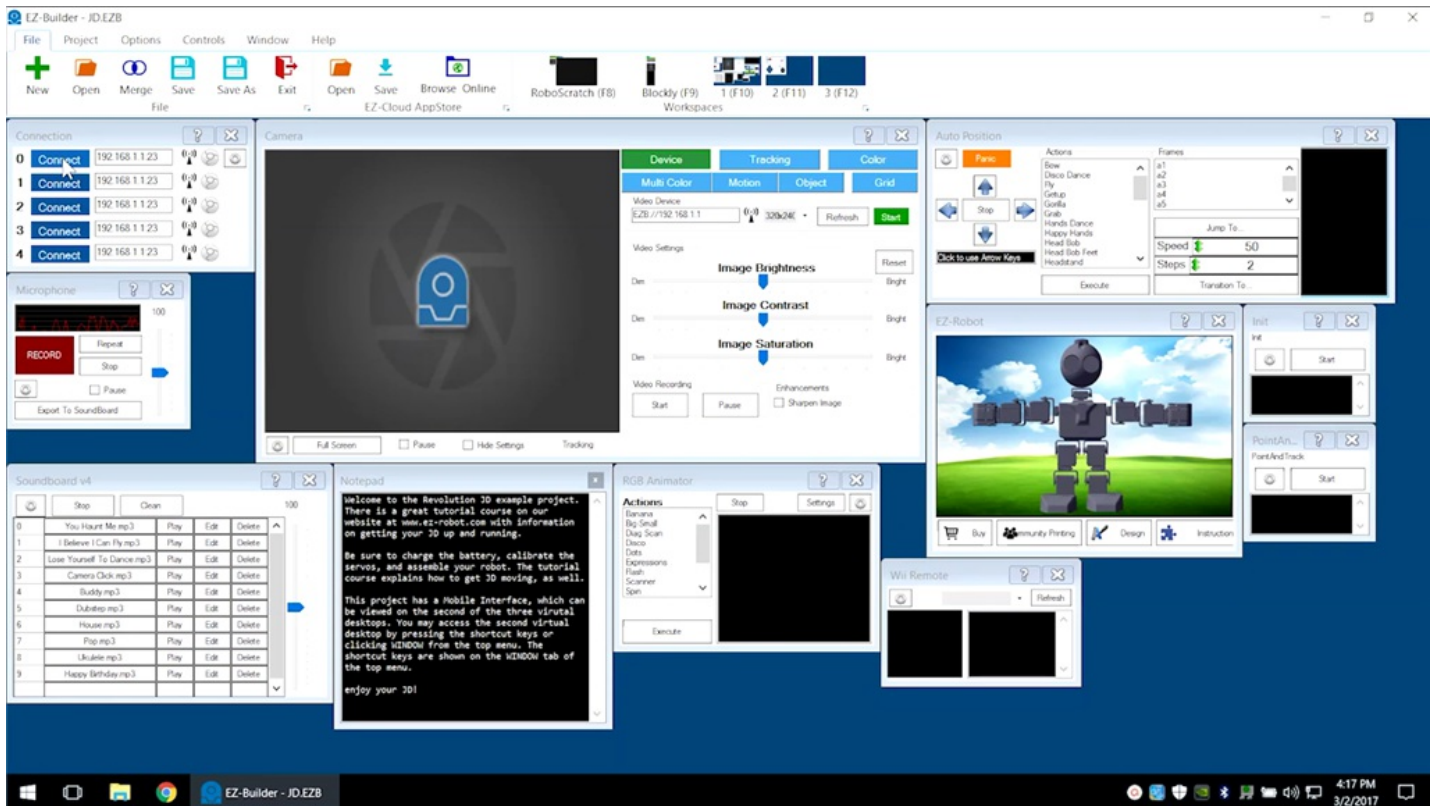
Zoom Out Zoom In

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



Connecting

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.

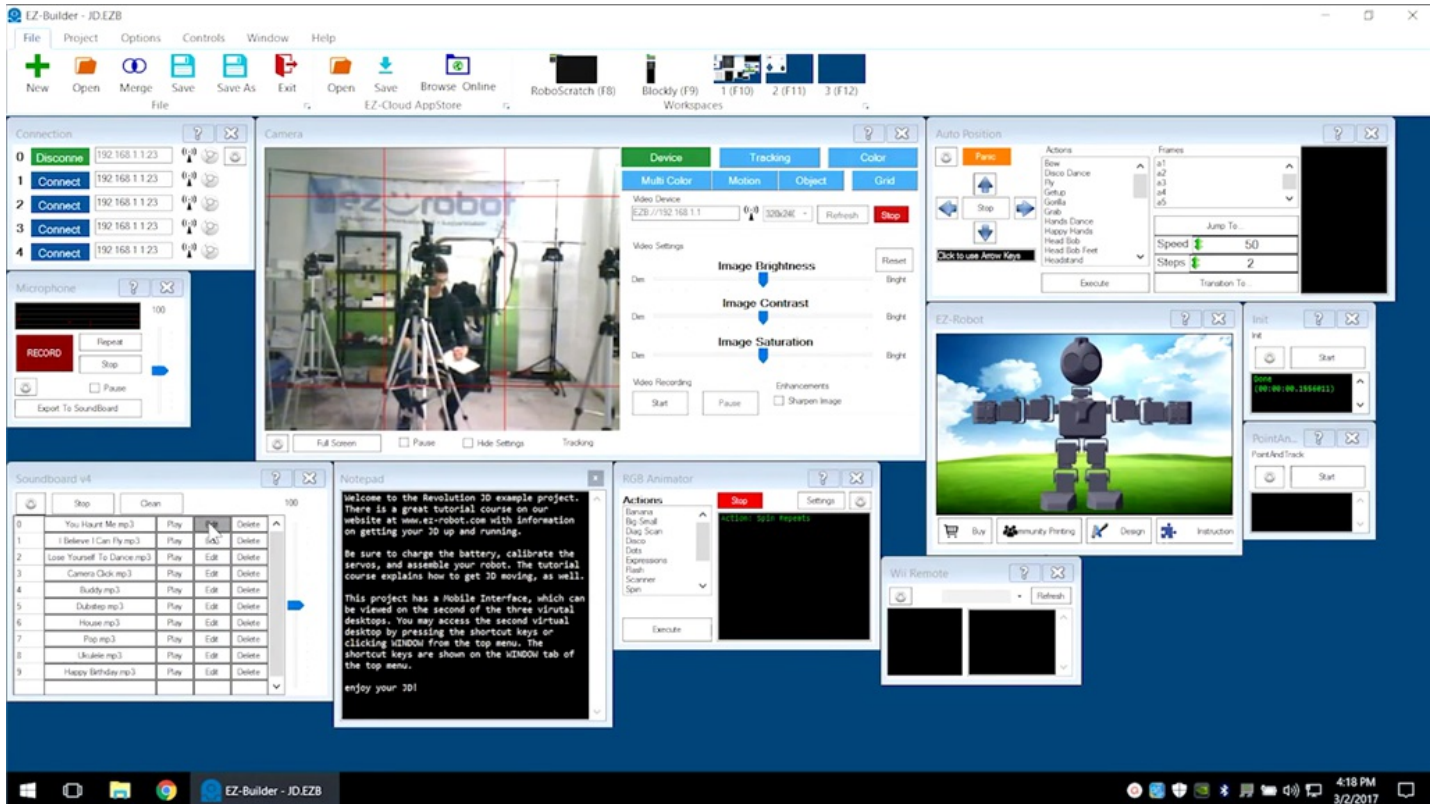
The screenshot displays the EZ-Builder software interface with the following components:

- Connection:** A list of four connection points, each labeled 'Connect' and showing the IP address 192.168.1.123.
- Microphone:** A control window with a volume slider set to 100, a red 'RECORD' button, and a 'Pause' button.
- Camera:** A central window showing a live video feed of a person at a desk with a camera on a tripod. It includes 'Image Brightness', 'Image Contrast', and 'Image Saturation' sliders, and a 'Video Recording' section with 'Start' and 'Pause' buttons.
- Auto Position:** A window with 'Actions' (Bow, Disco Dance, Fly, GetUp, Goofa, Grab, Hands Dance, Happy Hands, Head Bob, Head Bob Feet, Headstand) and 'Frames' (a1-a5) lists. It also has 'Speed' and 'Steps' sliders.
- EZ-Robot:** A window showing a 3D model of a robot on a green field under a blue sky.
- Soundboard v4:** A table with columns for 'Step', 'Clean', 'Play', 'Edit', and 'Delete'. It lists 9 audio files such as 'You Heart Me.mp3', 'I Believe I Can Fly.mp3', etc.
- Notepad:** A text area containing a welcome message for the Revolution 3D example project.
- RGB Animator:** A window with an 'Actions' list (Banana, Big Smile, Diag Scan, Data, Expressions, Flash, Scanner, Spin) and a 'Stop' button.
- Wii Remote:** A window with a 'Refresh' button.

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

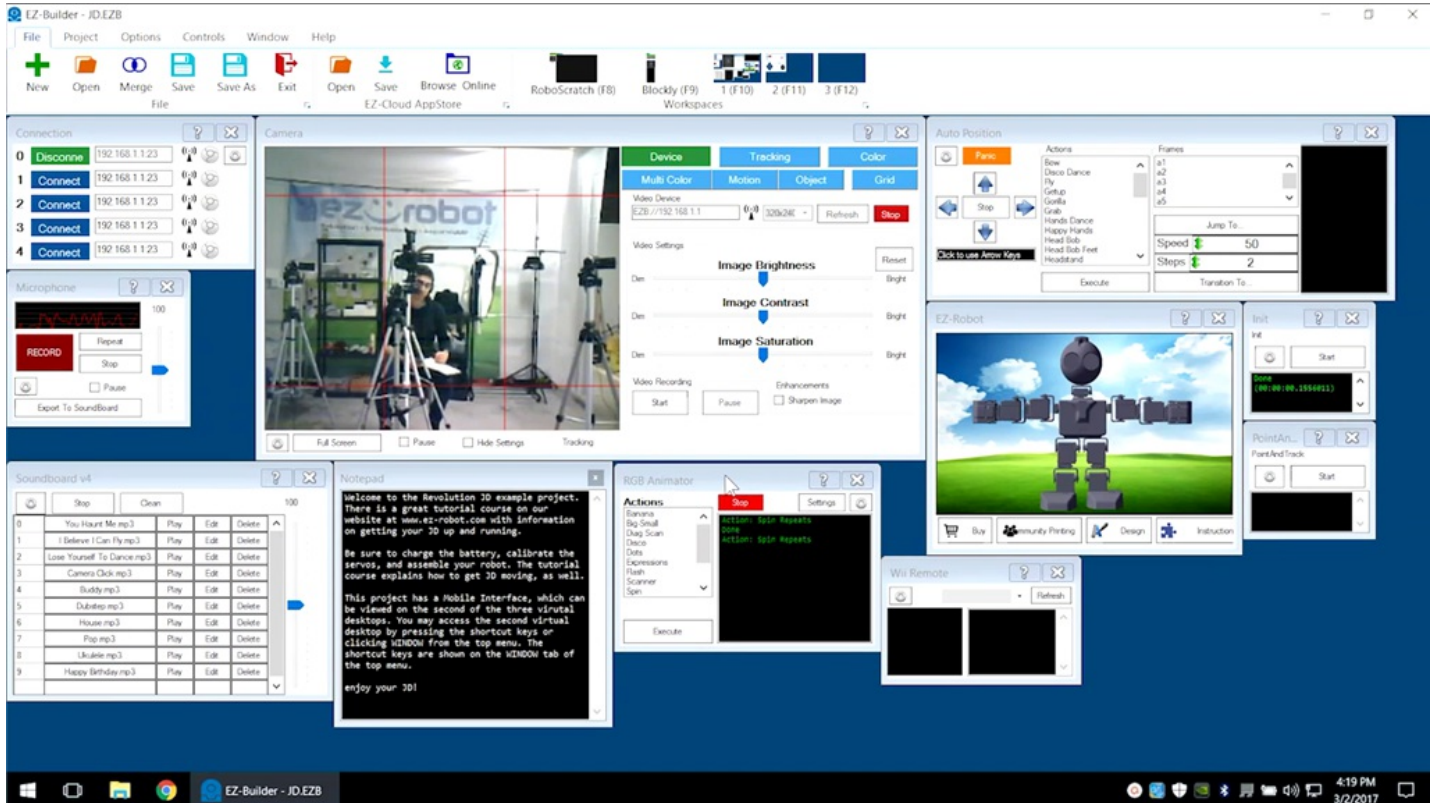
Soundboard Control

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

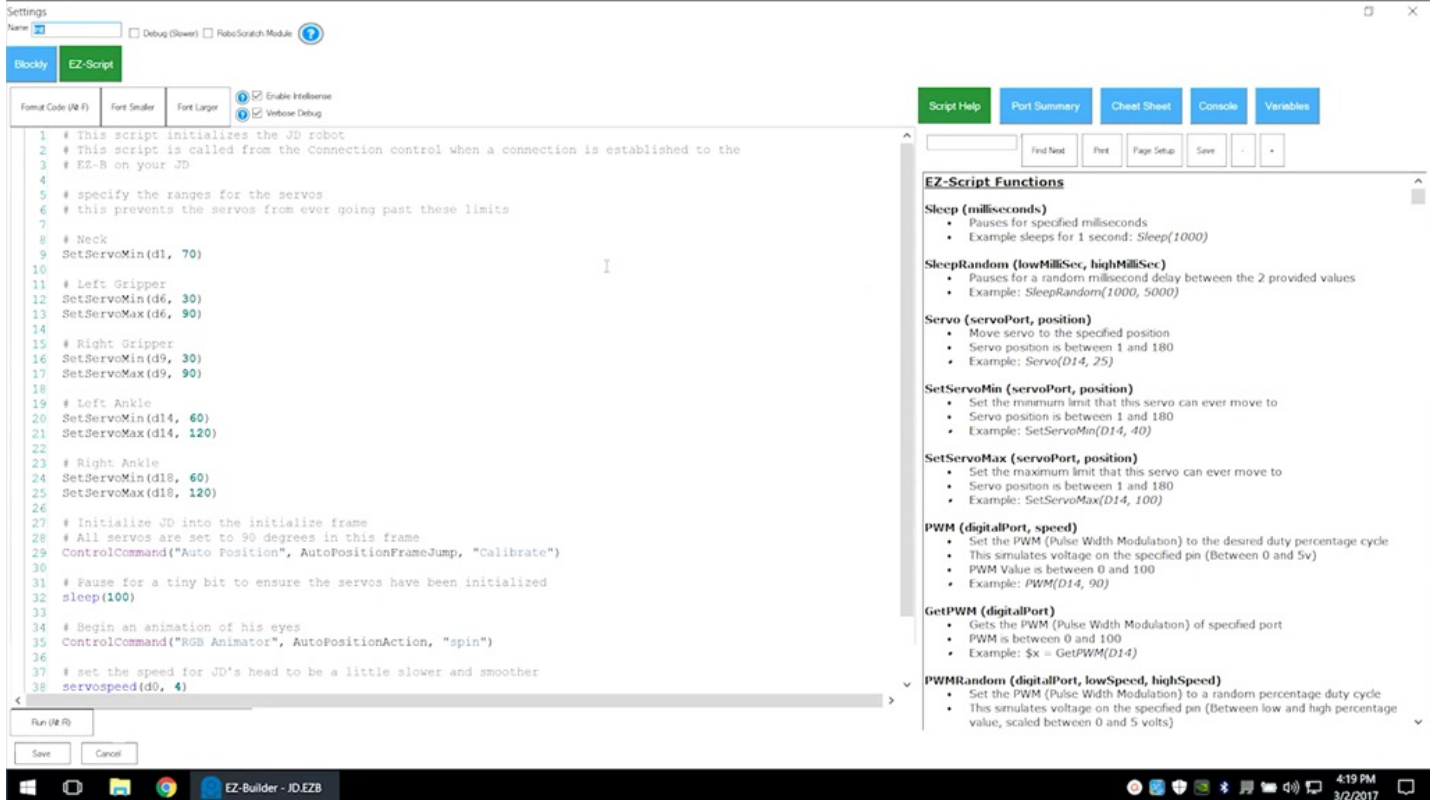


Commonly Used Controls

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



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



The screenshot displays the EZ-Script editor interface. The main window shows a script for initializing a robot's servos. The script includes comments and function calls for setting servo ranges and initializing the robot's frame. The right-hand panel provides a reference for EZ-Script functions, including Sleep, SleepRandom, Servo, SetServoMin, SetServoMax, PWM, GetPWM, and PWMRandom.

```
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)
```

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)

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

The screenshot displays the EZ-Builder software interface for a robot camera. 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, EZ-Cloud AppStore, RoboScratch (F8), Blockly (F9), and Workspaces (F10, F11, F12).

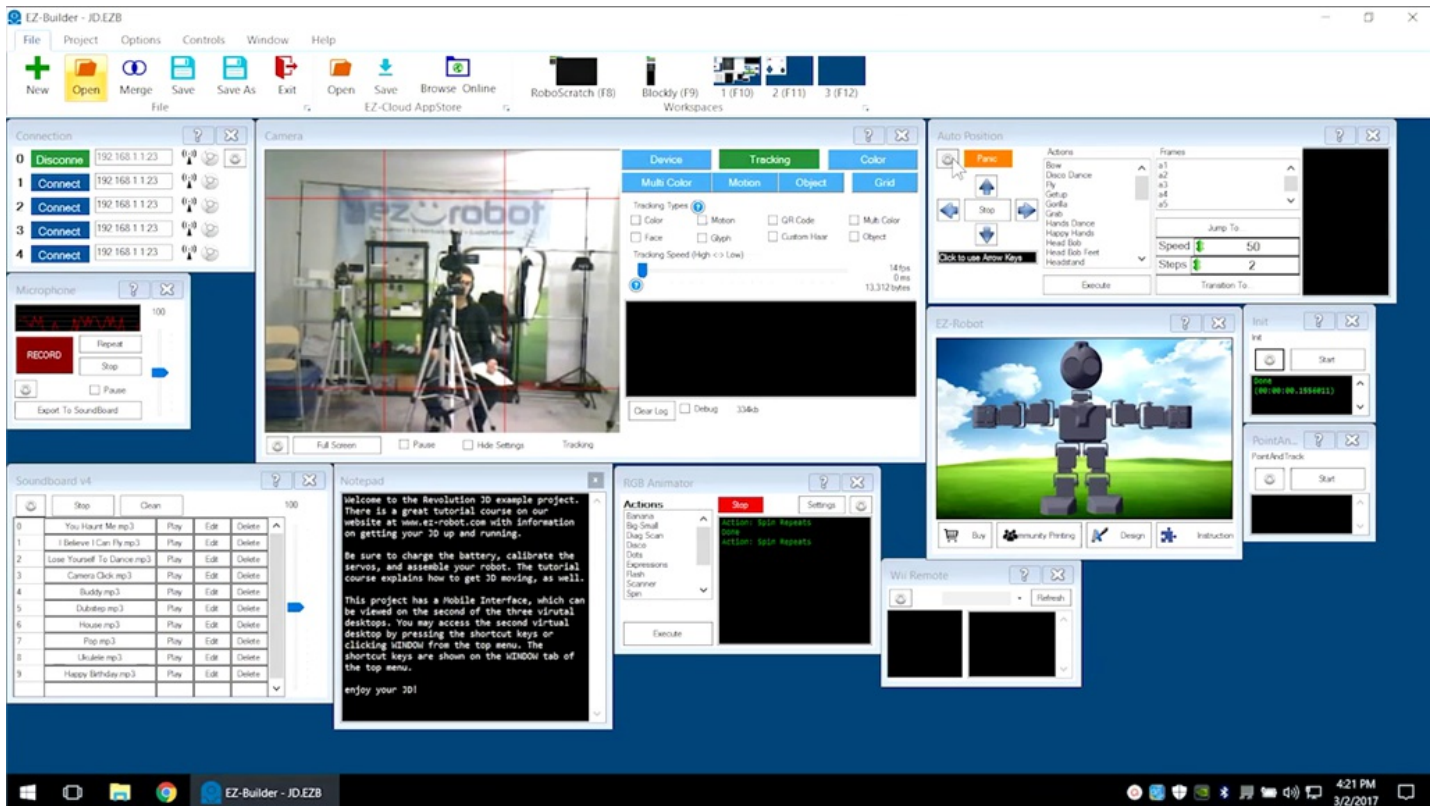
The interface is divided into several panels:

- Connection:** A list of connection attempts with IP addresses (192.168.1.123) and status indicators (Disconnect, Connect).
- Microphone:** A control panel with a volume slider, a "RECORD" button, and "Repeat", "Stop", and "Export To Soundboard" options.
- Soundboard v4:** A table with columns for "Stop", "Clean", "Play", "Edit", and "Delete". It contains a list of audio files (e.g., "You Hear Me.mp3", "I Believe I Can Fly.mp3").
- Camera:** A central panel showing a live video feed of a robot in a room. It includes a "Tracking" tab with sub-tabs for "Device", "Motion", "Object", and "Grid". The "Tracking" sub-tab is active, showing "Tracking Types" (Color, Motion, QR Code, Multi Color) and "Tracking Speed (High <-> Low)" set to 12 fps. A "Clear Log" button and "Debug" checkbox are also present.
- Auto Position:** A panel with "Action" and "Frames" sections. The "Action" section includes buttons for "Go", "Stop", and "Click to use New Keys". The "Frames" section has a list of frames (a1-e5) and a "Jump To" field.
- EZ-Robot:** A 3D model of the robot in a virtual environment. It includes buttons for "Buy", "Community Posting", "Design", and "Instruction".
- RGB Animator:** A panel with an "Actions" list (Turn on, Big Smile, Diag Scan, Dance, Dots, Expressions, Flash, Scanner, Spin) and a "Stop" button.
- Wii Remote:** A panel with a "Refresh" button and two empty slots.
- Notepad:** A text area containing a welcome message and instructions for the "Revolution 3D" project.
- Int:** A panel with a "Start" button and a "Done" indicator.
- PointAn...:** A panel with a "Start" button and a "PointAndTrack" indicator.

The Windows taskbar at the bottom shows the system tray with the time 4:20 PM and date 3/2/2017.

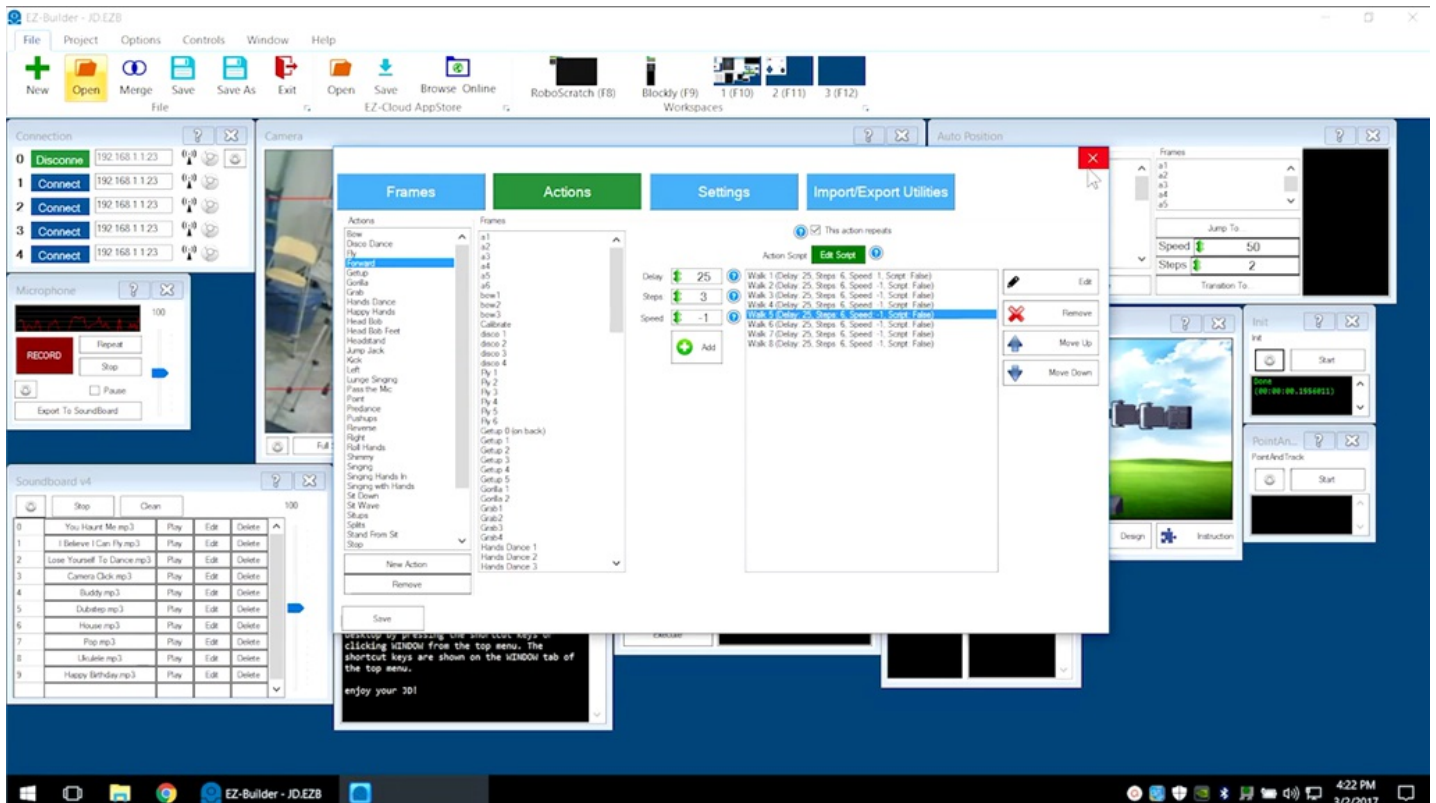
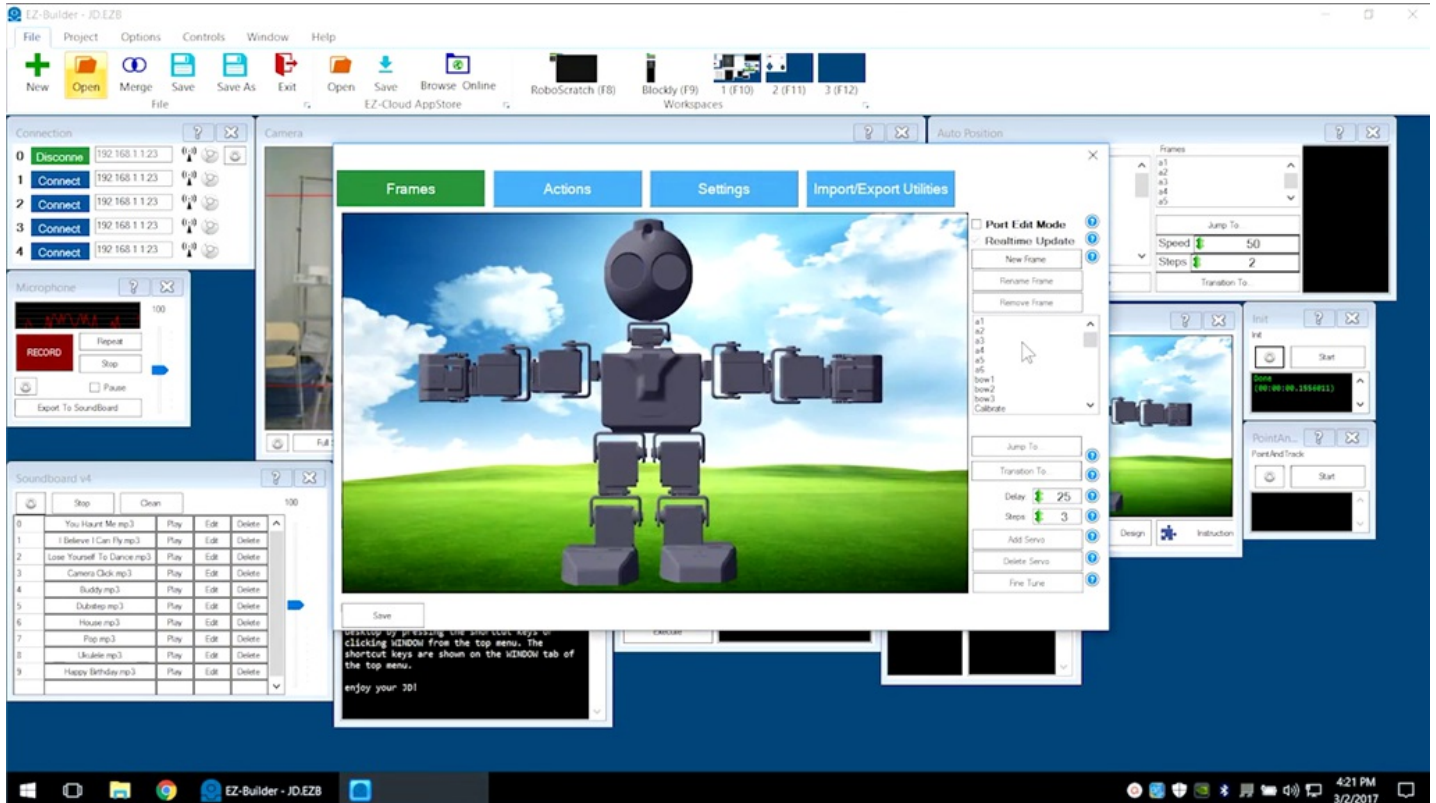
Auto Position Control

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



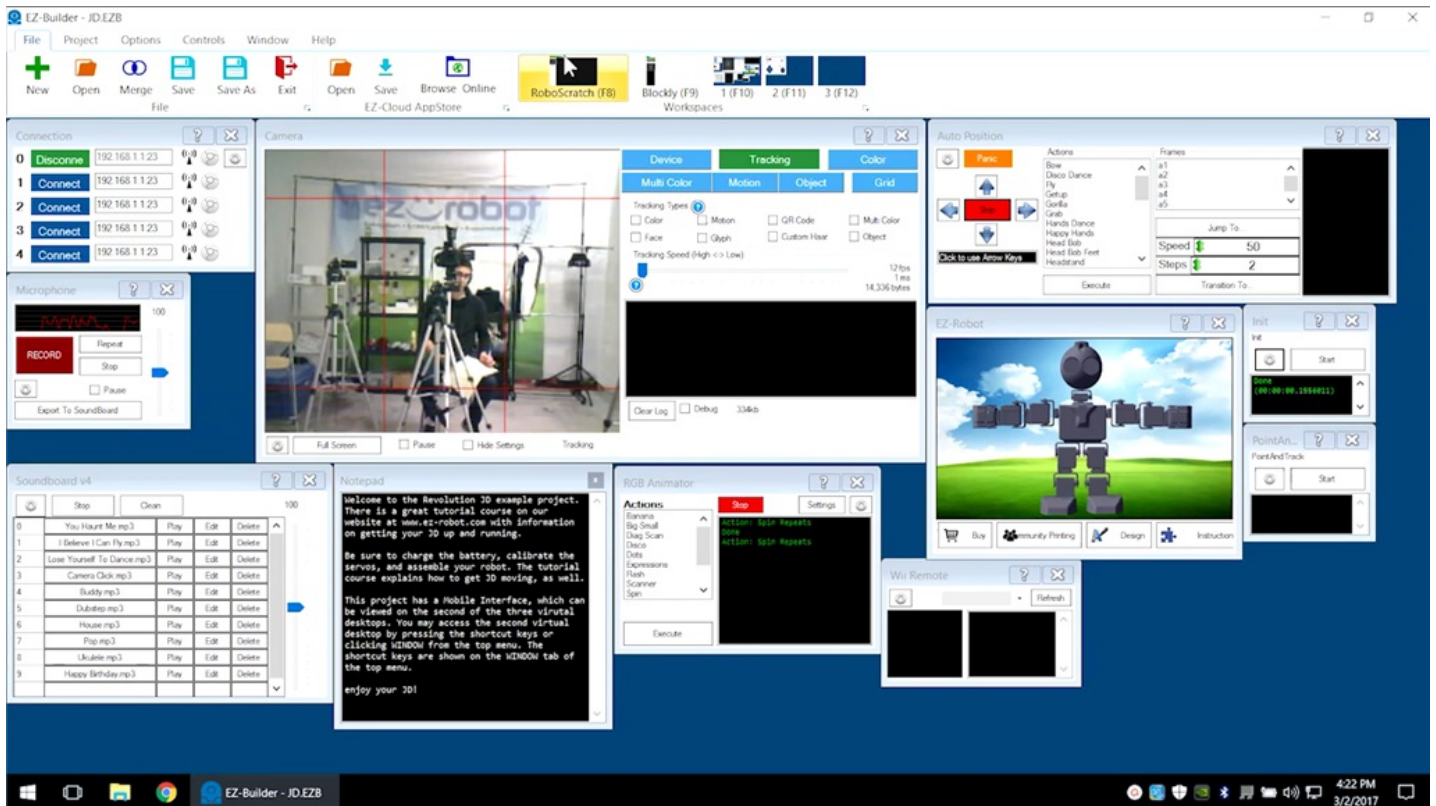
Frame Creation

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



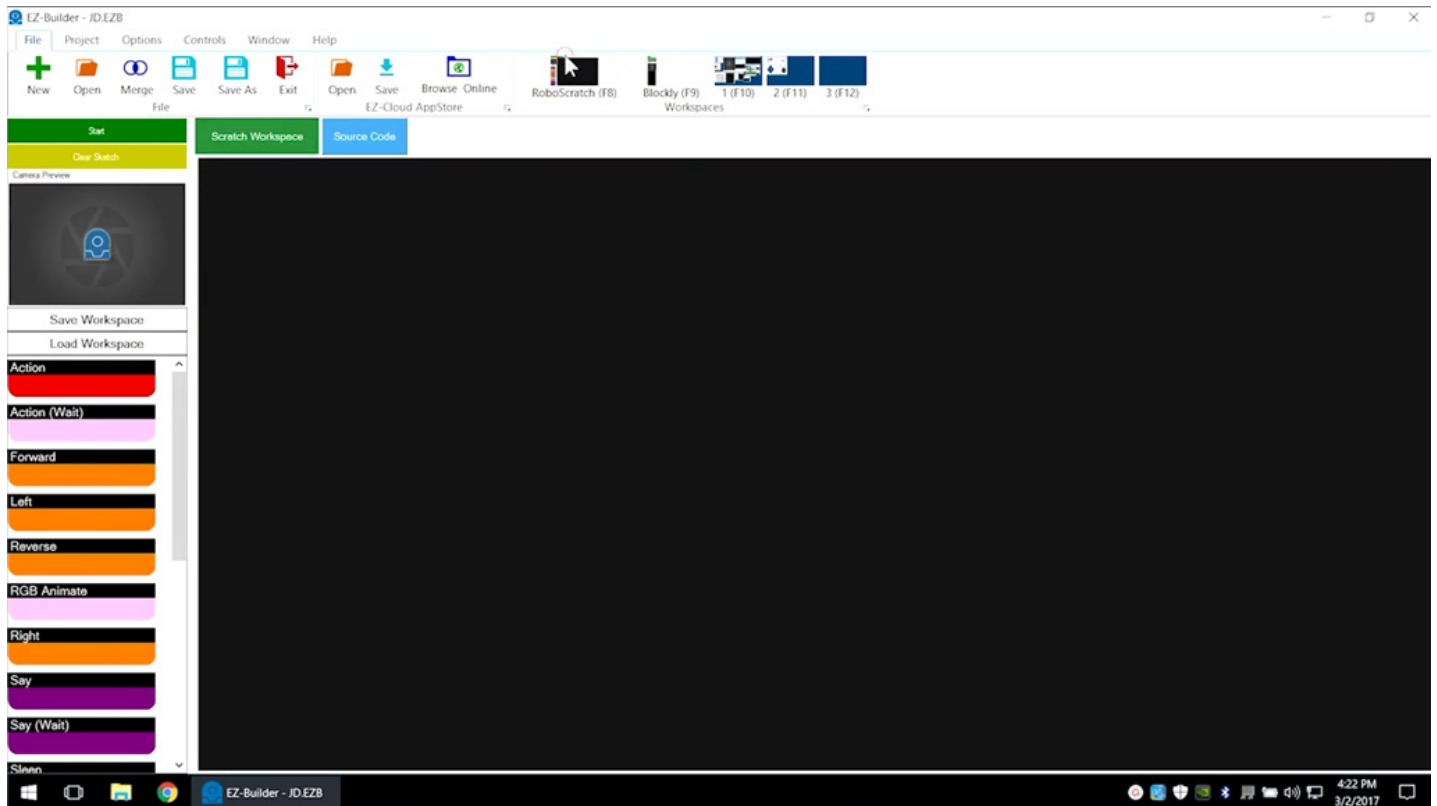
Coding Workspaces

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



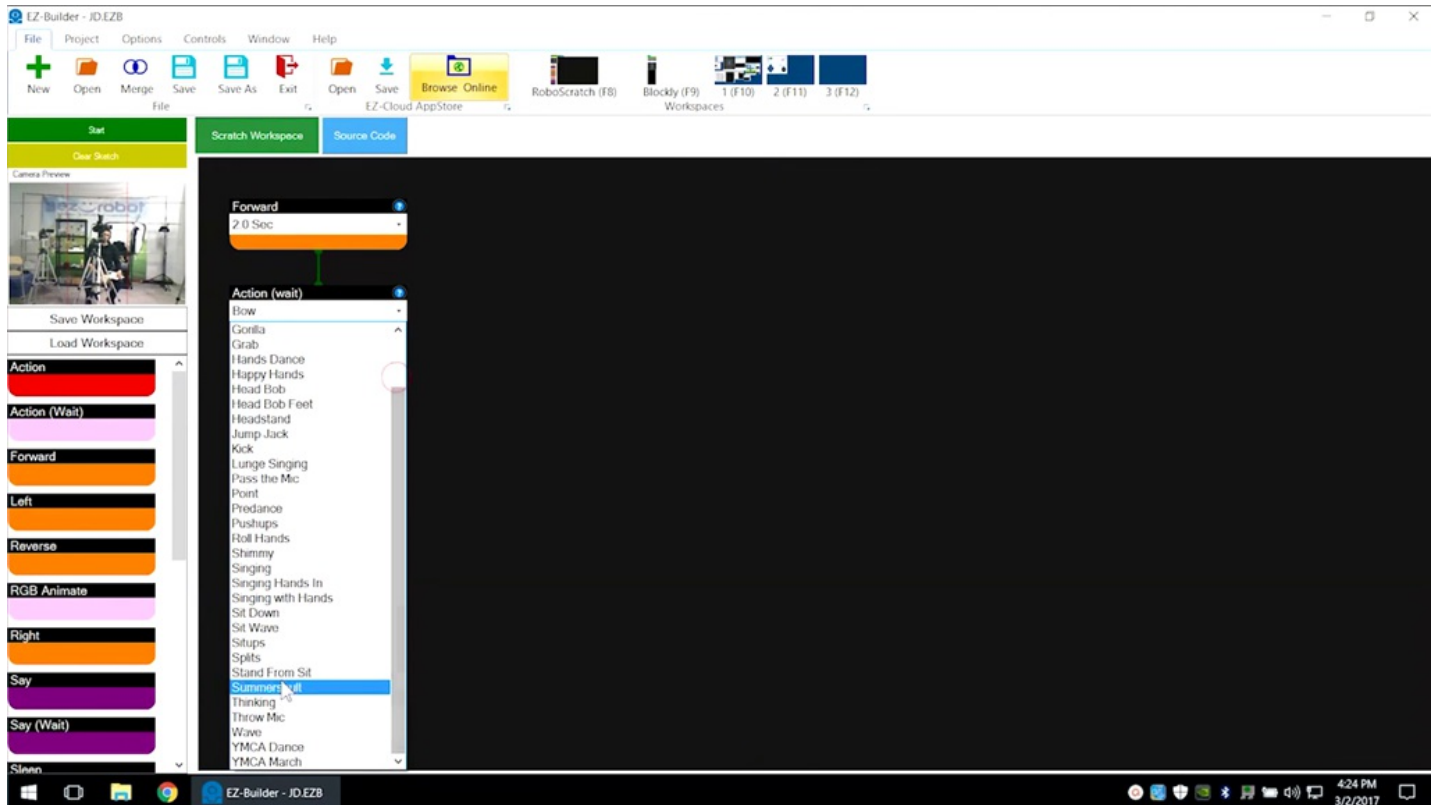
RoboScratch Workspace

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



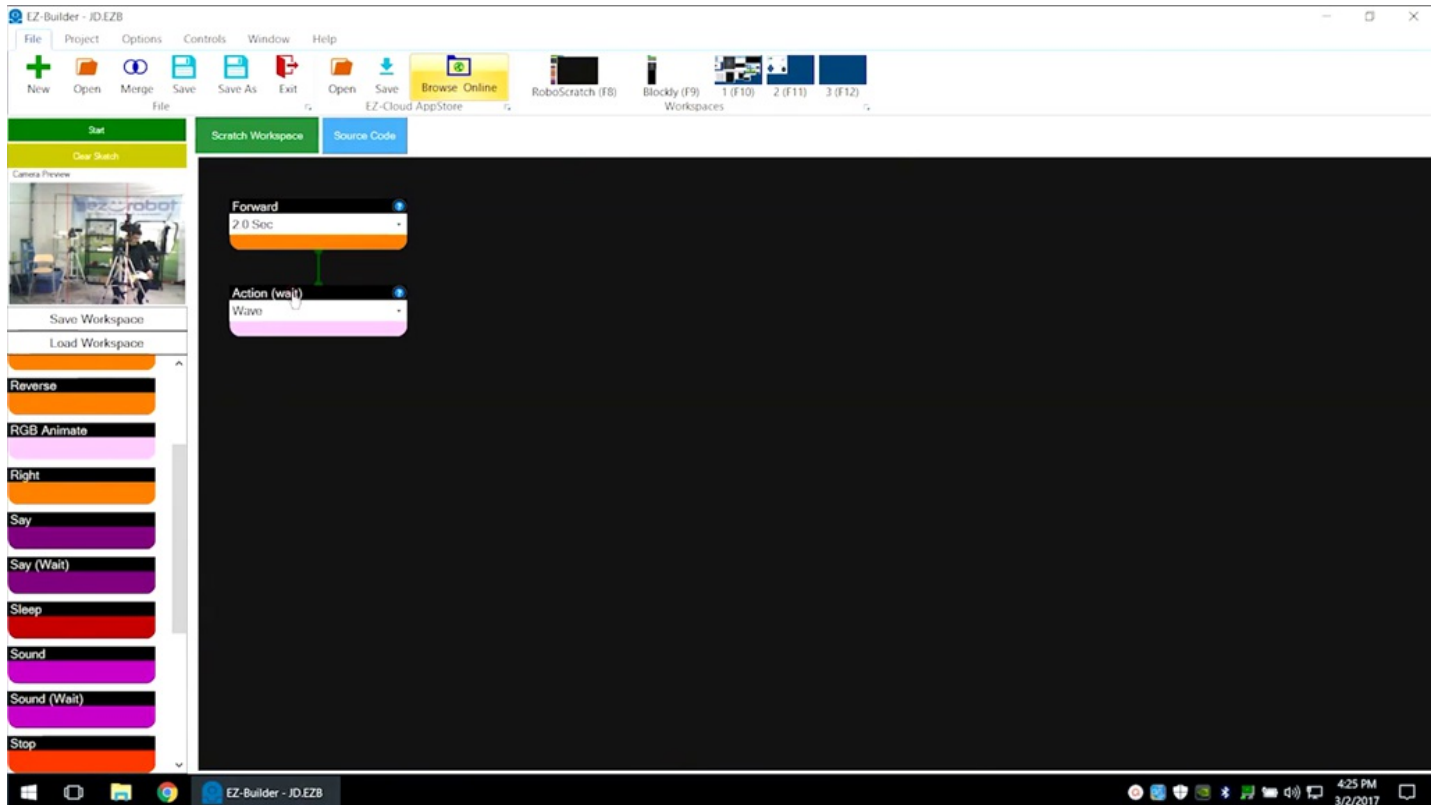
RoboScratch Commands

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



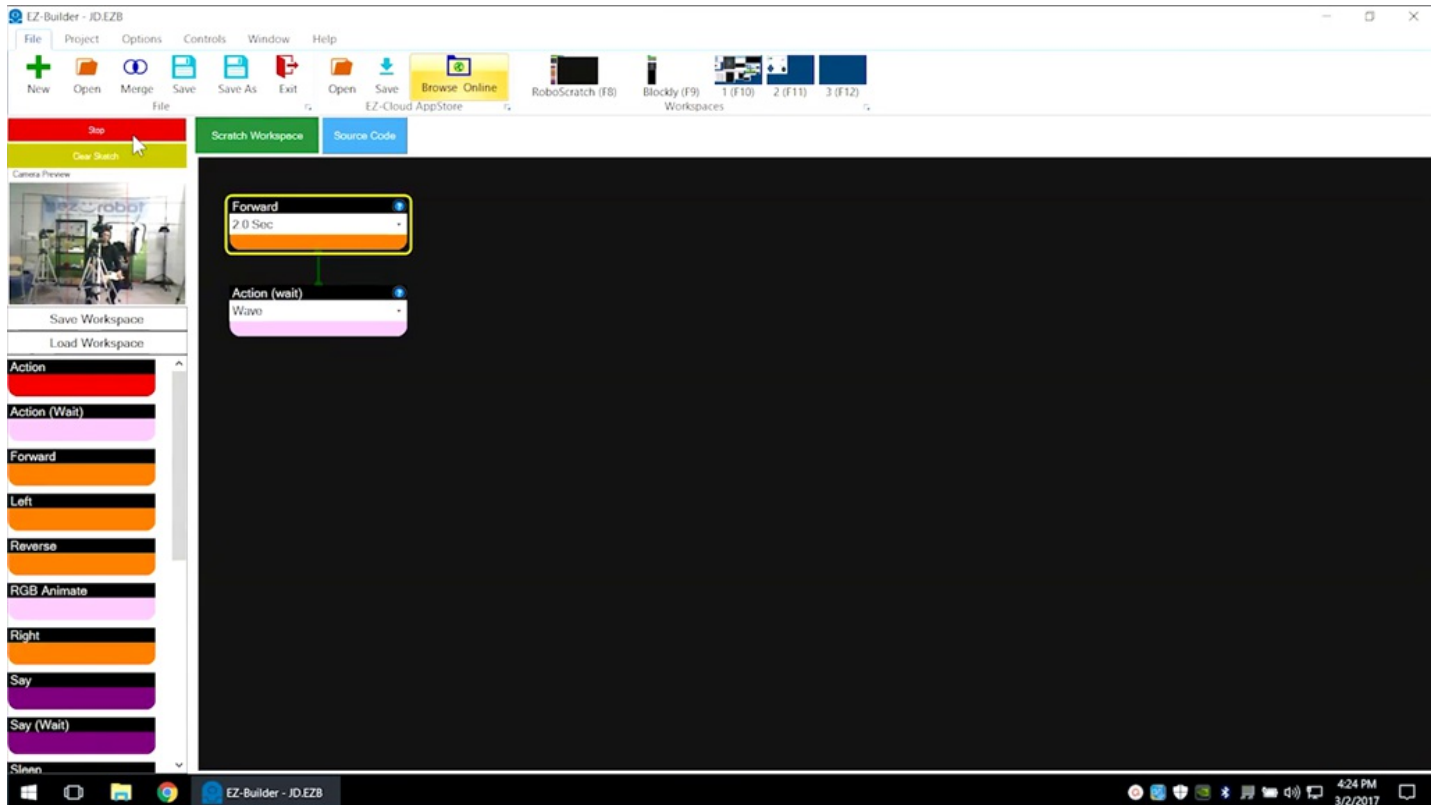
Using Wait

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



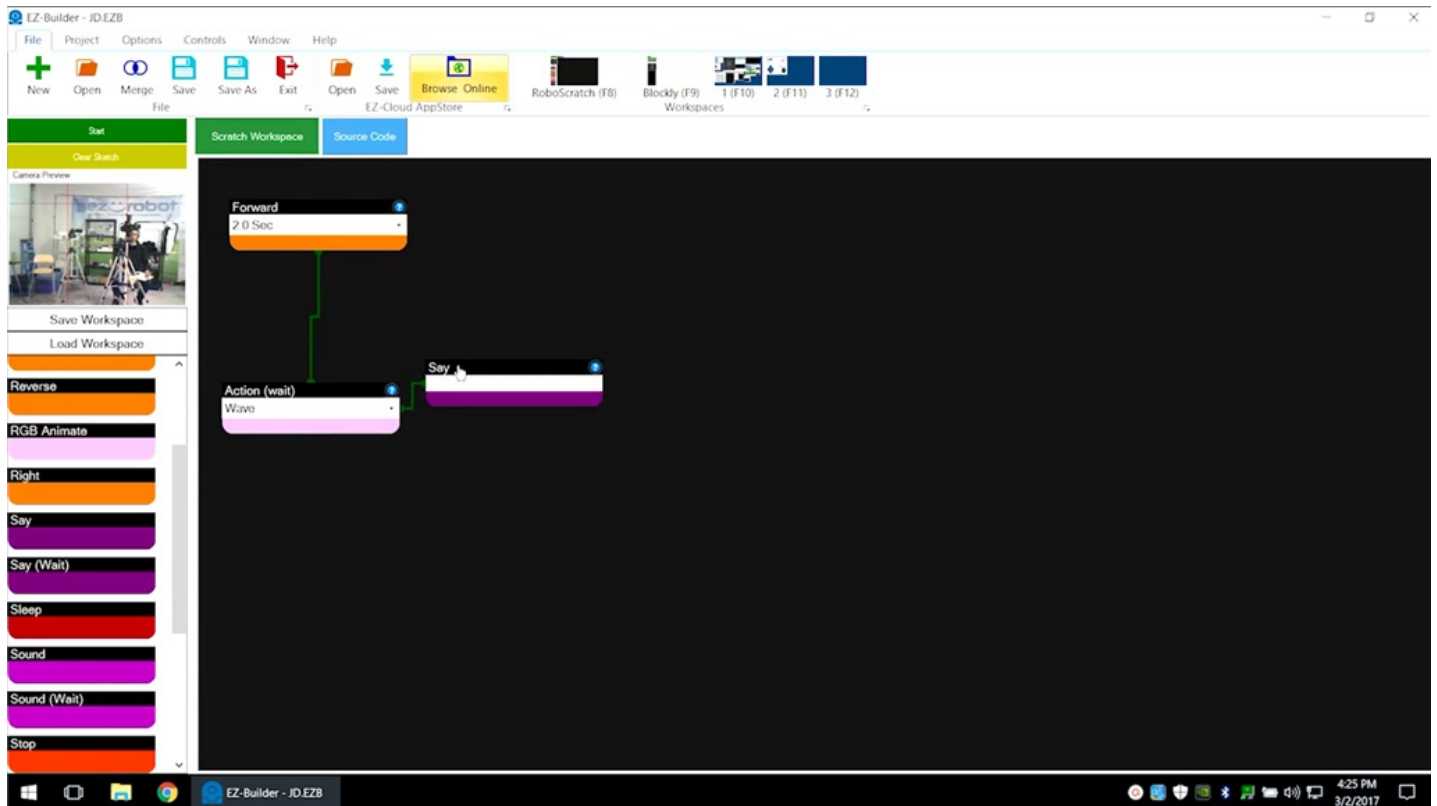
RoboScratch Program Execution

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



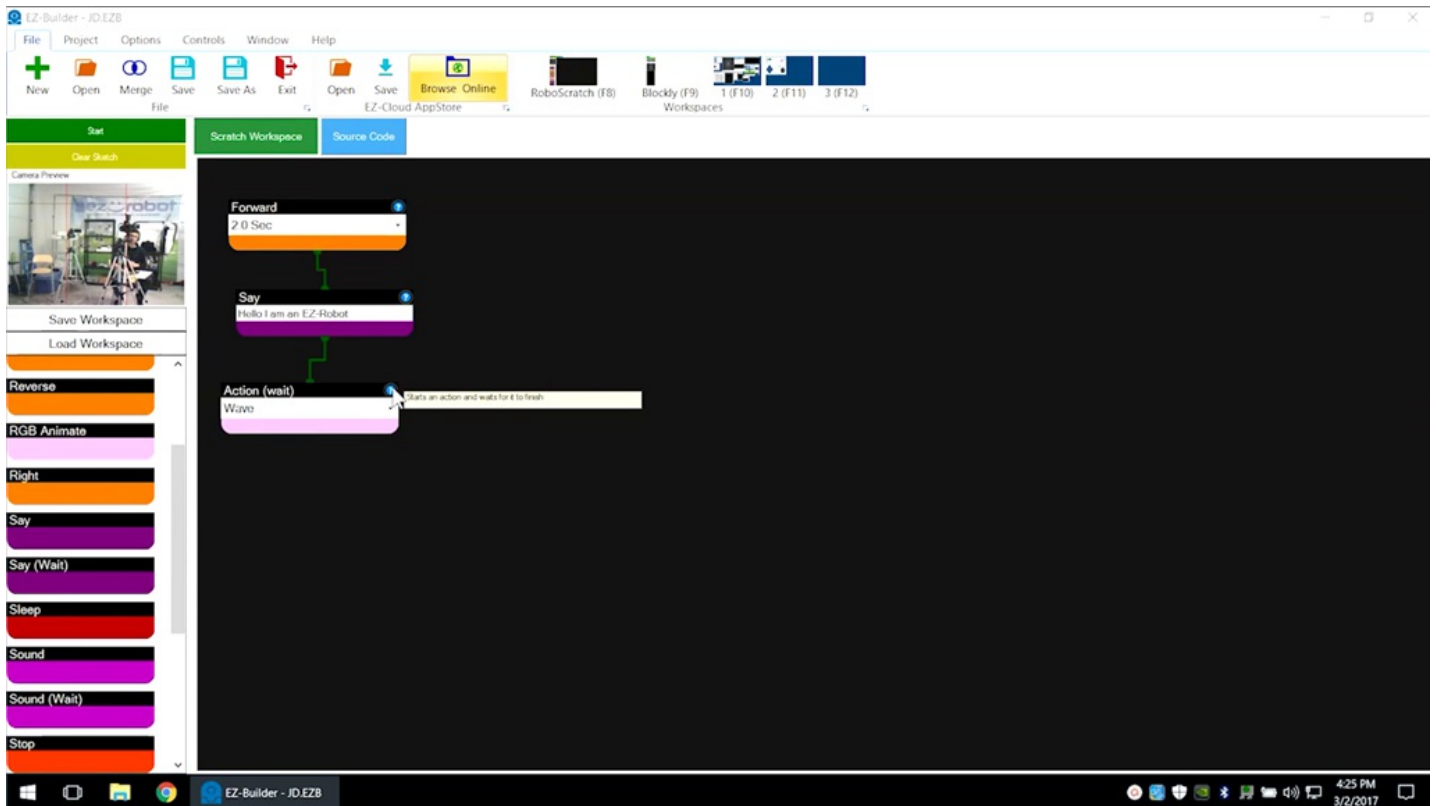
RoboScratch Program Flow

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



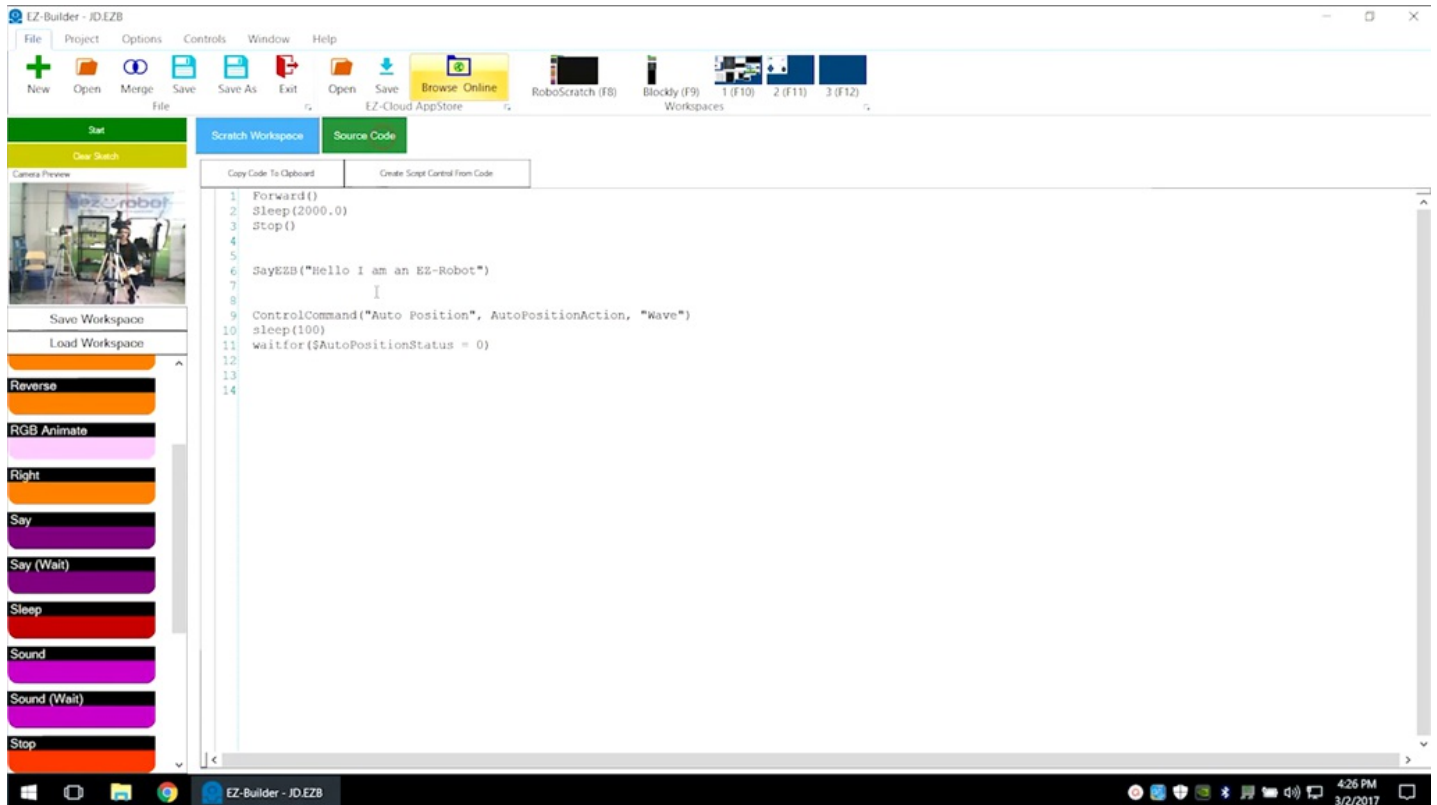
Blue Question Marks

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



RoboScratch Source Code

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



The screenshot displays the EZ-Builder software interface. The main window is titled "EZ-Builder - JD.EZB" and features a menu bar with "File", "Project", "Options", "Controls", "Window", and "Help". Below the menu bar is a toolbar with icons for "New", "Open", "Merge", "Save", "Save As", "Exit", "Open", "Save", "Browse Online", "EZ-Cloud AppStore", "RoboScratch (F8)", "Blockly (F9)", and three workspace icons labeled "1 (F10)", "2 (F11)", and "3 (F12)".

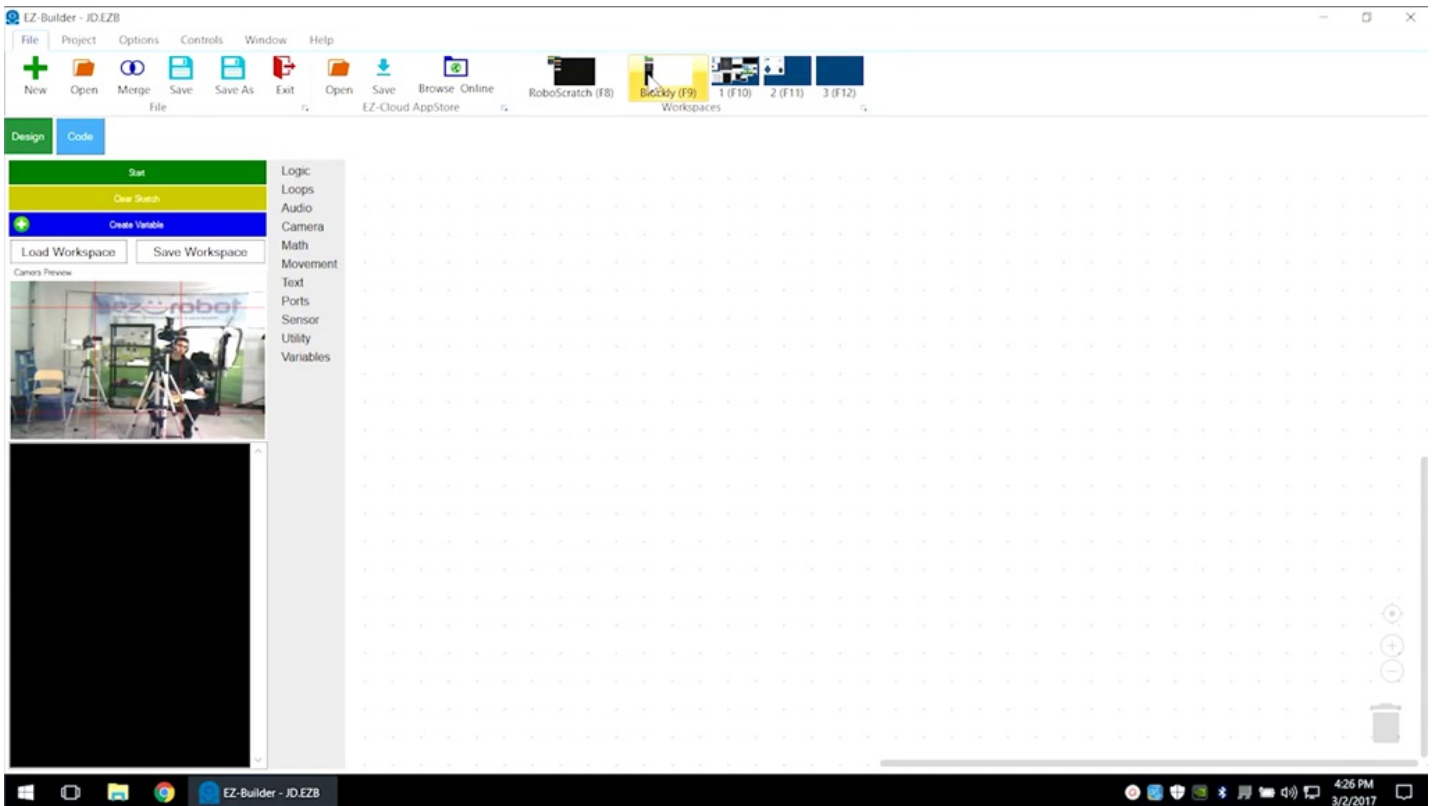
The interface is divided into several sections:

- Left Panel:** Contains a "Camera Preview" window showing a robot in a room. Below it are "Save Workspace" and "Load Workspace" buttons. A vertical list of action buttons includes "Reverse", "RGB Animate", "Right", "Say", "Say (Wait)", "Sleep", "Sound", "Sound (Wait)", and "Stop".
- Top Panel:** Includes "Scratch Workspaces" and "Source Code" tabs.
- Code Editor:** The "Source Code" tab is active, showing a script with the following code:

```
1 Forward()
2 Sleep(2000,0)
3 Stop()
4
5
6 SayEZB("Hello I am an EZ-Robot")
7
8
9 ControlCommand("Auto Position", AutoPositionAction, "Wave")
10 sleep(100)
11 waitfor($AutoPositionStatus = 0)
12
13
14
```

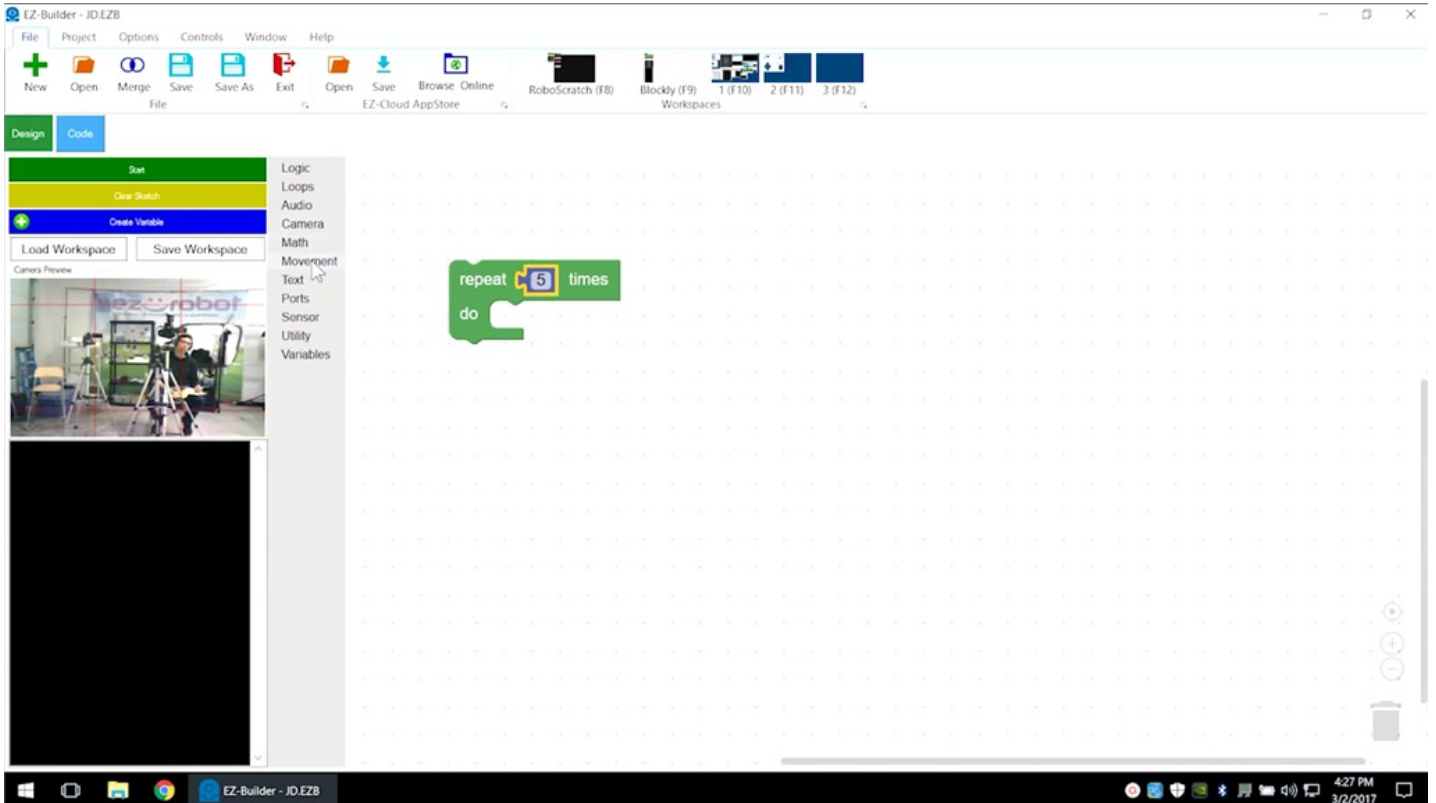
The Windows taskbar at the bottom shows the system tray with the time "4:26 PM" and date "3/2/2017".

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

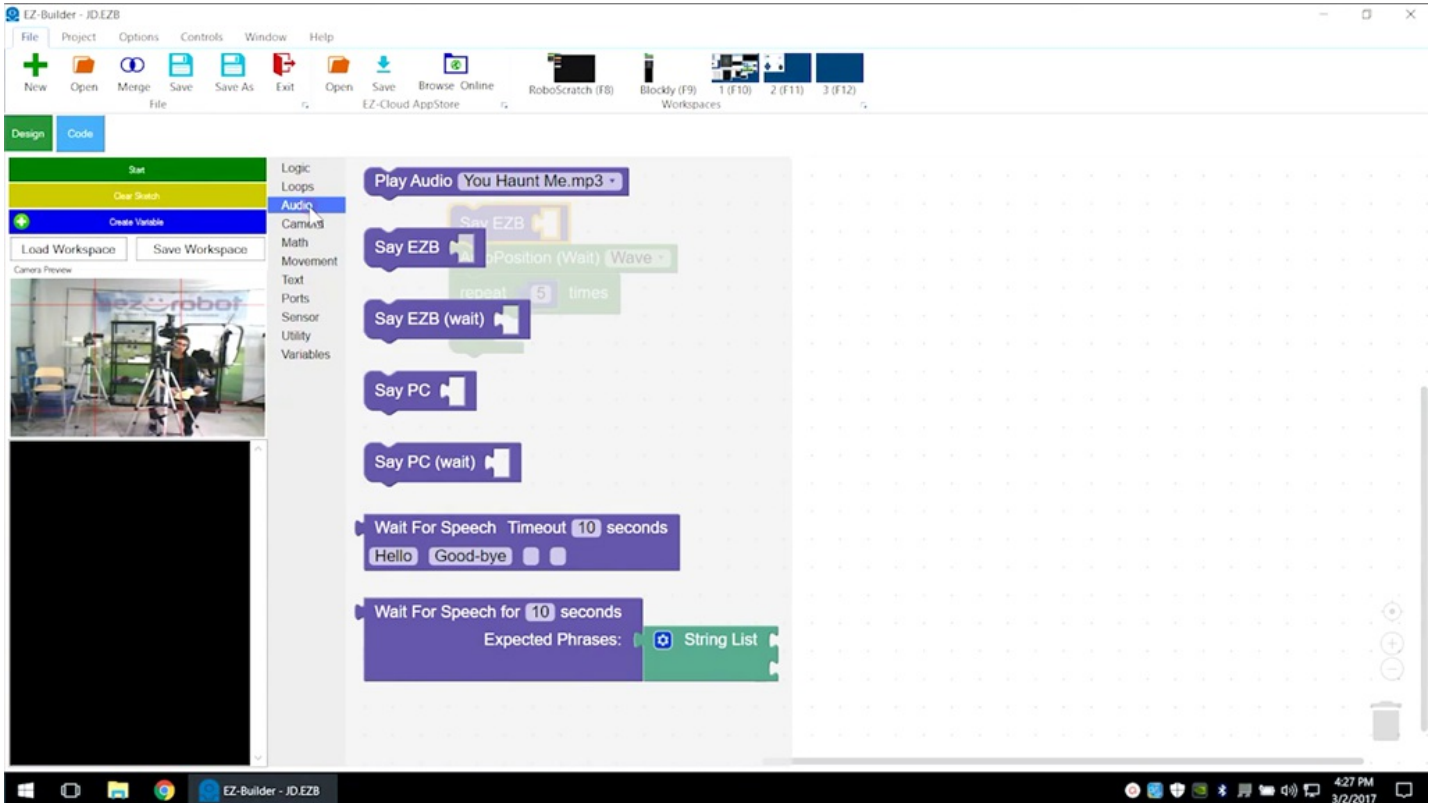


Blockly Commands

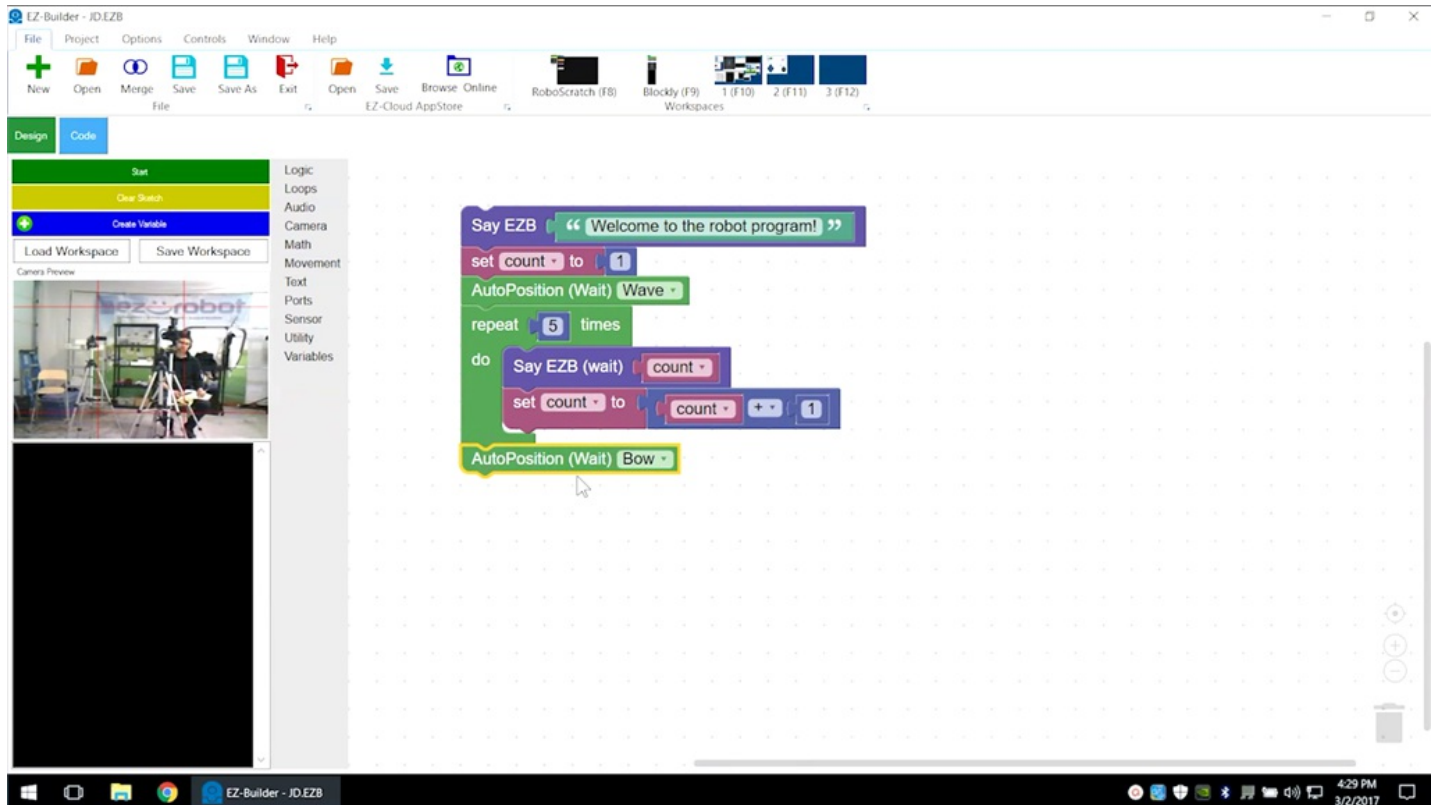
Click on commands, drag into position, and edit the parameters as desired.



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.



Blockly Execution

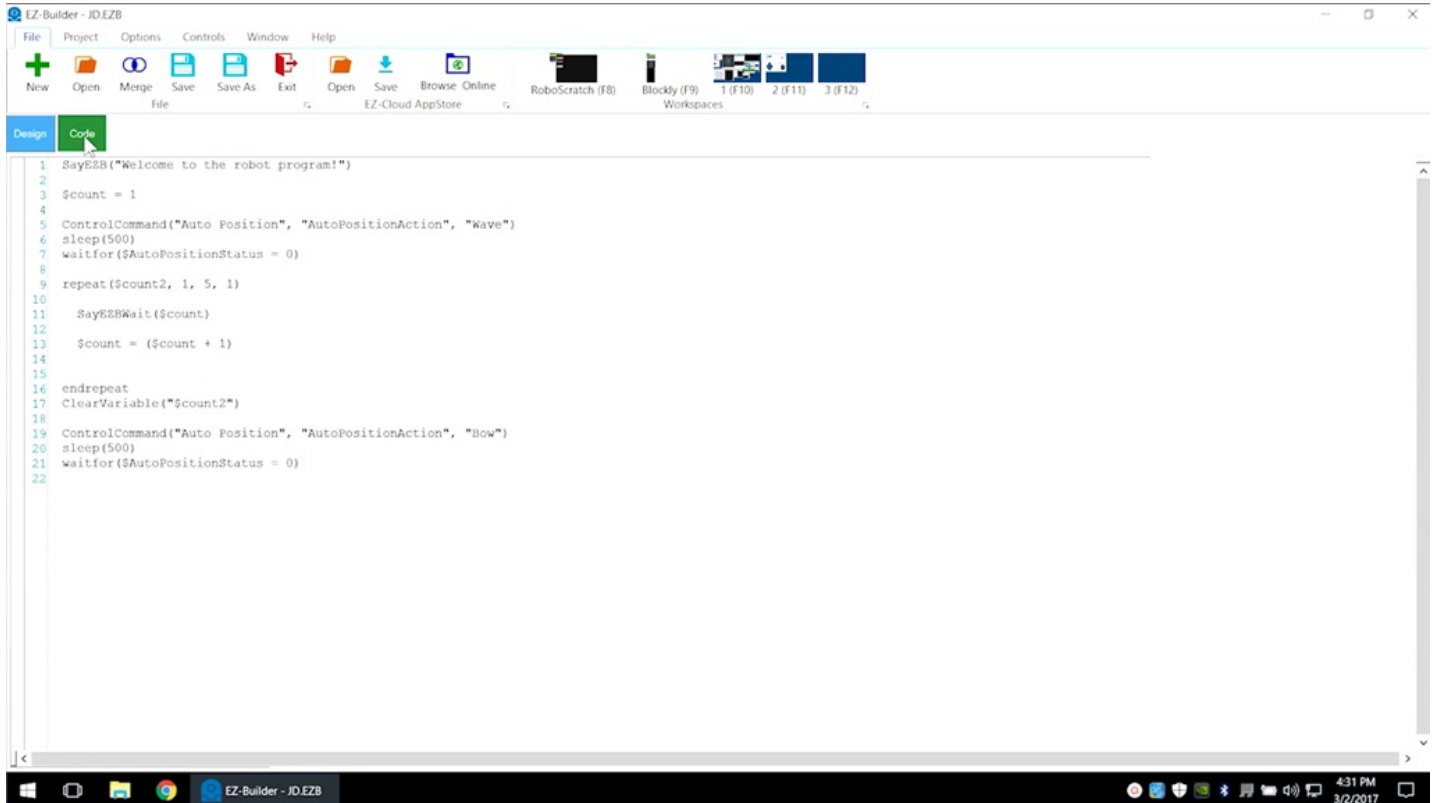
Click on the green **Start** button to execute the program.

The screenshot displays the EZ-Builder software interface. The top menu bar includes File, Project, Options, Controls, Window, and Help. Below the menu is a toolbar with icons for New, Open, Merge, Save, Save As, Exit, Open, Save, Browse Online, and EZ-Cloud AppStore. The main workspace is divided into Design and Code tabs. The Code tab is active, showing a Blockly script on a grid background. The script consists of the following blocks:

- Say EZB "Welcome to the robot program!"
- set count to 1
- AutoPosition (Wait) Wave
- repeat 5 times
- do loop containing:
 - Say EZB (wait) count
 - set count to count + 1
- AutoPosition (Wait) Bow

The left sidebar contains a list of categories: Logic, Loops, Audio, Camera, Math, Movement, Text, Ports, Sensor, Utility, and Variables. Below the sidebar is a camera preview window showing a robot in a room. The bottom status bar shows the time as 4:30 PM on 3/2/2017.

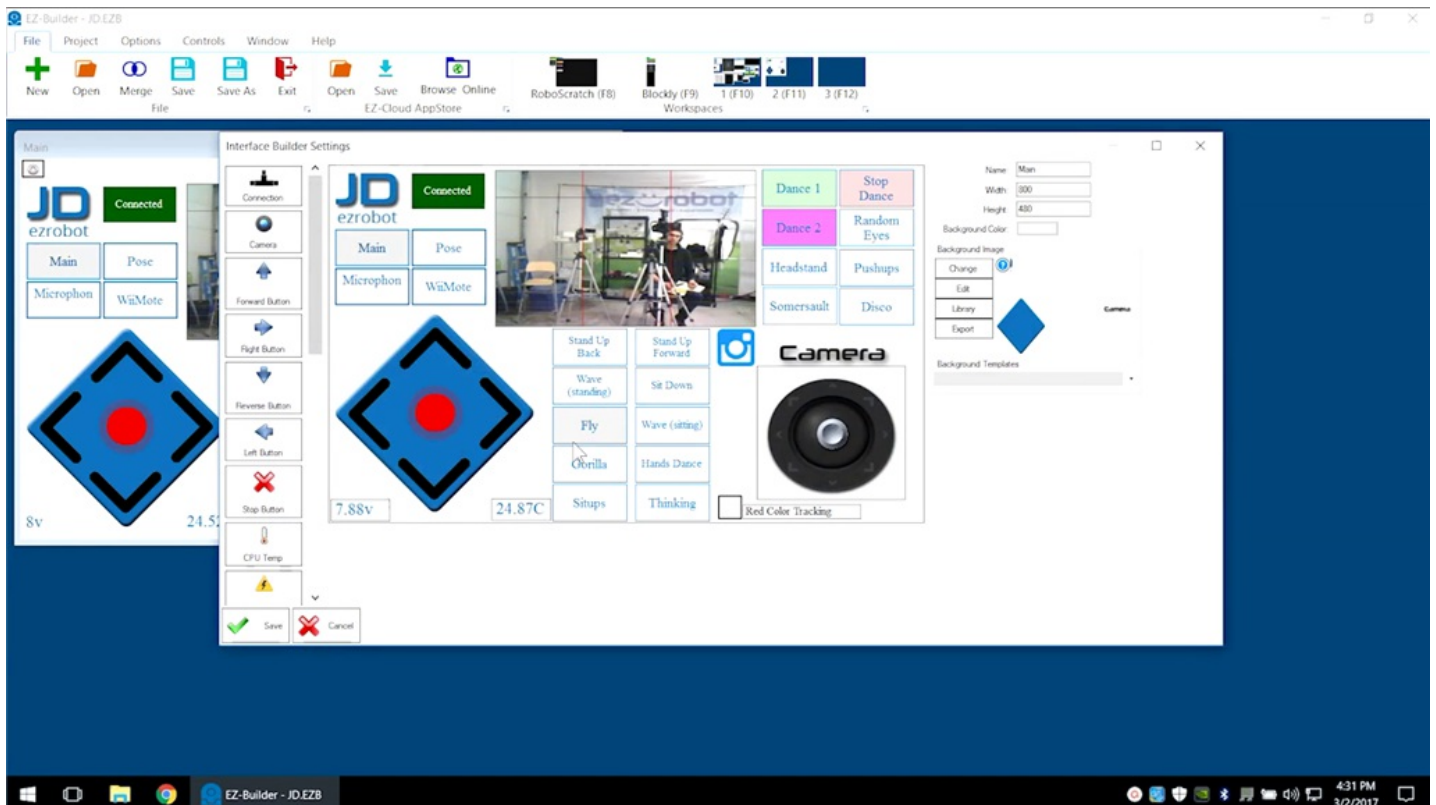
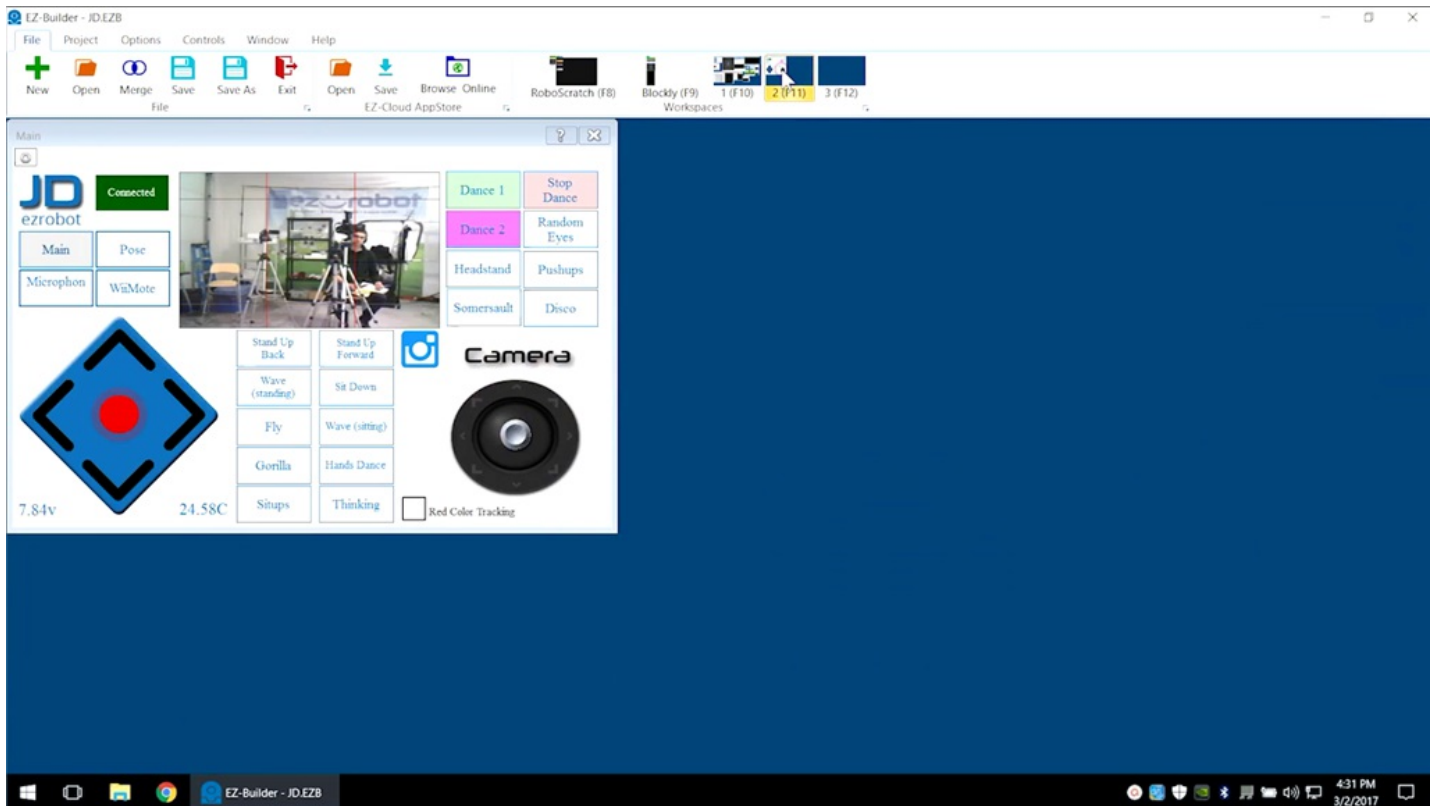
Click on the **Code** button to view the generated **EZ-Script** code.



```
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
```

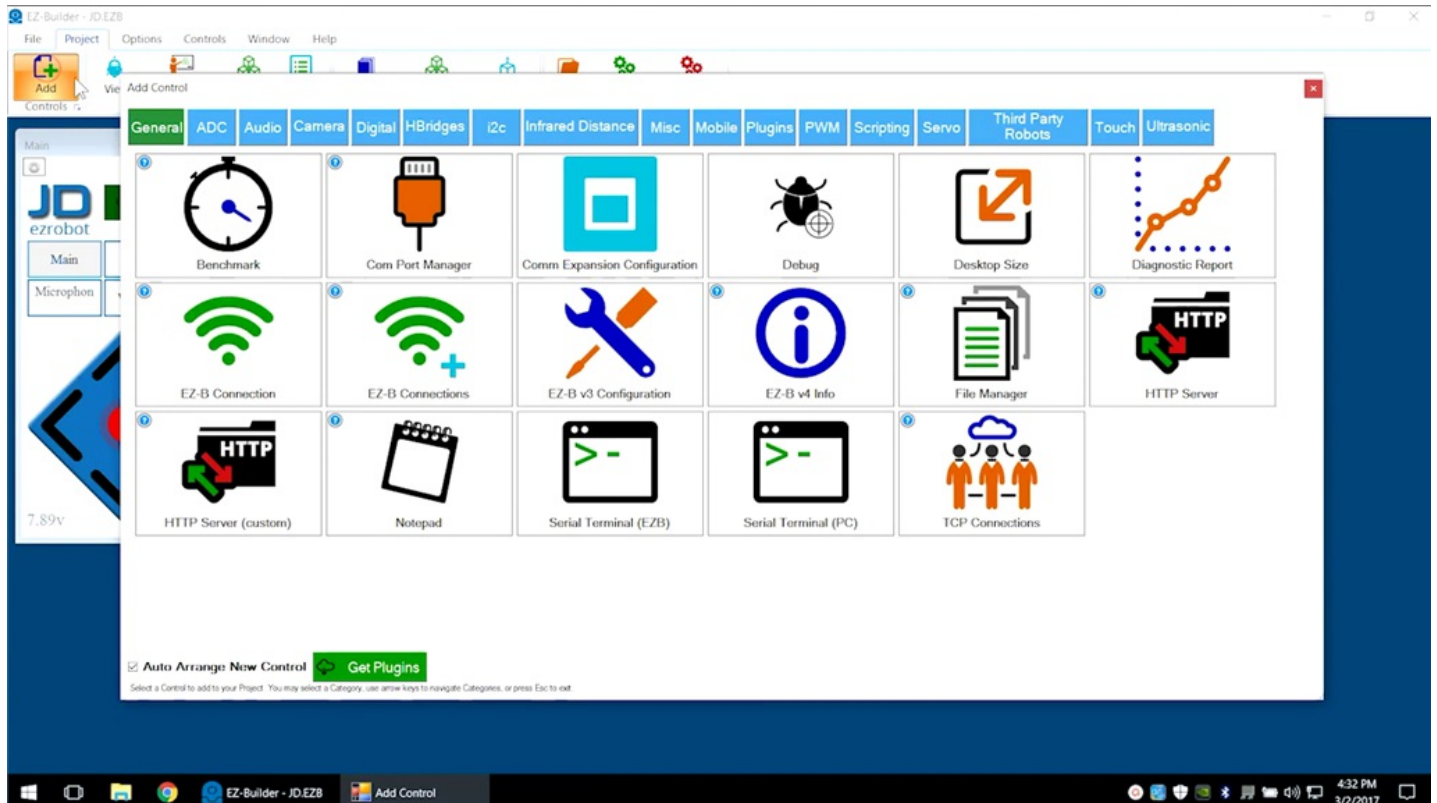

Additional Workspaces

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.



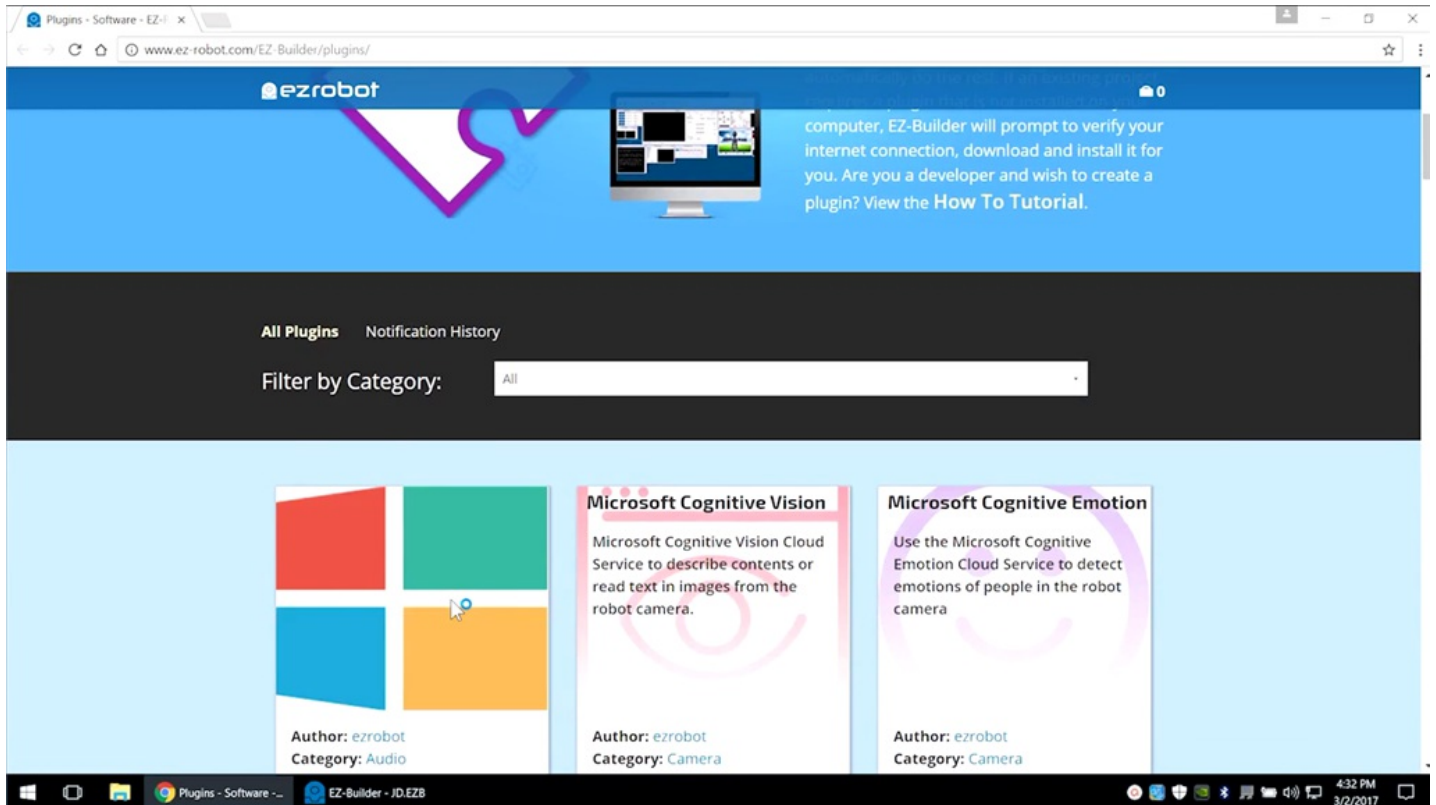
Adding Controls

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



⑤ Third Party Plugins

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



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 area 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 open under the `say` command, showing the following options:

- Say("text to speak")
- SayZB("text to speak")
- SayZBWait("text to speak")
- SayZB("text to speak")

The right-hand sidebar features a 'Script Help' panel titled 'EZ-Script Functions'. It lists several functions with their parameters and examples:

- 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 bottom of the window shows the Windows taskbar with the application name 'EZ-Builder - JD.EZB' and the system clock displaying '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-Builder software interface. On the left, a script editor contains 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
10
11
```

A right-click context menu is open over the selected line of code, listing various controls. The 'Auto Position' control is highlighted. The 'Cheat Sheet' window on the right displays the following information:

- GetDigital(Port)**
 - Returns the Digital value of the specified port as a 0 or 1
 - Example: $\$x = \text{GetDigital}(d0)$
- ASin(value)**
 - Returns the math Asin() function (also called ArcSin)
 - Example: $\$x = \text{ASin}(27)$
- ACos(value)**
 - Returns the math ACos() function (also called ArcCos)
 - Example: $\$x = \text{ACos}(27)$
- Sqrt(value)**
 - Returns the math Square Root function
 - Example: $\$x = \text{Sqrt}(9)$
- Map(input, inputMin, inputMax, containerMin, containerMax)**
 - Returns a scaled value of input between containerMin and containerMax. Specify the known input's minimum and maximum values, and the known container (output) values. For example, if you were to scale a value between -1 and 1 to 0 and 180, enter this, which returns 135.
 - Example: $\$x = \text{Map}(0.5, -1, 1, 0, 180)$
- Power(value, power)**
 - Returns the math Power() function
 - First parameter is the input value
 - The second parameter is the power
 - Example: $\$x = \text{Power}(2, 4)$
- Sin(value)**
 - Returns the math SIN() function
 - Example: $\$x = \text{Sin}(27)$
- Cos(value)**
 - Returns the math COS() function
 - Example: $\$x = \text{Cos}(27)$
- Abs(value)**
 - Returns the absolute value of a number

The screenshot shows the EZ-Builder software interface with the 'Cheat Sheet' window open. The script editor on the left contains 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 ControlCommand("Camera", CameraFaceTrackingEnable)
10
11
```

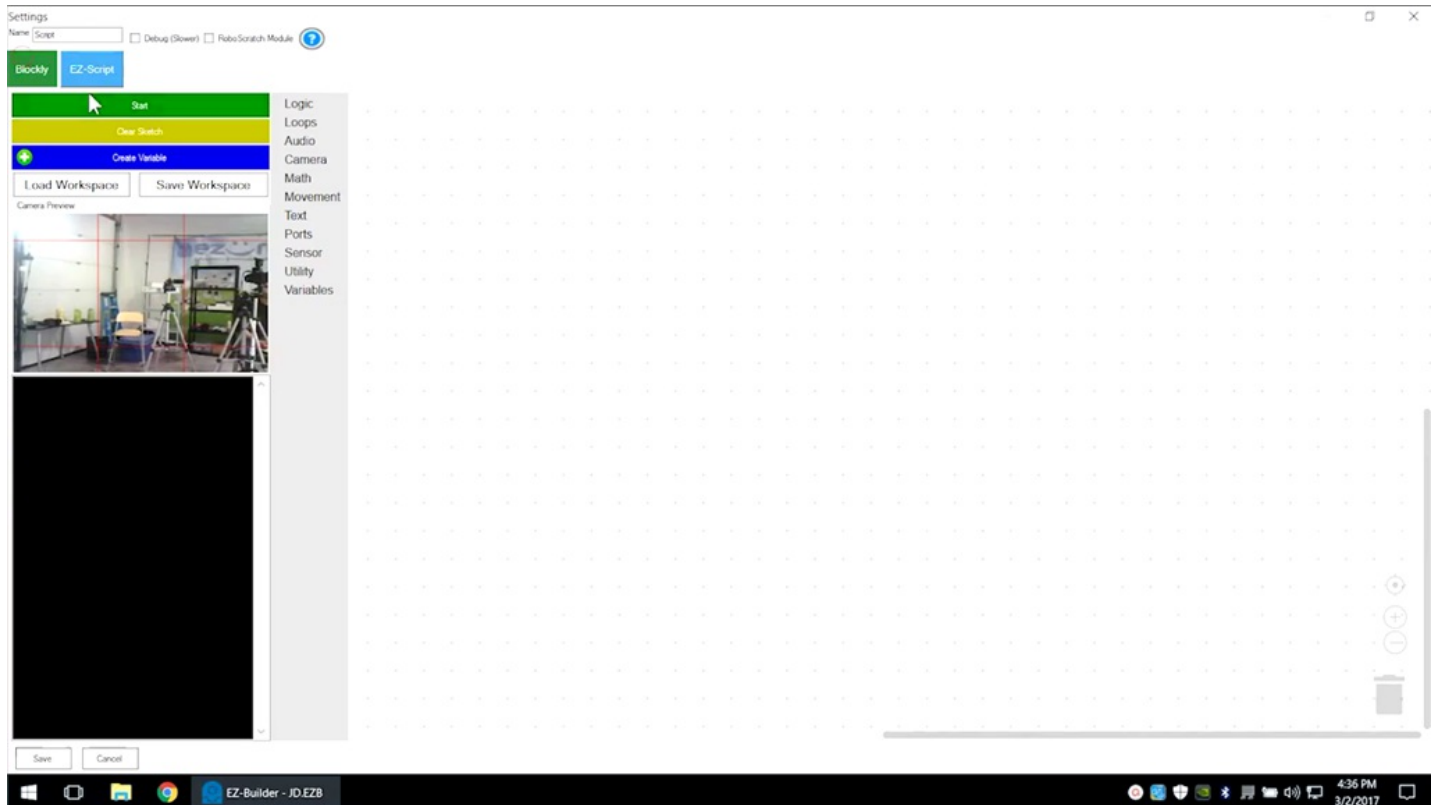
The 'Cheat Sheet' window displays the following information:

Use this list to display all available ControlCommand() functions for each control. You may click on an item to have it add automatically to your Multiline EZ-Script editor.

- Show Controls**
 - ShowDesktop(1)
 - ShowDesktop(2)
 - ShowDesktop(3)
 - ShowControl("RoboScratch")
 - ShowControl("Auto Position")
 - ShowControl("Camera")
 - ShowControl("Connection")
 - ShowControl("EZ-Robot")
 - ShowControl("Init")
 - ShowControl("Main")
 - ShowControl("Microphone")
 - ShowControl("Notepad")
 - ShowControl("PointAndTrack")
 - ShowControl("RGB Animator")
 - ShowControl("Script")
 - ShowControl("Soundboard v4")
 - ShowControl("Wii Remote")
- Auto Position**
 - 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, "Happy Hands")
 - ControlCommand("Auto Position", AutoPositionAction, "Head Bob")
 - ControlCommand("Auto Position", AutoPositionAction, "Headstand")
 - ControlCommand("Auto Position", AutoPositionAction, "Jump Jack")
 - ControlCommand("Auto Position", AutoPositionAction, "Kick")
 - ControlCommand("Auto Position", AutoPositionAction, "Left")
 - ControlCommand("Auto Position", AutoPositionAction, "Lunge Singing")
 - ControlCommand("Auto Position", AutoPositionAction, "Pass the Mic")
 - ControlCommand("Auto Position", AutoPositionAction, "Point")
 - ControlCommand("Auto Position", AutoPositionAction, "Pretance")
 - ControlCommand("Auto Position", AutoPositionAction, "Pushups")
 - 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, "Situps")
 - ControlCommand("Auto Position", AutoPositionAction, "Splits")
 - ControlCommand("Auto Position", AutoPositionAction, "Stand From Sit")
 - ControlCommand("Auto Position", AutoPositionAction, "Stop")
 - ControlCommand("Auto Position", AutoPositionAction, "Summersault")

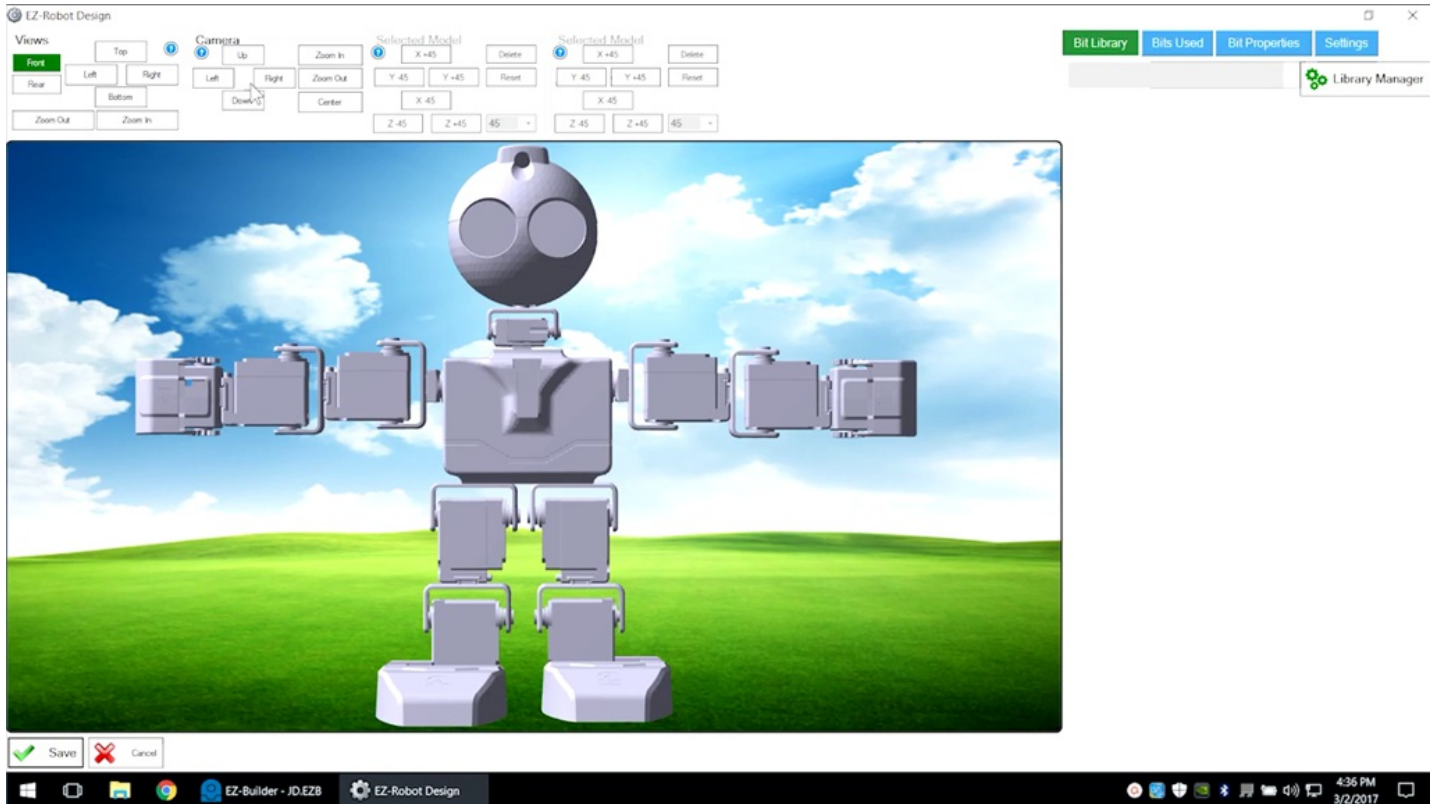
EZ-Script and Blockly

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



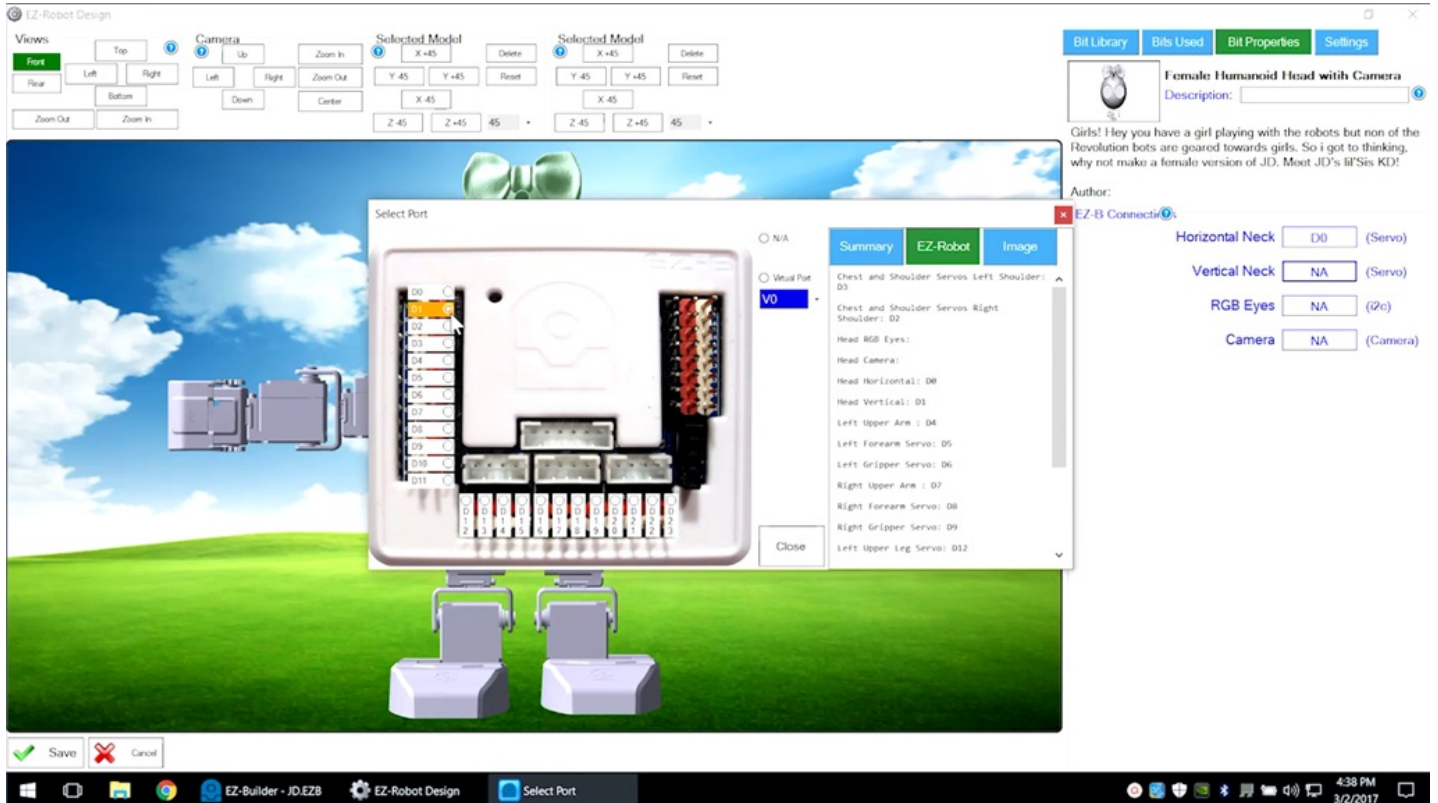
Design Mode

Enter **Design Mode** by selecting **Project -> Design**. Design options are available through the **Bit Library**.



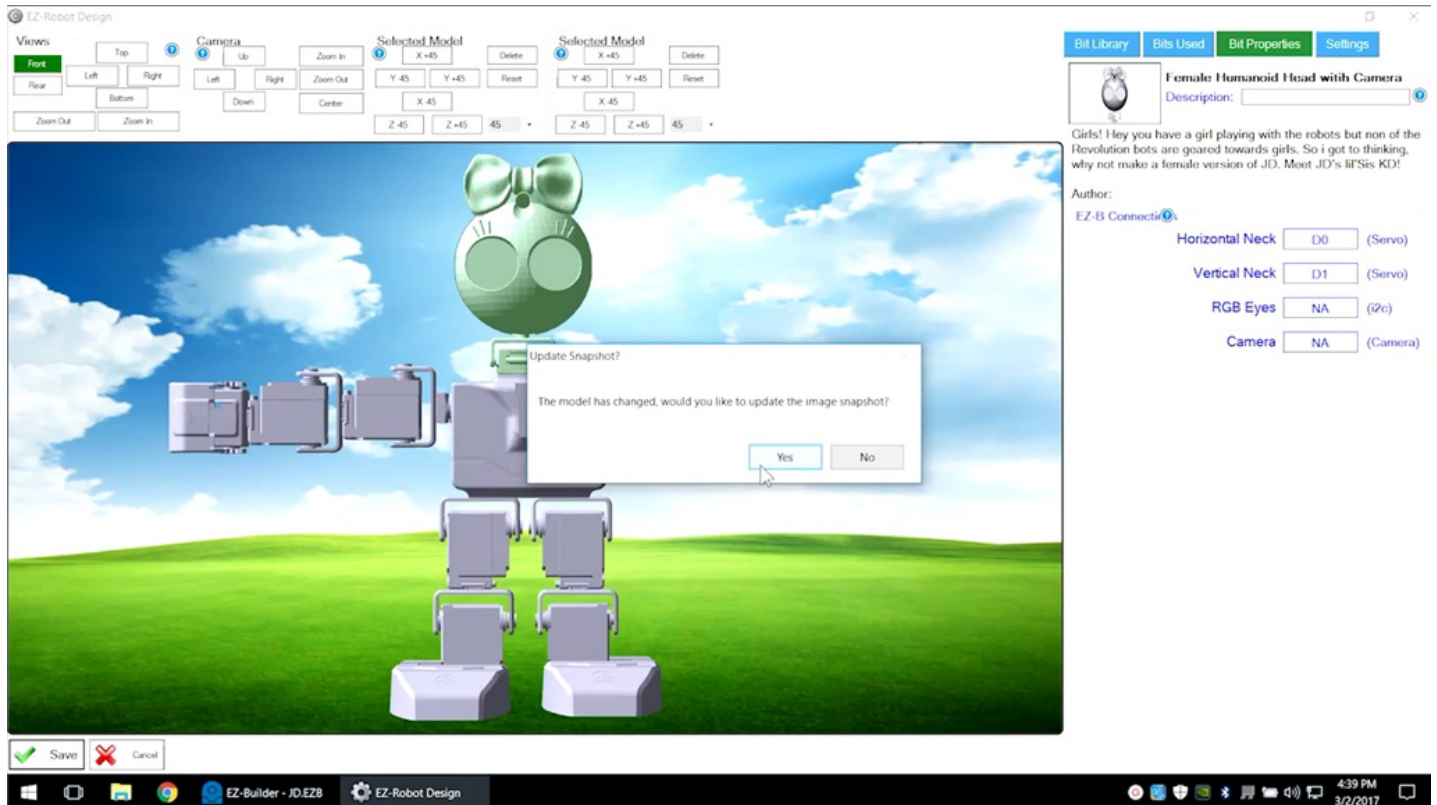
Connecting Custom Components

Choose which ports will be used for connecting custom components.



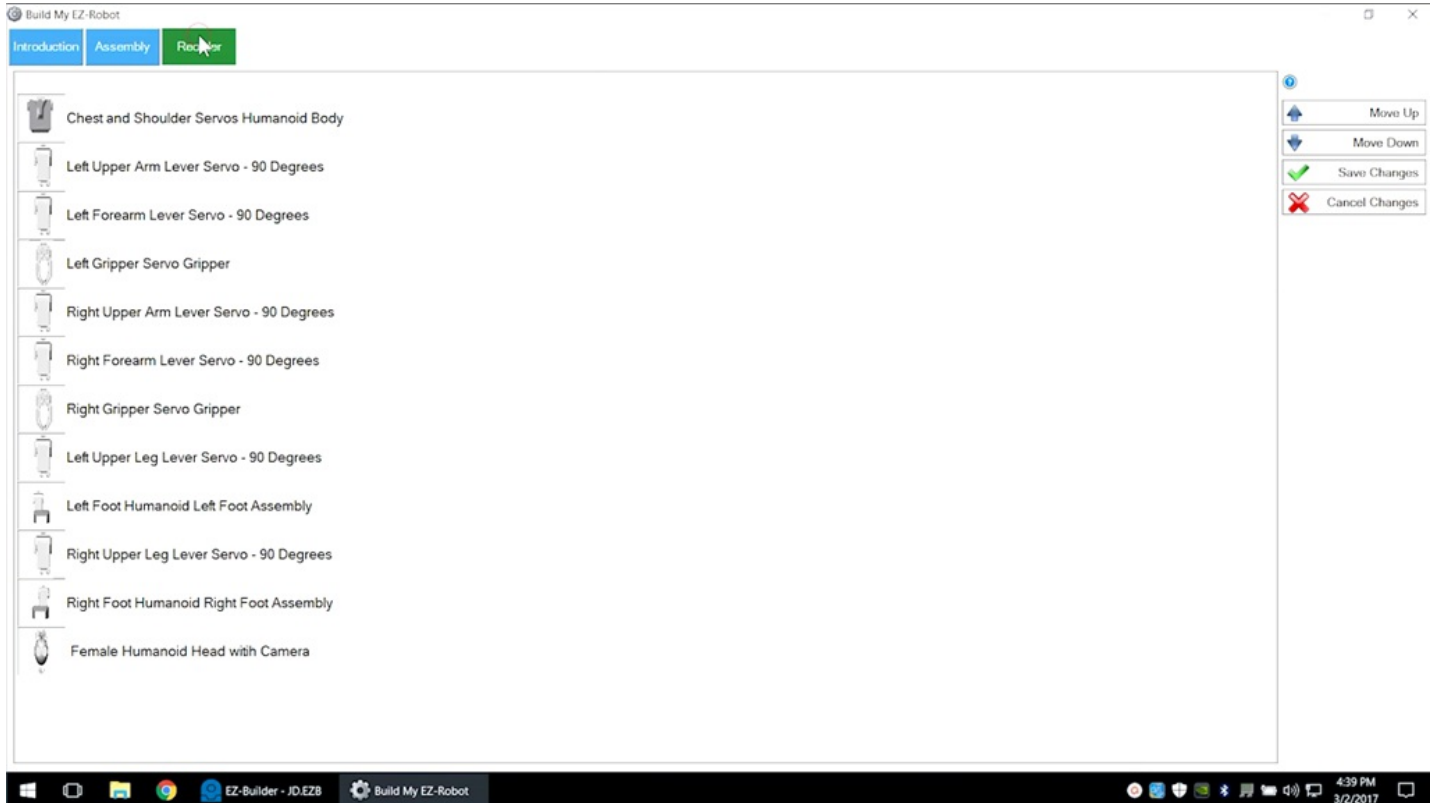
Updating Build Instructions

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.



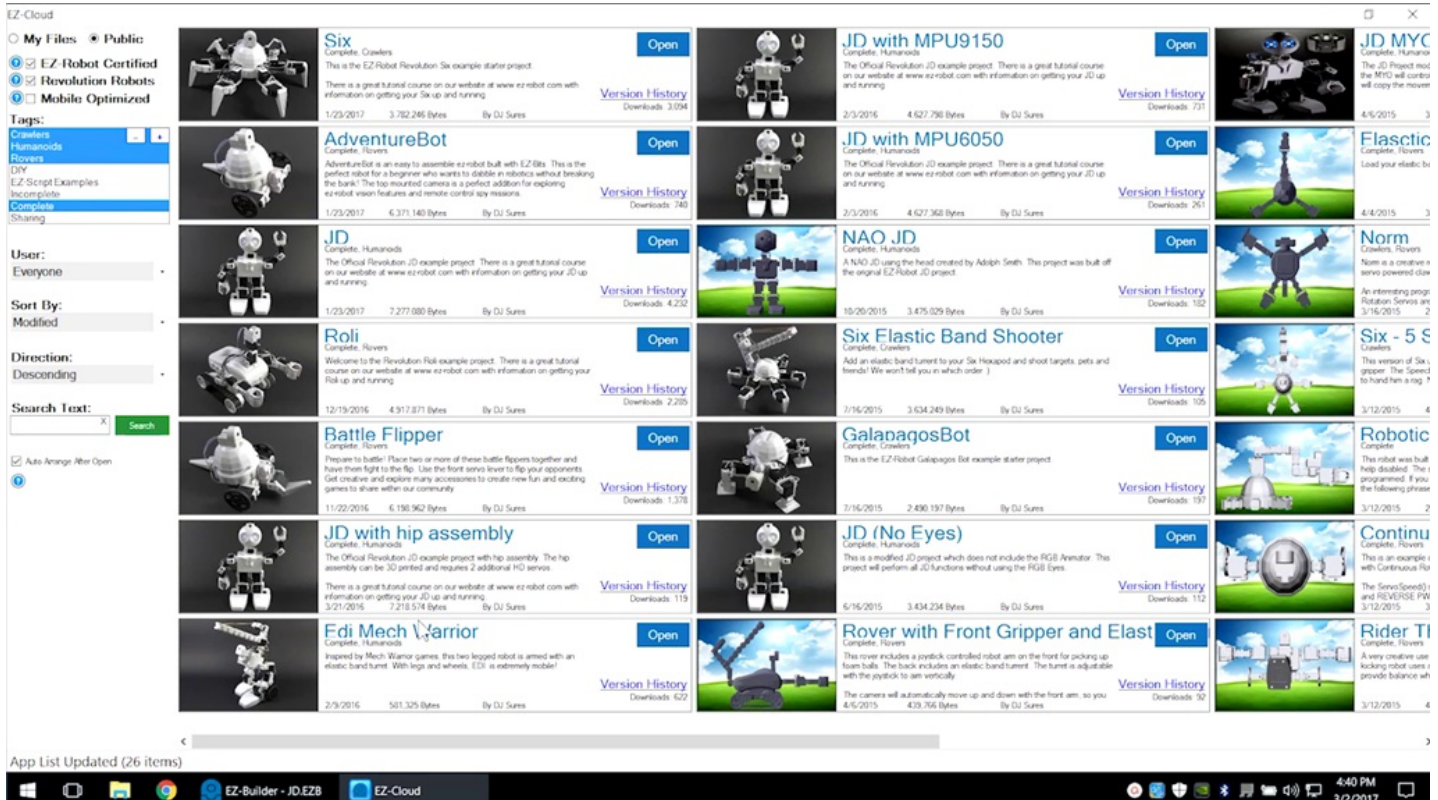
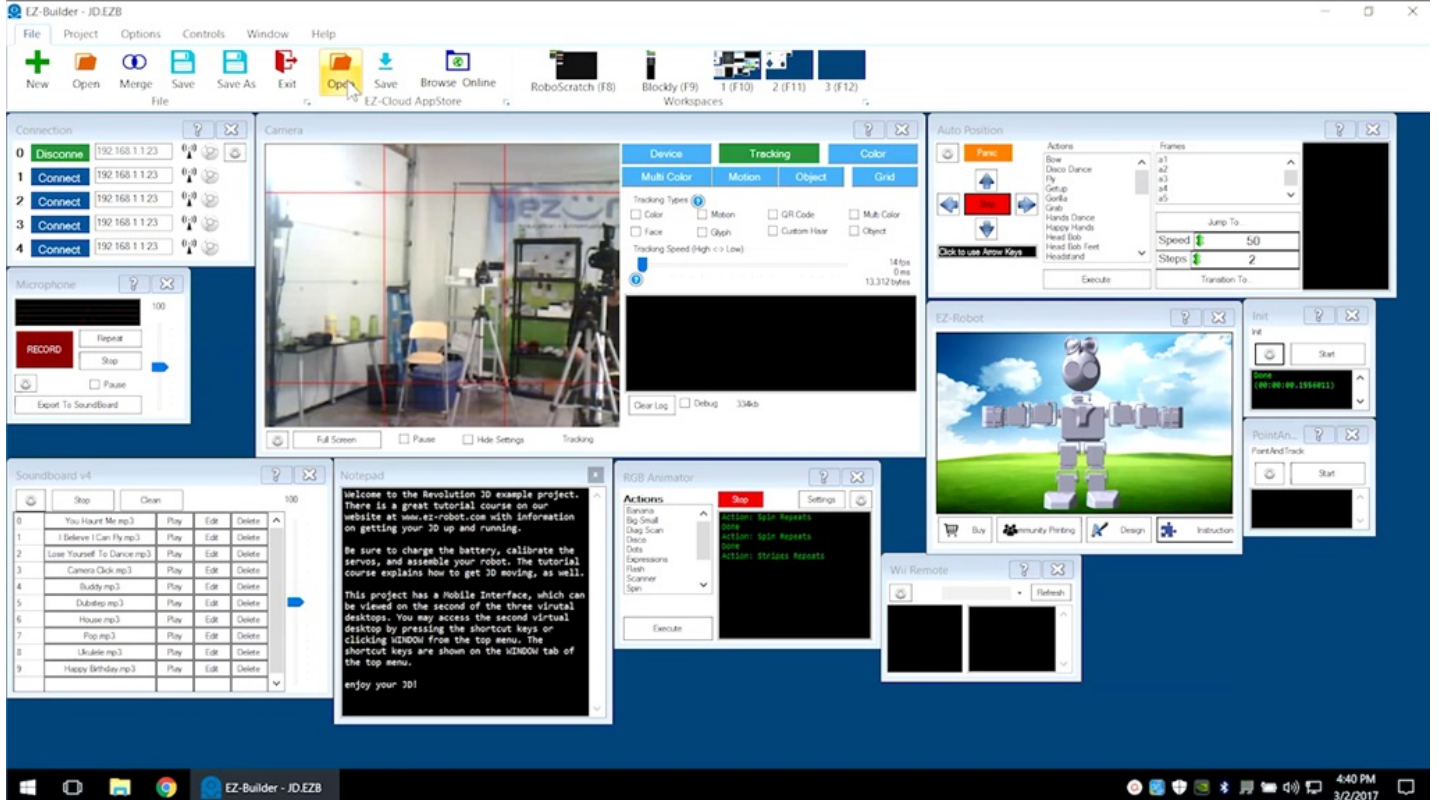
The screenshot shows the 'Build My EZ-Robot' application window. At the top, there are three tabs: 'Introduction', 'Assembly', and 'Reorder', with 'Reorder' being the active tab. The main area contains a list of build steps, each with a small icon and a text label:

- 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

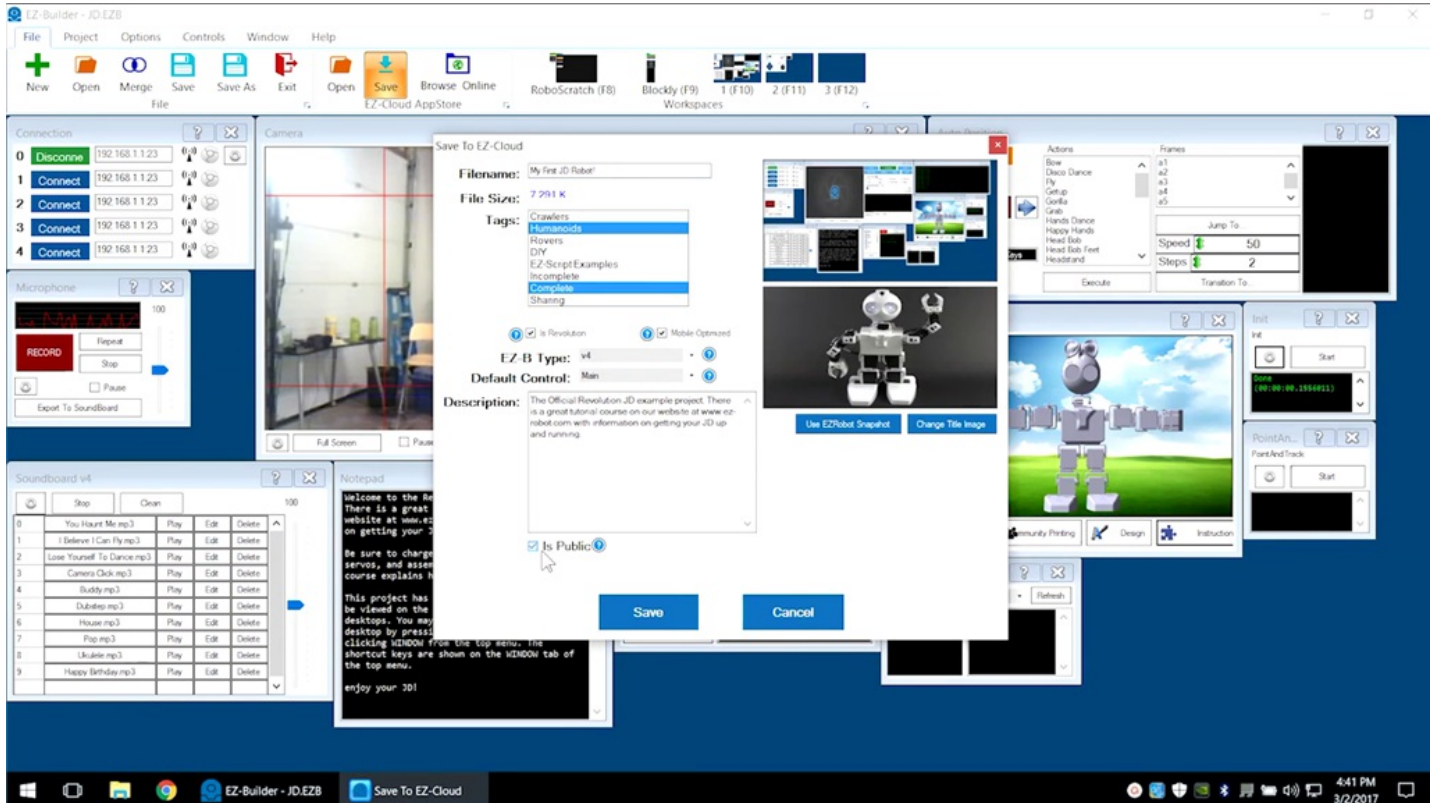
On the right side of the list, a context menu is open, showing four options: 'Move Up' (with an up arrow icon), 'Move Down' (with a down arrow icon), 'Save Changes' (with a green checkmark icon), and 'Cancel Changes' (with a red X icon).

The Windows taskbar at the bottom shows the current time as 4:39 PM on 3/2/2017. The taskbar includes icons for various applications, including 'EZ-Builder - JD.EZB' and 'Build My EZ-Robot'.

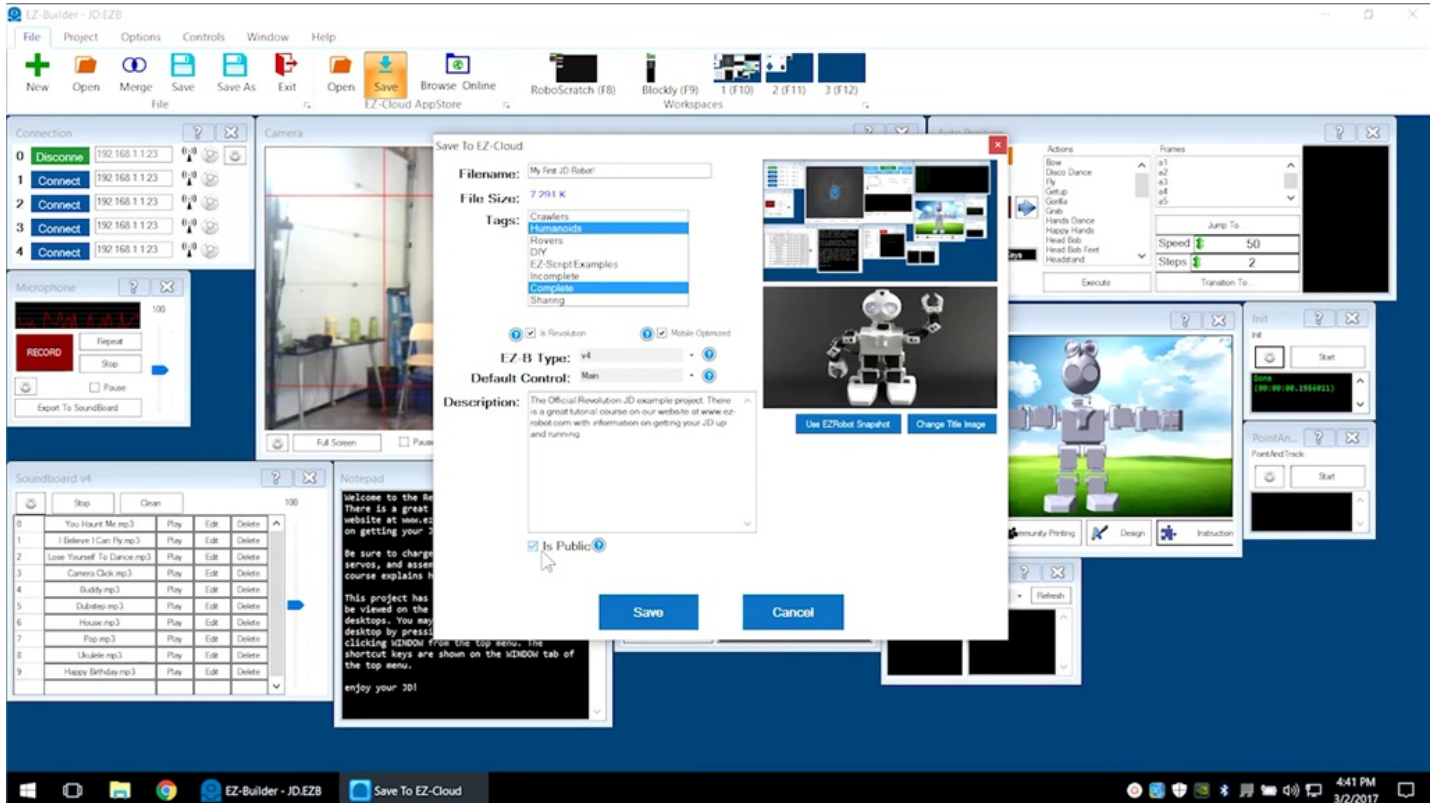
Access the **EZ-Cloud** through **File -> Open** to save and share programs.



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




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



Ⓢ Load Revisions

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

JD



JD

The Official Revolution 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; font-weight: bold; font-size: 8px;">Load</div> <p style="font-size: 8px; margin: 0;">1/23/2017 2:44:38 PM 7,277,080 Bytes</p> <p style="font-size: 8px; 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; font-weight: bold; font-size: 8px;">Load</div> <p style="font-size: 8px; margin: 0;">10/31/2016 6:00:42 PM 7,209,549 Bytes</p> <p style="font-size: 8px; margin: 0;">Change Log Disable all tracking types when the checkbox is disabled</p>	<div style="background-color: #007bff; color: white; padding: 2px; font-weight: bold; font-size: 8px;">Load</div> <p style="font-size: 8px; margin: 0;">2/7/2016 1:44:53 AM 7,198,023 Bytes</p> <p style="font-size: 8px; margin: 0;">Change Log addod rgb eye animation to happy birthday dance</p>	<div style="background-color: #007bff; color: white; padding: 2px; font-weight: bold; font-size: 8px;">Load</div> <p style="font-size: 8px; margin: 0;">2/3/2016 11:09:22 AM 4,621,855 Bytes</p> <p style="font-size: 8px; margin: 0;">Change Log increased i2c speed to 300,000 from 100,000</p>
<div style="background-color: #007bff; color: white; padding: 2px; font-weight: bold; font-size: 8px;">Load</div> <p style="font-size: 8px; margin: 0;">11/7/2016 12:42:07 PM 7,246,241 Bytes</p> <p style="font-size: 8px; margin: 0;">Change Log camera control uses latest analog joystick mobile control</p>	<div style="background-color: #007bff; color: white; padding: 2px; font-weight: bold; font-size: 8px;">Load</div> <p style="font-size: 8px; margin: 0;">10/30/2016 11:44:07 PM 7,209,554 Bytes</p> <p style="font-size: 8px; margin: 0;">Change Log made color tracking checkbox larger</p>	<div style="background-color: #007bff; color: white; padding: 2px; font-weight: bold; font-size: 8px;">Load</div> <p style="font-size: 8px; margin: 0;">2/6/2016 10:41:22 PM 7,200,541 Bytes</p> <p style="font-size: 8px; margin: 0;">Change Log updated default ip address</p>	<div style="background-color: #007bff; color: white; padding: 2px; font-weight: bold; font-size: 8px;">Load</div> <p style="font-size: 8px; margin: 0;">2/3/2016 11:03:13 AM 4,618,284 Bytes</p> <p style="font-size: 8px; margin: 0;">Change Log Changed capitalization on rgb animator actions</p>
<div style="background-color: #007bff; color: white; padding: 2px; font-weight: bold; font-size: 8px;">Load</div> <p style="font-size: 8px; margin: 0;">11/2/2016 3:45:30 PM 7,147,727 Bytes</p> <p style="font-size: 8px; margin: 0;">Change Log Vertical camera up/down control inverted</p>	<div style="background-color: #007bff; color: white; padding: 2px; font-weight: bold; font-size: 8px;">Load</div> <p style="font-size: 8px; margin: 0;">10/30/2016 11:40:01 PM 7,211,259 Bytes</p> <p style="font-size: 8px; margin: 0;">Change Log Added color tracking checkbox</p>	<div style="background-color: #007bff; color: white; padding: 2px; font-weight: bold; font-size: 8px;">Load</div> <p style="font-size: 8px; margin: 0;">2/6/2016 10:34:01 PM 7,200,477 Bytes</p> <p style="font-size: 8px; margin: 0;">Change Log smaller font size to fit words on mobile interface</p>	<div style="background-color: #007bff; color: white; padding: 2px; font-weight: bold; font-size: 8px;">Load</div> <p style="font-size: 8px; margin: 0;">2/3/2016 10:39:37 AM 4,617,391 Bytes</p> <p style="font-size: 8px; margin: 0;">Change Log capitaized each first letter of Actions in actions</p>
<div style="background-color: #007bff; color: white; padding: 2px; font-weight: bold; font-size: 8px;">Load</div> <p style="font-size: 8px; margin: 0;">11/2/2016 3:42:00 PM 7,147,726 Bytes</p> <p style="font-size: 8px; margin: 0;">Change Log servo pad UI for JD's head</p>	<div style="background-color: #007bff; color: white; padding: 2px; font-weight: bold; font-size: 8px;">Load</div> <p style="font-size: 8px; margin: 0;">7/29/2016 11:27:39 PM 7,206,354 Bytes</p> <p style="font-size: 8px; 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; font-weight: bold; font-size: 8px;">Load</div> <p style="font-size: 8px; margin: 0;">2/6/2016 10:31:25 PM 7,200,493 Bytes</p> <p style="font-size: 8px; 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; font-weight: bold; font-size: 8px;">Load</div> <p style="font-size: 8px; margin: 0;">2/1/2016 12:00:40 PM 4,617,947 Bytes</p> <p style="font-size: 8px; margin: 0;">Change Log JD's hands no longer close automatically will keep their last position during walking have him holding something.</p>

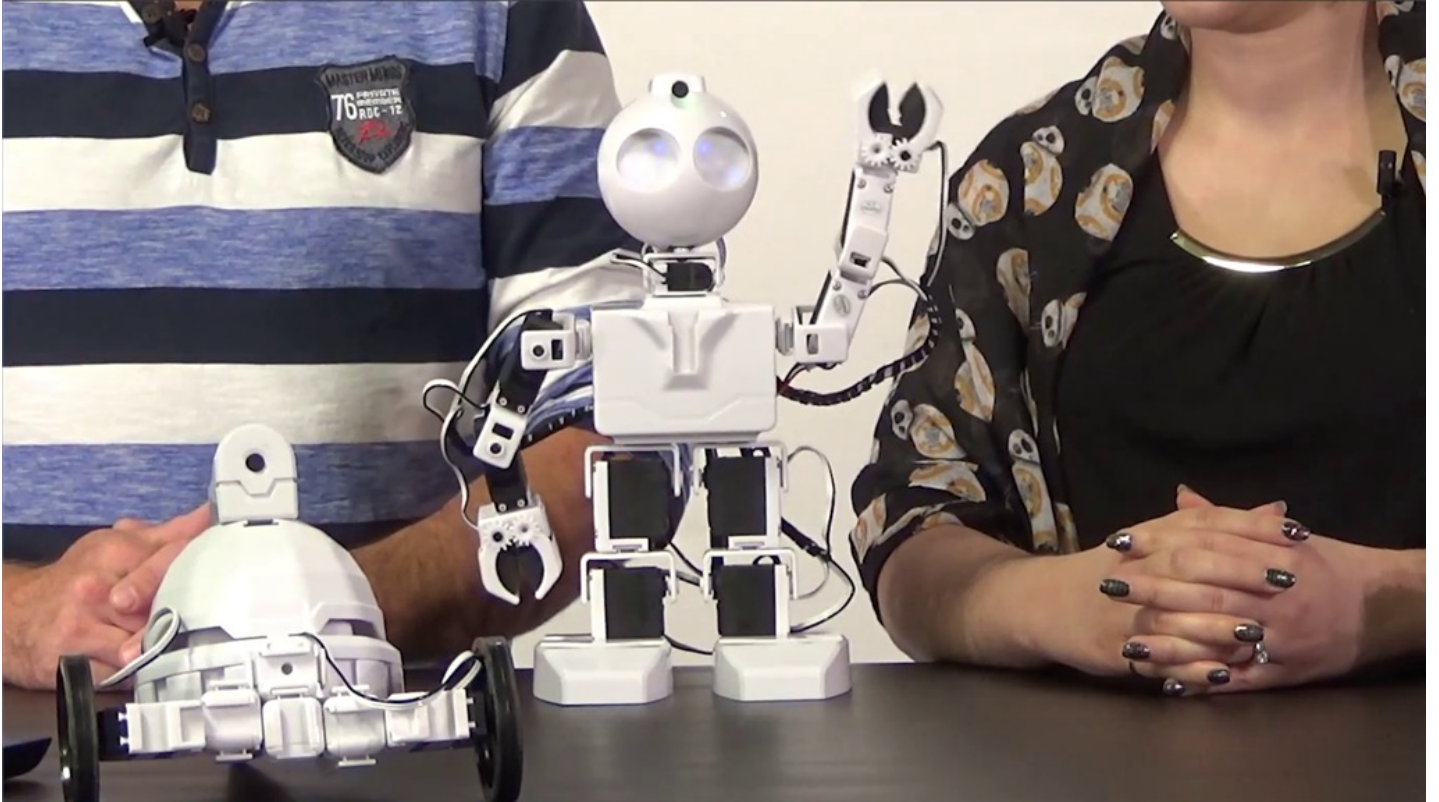
⏪⏩

Close

Windows EZ-Builder - JD.EZB EZ-Cloud JD 4:42 PM 3/2/2017

Learn More

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



Quiz

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.