Copyright 2026 Synthiam Inc.

Get Started

Get Started With Synthiam

Learn about Synthiam products on this page, or skip to the Getting Started Guide.

Skip to Getting Started Guide

Synthiam Products Videos

Programming robots can be frustrating and time-consuming.

We understand the pain of programming a robot, so we created ARC Robot Control Software to make it easier for anyone to program a robot in minutes.

With ARC, you'll spend less time writing code and more time getting things done with your robot. It's easy to set up, simple to use, and works right out of the box with any compatible device or computer. Get started today using this guide!

Individual Products Overview

ARC (Autonomous Robot Control) Software

ARC is a graphical and easy-to-use robot development software suite.
ARC provides an abstraction of common robot behaviors as reusable modules called skills. These skills are created by industry experts and published in Synthiam's Skill Store to use robot builders and programmers.

Learn About ARC

Skill Store

The Skill Store functions like an App Store, apart from hosting robot skills vs. apps.
Each reusable module is a technology, cloud service, or sensor driver wrapped in a graphical skill control.
Robot builders and programmers add skills from the Skill Store to their robots.
This method of delivering technologies as skills enable users to quickly experiment and integrate features at an escalated rate vs. traditional syntax programming with APIs and SDKs.

Visit the Skill Store

Exosphere

Robots learn from examples and are built to help us perform mundane or dangerous tasks.
But, to do their job right, they need to be taught.

Exosphere was developed to allow all robots to perform autonomous tasks by controlling as much or as little as possible to help robot companies achieve their goals.

All while training the global artificial intelligence knowledge base of autonomous tasks with real-world examples.

Try Exosphere

Roll Out

Synthiam publishes open-source hardware reference designs on our GitHub Repo to help robot companies and manufacturers make new advanced hardware products.

Our reference designs include advanced navigation systems, sensor arrays, EZB robot controllers, and more.

Save time and development costs using Synthiam source-available reference designs to make robot hardware products!

View Rollout Reference Designs

EZB Robot Firmwares

There are many microcontrollers and developer kits available in the market.

Synthiam has developed a variety of firmware that install on these devices to make them compatible with our ARC Software.

When a microcontroller is connected to ARC, it is called an *EZB*.

An EZB is any robot computer or microcontroller that accepts connections from Synthiam's ARC software to control servos, sensors, and more.

View Compatible Hardware

Support & Documentation

Synthiam strives to provide up-to-date documentation and guides for programming robots.

The support & docs section is your one-stop shop for getting started and onward.

If anyone is struggling with a challenge, Synthiam also provides a premium support service that gives you direct access to our robot experts to help solve the problem.

Explore Docs & Support

Let's Get Robot Build'n!

Now that you're familiar with Synthiam products that will make your robot awesome, we can move on to the robot-building tutorial.

Build a Robot

BACKUP STUFF

ARC provides an abstraction of common robot behaviors as re-usable modules, called skills. These are built by industry experts and published in Synthiam’s Skill Store to be used by robot builders and programmers. The approach is inspired by the democratization of complicated technologies that make them available to everyone. Such as Windows for personal computers, Unity for creating video games, and HTML for using the internet. (Meet ARC)

How ARC Works

ARC runs on an x86 Windows-based PC, Laptop, or SBC (single board computer) to leverage the PC's powerful architecture and hardware. The connectivity between ARC and sensors, servos, and peripherals is through a supported I/O controller. Also, the PC provides USB extensibility for additional peripherals, such as joysticks, cameras, audio devices, and more. However, the location of the PC for the robot (remote or embedded) will determine connectivity to the I/O controller and possible USB peripheral limitations.
This is stuff from the first page of getting started which has a link to every step in the getting started guide. replaced with NEXT button

Steps to Make a DIY Robot

With your pencil and paper ready, let us begin choosing options for making a robot in the next steps of this Getting Started guide.

2. Choose a Computer

ARC requires a computer running the Windows operating system. The computer can be embedded within the robot or used remotely over WiFi away from the robot.

3. Choose EZB Microcontroller

ARC's PC connects to an I/O controller (EZB). There are USB and WiFi EZBs.

4. Choose Power

Just as we need food to survive, a robot requires a power source to provide energy to all subsystems.

5. Robot Skills

A short overview of a robot skill and how you will use them in your ARC project.

6. Moving Servos

Moving servos is the primary goal of any robot builder. Having a servo move is giving your robot animated life, like Frankenstein!

7. Choose a Movement Style

A robot requires a movement type to interact with the physical world. Movement type is closely related to the locomotion style, for example,
walking with servos, driving with wheels, or flying.

8. Choose Vision Camera

A robot requires a camera to support vision-tracking capabilities. As sight is the most important sense for humans, a camera is a highly versatile robot sensor.

9. Choose Audio

A robot requires audio hardware to receive verbal commands and have the ability to speak audibly.

10. Remote Control

Remote control of the robot with a joystick, keyboard, WiiMote, etc., will allow you to test and ensure it is correctly operating.

11. User Interface

Create a custom user interface for controlling the robot.

12. Autonomous Navigation

Giving a robot the ability to navigate gives it a personality. It also allows it to accomplish goals by knowing where and where it is going.

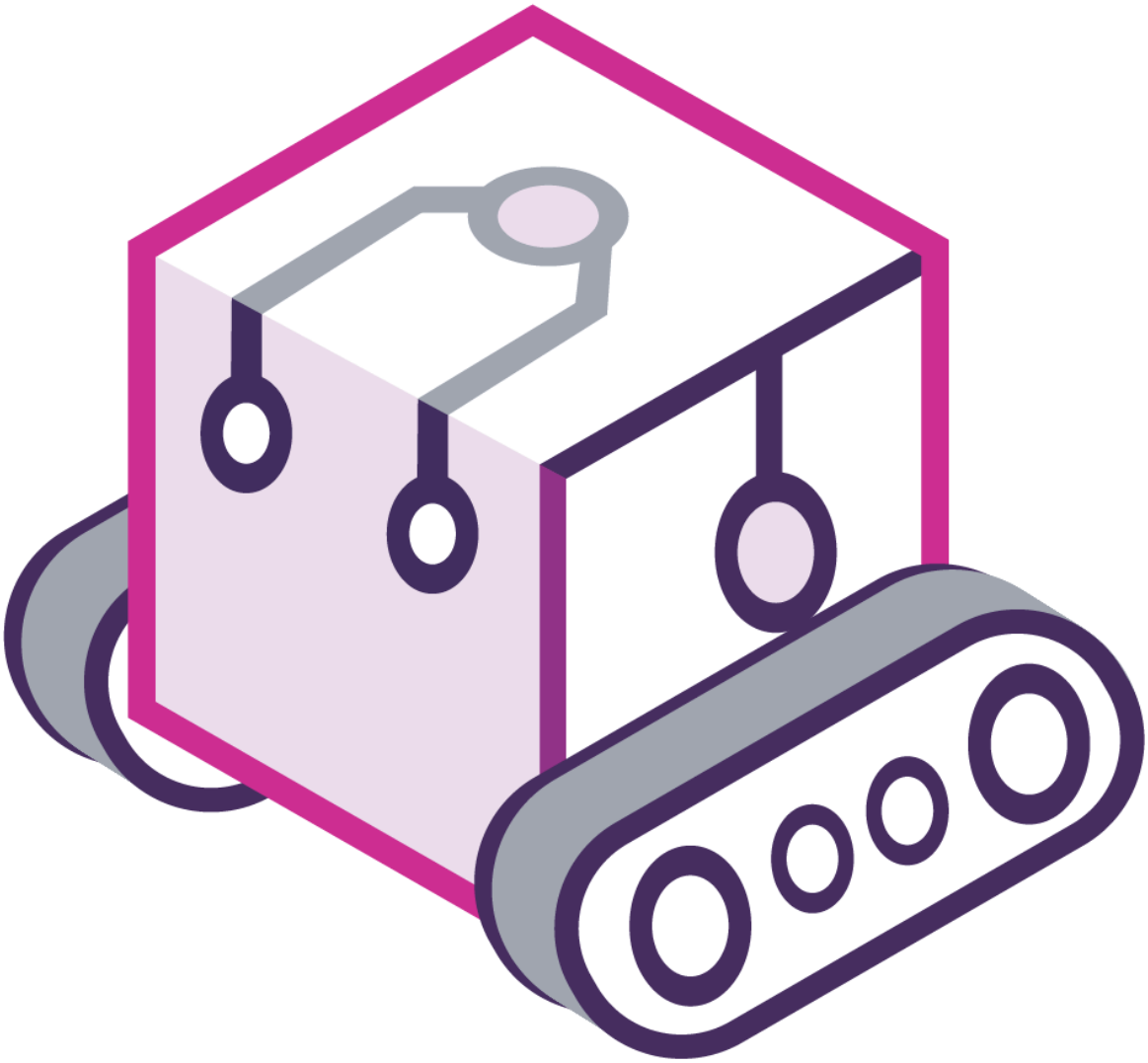
How to Make a Robot

Choose Your Robot

You are about to use the most accessible guide for building robots with science-fiction-inspired features.
We guarantee that anyone can build a robot on par with that of heavily funded corporate organizations.
Synthiam has packaged the most cutting-edge technologies and made them available to everyone.

Need Some Inspiration?

Do you need some inspiration before we begin? With over 4 million robot connections on Synthiam's platform, we have plenty of community robot examples.



View Example Robots

What kind of robot do you want to program?

Synthiam's ARC platform will connect to many robot products, toys, and DIY kits. Let's start by selecting what type of robot to make. There are options for using a robot kit or quickly building your dream DIY robot.

1. Use a Robot Product

Skill Level: *Beginner*
Program a pre-built robot product or a kit with Synthiam ARC. You can program a robot directly out of the box with low effort. Here are a few popular ARC-compatible robot products. Select a robot product to view purchase links and manuals.
[opt:hardwareType:2]

2. Make a DIY Robot

Skill Level: *Intermediate, Advanced*

Making a DIY robot has traditionally been a daunting task. Anyone can make DIY robots with Synthiam's ARC robot software.

We have created a step-by-step Getting Started Guide to making a DIY robot - press the button and get started!

Make a DIY Robot

1. Make a DIY Robot!

Plan Your Robot on Paper

Before assembling hardware or writing code, sketch your robot and write down its goals. Clear drawings and concise objectives help you visualize component placement, plan wiring and mounting, and identify potential mechanical or sensor conflicts early. Paper plans also create a record you can revisit as the project evolves — they are useful when troubleshooting, iterating on the design, or sharing progress with others.
During the following steps we provide hardware and software options that will help your robot achieve its goals, whether the project is a serious build or an experimental, fun prototype. Spend a little time now to imagine the final robot and note the key functions you want it to perform.
Tip: If you have questions at any point, ask in our friendly community for advice, feedback, or troubleshooting help.

Ask The Community Anything

Document the Physical Appearance and Layout

Sketch the robot’s exterior and internal layout. Mark where you will mount sensors, motors, controllers, batteries, and wiring harnesses. A clear diagram helps you estimate space requirements, avoid component interference, and plan cable routing. When you share these sketches on forums, contributors can give more precise suggestions when they clearly see

your intended layout.
Below is an example sketch that illustrates component placement and physical arrangement.
Example sketch showing component placement and basic layout for a robot design.

Break the Robot Down into Micro-Goals

Complex behaviors become manageable when you break them into smaller, testable tasks. List each discrete capability your robot needs to perform the overall feature, then refine those into sub-tasks that a robot can follow precisely. This makes programming, testing, and debugging much easier.
For example, consider the goal "retrieve a peanut butter jar from a cupboard." You would decompose it like this:

1. Know where the peanut butter jar is (identify via label, visual marker, or known coordinates).
2. Plan and move to the jar's approximate location.
3. Detect whether a cupboard door obstructs access; if so, open the door.
4. Visually locate the jar on the shelf (confirm it is visible and reachable).
5. Clear any obstructing items and grasp the jar securely.
6. Determine the work area for placing the jar (a table or counter coordinate).
7. Navigate to the work area while avoiding obstacles.
8. Place the jar at the specified location and release grip safely.
9. Perform any cleanup or return motions as required.

The number and complexity of micro-goals will depend on your robot's capabilities and the environment. Write each desired feature down and continue breaking it into smaller steps. Remember that robots do not possess the general world knowledge humans take for granted, so explicitly define any assumptions or environmental constraints the robot requires to succeed.

Next Step

2. Computer

Let's select the computer you will use with the robot. ARC runs on a Windows PC and connects to an I/O Controller (EZB) for moving servo motors and reading sensors. This step will determine what type of computer your robot will use.
A robot computer can be a PC laptop or Single Board Computer (SBC) that runs Microsoft Windows and the ARC software. If the computer is an SBC, it will be small enough to install inside the robot. Otherwise, the computer will be remote outside the robot, such as a laptop or desktop. No matter what type of computer you choose, it must be connected to an EZB Microcontroller for moving servos, motors, LEDs, and sensors. The computer's location will determine the connection type to the Input/Output (I/O) controller (USB or Wi-Fi). Any I/O controller compatible with ARC is called an EZB (*more information below*). Decide which of these two EZB/Computer configurations works with your robot build.

Embedded PC

(Computer in the robot)

An embedded PC means mounting the computer on board the robot. The computer can be a *laptop, single-board computer (SBC), or tablet*. The connection to I/O EZB microcontrollers is through USB. The benefit of this configuration is that USB devices, such as joysticks, cameras, and sensor peripherals, can be connected directly to the PC on the robot.

-or-

Remote PC

(Computer connects wireless to the robot)

Remote PC, the computer is not mounted on the robot and most likely sits on your desk or workbench and connects to the robot via wifi or Bluetooth. The computer can be a *laptop, tablet, or personal computer (PC)*. The downside to this configuration is that USB devices (Cameras, sensors, etc.) can not be mounted on the robot because they need to be connected to the PC, which will not be mounted on the robot.

Natively Supported Embedded PCs (SBC)

Did you choose to run ARC on an SBC? ARC runs any computer that supports Windows 10. However, when dealing with SBCs, there are performance considerations. CPUs in portable-style computers have less performance than a laptop.
We have documented and verified these SBCs that work well with ARC. If you choose an SBC, follow our tips for making a robot to get the most out of the robot's dedicated computer.

Get ARC Windows 10 ISO

[opt:hardwareType:3]

Previous Step

Next Step

3. EZB Microcontroller

When a generic I/O controller (Arduino, Microbit, etc.) is programmed with Synthiam firmware compatible with ARC, we refer to it as an EZB. The firmware then allows the EZB I/O controller to use additional capabilities available in ARC. An EZB connects sensors, motors, and peripherals to the computer.

What is an EZB Microcontroller?

a low-power device that allows sensors, servos, LEDs, and motor controllers to be connected. The EZB provides an interface between your computer and the robot hardware. Popular EZB Microcontrollers are Arduinos or EZ-Robot EZB v4/IoTiny, etc. You will find a list of compatible microcontrollers below in this guide.

Each EZB will provide various hardware and custom commands. To identify the capabilities

of an EZB, ARC provides abstract methods that represent the EZB's capabilities. For example, if the connected EZB supports a hardware UART, the respective EZB index in ARC will provide hardware UART commands. The Firmware Capability Manager determines the capabilities of each EZB. An EZB will report its capabilities to ARC. ARC will notify you if an unsupported command is issued to an EZB.

Natively Supported EZBs

Based on the type of computer selection (embedded SBC or remote PC), EZBs will support different connection types.
For example, a connection type of USB requires the computer to be embedded in the robot. An EZB that has Wi-Fi supports a remote computer or embedded computer configuration.

[opt:hardwareType:1]

Don't see your hardware listed?

Add any robot product, hardware, PLC, or microcontroller to the Synthiam-supported EZB list. We have published the EZB communication protocol for anyone to use.

EZB Communication Protocol Details

Previous Step

Next Step

4. Power

Let's identify how your robot will be powered!
Most mobile robots are powered by a battery rather than tethered by a power cable connected to an outlet. Understanding power requirements for a robot can be understood by the current and voltage required. Begin the robot design by documenting voltage and amperage requirements for motors, sensors, computers, and peripherals when choosing an appropriate battery. Whichever power option is used for the robot, ensure it provides enough amperage for the robot application.

How Many Amps?

Amperage is how current is measured. Peripherals such as servo-motors require high current (amperage), not voltage. Depending on size, a single servo-motor generally consumes between 1-3 amps (1,000 milli-amps) when moving quickly. When a robot has more than one servo moving simultaneously, the amperage requirement could be 5-10 amps. When planning the robot, you will need to consider this because a current shortage causes nearly every issue with EZBs restarting when servos move.

How Many Volts?

Considering the required voltage is generally an easy answer. Because most electronic devices operate at either 3.3v or 5v, the most common external power supply is 5v, and the most common battery is 7.4v.
When getting a battery or power supply for the robot, get either in the range of 5v - 7.4v, which will encompass nearly all robot hardware. However, it is still a good idea to check the sensors and motors for your robot to identify what voltage they recommend.

- Heavy Duty 7.4v 5.2 amp LiPo battery (for many, many servos)
- Lite-Duty 7.4v 1.5 amp LiPo battery (for a few servos. This type of battery is used in many everyday robot products, such as EZ-Robot JD)

Note: When powering servos or motors, take note of the amperage of your power supply because it is far more critical than the voltage. However, both need to be within an acceptable range; the most significant cause of robot failure is a lack of amperage in the power supply.

Battery Current vs. Power Supply Current (milliamps per hour)

The current provided by a battery is a little different from a power supply. A power supply limits the current it can produce (i.e., 3amps). However, a battery has an additional rating called a C rating. That rating is the capacity of energy the battery can safely discharge, represented as a multiple of its overall capacity. A battery with a higher C rating delivers more power, which means higher performance.

Batteries are rated in milliamps per hour. So they can provide X amount of milliamps in an hour. If your battery was 5,000 mAh and the robot required 1,000mAh continuously, the battery would theoretically last 5 hours. Also, if the robot were drawing 5,000 mAh, then the battery would last 1 hour. However, the C rating means that your battery can discharge more current than the mAh rating. A battery may have a 5,000 mAh capacity, but if the robot were to draw 10,000 mAh, then the battery would last 30 minutes. So a 5,000 mAh battery does not mean that is the maximum output; it is the current per hour. The robot can draw more, but the battery will drain more quickly.

A higher C rating of 2 means that a 7.4v 5200 mAh lipo battery can deliver ten continuous amps for 30 minutes. However, it is unlikely that any servo can withstand the heat that requires ten continuous amps. When servos are not moving, the current is minimal, giving them time to cool down after strenuous movements.

Power Efficient Servos

Not all servo-motors are alike. Servo motors come in many sizes and price ranges, which reflect their power consumption. The most inefficient servo-motor is low-cost analog-style hobby servo-motors, such as the popular model numbers starting with MG-xxx. The

inefficiency of analog servo motors does not contain digital logic that controls the motor's positioning with intelligent algorithms. For example, Dynamixel or EZ-Robot servos are digital and have the intelligence to use high-frequency algorithms that control the servo motor.

For example, a video of the EZ-Robot HDD servos demonstrates their efficiency and built-in shutdown features. While we use this EZ-Robot HDD servo video as an example, many other servo manufacturers have similar characteristics. We didn't have a video of other servo-motors being demonstrated to the level of detail that this video does. Another servo to consider would be Dynamixel from Robotis.

Select Battery Source

Battery

This type allows the robot to be completely mobile and portable. Popular battery options are Lead-acid, LiPo, NiCAD, NiMH, and LiFePO4. The battery is charged using an external charger. More advanced configurations can have a docking system for the robot to dock with the charger automatically. Make a note of the power requirements and what size battery. For example, you may require only a general-use 7.4v 2,000 mah LiPo battery or a heavy-duty 12v 10,000 mah battery, depending on the number of servos and motors used in the robot.

AC-DC Power Adapter

This type requires the robot to be tethered to a power outlet. Power adapters called "wall-warts" do not provide enough amperage for servo motors. If an AC-DC adapter is preferred, use a digital switching power supply with enough amperage to meet the robot's requirements. Here are some external power adapters that provide enough power for servo motors.

- 5v 15 amp power supply (for a few servos)
- 5v 30 amp power supply (for many servos)

Transistor Batteries (9v)

9 Volt batteries, also called transistor batteries are not designed for motors or high current devices. As you guessed, they're only intended for transistors. This may be a good option if your robot project does not have motors or servos because it can power the EZB, camera, and sensors. However, if a servo, motor, or anything mechanical is involved, these batteries will not work.

Wall Wart (power transformer)

Many are inclined to use one of these, as many lie around the home from old electronics. These generate power using a transformer, which is generally low current/amperage and, similar to the 9-volt battery, is designed for electronics, not motors. Even if you find a transformer that may advertise 1 or 2 amps, it will not be able to maintain the short spike or continuous draw required by servos and motors. We highly discourage people from using one of these power adapters as they fail to provide enough current and cause grief.

Battery and Power Supply Details

Because the discussion around power is such an extensive conversation, Synthiam's members made a great tutorial explaining more. We recommend reading this tutorial because it describes how voltage and amperage work. There's plenty of information on what kind of battery or power supply to use for the robot.

Battery and Power Supply Tutorial

Battery Monitor

ARC includes a battery monitor that disables the EZB's I/O outputs to reduce power consumption on supported I/O controllers when the battery voltage is lower than the specified threshold. By default, the battery monitor is configured for 7v to protect the safety of LiPo batteries. The battery monitor can be disabled in the connection control configuration. Also, some EZBs have a built-in battery monitor enabled on power-on, such as the EZ-Robot EZB v4 and IoTiny. The web interface menu of the EZ-Robot products allows disabling or configuring the power-on battery monitor.

To summarize, there is a battery monitor setting in the ARC connection control configuration. If using an EZ-Robot product, there is also a battery monitor setting for power-on that can be configured in the EZB web configuration menu.

Previous Step

Next Step

5. Robot Skills

Over the next few steps, we will be introducing various robot skills that will be used to give your robot life.

ARC project apps consist of Robot Skill Controls. Each skill is a behavior for the robot, similar to a process (or node). There are skills for Movements, Cameras, Speech Recognition, Machine Learning, and hundreds more. Skills can be added to a project workspace using the Add Skill option located in the Project tab of the main menu. By combining multiple skills, robots can perform advanced and complex tasks.

Virtual Workspaces to Organize Robot Skills

As a robot project grows, it will have several robot skills that clutter the workspace.
Synthiam ARC can customize virtual workspaces to organize robot skills.
The workspaces are located in ARC's top menu under the File tab. Pressing the ADD or REMOVE buttons allows you to add or remove workspaces—Right-click on a workspace to rename it.

In this screenshot example, we have organized the robot skills based on their function with the robot.
The Input workspace hosts interactive robot skills, such as joysticks and speech recognition.
The Camera workspace hosts the robot's camera and various tracking and computer vision robot skills.
The Processing workspace hosts scripts and AI Chatbot robot skills.
Finally, the Movement workspace hosts the robot skills responsible for moving the robot and interacting with the real world.

Example Project

Here is a screenshot of an example ARC project demonstrating multiple robot skill controls. Each skill control performs a specific function of the robot. An ARC Pro user's projects may contain an unlimited number of skills (as PC memory allows). Read more about robot skills here.

(Click to enlarge)

Manuals

Every robot skill will have a manual that can be accessed by pressing the "?" (question mark) on the top right of the robot skill's window.
This will bring you directly to the manual page for that individual robot skill.

Getting Robot Skills

The Add Robot Skill option within ARC can install robot skills.
However, you may wish to browse robot skills online and read detailed information about the function and how to use them—introducing Synthiam's Robot Skill Store. This is a place to browse robot skills before adding to your project.

Previous Step

Next Step

6. Moving Servos

Servos are used for high-precision movement of robot arm joints, camera gimbals, and more. This tutorial will provide technical information about servo motors and how they work. We made it easy to get a robot up and running. However, there are many fun and exciting things to learn about how the robot works. The more you know, the more you can get your robot to do it!

If your robot does not have servos, skip this step. For example, skip to the next step if your robot does not have an arm, gripper, or any servo requirement.

Types of Servos

ARC supports all servos, including the ability to define your servo driver using the Servo Script Robot Skill. The most common types of servos are...

- PWM Hobby servos that are connected directly to EZB digital ports
- Serial Smart Servos (Robotis Dynamixel, Lynx Motion, Feetech, Kondo KRS, LewanSoul, UB-Tech, and more)
- PWM servo extenders (SSC-32)

***Note:** You will find servo driver robot skills in the Servo Robot Skill Category. These skills are also listed at the bottom of this document.

Servo Interface Menu

ARC displays a standard configuration dialog for configuring servos across all robot skills. This manual will explain how to configure a servo (even multiple servos) to be moved from a robot skill. In this example, we will use a vertical servo robot skill, although this procedure applies to any robot skill that uses servos. Many robot skills use servos, including WiiMote, MYO, Camera, and dozens more...

The standard servo interface menu is the same across all robot skills that use servos. This menu will be displayed when selecting a servo in a robot skill's configuration.
You can choose the port, EZB Index, the MIN position value, and the MAX position value of each servo. You may also select the checkbox to invert the direction.

Advanced Servo Interface Menu

This new menu will display when the advanced button is pressed on the standard interface menu. This advanced menu allows specifying additional options and adding multiple servos.

The advanced interface is also used when configuring acceleration, velocity, or speed of servos that support those properties.

***Note:** *Hover your mouse cursor over the blue question marks to read about advanced options. Many advanced options require hardware support to work. So, by default, they are set to -1, which means ignore.*

Specify Servo Resolution

By default, the software will have a servo resolution of 180 positions. While most hobby servo controllers (i.e., Arduino, EZ-Robot EZB) are limited to 180 degrees, some servos support a much higher resolution (i.e., Dynamixel). You can configure ARC to support a higher servo resolution per project. The global servo resolution setting can be configured in the My Robot Project Properties menu. The range will be calculated to compensate if the value of servo positions is higher than the EZB supports. For example, if you set the ARC servo resolution to 360 while using an EZ-Robot EZB v4, the value of 360 will become the new maximum servo position.
*Note: The maximum number of servo positions ARC supports is **2,147,483,647**

Most Popular Servo Robot Skills

There are many robot skills control servos, such as the Camera Device, WiiMote, Joystick, and scripting, to name a few. There are a few of the most popular servo skills used by robot builders to get started.

Vertical/Horizontal Servo

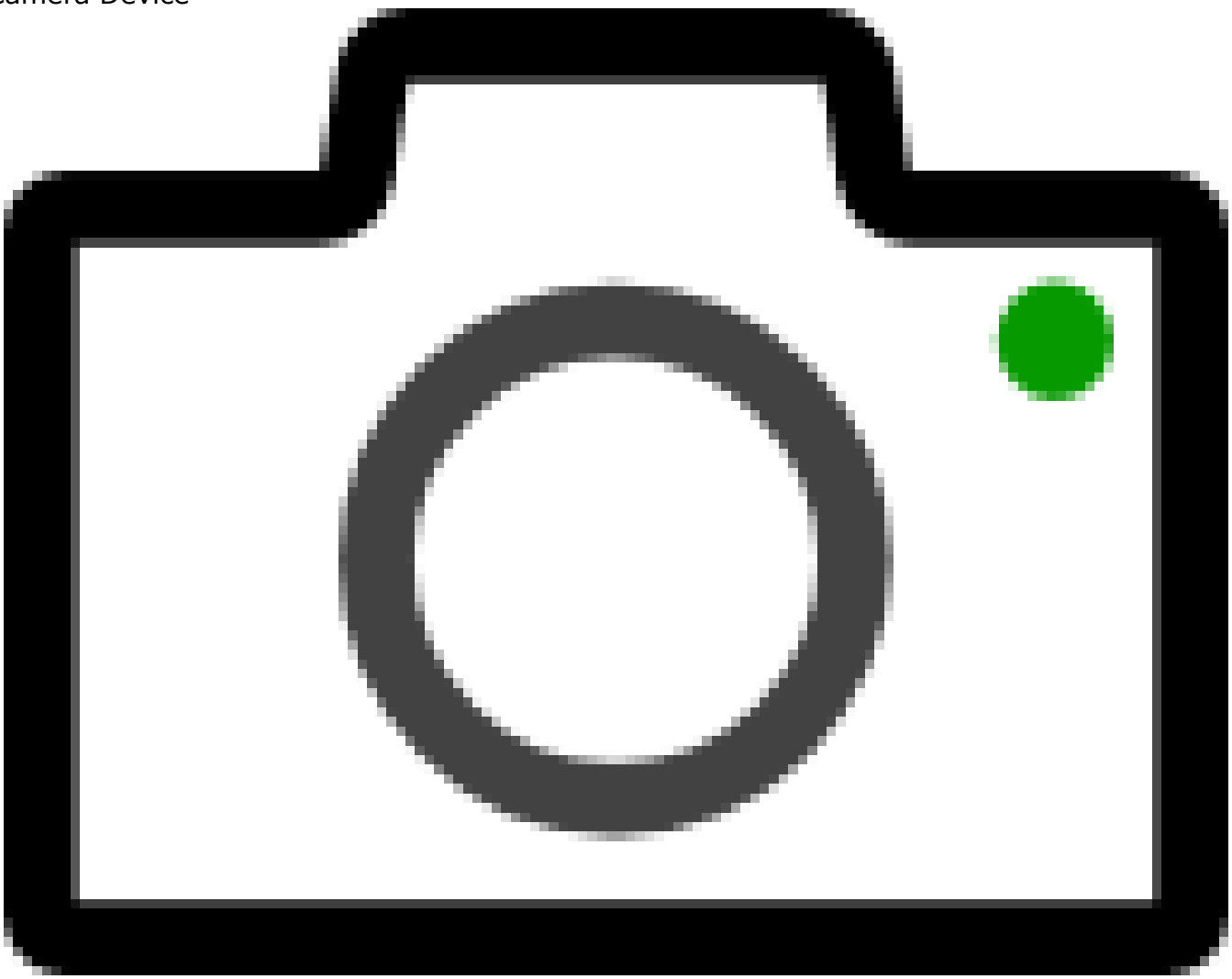


Great for testing and moving your first servo! These robot skills are excellent for testing your robot's movement range and ensuring the servos work. We always recommend using these robot skills when building your robot to ensure everything works correctly. At the same time, the two versions of this skill differ in how the user interface is presented by dragging vertically or horizontally. This does not have to matter based on the orientation of your servo in the robot itself.
These two skills can be used for testing as it is a matter of preference.

Horizontal Robot Skill

Horizontal Robot Skill

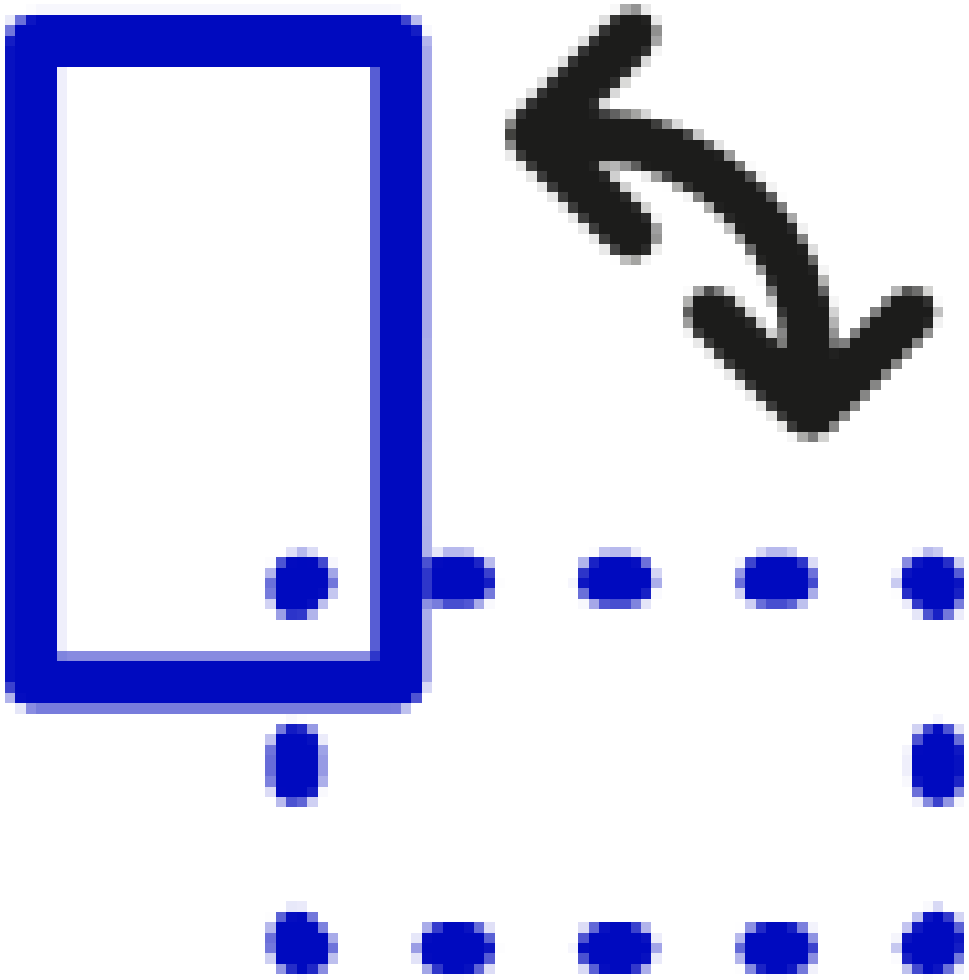
Camera Device



One of the most powerful and popular robot skills demonstrating ARC's power is the Camera Device.
This robot skill will use either the EZB remote camera (if supported) or a USB webcam mounted on the robot.
By configuring options, you can have the camera track objects, faces, and colors and even detect emotion or age with additional robot skills.
The Camera Device has settings in the options menu for controlling servos when a camera is mounted on a gimbal.
This feature allows the robot's camera to move and track the object.

Camera Device

Auto Position



Create servo animations to have your robot wave, dance, or pick up and move objects.
The Auto Position skill transforms a group of servos into custom positions (Frames) to create animations.
You can combine the frames into actions for your robot's animation.
This is done with "Inverse Kinematics" or "Motion Planning."

There are two types of the Auto Position robot skill. One is a movement panel that plays servo animations for robots that use servos to walk (i.e., humanoids or hexapods).
The other type is used for robots that use a different movement panel (i.e., hbridge, continuous rotation, etc.) but has servos for arms to be animated.

This is a popular robot skill for humanoid robots, such as the InMoov, Robotis Bioloid, or EZ-

Robot JD/Six.
For robots with many servos used in arms or legs, this allows creating animations that can be executed programmatically by other robot skills using the ControlCommand().
This means your robot can perform actions based on speech recognition or chatbot conversations.

Auto Position with movement panel

Auto Position for only servos

Joystick



There are two types of joystick robot skills based on the type of joystick used. The most common type is the XInput version, which supports the latest joysticks, including Xbox controllers.
The xInput has additional support for analog trigger buttons and multiple independent joysticks. These robot skills can control servos, allow scripts to be assigned to buttons, and optionally control the current movement panel to move the robot.

Direct Input Joystick

xInput Joystick

Virtual Reality



Being able to control your robot servos with your hands and move the robot's camera with your head is a fantastic experience. A few robot skills support a variety of VR headsets, including the Meta Oculus Quest and Steam VR (i.e., HTC Vive, Windows Mixed Reality, etc.). The hand tracking feature is one of the great features of using the Meta Oculus Quest robot skill. This feature lets you move servos without needing to hold the controllers in your hands.
You can map your fingers to individual servos, which is excellent for humanoid projects like the InMoov.

Direct Input Joystick

Hard Set Servo Limits

You can set global servo positions across all robot skills with the appropriate Javascript, EZScript or Python commands.

The robot skill settings for servos are for the individual robot skill only. Every robot skill has servo values (i.e., min and max). This means the Camera skill has different servo values than the Joystick skill. The values specified in a skill's configuration are specific to that skill.

There are commands for all languages (JavaScript, Python, EZ-Script, etc.) for specifying limits. For example, the EZ-Script command in an INIT script to specify servo positions globally across the entire application is...

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)

Here's an example from the EZ-Robot JD project that makes the left gripper not move further in either direction globally across the ARC software. Use the JavaScript, Python, or EZ-Script manual for the appropriate commands for setting global servo position values.

```
# Left Gripper
SetServoMin(d6, 30)
SetServoMax(d6, 90)
```

Example Testing a PWM Servo

Because ARC uses a standard dialog for configuring servos, the following steps will demonstrate how to use it. Any robot skill supporting moving servos will display the same "servo selection" dialog as this example.

In this example, we will use a standard PWM hobby servo connected to one of the EZB digital ports.

Step 1

Load ARC and connect to an EZB. Add a servo to the D0 port of the EZB.

***Note:** Ensure you have the latest ARC. When you load ARC with an internet connection, it will notify you if a newer is available.

Step 2

Press the Project tab from the top menu. Now press the Add Control button.

Step 3

The Add Control window will display. This window allows you to browse and select controls to add to your project. Press the SERVO tab to view servo-specific robot skills.

Step 4

We are going to use the Vertical Servo control for this example. Many kinds of robot skills interact with servos, even more than you can see on this page. Nearly every skill control uses servos; however, only the specific servo skill controls are listed on this page. Even the Camera, WiiMote, MYO, and more use servos. Click the Vertical Servo button to add the vertical servo skill control to your project.

Step 5
You will add the Vertical Servo skill control to the workspace. This skill allows sliding the mouse vertically to move a servo position. Alternatively, a Horizontal Servo skill control enables the mouse to be dragged horizontally to move the position.

Step 6

As mentioned in the Controls Tutorial, every control has a gear button. You can press this gear button to load the configuration menu. Each control has a unique configuration menu.

Step 7
ARC will now display the configuration menu

Step 8

Each configuration control will have many options. Any control that uses Servos will have a similar servo configuration interface. Some controls may have two or more servo configuration interfaces (usually for horizontal and vertical servo control). In this control, there is only one servo interface.

Name: This is the name of the control.

Board Index: ARC can connect 5 EZ-Bs to the ARC Software. This specifies which EZ-B to send the servo command to.

Port: The EZ-B port of the servo. Pressing this button will display the EZ-B to select the respective port.

Min/Max: Min and Max limits in degrees of the servo. The servo can move between 1 and 180 degrees. The value for Min must be less than the value of Max in all cases, even when Invert is checked.

Invert: If the servo is moving in the wrong direction, checking this box will reverse/invert the direction of the servo.

Multi Servo: If more than one servo moves, you can specify multiple servos.

Step 9

Press the PORT button, and ARC will display the EZ-B port configuration. You may select the port and press the Close button in this dialog.

Step 10

Now that you have selected a port, we can move the servo to specify the MIN and MAX limits. These numbers are in degrees between 1 and 180. Start with the MIN by pressing the mouse button while dragging the cursor UP or DOWN. The MIN value must be less than the MAX value. In this example, set the MIN to a low number, and the servo will move in real time if connected to ARC and an EZ-B.

Step 11

Move the MAX by pressing the mouse button while dragging the cursor UP or DOWN. The MIN value must be less than the MAX value. In this example, set the MAX to a low number, and the servo will move in real time if connected to ARC and an EZ-B.

Step 12

If you want this control to move more than one servo simultaneously, the Multi Servo button

does just that. Press the button to display the Multi Servo dialog.

Step 13

To add multiple servos, press the ADD SERVO button. Each time the button is pressed, a new servo entry is added. You can remove the servo by pressing the X on the respective servo. Use the PORT, MIN, and MAX to configure the limits. The multi-servo option also allows a ratio to be specified. The ratio is based on the primary value for the first servo in the list. If the servo degree position for the first servo is defined as ten and the ratio is 2, the respective servo will move to degree position 20.

Step 14

Close the configuration dialog for the servo control and return to the workspace. Now that the servo has been configured, you may move it between your specified MIN and MAX limits. Click in the servo position, and the cursor will change while holding the mouse button. Slide the mouse up and down (for vertical control) to move the servo between the specified limits.

***Note:** Even though many controls may move servos, the configuration of those servos is only valid for that control. If you configure a camera control to move servos to specific min/max ranges, those ranges only apply to the camera control. Every control has its own range and settings for versatility.

Servo Robot Skill Drivers

While hundreds of robot skills use servos, that may not be their primary focus, and they are in different categories.
For example, the Camera Device is a popular robot skill, but its primary focus is the camera and tracking, so it is located in the Camera category.
The Servo category generally lists robot skills that are drivers or servo-specific. You will find the robot skill driver in the Servo category if you use a smart servo, such as the Robotis Dynamnixel.
Below is a list of all robot skills from the Servo robot skill category.

[opt:skillcategory:servo]

Previous Step

Next Step

7. Movement Style

Let's determine how your robot will move!
ARC uses movement panels to control a robot's movement. Each ARC project is limited to one movement panel because robots only have one method of locomotion, whether wheeled, crawling, flying, bipedal, etc. The movement panel will register itself with the movement manager service. This allows any robot skill to share the robot's control movements.
For example, by adding a Joystick or WiiMote robot skill, the robot can automatically begin moving because the current movement panel handles the movements.

Tip: There is a section for additional details on how ARC uses movement panels and what a movement panel is.

Movement Panel Manual

Advantages of Movement Panels

The convenience of the movement manager service is that any ARC robot project can control any other robot product, irrelevant of the locomotion type (flying, walking, two-wheel driving, etc.). If your ARC project was created for a wheeled robot, you could replace the wheeled movement panel with a drone movement panel. Directions (forward, left, right, etc.) are sent to the movement manager service by any robot skill and handled by the movement panel to have the robot move.

Directions

The ARC framework provides several pre-programmed directions: forward, left, right, reverse, stop, roll right, roll left, up, down, and custom. Movement panels will support the directions that are appropriate to their movement type. For example, an HBridge movement panel will not support UP or DOWN because the robot must have wheels and, therefore cannot fly. You can see what directions are supported on a movement panel by the buttons visible on the interface. This diagram demonstrates how the various directions are expected to behave by movement panels. For example, turning left or right is expected to rotate on the spot. However, to turn slightly right while moving forward will require the robot to move forward with the right wheel speed somewhat slower than the left.

Choose a Movement Panel Type

Here is a list of movement panels for various robot configurations, including robots that walk with gait humanoids/hexapods and more. Select a movement panel to learn how to use it and add it to the project.
This list can also be found in the Movement Panels category of the robot skill store or within the ARC software.
[opt:skillcategory:movement panels]

Previous Step

Next Step

8. Vision

Let's choose your robot's vision system! If your robot does not have a camera, skip to the next step.
Robots with cameras provide navigation, tracking, and interactive benefits. Synthiam ARC's software is committed to making robot programming easy, including computer vision tracking. The ARC software includes a robot camera skill to connect WiFi, USB, or video capture devices. ARC's camera device skills include tracking types for objects, colors, motions, glyphs, faces, and more. You can add additional tracking types and computer vision modules from the skill store.

Choose the Camera Type

Wired USB Camera

Connects directly to a computer with a USB cable. You can only use this type of camera in an embedded computer configuration. This is because the USB cable will tether the camera to the PC. You can use any USB camera in ARC. Some advantages of using USB cameras are high resolution and increased framerate.

Wireless Camera

Connects wirelessly to a PC/SBC over a WiFi connection. Generally, this approach is only used in remote computer configurations. Few I/O controllers support a WiFi wireless camera transmission due to latency causing low resolution and potentially unreliable radio interference. For a wireless camera application, the most popular I/O controllers are the EZ-Robot EZ-B v4 and IoTiny.

Add Camera Device Robot Skill

Whichever camera type you choose, the robot skill needed to connect to the camera is the Camera Device Robot Skill. Add this robot skill to connect to the camera and begin viewing the video feed. Reading the manual for this robot skill, you will find many options for tracking objects.

Add Camera Device Robot Skill

Additional Computer Vision Robot Skills

Now that you have the camera working on your robot, you may wish to add additional computer vision robot skills.
Computer vision is a general term for extracting and processing information from images.

The computer vision robot skills will use artificial intelligence and machine learning to track objects, detect colors, and recognize faces. There are even robot skills to detect facial expressions to determine your mood.

[opt:skillcategory:camera]

Previous Step

Next Step

9. Audio

Let's plan how your robot will speak and hear!
Robots with audio capabilities provide a very interactive experience. The ARC software includes multiple skills that connect to WiFi, Bluetooth, or USB audio input and output devices. ARC has speech recognition, text-to-speech synthesis, and robot skills for playing music and sound effects!

Choose an Audio Input Device

Wired Microphone
Connects directly to a computer with a USB cable or through an existing soundcard's input port. It is only used in an embedded robot configuration.

Wireless Microphone
Connects wirelessly to a computer with an RF (radio frequency) or Bluetooth connection

Choose an Audio Output Device Type

Wired Speaker
Connects directly to a computer with a USB or Audio cable. It is only used in an embedded robot configuration. You can select whether the speaker is connected via an audio cable to an existing sound card or connects with a USB.

Wireless Speaker
Connects wirelessly to a computer over Bluetooth or WiFi.

EZB Speaker

Use an EZB that supports audio output (i.e., EZ-Robot IoTiny or EZ-B v4). Note that using this option, the EZB on index #0 is preferred. This is also necessary if you wish to use the Audio Effects robot skill found
here
.

Add Generic Speech Recognition



Once you have selected the type of microphone to use, the next step is to experiment with speech recognition. Many speech recognition robot skills range from Google, Microsoft, and IBM Watson.
However, the most popular robot skill to get started is the generic speech recognition robot skill. This uses the Microsoft Windows built-in speech recognition system.
Therefore, it's easy to configure, and you can get it up and running with little effort.

Using the Microsoft Windows Speech Recognition Engine, this skill uses your computer's default audio input device and listens for known phrases. Phrases are manually configured in the Settings menu, and custom actions (via script) are assigned to your phrases.

Get Speech Recognition

Audio Robot Skills

Now that you have selected the audio device your robot will use, many robot skills should be considered. They range from speech recognition to audio players. You can have as many audio robot skills as your robot needs to achieve the goals.

[opt:skillcategory:audio]

Previous Step

Next Step

10. Remote Control

Once the robot is built, powered, and connected to ARC, you may test it with remote control before adding autonomous behaviors. Remote controlling the robot with a joystick, keyboard, WiiMote, etc., will allow you to test the functionality and ensure it is correctly operating.

***Hint:** Your project needs to have a movement panel added and configured before these remote control robot skills can control your robot's movements. Review Step #7 in this getting started guide if you have not added a movement panel.

Remote Control Robot Skills

[opt:skillcategory:remote control]

[opt:skillcategory:virtual reality]

Previous Step

Next Step

11. User Interface

Using remote control features, such as joysticks, is great fun.
However, you may wish to create a custom user interface (UI) for using the PC in tablet mode or with the mouse.
ARC has a user interface builder that allows the interfaces to be displayed fullscreen to the user. In addition, the ARC Remote UI option enables mobile devices to operate the robot remotely using the custom interfaces you have created. The interfaces can have multiple pages that are switched by buttons. These interfaces are optionally displayed on an Android or iOS mobile device using the ARC Remote UI app.

Interface Builder Robot Skill



The Interface builder skill empowers you to create a touch-screen user interface (UI) for your robot. Use buttons, labels, pads, sliders, drop-downs, checkboxes, and displays to make a control panel to activate features. It's a powerful way to interact with your robot.

Detailed information on the Interface Builder manual page will help you configure a custom interface.

Interface Builder Manual

Designing Interfaces

The interface builder provides a graphical designer. You can add and move components on the display in real time.
Add custom scripts to buttons, add joysticks, and more.
Up to 127 interfaces can be defined, and ShowControl() commands allow the user to

navigate through the interfaces.

Default Interface

You can have the user interface automatically displayed to the user when the project is loaded by selecting the option in the Project Properties.

Full-Screen Interface

When creating a custom interface and the "Use Default Interface" option is checked, the current interface will be displayed full-screen in the ARC virtual window tab.

Remote UI

The ARC Remote UI option allows mobile devices to operate the ARC robot remotely using your created custom interfaces. You can add as many interfaces as you'd like using the Interface Builder robot skill. The interfaces can have multiple pages that are switched by a button. These interfaces are displayed on an Android or iOS mobile device using the ARC Remote UI app.

Previous Step

Next Step

12. Autonomous Navigation

Based on your skill level, there are beginner and advanced solutions for navigation and object avoidance. This page provides details and links to various navigation solutions that fit your robot's requirements and budget.

This video tutorial shows you how to add navigation with waypoints to a robot using a 360-Degree Lidar. This same result can be achieved using ARC's Lidars on any robot with a movement panel.

Navigation solutions range from low-cost single sensors to 3D depth cameras and 360-degree lidars.
You can decide what type of sensors to add to your robot and choose the appropriate robot skills.

1. Simple Object Avoidance

The simplest and lowest-cost navigation method uses distance sensors to avoid walls and other objects in the robot's path. This can be done with a single sensor or several sensors. Popular sensors for this simple solution are Ping Ultrasonic and IR Distance.

Ultrasonic Distance Robot Skills

Infrared Distance Robot Skills

2. Autonomous Navigation

ARC contains a messaging system for navigation called the Navigation Messaging System (NMS for short). While there are many navigation sensors, they vary in cost and accuracy.

NMS and compatible sensors.

ARC's most popular SLAM for autonomous navigation system is The Better Navigator, which can navigate between stored points. The Better Navigator takes the average 3D point of a set of waypoints and uses that to navigate.

The sensor data input type is essential for The Better Navigator to perform the desired operation. We recommend combining a 360-degree lidar with The Better Navigator robot skill. See the list below for supported Lidars, depth cameras, and many other navigation robot skills.

Add The Better Navigator robot skill.

SLAM Navigation Robot Skills

Here is a list of ARC robot skills for navigation, including autonomous navigation and remote control. In many cases, it may be necessary to combine autonomous navigation with remote control if the robot gets stuck. Another option is to use Exosphere and have the robot request user assistance when it gets stuck.
[opt:skillcategory:navigation]

Previous Step

Next Step

13. Project Example

Now that you have completed the above steps of the Getting Started guide, you should have the following completed...

- A PC with ARC installed
- ARC connected to an EZB
- Movement Panel robot skill added to the ARC project, configured, and operational
- Camera Device robot skill added to the ARC project and operational
- Optionally, a microphone on the PC with generic speech recognition robot skill added to the ARC project
- The robot is powered by a suitable battery or power supply

Next Steps

Let's begin adding new robot skills that extend your robot's capabilities. The next step of this tutorial will highlight a few of the most popular robot skills.

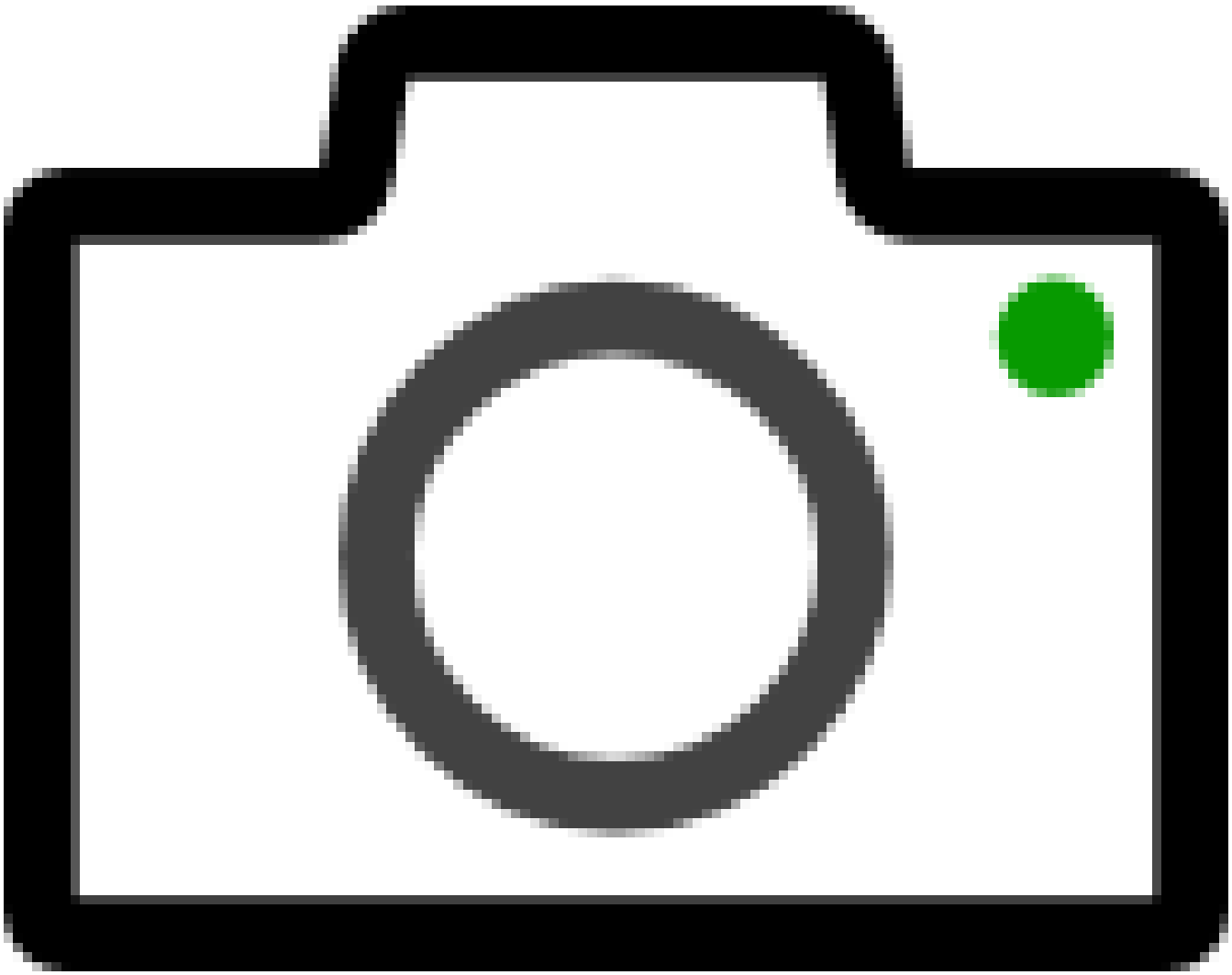
Previous Step

Next Step

14. Most Popular Robot Skills

By reaching this point of the Getting Started Guide, your robot has a movement panel, servos, camera, and possibly speech recognition.
However, hundreds of robot skills are available in the Robot Skill Store to extend your robot's capabilities.
We will cover the most popular robot skills on this page.
Remember that robot skills covered in previous steps of this guide will not be included in this list (i.e., movement panels or speech recognition).

Camera Device

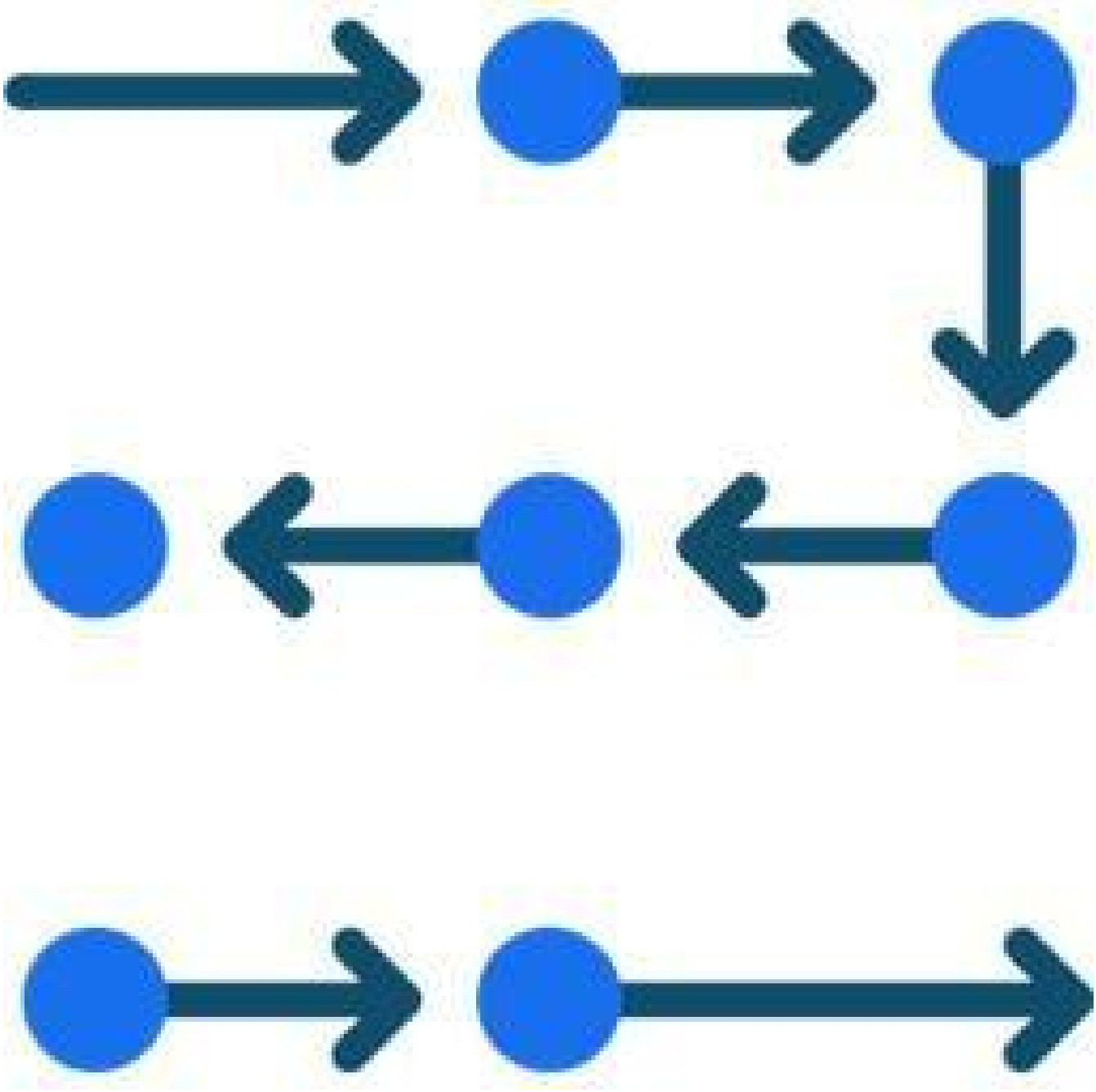


Previously covered in step 8 of this guide, one of the most powerful and popular robot skills demonstrating ARC's power is the Camera Device.
This robot skill will use either the EZB remote camera (if supported), or USB webcam mounted on the robot.
By configuring options, you can have the camera track objects, faces, and colors, and even detect emotion or age with additional robot skills.
The Camera Device has settings in the options menu for controlling servos when a camera is mounted on a gimbal.
This feature allows the robot's camera to move and track the object.

Get Camera Device

All Camera Robot Skills

Task Script



This enables you to assign tasks to your robot. A task could be something like going to the kitchen and returning with a beverage.
The task could be split up into many steps. This robot skill allows your robot to operate in a "task mode" by executing many steps to achieve a task.

Split complex tasks into several simple scripts. Each script is a stage of completing a task, executed one after another. For example, you can use this to have your robot navigate into a room, look for an item, move toward the object, pick up the article, and return to another point. Each task step is split into stages with a script for each stage.

Exosphere



Exosphere combines telepresence, artificial intelligence, machine learning, and human assistance to give autonomous behaviors to cloud robotics.

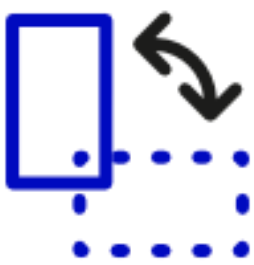
Exosphere is a game-changing product that allows users to control robots remotely to complete tasks anywhere with an internet connection. You can use your phone or tablet to control a robot or add your robot to be controlled by others. This product gives you the

power to help make the world safer and improve business efficiency.

Get Exosphere
Try Exosphere Playground

All Remote Control Robot Skills

Auto Position



Create servo animations to have your robot wave, dance, or pick up and move objects.
The Auto Position skill transforms a group of servos into custom positions (Frames) to create animations.
You can combine the frames into actions for your robot's animation.
This is done with "Inverse Kinematics" or "Motion Planning."

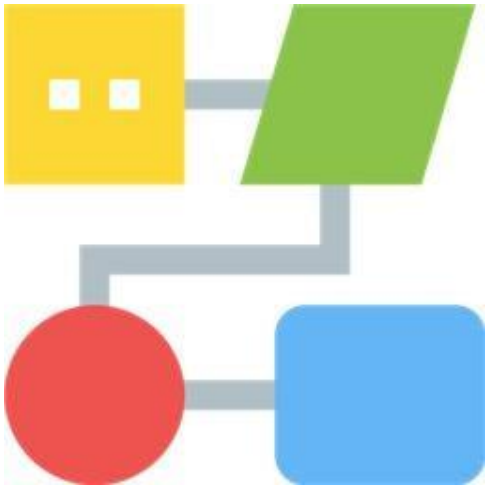
There are two types of the Auto Position robot skill. One is a movement panel that plays servo animations for robots that use servos to walk (i.e., humanoids or hexapods).
The other type is used for robots that use a different movement panel (i.e., hbridge, continuous rotation, etc.) but has servos for arms to be animated.

This is a popular robot skill for humanoid robots, such as the InMoov, Robotis Bioloid, or EZ-Robot JD/Six.
For robots with many servos used in arms or legs, this allows creating animations that can be executed programmatically by other robot skills using the ControlCommand().
This means your robot can perform actions based on speech recognition or chatbot conversations.

Auto Position with movement panel

Auto Position for only servos

Conversation Menu



Have a conversation with your robot using your voice by navigating through menu options so your robot can perform tasks. This is similar to how menu trees on phone systems work. You can define the menu tree in the configuration to run scripts based on options selected by the user with their voice. The menu tree can have as many branches as necessary.

The robot will speak each menu prompt and optionally speak each option to the user. The microphone on the robot or PC is used for the user's response. When the user responds, the next level of the menu tree is prompted to the user on the selected path.

Get Conversation Menu

Open AI



Give your robot a personality and conversational ability. This robot skill adds chatbot capability to your robot so you can have meaningful conversations that it will remember. Configure the personality in any way you wish so the robot can prefer specific subjects or have particular opinions.

Get Open AI

All AI Chatbot Robot Skills

Tip: Hundreds of robot skills are available in the Robot Skill Store. Visit the robot skill store to explore the entire library of robot skills.
Visit Robot Skill Store

Previous Step

Next Step

15. Programming Introduction

Synthiam has developed the ARC software to scale between beginner and advanced users. Our mission is to make robot programming accessible and increase the likelihood of success. Within the configuration dialog of nearly all skill controls is the ability to edit scripts to define specific programmatic behaviors, such as when to start tracking using the camera. The script editors are input fields with a pencil icon to the right. In this example, we are displaying the configuration dialog for the WiiMote.

Editor Dialog

The editor dialog consists of an input section on the left and reference assistance on the right. The window can be resized or maximized; remember the last size for future use.

Program in Multiple Languages

The programming language tab will be selected if there is existing code in the editor when it loads.
Alternatively, if you are editing a blank script, there will not be any code, and you can select the tab for the programming language you wish to use.
If there is existing code, changing the programming language will present a confirmation that you want to erase the current code and start over.

Introducing the Control Command

Programming in ARC is straightforward because you will leverage robot skills' functionality. Whichever programming language you choose below, most of your programming will link robot skills together using the ControlCommand.

You are familiar with Robot Skills by reaching this far in the Getting Started guide. There are plenty of windows within the ARC project desktop, and each window is a robot skill control. A robot skill control is a little program that runs within ARC created by one of our many partners. For example, hundreds of robot skill controls are available in the Robot Skill Store, such as gpt-3, vision detection (face, emotion, color, glyph, etc.), servo gait/animation, speech recognition, etc. Each robot skill control is a behavior that gives your robot more ability. These robot skill controls are separate programs that do something specific, such as processing video image data from the camera or moving servos in animations. Because each robot skill control is a particular program, ARC provides a mechanism for robot skills to *talk to each other*. This mechanism is called the **ControlCommand()**.

Using control commands, an event of one robot skill can instruct another robot skill to do something. For example, if the Speech Recognition control detects the phrase "Follow My Face," the respective code may be instructed to inform the Camera control to enable Face Tracking.

This detailed example detects speech recognition text and uses cognitive services for more extensive messages. The manuals for each robot skill will be essential to review when creating projects.

For further reading about the ControlCommand and how it works, we recommend reading its dedicated manual page here.

Choose Programming Environment

We have provided access to various programming methods to accommodate all users. Once you are ready to begin programming a robot, select a programming skill level.

Roboscratch

(Very easy)

Designed for learning the basics of programming, RoboScratch introduces a programming interface exclusive to ARC. With RoboScratch, function blocks are added to the workspace to create a sketch.
Some blocks include waiting for the camera to see an object, waiting for speech, or executing a movement action.
Link blocks to instruct robots to perform behaviors and complete tasks.

Learn More

Blockly

(Easy)

Blockly programming provides the user with graphical blocks representing programming functions or subroutines.
This allows users to access advanced technologies creatively without knowing the programming syntax by typing with a keyboard.

Learn More

Scripting

(Intermediate)

Syntax programming is available in three scripting languages JavaScript, EZ-Script, and Python. All robot skills support scripting languages interacting with each other for custom behaviors.
Using the powerful ControlCommand messaging system, have events from robot skills send triggers to other robot skills within the project.

Javascript

EZ-Script

Python

Programming
(Advanced)

Compile programs and libraries which interface with the ARC API. Make robot skills to distribute in the robot skill store or use the existing framework to accelerate robot development.
Access existing robot skills or the ARC API framework with C++, C, C#, VB, and more.

Learn More

Scripting Robot Skills
Nearly all robot skills have the option to execute a script. However, there are robot skills specific to scripting. These scripting robot skills allow you to run programs or customize features, such as creating a custom movement panel.
[opt:skillcategory:scripting]

Tutorials

AI Robot Chat

This tutorial shows how to connect speech recognition to an AI chatbot so your robot can hear what you say, send that text to a PandoraBot, and speak the response. You can follow these steps in an existing project or start from a blank project.

Example of a robot conversation using speech recognition and PandoraBot.

Requirements

- Synthiam ARC installed
- Working microphone
- Sound card or an EZ-B with audio support (see EZB hardware overview to check compatibility)
- An active internet connection

Robot Skills Used

The tutorial uses these robot skills. Click any item for the skill manual:

- Bing Speech Recognition
- PandoraBots

Programming Concepts Used

This tutorial demonstrates these ARC programming concepts and APIs:

- JavaScript in ARC
- ControlCommand()

Steps

1. Add the Bing Speech Recognition skill
   In ARC, add the Bing Speech Recognition robot skill from the top menu: Project > Add Robot Skill > Speech Recognition > Bing Speech Recognition.

   Selecting the Bing Speech Recognition skill from the Add Robot Skill menu.

2. Add the PandoraBot skill
   From the top menu, add the PandoraBot skill: Project > Add Robot Skill > Artificial Intelligence > PandoraBots.

   Selecting the PandoraBots skill from the Add Robot Skill menu.

3. Open the Bing Speech Recognition config
   Open the Bing Speech Recognition skill and press the **Config** button in its menu bar to open its configuration window.

Click the Config button to access the Bing Speech Recognition settings.

4.

Edit the "All Recognized Scripts"

Click the **All Recognized Scripts** button. This script runs for every phrase the speech recognition translates. Note the variable shown — you will reference it shortly.

The All Recognized Scripts area handles every recognized speech phrase.

5.

Switch to the JavaScript editor

When the script editor opens, switch to the **JavaScript** tab to insert code that will forward recognized text to PandoraBot.

Select the JavaScript tab to write the forwarding script.

6.

Insert the PandoraBot SetPhrase command

Right-click in the editor, choose **PandoraBot**, then select the **SetPhrase** option. This inserts the ControlCommand call used to send text to the chatbot.

Use the context menu to insert the PandoraBot SetPhrase command into your script.

7.

Replace the placeholder phrase

With the inserted code visible, highlight the placeholder text ("[phrase]"). You will replace it with the speech variable that contains the translated text.

Highlight the placeholder text to prepare for the variable insertion.

8.

Insert the $BingSpeech variable

Open the **Global Variables** tab, find the `$BingSpeech` variable, and click it. This inserts the variable at the cursor location, replacing the placeholder. The variable contains the text recognized by the Bing skill.

Insert the $BingSpeech global variable so the recognized text is forwarded to PandoraBot.

9.

Resulting code

The final code instructs ARC to send the contents of `$BingSpeech` to the PandoraBot skill using ControlCommand. It should look like this:

```
ControlCommand("PandoraBot", "SetPhrase", getVar("$BingSpeech"));
```

This line uses ControlCommand() to call the PandoraBot skill's SetPhrase action with the recognized speech text.

10.

Save the script

Press **SAVE** in the script editor to store your changes.

Save your JavaScript changes in the editor.

11.

Enable voice activity detection (optional but recommended)

Check **Auto Record using VAD** to let ARC automatically detect human speech and begin recording when you speak. Hover the blue question mark for details on how VAD works.

Enable Auto Record using VAD to automatically start recording when speech is detected.

12.

Save the Bing Speech configuration

Press **SAVE** in the Bing Speech Recognition configuration to apply all changes to the robot skill.

Save the configuration to activate the script and settings.

13.

Optional: Use the EZ-B speaker for PandoraBot TTS

If you have an EZ-B with supported audio (for example EZ-Robot EZ-B v4 or IoTiny), you can configure PandoraBot to play speech through the EZ-B speaker. Open the PandoraBot config (three dots in the title bar), check **Use EZ-B v4 Speaker**, and press save.

Enable the EZ-B speaker option to have PandoraBot responses play through the EZ-B device.

14.

You're done — test the conversation

Speak to the robot. The Bing Speech Recognition skill will translate your speech to text, the script will forward that text to PandoraBot, and PandoraBot will respond. Verify recording and audio output are working on your system.

When configured correctly, your robot can converse by forwarding recognized speech to PandoraBot and playing the response.

Notes & Troubleshooting

- Ensure your microphone and audio output are working before testing.
- If responses are not heard, confirm the PandoraBot TTS settings and your audio routing (local speakers or EZ-B speaker).
- Use the global variable `$BingSpeech` whenever you need the translated text from Bing Speech Recognition.

Detect Face Expression

Init Script

An initialization script is a Script Robot Skill used to specify settings for the robot and initialize other robot skills. Generally, this type of script is executed when the EZB connection is successful. Or, this script can be executed when ARC loads the project after Windows starts up. This initialization script will start when Windows loads and the script establishes a connection to the EZB. We'll cover both examples of an initialization script here.

Requirements

- ARC project with a working robot

Robot Skills Used

The following robot skills are used in this tutorial. Click any of these to further read the manuals for robot skills.

- Script
- EZ-B Connection

Init Script Robot Skill

An initialization script will be a Script Robot Skill. This is a popular robot skill for creating scripts for writing and executing scripts. The script robot skill for a project is highlighted in the image above.

Script Robot Skill Manual

a) Init Script Example For Windows Startup

This initialization script will be executed when Windows starts or when a shortcut is clicked. The ARC shortcut creator can be configured to execute a script when a project shortcut is loaded. With a init script executed when Windows loads, you may want the script to establish a connection to the EZB. This is different behavior than the other use of an init script after the connection is established. A project could consist of both, but it's up to your requirements.

Shortcut Creator Manual

This is an example of an init script for when Windows starts up. This script will connect to the EZB and then configure the project. In this script, the first command establishes a connection to the EZB. Once connected, the servo limits are specified. The servo limits prevent the servos from moving past the physical positions of the robot. Then, the auto

position is instructed through a control command to move the robot into a "Standing" frame. The speeds of a few servos are specified, the speed of the movement is specified, the camera is initialized, and the camera motion tracking is enabled. Finally, the robot will speak at the end of the script and mention that it is ready.

When the shortcut creator wizard is used, you can specify the script to execute when the shortcut is clicked (or when Windows starts). This script will execute and initialize the robot. This is an example init script for a robot with a dedicated PC or embedded computer.

```
// Connect to the EZB
ControlCommand("Connection", "Connect0");

// specify the ranges for the servos
// this prevents the servos from ever going past these limits

// Neck
Servo.setMinPositionLimit(d1, 70);

// Left Gripper
Servo.setMinPositionLimit(d6, 30);
Servo.setMaxPositionLimit(d6, 90);

// Right Gripper
Servo.setMinPositionLimit(d9, 30);
Servo.setMaxPositionLimit(d9, 90);

// Left Ankle
Servo.setMinPositionLimit(d14, 60);
Servo.setMaxPositionLimit(d14, 120);

// Right Ankle
Servo.setMinPositionLimit(d18, 60);
Servo.setMaxPositionLimit(d18, 120);

// Initialize JD into the initialize frame
// All servos are set to 90 degrees in this frame
ControlCommand("Auto Position", "AutoPositionFrameJump", "Standing");

// Pause for a tiny bit to ensure the servos have been initialized
sleep(100);

// Begin an animation of his eyes
ControlCommand("RGB Animator", AutoPositionAction, "spin");

// set the speed for JD's head to be a little slower and smoother
Servo.setSpeed(d0, 3);
Servo.setSpeed(d1, 3);

// Initialize the movement speed
Movement.setSpeed(255);

// start the camera so the user doesn't have to press the start button on the camera
if (!getVar("$IsCameraActive"))
  ControlCommand("Camera", "CameraStart");

// Enable camera motion tracking for security warning
ControlCommand("Camera", "CameraMotionTrackingEnable");

Audio.sayEZB("I am initialized and ready for action");
```

b) Init Script Example For EZB Connection

The other example for using an init script after an EZB establishes a connection. This type of

script is used when a robot is to be initialized manually after a successful EZB connection is made. The ARC connection control will send a Control Command to the init script.

The image above shows how the connection control will command the init script control when an EZB connection is established. First, the Init script will need some code. In the example below, you will notice the script is similar to the previous example, with the difference being there is no command to establish an EZB connection. There is no control command to establish an EZB connection because ARC will execute this script after the connection control connects the EZB.

```
// specify the ranges for the servos
// this prevents the servos from ever going past these limits

// Neck
Servo.setMinPositionLimit(d1, 70);

// Left Gripper
Servo.setMinPositionLimit(d6, 30);
Servo.setMaxPositionLimit(d6, 90);

// Right Gripper
Servo.setMinPositionLimit(d9, 30);
Servo.setMaxPositionLimit(d9, 90);

// Left Ankle
Servo.setMinPositionLimit(d14, 60);
Servo.setMaxPositionLimit(d14, 120);

// Right Ankle
Servo.setMinPositionLimit(d18, 60);
Servo.setMaxPositionLimit(d18, 120);

// Initialize JD into the initialize frame
// All servos are set to 90 degrees in this frame
ControlCommand("Auto Position", "AutoPositionFrameJump", "Standing");

// Pause for a tiny bit to ensure the servos have been initialized
sleep(100);

// Begin an animation of his eyes
ControlCommand("RGB Animator", AutoPositionAction, "spin");

// set the speed for JD's head to be a little slower and smoother
Servo.setSpeed(d0, 3);
Servo.setSpeed(d1, 3);

// Initialize the movement speed
Movement.setSpeed(255);

// start the camera so the user doesn't have to press the start button on the camera
if (!getVar("$IsCameraActive"))
  ControlCommand("Camera", "CameraStart");

// Enable camera motion tracking for security warning
ControlCommand("Camera", "CameraMotionTrackingEnable");

Audio.sayEZB("I am initialized and ready for action");
```

We will configure the connection control and specify it to start the init script when an EZB connection is established. Press the configure button on the connection control.

Select the script for the first EZB established connection.

Right-click in the editor, select the Init script menu item, select the option to start the script, and insert the Control Command code. Save the changes to this connection control, and now when the EZB is connected, ARC will execute the init script.

Phone Notification from Exosphere

This example will demonstrate how to have Exosphere push notifications to your mobile device (iOS or Android). The notification will alert you when someone has started and stopped controlling the robot. This tutorial expects the robot project to already be operational by containing both a Movement Panel and a Camera.

Requirements

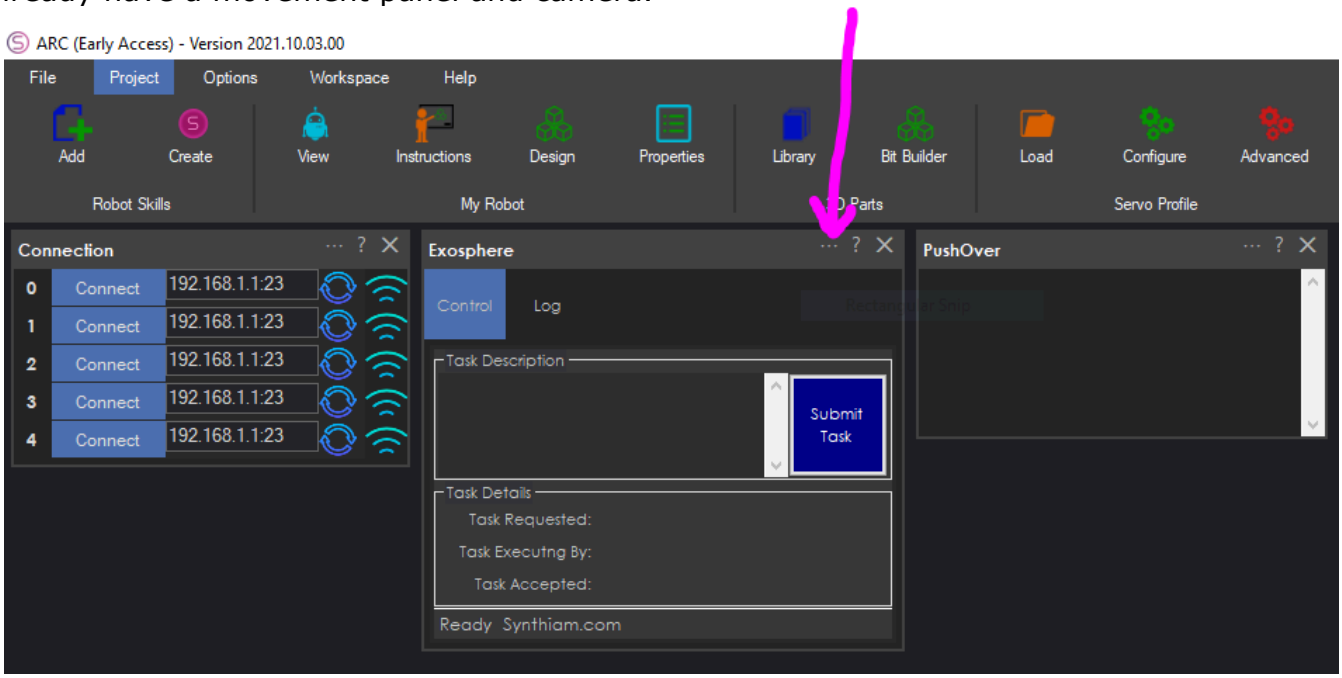
- ARC Project with configured Movement Panel & Camera
- PushOver robot skill (configured from the manual)
- Internet Connection

Robot Skills Used

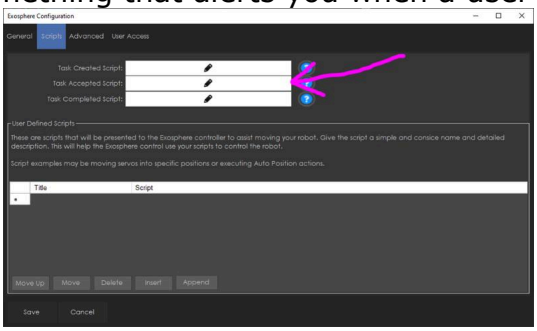
The following robot skills are used in this tutorial. Click any of these to further read the manuals for the robot skills.

- Exosphere
- Camera Device
- PushOver

1) Add the Exosphere and PushOver robot skills to your ARC project. The ARC project must already have a movement panel and camera.

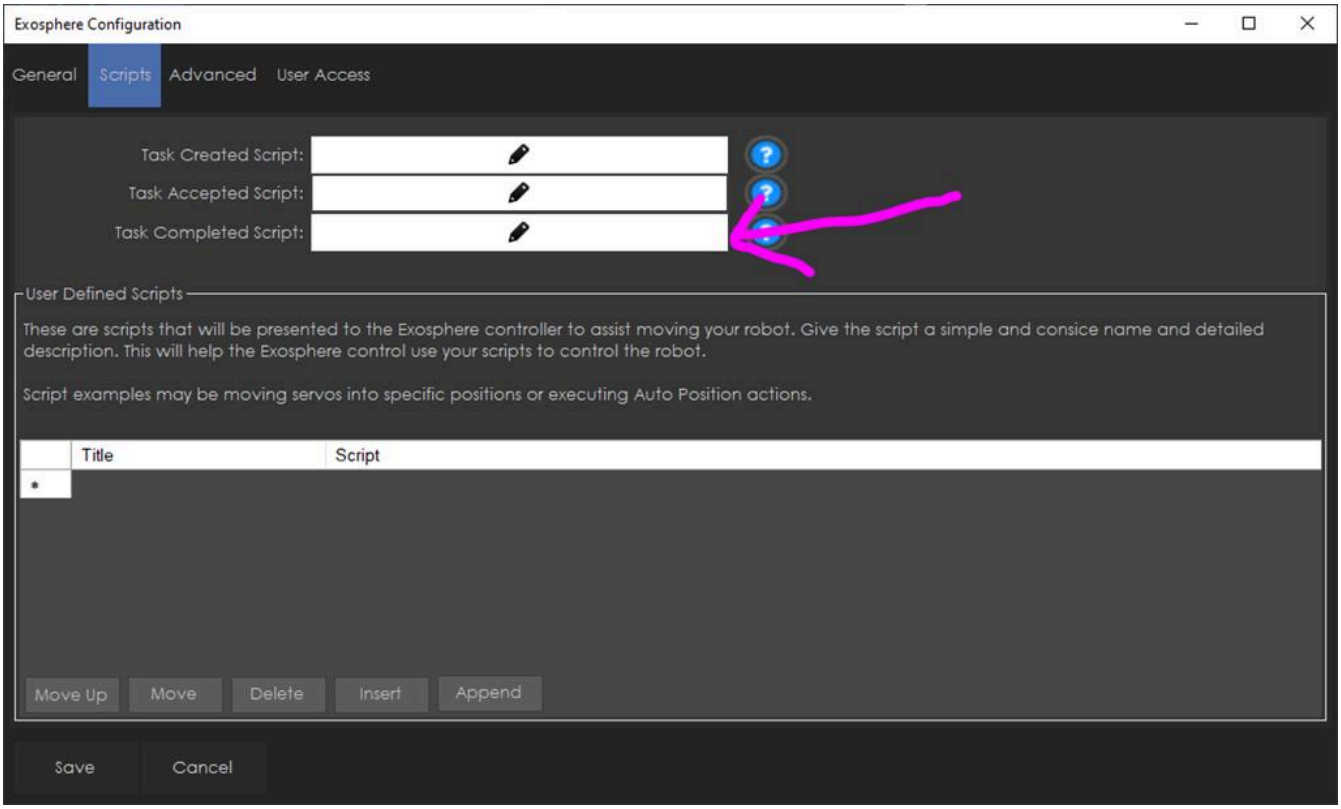


2) In Task Accepted, add something that alerts you when a user chooses the task...



[code]
ControlCommand("PushOver", "Push", "Exosphere task started by " +
getVar("$ExospherePuppeteerUsername"));
[/code]

3) Finally, add another script for the Task Completed. This will notify you when the task is over



[code]
ControlCommand("PushOver", "Push", "Exosphere task completed by " +
getVar("$ExospherePuppeteerUsername"));
[/code]

Moving a Servo

The first and most important action of any robot is moving a limb with a servo. Even if your robot is not complete, it's a good idea to practice moving a servo with ARC to become familiar with how easy it is. Because it is easy to move servos with ARC, you can now focus on the creative aspect of the robot. This tutorial will add a servo robot skill, configure the servo robot skill, and move a servo.

Requirements

- Connected to an EZB
- Servo connected to the EZB

Robot Skills Used

The following robot skills are used in this tutorial. Click any of these to further read the manuals for the robot skills.

- Horizontal Servo
- Servo

Quick Tutorial

Press the Project button from the top menu, and select Add Robot Skill.

The Robot Skills window will display. Navigate to the Servo tab and select Horizontal Servo.

Now the Horizontal Servo robot skill is visible in your project. Press the Config button on the Horizontal robot skill.

Select the Servo Port that your servo is connected to on the EZB. The servo port will vary based on the type of EZB used.

Either drag the Min and Max values to specify the range, or right-click and enter the range with the keyboard. By default, the global range in ARC is between 1-180 to accommodate PWM hobby servos. You can alter this global value for high precision servos in the ARC settings menu.

Press SAVE on the config window to save the settings.

Click the mouse on the horizontal robot skill servo position display, and drag LEFT and RIGHT. Watch as the servo moves.

This is only the beginning...

Moving a servo with this robot skill is exciting, but the following steps are even more exciting. When several servos are connected to a robot for arms or grippers, you can configure the Auto Position robot skill to re-create animations. This is a popular robot skill for humanoid robots, such as the InMoov. Also, the Getting Started guide has a movement page that lists the different wants you can make a robot move.

Check out any of these related links...

[opt:skillcategory:servo]

Moving Servo with Speech Recognition

This tutorial is mainly focused on speech recognition. Making a servo move is an example of adding user-defined phrases to the speech recognition robot skill. While there are many speech recognition robot skills available in ARC, we will be using the default Speech Recognition robot skill. This tutorial will add the speech recognition robot skill and configure it to move the servo in a few positions.

*Note: While a built-in computer microphone will work, it is recommended to use a handheld, headset, or dedicated microphone that doesn't pick up ambient sound. The speech recognition system works best when it only hears your voice. Other speech recognition robot skills, such as the Bing Speech Recognition, have higher quality recognition and can successfully detect speech in a noisy environment. The speech recognition robot skill in this tutorial is not as accurate, and therefore we recommend the dedicated microphone.

Requirements

- Connected to an EZB
- Servo connected to the EZB
- Computer with a soundcard and microphone

Robot Skills Used

The following robot skills are used in this tutorial. Click any of these to further read the manuals for the robot skills.

- Speech Recognition
- JavaScript Programming Language
- Servo

Let's Do It

Add the Speech Recognition robot skill. Do this by pressing the Project -> Add Robot Skill from the top menu.

When the Add Robot Skill window opens, select the Audio tab and click the Speech Recognition robot skill. This will add the robot skill to the project.

Now that the robot skill is added to your project press the configuration button on the speech recognition robot skill. This will open the configuration window for the skill.

In the configuration window, there will be several default phrases with code. These default phrases are bound to movements to control a movement panel. We'll be adding two of our phrases to recognize.

Enter the phrase "Move Servo Left" in the first available empty phrase.

Press the Command button next to the new phrase to open the script window.

Select the Javascript tab. And enter the following command—substitute port D0 with the port of your servo. Also, if you are aware that position 1 extends the limits of your robot servo, use a safe position value. Press the SAVE button

```
Servo.setPosition(d0, 1);
```

Enter the phrase "Move Servo Right" in the first available empty phrase.

Press the Command button next to the new phrase to open the script window.

Select the Javascript tab. And enter the following command—substitute port D0 with the port of your servo. Also, if you are aware that position 1 extends the limits of your robot servo, use a safe position value. Press the SAVE button

```
Servo.setPosition(d0, 180);
```

Press SAVE on the speech recognition configuration window

Speak the commands "Move Servo Left" and "Move Servo Right". Notice the servo moving.

More Stuff

There are other robot skills to explore for speech recognition. The recommended robot skill for the highest quality is the Bing Speech Recognition robot skill. The configuration of this robot skill is similar to this tutorial. One of the advantages of the Bing Speech Recognition robot skill is that it will translate any speech, not only the defined phrases. You can feed the output to a chatbot to give your robot conversational capabilities by detecting any speech.

- Bing Speech Recognition robot skill

  The recommended robot skill provides higher quality detection and more capabilities for customization.

- Artificial Intelligence robot skills

  This category of robot skills contains several chatbots that give the robot conversational capabilities.

- Robot Chat Video Example

  This video is an example of combining Bing Speech recognition with a chatbot, camera, and cognitive services.

- ControlCommand() manual

  When you're ready to have speech recognition send commands to other robot skills, this manual is required reading. The ControlCommand is a programming command that simulates pressing a button on another robot skill. For example, you would use this command in a detected phrase to enable camera motion tracking. Or use this command in speech recognition to trigger an Auto Position action.

Auto Position Action from Speech Recognition

This tutorial is mainly focused on speech recognition. We will demonstrate how to have the speech recognition robot skill trigger an auto position action. This is an example of adding user-defined phrases to the speech recognition robot skill. While there are many speech

recognition robot skills available in ARC, we will be using the default Speech Recognition robot skill.

*Note: While a built-in computer microphone will work, it is recommended to use a handheld, headset, or dedicated microphone that doesn't pick up ambient sound. The speech recognition system works best when it only hears your voice. Other speech recognition robot skills, such as the Bing Speech Recognition, have higher quality recognition and can successfully detect speech in a noisy environment. The speech recognition robot skill in this tutorial is not as accurate, and therefore we recommend the dedicated microphone.

Requirements

- Connected to an EZB
- Configured Auto Position with defined actions
- Computer with a soundcard and microphone

Robot Skills Used

The following robot skills are used in this tutorial. Click any of these to further read the manuals for the robot skills.

- Auto Position or Auto Position Movement Panel
- Speech Recognition
- JavaScript Programming Language
- ControlCommand()

Let's Do It

Add the Speech Recognition robot skill. Do this by pressing the Project -> Add Robot Skill from the top menu.

When the Add Robot Skill window opens, select the Audio tab and click the Speech Recognition robot skill. This will add the robot skill to the project.

Now that the robot skill is added to your project press the configuration button on the speech recognition robot skill. This will open the configuration window for the skill.

In the configuration window, there will be several default phrases with code. These default phrases are bound to movements to control a movement panel. We'll be adding two of our

phrases to recognize.

Enter the phrase "Robot do my action" in the first available empty phrase.

Press the Command button next to the new phrase to open the script window.

Press the Javascript tab. In the editor, right-click and select the auto position. Scroll and select the action item with the mouse you wish to execute.

Press the SAVE button on the script window.

Press SAVE on the speech recognition configuration window

Speak the commands "Robot do my action". Notice the auto position action will execute.

More Stuff

There are other robot skills to explore for speech recognition. The recommended robot skill for the highest quality is the Bing Speech Recognition robot skill. The configuration of this robot skill is similar to this tutorial. One of the advantages of the Bing Speech Recognition robot skill is that it will translate any speech, not only the defined phrases. You can feed the output to a chatbot to give your robot conversational capabilities by detecting any speech.

- Bing Speech Recognition robot skill

  The recommended robot skill provides higher quality detection and more capabilities for customization.

- Robot Chat Video Example

This video is an example of using the ControlCommand to control other robot skills for conversation interaction. Such as the Bing Speech recognition with a chatbot, camera, and cognitive services.

Move a servo with speech recognition (Blockly)

This tutorial is mainly focused on speech recognition. Making the servo move is an example of adding user-defined phrases to the speech recognition robot skill. While there are many speech recognition robot skills available in ARC, we will be using the default Speech Recognition robot skill. This tutorial will add the speech recognition robot skill and configure it to move the servo in a few positions.

*Note: While a built-in computer microphone will work, it is recommended to use a handheld, headset, or dedicated microphone that doesn't pick up ambient sound. The speech recognition system works best when it only hears your voice. Other speech recognition robot skills, such as the Bing Speech Recognition, have higher quality recognition and can successfully detect speech in a noisy environment. The speech recognition robot skill in this tutorial is not as accurate, and therefore we recommend the dedicated microphone.

Requirements

- Connected to an EZB
- Servo connected to the EZB
- Computer with a soundcard and microphone

Robot Skills Used

The following robot skills are used in this tutorial. Click any of these to further read the manuals for the robot skills.

- Speech Recognition
- Blockly
- Servos

Let's Do It

Add the Speech Recognition robot skill. Do this by pressing the Project -> Add Robot Skill from the top menu.

When the Add Robot Skill window opens, select the Audio tab and click the Speech Recognition robot skill. This will add the robot skill to the project.

Now that the robot skill is added to your project press the configuration button on the speech recognition robot skill. This will open the configuration window for the skill.

In the configuration window, there will be several default phrases with code. These default phrases are bound to movements to control a movement panel. We'll be adding two of our phrases to recognize.

Enter the phrase "Move Servo Left" in the first available empty phrase.

Press the Command button next to the new phrase to open the script window.

Select the Blockly tab. Specify the following block as in the image. The servo position block is found in Movement, and the number for the position is found under Math. Substitute port D0 with the port of your servo. Also, if you are aware that position 1 extends the limits of your robot servo, use a safe position value. Press the SAVE button

Enter the phrase "Move Servo Right" in the first available empty phrase.

Press the Command button next to the new phrase to open the script window.

Select the Blockly tab. Specify the following block as in the image. The servo position block is found in Movement, and the number for the position is found under Math. Substitute port D0 with the port of your servo. Also, if you are aware that position 180 extends the limits of your robot servo, use a safe position value. Press the SAVE button.

Press SAVE on the speech recognition configuration window

Speak the commands "Move Servo Left" and "Move Servo Right". Notice the servo moving.

More Stuff

There are other robot skills to explore for speech recognition. The recommended robot skill for the highest quality is the Bing Speech Recognition robot skill. The configuration of this robot skill is similar to this tutorial. One of the advantages of the Bing Speech Recognition robot skill is that it will translate any speech, not only the defined phrases. You can feed the output to a chatbot to give your robot conversational capabilities by detecting any speech.

- Bing Speech Recognition robot skill

  The recommended robot skill provides higher quality detection and more capabilities for customization.

- Artificial Intelligence robot skills

  This category of robot skills contains several chatbots that give the robot conversational capabilities.

- Robot Chat Video Example

  This video is an example of combining Bing Speech recognition with a chatbot, camera, and cognitive services.

- ControlCommand() manual

  When you're ready to have speech recognition send commands to other robot skills, this manual is required reading. The ControlCommand is a programming command that simulates pressing a button on another robot skill. For example, you'd use this command in a detected phrase to enable camera motion tracking or use this command in speech recognition to trigger an Auto Position action.

Robot Has Conversation

The first thing you will want to do is merely experiment with what chatbot you'd like to use.

There are a number of them on Synthiam's platform, as you can see in the skill store: https://synthiam.com/Products/Controls/Artificial-Intelligence
.

For beginners, I would probably recommend using the AIMLBot because it is very configurable and has a feature that you require to understand who is looking at the robot by the camera. So, install the AIMLBot here: https://synthiam.com/Support/Skills/Artificial-Intelligence/AimlBot?id=16020
.

Make Chatbot Speak

The chatbot won't speak by default. It'll display the output in the log window. Let's edit the chatbot to add some python code to make it talk out of the PC speaker OR EZB speaker - whatever you choose. View the aimlbot configuration and select the response script.

Now under the Python tab, add one of these... depends if you want the audio out of the PC or EZB.

```
Code:// speak out of the PCAudio.Say(getVar("$BotResponse"));// speak out of the
EZBAudio.SayEZB(getVar("$BotResponse"));
```

Speech Recognition

Now you need to speak to the robot. There are dozens of speech recognition modules, but Bing Speech Recognition is preferred. That is very reliable and configurable for things like this. You can install it here: https://synthiam.com/Support/Skills/Audio/Bing-Speech-Recognition?id=16209
.

Connect Speech Recognition to Chatbot

Now you need to connect the speech recognition to the chatbot. So that when you speak, it pushes the detected phrase into the AIML chatbot. View the bing speech recognition configuration screen and add this code to the All Recognized Scripts. Since you're using python, I used the python tab.

```
Code:ControlCommand("AimlBot", "SetPhrase", getVar("$BingSpeech"));
```

Once you save that configuration setting, you can start talking to the robot, and the chatbot will print responses back.

Now that you have the chatbot working let's add the camera stuff to make the robot change who it is seeing...

The next step is to make the head move. Just use the
Auto Position
robot skill, and that will be best. Install it from here:
https://synthiam.com/Support/Skills/Servo/Auto-Position-Gait?id=20314

There's more than enough information in the
Auto Position
manual
to explain how to create your animations.

Here is the project all working:
aiml chatbot.EZB
. All you need to add next is the
Auto Position
for having the robot head servos move around. You can also configure the camera device to move the robot head's eyes to follow the face. That are just a few settings in the camera device.

Select the Track by relative position option if the camera is stationary and not moving. But here are the camera settings you need to have the robot's eyes move and follow the face.

Record Servo Positions To Auto Position Frames

In this tutorial, we'll demonstrate how to create Auto Position robot skill animations by recording the servos' positions after you manually move them. The Auto Position robot skills create animations and gait for walking. While this isn't a manual for the auto position robot skill, we'll be using the Auto Position robot skill in this tutorial.

Prerequisite

- Servo Manual

Not all servos are alike, so we have written a fantastic manual explaining the differences. It is highly recommended that the servo manual be read before building a robot.

Introduction To Servo Motors

- Auto Position

It's a good idea to read the Auto Position Robot Skill manual to familiarize yourself with how it works. This ensures you know how to create and link Frames into Actions. You should also know the two types of Auto Position Robot Skills to ensure you use the correct one. One is a Movement Panel (walking forward, etc.), and the other is for simple animations complimenting an existing movement panel. If this doesn't sound familiar, the auto position manual will teach you what those terms mean.

Auto Position Movement Panel Manual

Auto Position Manual

***Note:** *This tutorial will require adding an Auto Position robot skill and configuring the servos. Use the Auto Position manual to accomplish those steps.*

Compatible Servos

It is important to note that not all servos are compatible with the ability to transmit their position. This is because the most popular type of servo is a PWM Hobby Servo, and they only receive a position command but do not transmit their position. The type of servos that will transmit their position are Serial Smart Servos, such as Dynamixel, LewanSoul, LynxMotion, etc. This is documented in the Introduction To Servo Motors, which is mentioned as a pre-requisite above.

- PWM Servos

The PWM Hobby Servos can be modified to transmit their position. ARC can decipher the servo position using a compatible robot skill. The manual for the robot skills demonstrates how to modify the PWM servo for this use case.

PWM Servo Feedback (ADC)

PWM Servo Feedback (i2c)

-

Serial Smart Servos

As documented in the Introduction To Servo Motors mentioned in the pre-requisite section of this tutorial, smart servo motors can transmit their position. This is because these types of servos use a bi-directional communication protocol. Here are links to the most popular ARC-supported smart servo motors.

Dynamixel (most popular)

FeeTech RS485

FeeTech Serial Bus

Kondo KRS ICS

LewanSoul

Lynx Motion

Tutorial Steps

1.
Configure Servos

The first step of this tutorial is to verify that you already have an ARC project with servos that move. Also, if the servos are smart servos, the servos are connected correctly for bi-directional communication. Or, if they're PWM Hobby Servos, the selected PWM Servo robot skill is selected.

2.
Add Auto Position

Using the information from the pre-requisite above, add the Auto Position robot skill that best suits your needs.

3.
Add New Frame

1) Press the CONFIGURE button on the Auto Position to load the configuration window.

2) Press the NEW FRAME button and enter *Test 1* as the new frame name

3) Select the *Test 1* new frame that you just added

4.
Release All Servos

Press the Release All Servos button to have the servos stop holding their position. This is required so that you can manually move the servos into position. Without doing this, the servos will hold their position, and it will be impossible to move them by hand.

5.
Move Servo By Hand

Move the servos into a position by hand.

6.
Get & Set All Positions

Press the "Get & Set All Positions" button. This will retrieve all servo positions and set their values for the selected frame.

7.
Add More Frames

Continue to repeat the previous steps by adding multiple frames with the servos in each position.

8.
Add Frames To An Action

Now that you have created all the frames, you can play them back in any order by adding them to a new Action.

Speech Recognition With Push Button Microphone

This tutorial shows how to use a physical walkie-talkie-style button connected to your EZB to control when the

Bing Speech Recognition robot skill listens to and transcribes speech in Synthiam ARC. You can use other speech recognition robot skills by substituting the variables and control commands. Instead of relying on wake words or always-listening Voice Activity Detection (VAD), you'll build a

reliable *push-to-talk* style system: press and hold the button to record, release to stop, and

transcribe into `$BingSpeech`.

Example Use Case:

1. Press and hold the microphone button to call `ControlCommand("Bing Speech Recognition", "StartListening")`.
2. Releasing the button calls `ControlCommand("Bing Speech Recognition", "StopListening")`, which ends the recording, updates the `$BingSpeech` variable with the recognized text, and Bing executes the "*All Recognized*" script.

***Tip:** Bing's *All Recognized* script can send another ControlCommand with the $BingSpeech variable as a parameter to any other robot skill, such as Chat GPT or Autonomous AI, so you can have a fluent conversation with your robot without disruptions and being rushed while it listens to transcribe.

Why Use a Button Instead of Always-On Listening?

Some robot skills and speech systems try to guess when someone is talking using

Voice Activity Detection (VAD). While convenient, VAD cannot easily distinguish whether the

speech is intended for the robot or is just background conversation, music, or other noise. This can cause:

- Accidental triggers when people nearby are talking.
- Partial sentences if the robot starts or stops listening at the wrong time.
- Overlapping or repeated responses when speech is detected while the robot is still replying.

A **push-to-talk button** fixes this by giving you a clear, intentional signal:

the robot only listens while the button is held down. This is similar to how a walkie-talkie works

and is the recommended method for reliable conversational input with robots.

What You'll Need

- Synthiam **ARC** (latest version) installed on Windows.

- An **EZB-compatible controller** (e.g. EZ-B v4, IoTiny, Arduino via EZB firmware, etc.).
- A **momentary push button** (normally open).
- Wiring to connect the button to a digital port (e.g. `D0`) and ground.
  Use a **pull-up resistor** so the input is high when released, and low when pressed.
- A **microphone** connected to the PC running ARC, or an EZB board with audio input.
- An **internet connection** (Bing Speech Recognition uses an online service).

How Bing Speech Recognition Works in ARC

The **Bing Speech Recognition** robot skill listens to audio through your configured microphone and sends it to Microsoft's speech service. When recognition completes, the translated text is stored in a variable called `$BingSpeech`. Other robot skills (such as AI chatbots, movement controllers, or custom scripts) can read this variable to determine how the robot should respond or move.

ARC's **ControlCommand()** messaging system lets one robot skill send commands to another.
In this tutorial, a **Script** robot skill will send `StartListening` and `StopListening` commands to the Bing Speech Recognition skill whenever the hardware button is pressed or released.

Step 1 — Add the Bing Speech Recognition Robot Skill

1. In ARC, open your robot project (or create a new one).
2. In the top menu, select **Project > Add Robot Skill**.
3. Browse to the **Audio** category and add **Bing Speech Recognition**.
4. Once added, click the **Config** (gear) icon on the Bing Speech Recognition robot skill to open its configuration.

Inside the configuration, you'll find options such as the wake word, recording time limits, and script
callbacks (e.g. *All Recognized Script*). The recognized transcription text is exposed via the `$BingSpeech` variable, so any script or robot skill can use it.

Step 2 — Configure Bing Speech for Push-to-Talk

Because we want the robot to listen *only* when the button is held, you should configure Bing Speech
Recognition for a manual, software-controlled workflow:

- Optional: **Turn off or ignore the wake word** if you don't plan to use it. This prevents the
  robot from accidentally starting to listen during background conversations.

- Set the **maximum recording time** to the longest allowed duration. This ensures that listening
  does not stop prematurely while the button is held. When you release the button, our script will send
  `StopListening` explicitly to end the recording.

- If you plan to send recognized text to an AI service or movement script, configure the **All Recognized Script** in the Bing Speech skill. That script will run every time `$BingSpeech` is updated.

Step 3 — Wire the Physical Button to the EZB

Next, wire a button to one of the EZB's digital ports, such as `D0`. The goal is:

- **Released button** = input is **high** (logic 1).

- **Pressed button** = input is **low** (logic 0).

To achieve this, use a **pull-up resistor** on the input:

- Connect one side of the button to EZB digital port `D0`.

- Connect the other side of the button to **GND**.

- Add a pull-up resistor between `D0` the EZB's **+5 V** (or 3.3 V, depending on
  your controller). When the button is not pressed, `D0` it is pulled high. Pressing the button connects
  `D0` to ground, making it low.

This wiring matches the logic used in the script below:
released = high, pressed = low.

Step 4 — Add a Script Robot Skill and Select JavaScript

1. In ARC, click **Project > Add Robot Skill**.

2. Add the **Script** robot skill (Scripting category).

3. Open the Script robot skill and click the **Script** or **Edit** button to open the code editor.

4. At the top of the editor, select the **JavaScript** tab. ARC's Script skill supports JavaScript, Python , and more, but this example uses the JavaScript API.

Step 5 — Push-to-Talk Button Script (JavaScript)

The script below runs in an infinite loop. It waits for the button to be pressed (input goes low), starts Bing Speech Recognition, then waits for the button to be released (input returns high) and stops listening. At that moment, Bing Speech transcribes the audio and updates `$BingSpeech`.

Example Script Robot Skill — JavaScript Code

```
// Push-to-talk Bing Speech Recognition using a button on D0
// Button released = high (pulled up), pressed = low.

while (true) {

  // Wait for the button press (pulls D0 low)
  Digital.wait(D0, false, 10);

  // Start Bing Speech Recognition
  ControlCommand("Bing Speech Recognition", "StartListening");

  // Small delay in case the button is pressed and released very quickly
  sleep(200);

  // Wait for the button release (D0 returns high)
  Digital.wait(D0, true, 10);

  // Stop Bing Speech Recognition and start transcription
  ControlCommand("Bing Speech Recognition", "StopListening");
}
```

A few important notes about this script:

- **`Digital.wait(D0, false, 10)`** blocks the script until the digital port `D0` reads low (button pressed). The third parameter (`10`) tells ARC to check the port every 10 ms, which is efficient and avoids busy-looping.

- The short `sleep(200)` helps handle rapid taps by giving Bing Speech a moment to engage, mimicking how a person naturally holds a push-to-talk button for at least a fraction of a second.

- `Digital.wait(D0, true, 10)` then waits until the button is released (input returns high).

- When `StopListening` is called, the Bing Speech Recognition skill finishes the recording, sends the audio to the cloud service, and updates `$BingSpeech` with the recognized text.

Step 6 — Using the $BingSpeech Variable

Once you release the button and Bing finishes processing, the recognized phrase is available in `$BingSpeech`. You can use this in many ways:

- In the Bing Speech skill's **All Recognized Script**, call an AI chat robot skill (for example, OpenAI ChatGPT or PandoraBots) using `ControlCommand()` and pass `$BingSpeech` as the user's phrase.

- Use conditional logic to trigger movements, animations, or servo actions. For example, if `$BingSpeech` contains "wave hello", trigger an Auto Position frame to wave the robot's arm.

- Log recognized phrases to a file or a variable array for diagnostics and debugging.

Tips and Variations

- **Place the button on the microphone.** Mounting the button directly on, or next to, the microphone makes the interaction intuitive: press where you speak.

- **Combine with wake words.** You can still keep a wake word enabled for hands-free use, while using the hardware button when you need strict control, such as in noisy environments.

- **Guard against overlapping responses.** In more advanced projects, use a global variable `$IsSpeaking` to prevent new questions from being sent to an AI skill while the

robot is
still talking. Only allow new input when the robot finishes speaking.

- **Support multiple EZB boards.** If your button is on a different EZB index, pass the `ezbIndex` parameter to `Digital.wait()` so it reads the correct board.

Result:

You've now built a robust push-to-talk speech pipeline in Synthiam ARC. Your robot only listens when you
press the button, eliminating most accidental triggers from VAD and background conversations while still
taking advantage of Bing's cloud-based speech recognition and ARC's powerful robot skill messaging system.