Overview

This section has troubleshooting tips and steps to help with common problems.

- Troubleshoot WiFi Connections
  Use this if you're having trouble connecting to an EZB over WiFi.
- Troubleshoot USB Connections
  Use this if you're having trouble connecting to an EZB over USB.
- Connecting EZB Over the Internet
  Information on connecting to an EZB over the internet by providing access through your router.
- Battery and Power Supply
  If the EZB requires more power, this page provides information on how to learn about batteries and power supplies.
- EZB Disconnecting
  Diagnose why the EZB is disconnecting while using the robot.
- ARC Screen Resolution
  Learn about ARC screen resolutions and scaling for the optimal experience.
- Slow Performance
  Optimize your robot project for performance if it is running slow.
- Camera & Microphone Issues
  Look here if you are experiencing issues with the microphone, camera, or speaker on your PC not working with ARC.

EZB Connection Troubleshooting

An EZB is any robot/microcontroller that runs the ARC-compatible firmware or speaks the ARC language. The connection type between EZBs will vary between USB, Bluetooth, or WiFi.

Select the type of EZB that you are having issues connecting with:

WiFi EZB

View WiFi Connection Diagnostic

These EZB controllers connect to the PC over a WiFi connection. The most popular WiFi EZBs are EZ-Robot IoTiny, EZ-Robot EZ-B v4, ESP32, ESP32Cam, Raspberry Pi, and more.

USB EZB

View USB Connection Diagnostic

These EZB controllers connect to the PC with a COM, Serial, or Bluetooth connection. The most popular EZBs of this type are Arduino, Robotis OpenCM, Robotis OpenCR, BBC Microbit, EZ-Robot EZ-B v3, and more.

Troubleshoot WiFi Connections

An EZB is any robot/microcontroller that runs the ARC-compatible firmware to use the ARC communication protocol. The connection type between EZBs will vary between USB, Bluetooth, or WiFi. This diagnostic page is for Wi-Fi-enabled EZBs. If you are not using a Wi-Fi-enabled EZB, select your EZB connection type from this index page.

Anti Virus for WiFi EZB Controllers

Third-party antivirus Software - This is the number one connection issue for WiFi controllers, such as

the EZ-Robot EZB v4 or IoTiny. There are many reports of connection issues with PCs running 3rd party (non-Microsoft) Antivirus Software, such as AVAST, McAfee, Norton, or Norton Internet Security. If your PC has 3rd party antivirus software installed, please disable or uninstall it. Nearly all WiFi communication issues are related to 3rd party anti-virus software. Disable any 3rd party (non-Microsoft) virus scanner before posting a message.

WiFi EZB Connection Diagnostic

The most common WiFi controllers are the ESP32, ESP32Cam, EZ-Robot EZB v4, and EZ-Robot IoTiny. This connection diagnostic does not include hardware support, so we recommend visiting the manufacturer's website for product support. However, we will provide connection debugging assistance here. In this connection diagnostic, we'll use the EZ-B in AP Mode (also called WiFi Access Point or AdHoc).

***Note:** *If using an EZB controller that does not support audio feedback or reset buttons, skip the appropriate steps (i.e., ESP32, Raspberry Pi).*

****Note:** *If you identify a network conflict when executing the diagnostic in the chart below, we have a tutorial for Two Network Interfaces.*

WiFi Channel/Signal Scan

Some EZB controllers are connected via WiFi, such as the ESP32, EZ-Robot IoTiny, and EZB v4. While WiFi is convenient by not requiring wires, it is the second most common cause of disconnections. The WiFi connection should operate over a channel that is not saturated and provides the most stability and throughput.

***Note:** *We recommend using a USB EZB connection rather than WiFi for robots in production environments.*

- Check for the WiFi channel saturation? Use THIS TOOLï»¿ to check and use a less saturated channel. If possible, consider hard-wiring the EZB to the PC.

Download the WiFi Scan Tool (WifiInfo.exe)

EZ-Robot Controllers

EZ-Robot has excellent tutorials on changing the WiFi mode, which can be accessed HERE.

Guide: Forum Posting For Assistance

If you still have trouble connecting to an EZ-B, the section below contains troubleshooting steps. If you are ready to post on the community requesting assistance with connection issues, you must answer these questions in your forum post.

1. Have you checked the manufacturer's website for instructions to use their product? Most manufacturers provide tutorials or instructions for their products. The manufacturer may also have a FAQ or troubleshooting guide on their website.
2. What product are you connecting to? (*i.e., Arduino, ESP-32, EZ-Robot JD, Raspberry Pi, Robotis Bioloid, etc.*).
3. What power source are you providing the robot? (*Battery, wall adapter, how many volts/amps, etc.*)
4. Does the robot make a startup sound when powered on (if applicable)? Some controllers have a verbal startup message or startup sound.
5. If WiFi device, are you connecting the correct WiFi network of the EZB/Robot?

6. Do you have two network adapters? (i.e., one for the internet and one for the robot)
7. What error message is displayed in the status window at the bottom of ARC? Press the COPY button on the status window and paste the contents into the forum post.

Additional Diagnostics

If the EZB connection drops, consider reading the EZB Disconnection Troubleshooting Guide.

Reset ARC User Registry

This process will clear all information about the current ARC user. The next time ARC is loaded, you will be prompted to log in with a new user account.

When Is This Necessary?

If an ARC Pro user has been changed or uses a different account, the ARC Pro may not load. This is because the currently logged-in user account does not have a subscription to load ARC Pro. The solution is to remove the existing user credential and reset ARC to default values. The next time ARC is loaded, it will prompt the user.

Steps

1. Close ARC
2. Download the *"Reset ARC Registry For Current User.zip"* file
3. Navigate to your downloads folder and extract the contents of the file
4. Double-click on the REG file that was extracted from the ZIP file
5. Follow the prompts to execute the registry changes
6. You may launch ARC and be prompted for new user credentials.

Reset ARC Registry For Current User.zip

Troubleshoot USB Connections

An EZB is any robot/microcontroller that runs the ARC-compatible firmware to use the ARC communication protocol. The connection type between EZBs will vary between USB, Bluetooth, or Wi-Fi. This diagnostic page is for USB-enabled EZBs. If you are not using a USB-enabled EZB, select your EZB connection type from this index page.

USB EZB Connection Diagnostic

Arduino and USB controllers have tutorials for connection on their respective manual pages. However, the baud rate is the leading cause of connection trouble with Arduino. Ensure you have followed the Getting Started guide links to configure ARC to the correct baud rate. Additionally, providing proper power to the controller is essential as well.

DTR/RTS Enable

Some USB devices may require DTR/RTS to be enabled. This can be done for the specified EZB index in the connection control configuration screen. Press the CONFIG button on the connection control, select the COM/USB Connection tab, and enable DTR/RTS checkboxes.

Guide: Forum Posting For Assistance

If you still have trouble connecting to an EZB, you may ask on the community support forum. If you are ready to post on the community requesting assistance with connection issues, you must answer these questions in your forum post.

1. What product are you connecting to? (*i.e., Arduino, ESP-32, EZ-Robot JD, Raspberry Pi, Robotis Bioloid, etc.*).
2. What power source are you providing the robot? (*Battery, wall adapter, how many volts/amps, etc.*)
3. Does the robot make a startup sound when powered on (if applicable)? Is there a verbal message spoken from the product on bootup (if applicable)? Have you checked the product manual?
4. If Wi-Fi device, are you connecting the correct WiFi network of the EZB/Robot?
5. Do you have two network adapters? (i.e., one for the internet and one for the robot)
6. What error message is displayed in the status window at the bottom of ARC? Press the COPY button on the status window and paste the contents into the forum post.

Completely Remove All ARC

Occasionally, ARC needs to be removed entirely from your system. This could be reinstalling from a fresh instance or if you are no longer using the software. These instructions will detail every step to remove ARC altogether.

***Note:** This will not explain how to remove the ARC projects folder in your My Documents. We do not instruct you to remove that folder because it contains your personal projects, and we do not want you to delete them accidentally. If you wish to remove that folder, do so at your own risk. Ensure you have the project files backed up to the Synthiam Cloud if you want to delete your project folder.

Follow these steps to remove Synthiam ARC from your computer completely.

Step 1: Close Any Instance of Synthiam ARC

Before uninstalling, make sure ARC is not running. If it is open:

- Click the **X** in the top-right corner of the ARC window.
- If it does not close, open the **Task Manager** by pressing `Ctrl + Shift + Esc`.
- Find **ARC.exe** in the list, right-click it, and select **End Task**.

Step 2: Reboot Your Computer

Restart your computer to ensure all ARC-related processes are fully stopped before uninstalling.

To restart:

- Click the **Start** button.
- Select **Power** > **Restart**.

Step 3: Uninstall Synthiam ARC

To remove ARC from your system:

- Press `Windows + R`, type `appwiz.cpl`, and press `Enter`.
- Find **Synthiam ARC** in the list.
- Right-click it and select **Uninstall**.
- Follow the on-screen instructions to complete the process.

Step 4: Delete the Remaining Files

Some files may remain after uninstallation. Manually delete them:

- Open **File Explorer** (press `Windows + E`).

- Go to `C:\Program Files (x86)` and delete the **Synthiam Inc** folder.
- Then, navigate to `C:\ProgramData` (hidden by default; enable hidden files in **View** settings).
- Delete the **ARC** folder.

Step 5: Remove Registry Entries

To ensure complete removal, delete ARC registry entries:

- Press `Windows + R`, type `regedit`, and press `Enter`.
- Navigate to `Computer\HKEY_CURRENT_USER\SOFTWARE\ARC by Synthiam`.
- Right-click the **ARC by Synthiam** folder and select **Delete**.
- Close the registry editor.

Warning: Be careful when modifying the registry. Deleting the wrong key may cause system issues.

Step 6: Reboot Again

Restart your computer one more time to ensure all changes take effect.

- Click the **Start** button.
- Select **Power** > **Restart**.

Optional: Reinstall Synthiam ARC

If you are removing ARC to reinstall a fresh copy, you can download the latest version here:

Download Synthiam ARC

Connecting EZB Over Internet

A Wi-Fi or Ethernet-enabled EZB can be connected remotely over the internet. In these cases, a robot could be at your house or work and remotely controlled from a different location. Because most home/office networks have private IP addresses behind a security firewall, this feature will require configuring the router/firewall to forward a port. The port that most EZBs use by default is TCP 23. This process to enable remote client connections would require the TCP port 23 to be forwarded to the IP Address of the EZB device.

Here is an example diagram of how a remote connection would be configured to an EZB on TCP port 23.

EZB Listening (server) Port

Because EZBs can configure their listening (server) port, we mention TCP port 23 in this document, but it may be changed with the EZB configuration. Check the manual for the EZB type you're using to see how the TCP listening (server) port can be changed.

IP Address

In the remote ARC instance, you will need to enter the IP Address of the Office/Home router/firewall. This is because the office/home router/firewall will be the endpoint that ARC will connect to. That router/firewall will forward the connection data to and from the EZB. You can obtain your office/home router/firewall's IP address in a few different ways...

Get IP Address from Google - Using this method, use a home/office network PC and search in google for the words "What Is My IP?". You will be presented with an IP address. That is the IP address that the remote ARC instance will use to connect to the robot on the home/office network.

Dynamic DNS Server - Services like Dyn DNS allow entering the IP address that the hostname rather than the IP address can reference.

Battery and Power Supply

If an EZB microcontroller is often disconnecting, the batteries or power source is not providing enough current for the servos. When this happens, it is called a "*Brown Out*." That means the EZB controller is not getting enough power when the servos move. In this case, the solution is to replace the power source

with one providing higher amperage. A wall-wart (power adapter) will **not** provide enough current for electro-mechanical servo motors. We recommend using the power supply tutorial and exploring options for powering robots.

Further Reading

- Choose a Power Type (battery or power adapter) <- RECOMMENDED
- How To Initialize Servos
- Josh's Power Supply Tutorial

EZB Disconnecting

An unwanted EZB disconnection is when ARC disconnects from the EZB without notice. In many cases, ARC may lock up and stop responding.

ARC release candidates run stress tests on several virtual machines before being released to the public. The most significant stress test for reliability uses ARC to control four robots 24/7 connected to Exosphere. Many people do not know that Synthiam's Exosphere robots are performing reliability/efficiency/stability tests for ARC. These robots perform multiple cameras with tracking, Wi-Fi and USB connections, NMS, and more. The Exosphere test robots are always online, so people worldwide use them at all times of the day. Generally, you can dismiss ARC core as causing disconnection/freezing issues.

1. Power

This is the first item to check because it is generally the most common problem with DIY robots. Robots that use motors and servo motors will often draw high currents. Sometimes, these robots draw more current than what is available from the battery or power supply.

- How many amps is the robot drawing? Check for the current draw of average use and peak use.
- How many amps is the power supply rated for? Verify the power supply provides enough current based on the measurement in the previous step.
- Are motors and servo motors directly connected to the EZB power pins? Some EZB controllers, such as the EZ-Robot IoTiny or EZ-B v4, have power pins for convenient connections. However, due to the small traces of these EZB controllers, they might not provide the current necessary and brown-out. A brown-out is when the EZB does not get enough power and shuts down or locks up.

View Power Supply Tutorial

2. Wi-Fi Connection

Some EZB controllers are connected via Wi-Fi, such as the ESP32, EZ-Robot IoTiny, and EZB v4. While Wi-Fi is convenient by not requiring wires, it is the second most common cause of disconnections. The Wi-Fi connection should operate over a channel that is not saturated and provides the most stability and throughput.

***Note:** *We recommend using a USB EZB connection rather than Wi-Fi for robots in production environments.*

- Check for the Wi-Fi channel saturation? Use THIS TOOLï»¿ to check and use a less saturated channel. If possible, consider hard-wiring the EZB to the PC.

Download Wi-Fi Scan Tool

3. Communication Timeouts

Sensors and peripherals connected to the EZB may require bi-directional communication. In many cases, this is most prevalent with i2c devices that need START and STOP acknowledgments while reading/writing data. Some I2c devices may time out when a connection is unstable due to a loose connector, electrical interference, or communication noise. The time-out could dramatically slow the EZB communication, which ARC will appear to have locked up. Or, ARC may entirely freeze while waiting for a response from an unresponsive EZB due to a communication timeout with a peripheral/sensor.

- Are there any i2c devices connected to the EZB? (i.e., RGB Eyes, Compass, Accelerometer) If so, ensure the wiring and connectors are secure. I2C devices with poor wiring/connections can lock up many microcontrollers.
- Common disconnects are reported from EZ-Robot JD Humanoid products when the RGB Eyes

connection is loose or faulty. This causes the EZB controller to lock up while sending I2C commands. If using an EZ-Robot product with RGB Eyes, have you checked the cable connection to ensure it is secure?

4. ARC Project

Lastly, examine the project for issues that may cause disconnections. In some circumstances, many scripts looping and reading/writing data can cause a lockup or disconnection. This happens when the data channel is flooded with requests, most commonly over Wi-Fi-enabled EZBs.

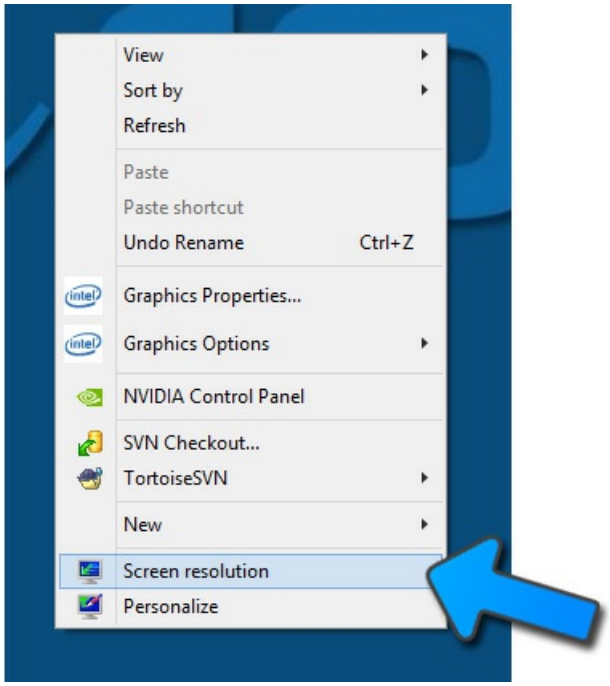
Screen Resolution

The minimum screen resolution for ARC is 1024x768 at 100% scaling. Ideally, the recommended resolution for ARC is 1920x1080 or higher at 100% scaling. For 4k monitors, a scaling of 150% or 175% may be sufficient. Scaling will increase the size of fonts and buttons on the screen, leaving less space for additional robot skills.

Resolution Scaling

Microsoft Windows has a feature to configure scale sizing, which will affect ARC's interface. If the scale sizing is set larger than recommended (100-125%), there will be less screen real-estate for robot skills, and scrollbars will be used. Restore the scaling size of Windows to the recommended sizing by following these steps...

Step 1



Right-click with the mouse on your desktop to display the properties menu. Select SCREEN RESOLUTION - or DISPLAY SETTINGS (depending on the version of Windows).

Step 2

Locate the option for changing the size of text, apps, and other items. Select a reasonable scaling option of either 100% or 125% based on the resolution.

Camera, audio, and microphone issues

In Windows ARC, having a camera and microphone as part of your robot lets you track objects, use telepresence exosphere, take pictures, record videos, and more. Many apps and services request and use the camera or microphone, and Windows settings give you control over which apps can use your camera or microphone.

Some worry about unknown apps, organizations, or malware using their camera or microphone. Whenever your camera or microphone is used, you should be in charge. To help you understand when your camera is turned on, the following indicators are provided:

- If your device comes with a camera light, the light will turn on when the camera is in use.
- If your system doesnâ€™t have a camera light, youâ€™ll get a notification to let you know when the camera turns on or off.

To help you understand when your microphone is in use, the following indicators are provided:

- A microphone icon will be displayed in the notification area of the taskbar.

See which apps currently use your camera or microphone or have recently accessed your camera or microphone.

- In Windows 10, go to:
  - **Start** > **Settings** > **Privacy** > **Camera**
  - **Start** > **Settings** > **Privacy** > **Microphone**
- In Windows 11, go to:
  - **Start** > **Settings** > **Privacy & security** > **Camera**
  - **Start** > **Settings** > **Privacy & security** > **Microphone**.

> **Note:** This functionality does not exist in Windows 10 versions 1809 and earlier.

Microsoft Windows also provides settings to keep you in control, as described in the following sections.

Camera or microphone controls might be turned off if youâ€™re using a device assigned to you by your workplace or if youâ€™ve added a work account to your device. If thatâ€™s the case, youâ€™ll see **Your organization manages some settings** at the top of the Camera or Microphone settings pages.

Three types of cameras

There are three types of cameras that apps and services may have access to, depending on what type of cameras come with or are installed on your device:

- A **color camera** is used for taking traditional color photos or videos.
- An **infrared camera** takes a grayscale (black and white) photo or video based on infrared intensity.

- A **depth camera** can see the shapes of items in front of it and how far they are from the device. For example, a depth camera can recognize when a person is in front of the device.

How to control which apps can use the camera

1. Do one of the following:
    - In Windows 10, go to **Start** > **Settings** > **Privacy** > **Camera**.
    - In Windows 11, go to **Start** > **Settings** > **Privacy & security** > **Camera**.
2. In Windows 10, ensure **Camera access for this device**, **Allow apps to access your camera**, and ensure the individual toggle for the Microsoft Store app you wish to use is turned on. If you do not see the app or website you're looking for in the list, it's likely a desktop app. Desktop apps cannot be individually toggled, but access for those apps can be controlled using **Allow desktop apps to access your camera**.
3. In Windows 11, ensure **Camera access**, **Let apps access your camera**, and provide the individual toggle for the Microsoft Store app you wish to use is turned on. If you do not see the app or website you're looking for in the list, it's likely a desktop app. Desktop apps cannot be individually toggled, but access for those apps can be controlled using **Let desktop apps access your camera**.

For more info about controlling camera access on your device, see Manage app permissions for your camera in Windows.

How to control which apps can use the microphone

1. Do one of the following:
    - In Windows 10, go to **Start** > **Settings** > **Privacy** > **Microphone**.
    - In Windows 11, go to **Start** > **Settings** > **Privacy & security** > **Microphone**.
2. In Windows 10, ensure that **Microphone access for this device**, **Allow apps to access your microphone**, and the individual toggle for the Microsoft Store app you wish to use is turned on. If you do not see the app or website you're looking for in the list, it's likely a desktop app. Desktop apps cannot be individually toggled, but access for those apps can be controlled using **Allow desktop apps to access your microphone**.
3. In Windows 11, ensure that **Microphone access**, **Let apps access your microphone**, and the individual toggle for the Microsoft Store app you wish to use is turned on. If you do not see the app or website you're looking for in the list, it's likely a desktop app. Desktop apps cannot be individually toggled, but access for those apps can be controlled using **Let desktop apps access your microphone**.

Slow Performance

ARC was designed to prioritize robot skills and their respective compilers in a multi-threaded environment. This ensures that each robot skill and separate event scripts are executed in isolated threads. This approach provides robot skills, and scripts distribute CPU time for a responsive UI and robot program. Even with ARC's multi-thread architecture, users can create scripts and project configurations that cause sluggish behavior. This robot support document outlines common issues that will result in low performance.

Contents

Tight Loops

A tight loop is a programming term. Such a loop heavily uses I/O or processing resources, failing to adequately share them with other programs running in the operating system. An example of a tight loop would be a script that loops indefinitely without sleep() to relieve the CPU and give other threads more CPU time. It is also worth noting that a tight loop may also affect EZB communication performance if the loop commands are reading from the EZB or the peripheral sensor is timing out (*see Flood EZB Communication Channel and Peripheral Timeout*).

JavaScript Example

Bad

```
while (true) {

  // Read the ADC value of the pot
  var adcVal = ADC.get(adc0);

  // Map the ADC value (0-255) to the servo degrees (1-180)
  var servoPos = Utility.map(adcVal, 0, 255, 1, 180);

  // Move the servo into the position
  Servo.setPosition(d0, servoPos);
}
```

Good

```
while (true) {

  // Read the ADC value of the pot
  var adcVal = ADC.get(adc0);

  // Map the ADC value (0-255) to the servo degrees (1-180)
  var servoPos = Utility.map(adcVal, 0, 255, 1, 180);

  // Move the servo into the position
  Servo.setPosition(d0, servoPos);

  // Give CPU time to other threads
  sleep(100);
}
```

The primary question for resolving tight loops by adding a sleep() is, "How long should my sleep command be for?". The priority of your software loop determines the answer to that question. The loop only needs to run four times a second or less in many cases. For a script that runs four times per second, a sleep(250) command would suffice. The sleep() command parameter is milliseconds (MS), and there is 1,000 MS in a second.

Top

Flooded EZB Communication Channel

EZBs can be connected through Serial USB or Wi-Fi. Generally, the Serial USB is fast with very low latency and rarely experiences a flooded communication channel. However, it is common for WiFi EZBs to slow ARC scripts and robot skills from flooding the communication channel if data is being read too often or read redundantly. When data is read too often with a WiFi EZB, the overhead of the TCP stack and WiFi protocol will cause slight delays. Because EZB read commands are blocking, they will block all other EZB communication until a response is returned. There are four common mistakes made that may flood the communication channel.

1. **Redundant Robot Skills**

A project contains many robot skills that pull data from the same port/peripheral. For example, two or more Read ADC robot skills may display ADC data from the same port. This will double the EZB communication and therefore double the latency. The solution would be to limit the number of robot skills querying data from the EZB by removing duplicates.

2. **Multiple Robot Skills/Scripts Reading Same Port/Peripheral**

   There are visual robot skills for displaying ADC and Digital port statuses. These robot skills are excellent for debugging and educational use by providing visual feedback. However, it is common for ARC users to add a Read ADC robot skill and use Read ADC commands in scripts. In this case, both the robot skill and scripts are querying the same port, doubling the EZB read commands on the communication channel and latency. There are a few solutions for this scenario.

   - You may remove the visual robot skill (i.e., Read ADC, Ultrasonic Distance) and continue using the script that references the port.
   - Check if the robot skill (i.e., Read ADC, Ultrasonic Distance) populates a variable with the port value and uses that port value in your script. Many robot skills, such as Read ADC, Ultrasonic Distance, and Read Digital, will populate variables that can be referenced in scripts.
   - If a GUI to present the port statuses is essential to the project, consider populating a global variable in a script and displaying the variable statuses on a custom User Interface Builder form.

3. **Tight loops**

   A tight loop may be querying a port too quickly and not providing CPU time or EZB communication access. (*See the Tight loops section.*)

4. **Slow Sensor Protocol**

   Some sensors and peripherals will have slow protocols. Two such examples are ultrasonic distance sensors and I2C devices. Ultrasonic distance sensors can take many milliseconds to respond with a valid value and even longer when experiencing a timeout. If an ultrasonic distance sensor does not detect an object because it is out of range, the sensor will timeout, which takes a long time. A timeout will block all other EZB communication until the timeout is completed. This scenario is further amplified when many robot skills and scripts attempt to read from the problematic sensor, as they will block all other communication.

   The solution is to choose sensors that suit the performance requirements of your robot or use a dedicated EZB for low-latency sensors. If the performance is affected by ultrasonic sensors, consider adding all ultrasonic distance sensors to an affordable USB Arduino EZB or WiFi ESP32 EZB would allow the primary EZB not to experience blocking issues.

5. **Sensor/Peripheral Timeout**

   A sensor or peripheral connected to the EZB may be timing out. (*See the Sensor/Peripheral Timeout section.*)

Top

Sensor/Peripheral Timeout

Sensors and peripherals connected to the EZB may use a protocol that consumes time or times out if not connected. Two such examples are ultrasonic distance sensors and I2C devices. Ultrasonic distance sensors can take many milliseconds to respond with a valid value and even longer when experiencing a timeout. If an ultrasonic distance sensor does not detect an object because it is out of range, the sensor will timeout, which takes a long time. A timeout will block all other EZB communication until the timeout is completed. This scenario is further amplified when many robot skills and scripts attempt to read from the problematic sensor, as they will block all other communication.

The solution is to choose sensors that suit the performance requirements of your robot or use a dedicated EZB for low-latency sensors. If the performance is affected by ultrasonic sensors, consider adding all ultrasonic distance sensors to an affordable USB Arduino EZB or WiFi ESP32 EZB would allow the primary EZB not to experience blocking issues.

Top

Abusing ControlCommand()

The ControlCommand() script command will send an instruction to another robot skill in the ARC project (*read more here*). In most robot skills, the available ControlCommands can also be accessed with GUI button presses. If a tight loop or script rapidly calls another robot skill via the ControlCommand, it would be similar to rapidly pressing the UI button. Consider how often your robot skill needs to call another robot skill to trigger an event. Specifically, ensure that your loop has given the other robot skill enough time to complete the task. Otherwise, you may interrupt the other job preventing it from ever completing.

Top

High Camera Resolution

Machine vision and computer recognition are very highly CPU-intensive processes. The cameras for computer vision provide much less resolution than what you, as a human, would use for recording a birthday party. If you were to run computer vision to recognize objects and decode frames at HD quality, your computer response would grind to a halt. Let us examine how much data is contained in a video stream at varying resolutions.

- 160x120 = 57,600 Bytes per frame = 1,152,000 Bytes per second
- 320x240 = 230,400 Bytes per frame = 4,608,000 Bytes per second
- 640x480 = 921,600 Bytes per frame = 18,432,000 Bytes per second

**Note: at 320x240, your CPU is processing complex algorithms on 4,608,000 Bytes per second. Soon as you move to a mere 640x480, it's 18,432,000 Bytes per second.*

To expand on this example, 4,608,000 Bytes per second is just the data, not including the number of CPU instructions per step of the algorithm(s). Do not let television shows, such as Person Of Interest, make you believe that computer vision and CPU processing areas are accessible in real time, but many of us are working on it! We can put 4,608,000 Bytes into perspective by relating that to a 2-minute MP3 file. Imagine your computer processing a 2-minute MP3 file in less than 1 second - that is what vision processing for recognition is doing at 320x240 resolution. Soon as you increase the resolution, the CPU has to process an exponentially more significant amount of data. Computer vision recognition does not require as much resolution as your human eyes, as it looks for patterns, colors, or shapes.

The solution is to choose a camera device with video quality that suits the object's needs to be recognized. If your robot project operates correctly with a resolution of 320x240, it is advised to continue using that resolution.

Top
