Troubleshooting ARC

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.

Startup Errors

When ARC is loading, a loading screen appears and the bottom-left corner displays the current initialization step. If loading seems to pause for a while, ARC may still be initializing or synchronizing a large amount of data. During this time, any robot skill updates in the Skill Store will also sync if an internet connection is available.
ARC loading screen with initialization progress information.

Cef Chromium Embedded Browser

ARC uses a Chromium Embedded Browser (WebView) for rich media content, including Blockly. If you see an error similar to the example below, a required redistributable library may be missing from your system.

```
=== Exception Information ===
Type: System.InvalidOperationException
Message: Cef.IsInitialized was false!. Check the log file for errors. See https://github.com/
cefsharp/CefSharp/wiki/Trouble-Shooting#log-file for details.
HResult: -2146233079
Source: CefSharp.WinForms
TargetSite: Void InitializeCefInternal()
```

1. Download the Required Runtime
   If ARC cannot initialize the Chromium Embedded Browser, install the Visual C++ Redistributable runtime below.

   [Latest VC Redist x86]
   [Latest Supported Visual C++ Redistributable Downloads]

2. Re-run The Installer
   Download the latest Synthiam ARC and reinstall. During installation, select the REPAIR option.

   [Download 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.

Screen Resolution & Display Issues

This document includes details on recommended display resolution and on diagnosing blurriness in Synthiam ARC in Microsoft Windows. Scroll through this document to identify the section that is relevant to your experience.

Proper screen resolution and Windows scaling settings are critical for getting the best experience when using Synthiam ARC.

Incorrect scaling can reduce usable workspace, force the use of scrollbars, and make complex robot projects harder to manage.

Minimum and Recommended Display Settings

The **minimum supported screen resolution** for ARC is
1024×768 with Windows scaling set to **100%**.
While ARC will run at this resolution, workspace will be limited, and many robot skills may require scrolling.

The **recommended resolution** for comfortable daily use is
1920×1080 (1080p) or higher at **100%** scaling.
This provides adequate space for multiple robot skills, configuration panels, and real-time feedback windows.

On **4K (3840×2160)** displays, Windows scaling is often required to keep text readable.
In these cases, a scaling value of **150%** or **175%** is typically sufficient.
Be aware that increasing the scaling enlarges text and buttons, reducing the available screen space for robot skills.

Higher scaling values trade usable workspace for readability.
If your monitor is physically large, running at 100% scaling may still be comfortable and will provide the maximum workspace.

Understanding Resolution Scaling in Windows

Microsoft Windows includes a feature called **display scaling** that adjusts the size of text,

apps,
and other interface elements independently of the screen resolution.
This directly affects how ARC’s interface is displayed.

When scaling is set above the recommended range (typically 100–125% for non-4K displays),
Windows enlarges UI elements, reducing available screen real estate.
In ARC, this results in:

- Less room for robot skills and panels
- More frequent use of scrollbars
- Reduced visibility of multiple skills at once

For best results, Windows scaling should be set to the lowest comfortable value for your display.

How to Adjust Screen Scaling

Step 1

Right-click on an empty area of your desktop to open the context menu.
Select **Display settings** (or **Screen resolution** on older versions of Windows).

Step 2

Locate the **Scale and layout option**, which controls the size of text, apps,
and other on-screen elements.
Choose a reasonable scaling value:

- **100%** – Recommended for 1080p and lower resolutions
- **125%** – Acceptable for higher-resolution displays if text is too small
- **150%–175%** – Common for 4K displays

After changing scaling, close and restart ARC to ensure the interface updates correctly.

Best Practices

- Use 100% scaling whenever possible for maximum workspace
- Avoid custom scaling values that Windows does not list
- Restart ARC after any scaling or resolution change
- Balance readability and workspace based on monitor size, not just resolution

Correct resolution and scaling settings ensure ARC remains sharp, readable, and efficient when building and controlling robots.

ARC Appears Blurry on 4K Displays

If Synthiam ARC looks blurry or slightly out of focus on a 4K or high-DPI display, this is not a bug in ARC itself.
The issue is caused by how Microsoft Windows handles display scaling for desktop applications.

Why This Happens

Windows was initially designed around 96 DPI displays (commonly 1080p at typical monitor sizes).
When 4K displays became common, Microsoft added *display scaling* to make text and UI elements readable.
Unfortunately, this system was bolted on after the fact and behaves inconsistently across applications.

Many desktop applications (including complex, high-performance tools like ARC) render their UI at a fixed resolution.
When Windows applies scaling on top of that rendering, it often uses bitmap upscaling instead of accurate DPI-aware rendering.
The result is a soft or blurry appearance, especially noticeable on text, icons, and fine UI details.

Custom scaling values (anything other than the standard 100%, 125%, 150%, etc.) make this problem worse.
In many cases, Windows applies a second scaling pass, further degrading visual clarity.

Recommended Fix: Enable "Fix Scaling for Apps"

Windows includes an option that attempts to correct blurry desktop applications by re-evaluating how scaling is applied.
While not perfect, this option often improves clarity for ARC and other desktop software.

How to Enable It

1. Open **Windows Settings**
2. Go to **System**
3. Select **Display**
4. Scroll down and click **Advanced scaling settings**
5. Under **Fix scaling for apps**, turn on:

   "Let Windows try to fix apps so they're not blurry."
6. Close and restart ARC.

This setting only applies to desktop applications and primarily affects the main display.

It may not fix every app, but it is the first and easiest step to try.

Additional Suggestions

Option 1: Set Scaling to 100%

The most precise results are achieved when Windows scaling is set to **100%**.
This disables DPI scaling entirely and allows ARC to render at native pixel resolution.

If text and UI elements appear too small at 100% scaling, consider the resolution option below.

Option 2: Run the Display at 1080p

Many users find the best balance of clarity and usability by running their monitor at **1920×1080** with scaling set to **100%**.

This avoids Windows scaling altogether and provides crisp, predictable rendering.
On large 4K monitors, 1080p at 100% scaling is often more comfortable than 4K with aggressive scaling.

Option 3: Avoid Custom Scaling Values

Custom scaling (for example 110%, 135%, or 175%) is **not recommended**.
These values frequently cause Windows to fall back to bitmap scaling, resulting in blurring.

Stick to standard scaling values or use 100% scaling whenever possible.

Summary

- ARC is not blurry by design; the issue is caused by Windows DPI scaling
- 4K displays expose weaknesses in Windows' scaling implementation
- Enable **"Fix scaling for apps"** in Advanced Scaling Settings
- For best clarity, use 100% scaling or run the display at 1080p
- Avoid custom scaling percentages

These steps provide the most reliable way to achieve sharp, readable visuals when running Synthiam ARC on high-resolution displays.

Troubleshoot WiFi Connections

An EZB is any robot or microcontroller running ARC-compatible firmware that uses the ARC communication protocol. EZB connection types can vary and may include USB, Bluetooth, or Wi-Fi. This diagnostic page is specifically for Wi-Fi-enabled EZBs. If you are not using a Wi-Fi-enabled EZB, select your EZB connection type from the connection troubleshooting index page.

Antivirus Software and Wi-Fi EZB Controllers

Third-party antivirus software is the most common cause of connection issues with Wi-Fi controllers such as the EZ-Robot EZB v4 and IoTiny. Many users report connection problems on PCs running third-party security software such as Avast, McAfee, Norton, and similar products.

If your PC has third-party antivirus or firewall software installed, disable or uninstall it before troubleshooting. These programs commonly block Wi-Fi communication, including port 23, which is required by many EZB devices.

For best results, use Microsoft Defender alone while testing the connection.

Wi-Fi EZB Connection Diagnostic

The most common Wi-Fi controllers include the ESP32, ESP32-CAM, EZ-Robot EZB v4, and EZ-Robot IoTiny. This diagnostic page focuses on connection troubleshooting and does not replace hardware support from the manufacturer. For hardware-specific problems, visit the manufacturer's website for documentation and support. In this guide, the EZ-B is assumed to be operating in **AP Mode** (also called **Wi-Fi Access Point** or **Ad Hoc mode**).

Note: If you are using an EZB controller that does not support audio feedback or does not have a reset button, skip the related steps. For example, some ESP32-based devices and Raspberry Pi systems do not provide these features.

Note: If you discover a network conflict during the diagnostic below, review this tutorial for help with using two network interfaces.

Connection Diagnostic Checklist

1. **Is the LED flashing blue?**

   If **No**, press the *Reset* button on the EZB.

2. **Was the start-up chime heard from the EZ-B?**

   If **No**, press the *Reset* button on the EZB and try again. If there is still no sound, the EZB may be damaged or the speaker may be disconnected.

3. **Is the PC connected to the EZB over Wi-Fi?**

   If **No**, press the *Reset* button on the EZB and connect to the EZB over Wi-Fi.

4. **Is third-party antivirus software or a third-party firewall installed?**

   If **Yes**, disable or uninstall the third-party antivirus or firewall because it may block port 23.

5.

Does the PC have two network interfaces? (one for internet and one for the EZB)

If **No**, review the connection tutorial to verify that everything appears operational.

6. **Is the router or internet network using 192.168.1.0/24?**

If **Yes**, change either the router/internet network or the EZB network to avoid an IP conflict.

7. **If everything appears operational but the connection still fails,** review the connection tutorial again and verify each step carefully.

Wi-Fi Channel and Signal Scan

Some EZB controllers connect over Wi-Fi, including the ESP32, EZ-Robot IoTiny, and EZB v4. Although Wi-Fi is convenient because it eliminates cables, it is also one of the most common causes of unstable connections and disconnects. For the best performance, the Wi-Fi connection should use a channel with minimal congestion and strong signal quality.

Recommendation: For production environments, a USB-connected EZB is generally more reliable than Wi-Fi.

- Check for Wi-Fi channel congestion. Use this tool to scan nearby wireless networks and choose a less congested channel. If possible, consider a hard-wired connection between the EZB and the PC.

Download the Wi-Fi Scan Tool
WifiInfo.exe

EZ-Robot Controllers

EZ-Robot provides tutorials for changing Wi-Fi modes, which can be accessed here. EZ-Robot EZB controllers also use different LED status indicators, which are documented in the EZB v4 datasheet.

One important status to recognize is a **solid red LED**. This usually indicates a connection or communication problem, often involving an I2C device or communication with the lower EZB board. Also verify that the EZB plays its start-up chime when powered on. If there is no audio on startup, the lower EZB board may be damaged. The startup sound of the v4 and iotiny is a diagnostic to let you know communication between the microprocessors is working.

Guide: Posting in the Community for Help

If you are still having trouble connecting to an EZ-B, follow the troubleshooting steps above first. When you are ready to post in the community for help, include answers to the questions below so others can assist you more effectively.

1. Have you checked the manufacturer's website for instructions on how to use the product? Most manufacturers provide setup guides, FAQs, and troubleshooting steps.

2. What product are you trying to connect to? (For example: Arduino, ESP32, EZ-Robot JD, Raspberry Pi, Robotis Bioloid, etc.)

3. What power source is powering the robot? (Battery, wall adapter, voltage, amperage, and so on.)

4. Does the robot make a startup sound when powered on, if applicable? Some controllers play a spoken startup message or a chime.

5. If it is a Wi-Fi device, are you connected to the correct EZB or robot Wi-Fi network?

6. Does your PC have two network adapters? (For example, one for internet access and one for the robot.)

7. What error message appears in the ARC status window at the bottom of the screen? Press the **COPY** button in the status window and paste the contents into your forum post. Also include the LED status shown on the EZB, if applicable.

Additional Diagnostics

If the EZB connection drops after it has connected successfully, review the EZB Disconnection Troubleshooting Guide.

Troubleshoot USB Connections

An EZB is any robot or microcontroller running ARC-compatible firmware that uses the ARC communication protocol. EZB connection types can include USB, Bluetooth, or Wi-Fi. This diagnostic page is specifically for **USB-enabled EZBs**. If you are not using a USB-enabled EZB, select your EZB connection type from the connection troubleshooting index page.

USB EZB Connection Diagnostic

Arduino and other USB controllers typically have setup tutorials in their respective manuals. For many USB devices, the most common connection issue is an incorrect **baud rate**. Make sure you have followed the Getting Started guide for your device and configured ARC with the correct baud rate. Also verify that the controller is receiving proper power, because insufficient or unstable power can prevent successful communication.

Connection Diagnostic Checklist

1. **Is this an ARC-supported USB EZB device?**

   If **No**, view the list of supported devices on the ARC product page or in the support section.

2. **Is the latest firmware programmed onto the device?**

   If **No**, open the hardware page for the device in the support section and review the firmware and setup instructions.

3. **Is the correct COM port selected in the connection control?**

   If **No**, click the ARC connection control to view the drop-down list of available COM ports and select the correct one.

4. **Is the correct baud rate selected in the connection control?**

   If **No**, review the hardware page for the device in the support section and select the correct baud rate in the connection control configuration.

5. **Does RTS/DTS need to be enabled in the connection control?**

   If **Yes**, enable the required handshake options in the connection control configuration. Some hardware requires these settings, and the hardware or firmware page will mention it.

6. **If everything appears correct but the device still does not connect,** re-check the hardware manual and support documentation for the specific device.

DTR/RTS Enable

Some USB devices require **DTR** and/or **RTS** to be enabled before communication will work correctly. These options can be configured for the selected EZB index in the connection control configuration screen.

To enable them, press the **CONFIG** button on the connection control, select the **COM/ USB Connection** tab, and enable the **DTR** and **RTS** checkboxes as required by the device documentation.

UART <-> USB EZB Connections (EZB v4.x / EZB v2)

Some USB-enabled EZBs, such as the **EZB v4.x** and **EZB v2**, internally use a **UART <-> USB bridge**. ARC communicates with these devices over USB at the software level, but electrically the communication between the EZB and the USB interface is still **TTL UART**. This creates physical wiring limitations that are often overlooked.

UART is not intended for long cable runs. Unlike USB, Ethernet, or RS-485, UART does not provide differential signaling, clock recovery, or strong built-in noise rejection. Because of this, cable length, routing, shielding, and nearby interference all have a significant effect on reliability.

Recommended UART Cable Requirements

- **Keep UART cables as short as possible**, ideally **under 12 inches (30 cm)**.
- **Use shielded cable** or **twisted-pair wiring**.
- **Avoid straight, parallel ribbon or jumper wires** for anything longer than a few inches.
- **Do not coil excess cable length** inside the robot shell.

Straight, untwisted wires can act like antennas. Inside a robot shell, they can easily pick up interference from:

- Motor noise and brush EMI
- Switching regulators and DC-DC converters
- Wi-Fi and Bluetooth radio emissions
- Static discharge and resonance from long wire lengths

This interference can corrupt UART data and cause problems such as:

- Random disconnects
- Partial or garbled commands
- Intermittent failures that appear to be software related
- Unstable or unreliable ARC connections

Using Twisted Pair with UART

When using twisted-pair cable, pair each signal with **ground**. This reduces loop area and greatly improves noise immunity.

- **TX ↔ GND** twisted together
- **RX ↔ GND** twisted together

This works because the return current flows beside the signal path, which helps cancel induced noise and reduces susceptibility to outside interference.

Grounding and Shielding Best Practices

- If using **shielded cable**, connect the shield to **ground on one end only**, preferably on the EZB side.
- Do **not** connect the shield on both ends, because that can create a ground loop.
- Ground unused conductors at one end so they do not act as floating antennas.
- Make sure the EZB and USB interface share a common ground reference.

Leaving extra conductors floating inside a robot can significantly increase EMI pickup. Grounding unused wires helps prevent resonance and reduces noise coupling into nearby signal wires.

Routing and Physical Placement

- Keep UART wiring away from motors, motor drivers, and high-current wiring.
- Avoid running UART wiring parallel to power cables over long distances.

- Cross power wiring at 90-degree angles when separation is not possible.
- Avoid tight loops, coils, or large circular wire paths.

If you need longer communication distances, UART is the wrong protocol for the job. Use USB directly, Ethernet, Wi-Fi, or a protocol such as RS-485 that is designed for noisy environments and longer cable runs.

Following these wiring practices will greatly improve connection stability, reduce unexplained communication errors, and eliminate many issues that are often incorrectly blamed on ARC, firmware, or drivers.

Guide: Posting in the Community for Help

If you still have trouble connecting to an EZB, you can ask for help in the community support forum. Before posting, work through the troubleshooting steps above. When you are ready to post in the community, include answers to the questions below so others can help you more effectively.

1. What product are you connecting to? (For example: Arduino, ESP32, EZ-Robot JD, Raspberry Pi, Robotis Bioloid, and so on.)

2. What power source is powering the robot? (Battery, wall adapter, voltage, amperage, and similar details.)

3. Does the robot make a startup sound when powered on, if applicable? Does it speak a startup message on boot, if applicable? Have you checked the product manual?

4. If the device also supports Wi-Fi, are you connected to the correct Wi-Fi network for that EZB or robot?

5. Do you have two network adapters? (For example, one for internet access and one for the robot.)

6. What error message is displayed in the status window at the bottom of ARC? Press the **COPY** button in the status window and paste the contents into your forum post.

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

Note: An unwanted EZB disconnection occurs when ARC disconnects from the EZB without notice during an active and valid connection.

This guide is intended for situations where a connection was previously working but is now disconnecting.
If you are unable to connect to the EZB at all, please use the [Connection Troubleshooting Guide]().

ARC release candidates undergo extensive stress testing across multiple virtual machines before being released to the public.
One of the most demanding tests involves controlling four robots continuously (24/7) connected to Exosphere.

These Exosphere robots perform ongoing reliability, efficiency, and stability testing for ARC.
They operate with multiple cameras, tracking systems, Wi-Fi and USB connections, NMS, and more.
Because these robots are always online and accessed globally, ARC itself is highly unlikely to be the root cause of random disconnections or freezing.

---

1. Power

Power issues are the most common cause of disconnections, especially in DIY robot builds.
Motors and servos can draw significant current, particularly under load.

- **Measure current draw:** Determine both average and peak current usage of your robot.
- **Verify power supply capacity:** Ensure your power source can handle peak current demands.
- **Avoid powering motors through EZB pins:**
  While controllers like the EZ-Robot IoTiny and EZ-B v4 offer convenient power pins, their internal traces may not support high current loads.
  This can cause a **brown-out**, where the controller shuts down or becomes unstable due to insufficient power.

View Power Supply Tutorial

---

2. Wi-Fi Connection

Wi-Fi-enabled EZB controllers (such as ESP32, EZ-Robot IoTiny, and EZ-B v4) provide flexibility but are also a common source of instability.
Wireless environments can introduce interference, congestion, and signal degradation.

Recommendation: For production or mission-critical robots, use a USB connection instead of Wi-Fi whenever possible.

- **Check Wi-Fi channel congestion:**
  Use a scanning tool to identify less crowded channels and switch accordingly.

- **Improve signal stability:**
  Reduce distance to the router, minimize interference, or consider a wired connection.

Download Wi-Fi Scan Tool

---

3. Communication Timeouts

Many sensors and peripherals communicate with the EZB using protocols such as I2C.
These devices often require precise timing and acknowledgment signals.

Loose wiring, electrical noise, or unstable connections can cause communication timeouts.
When this happens, the EZB may become unresponsive while waiting for a reply, making ARC appear frozen or disconnected.

- **Inspect I2C connections:**
  Ensure all wiring is secure and properly connected (e.g., RGB Eyes, compass, accelerometer).

- **Check known weak points:**
  For example, EZ-Robot JD Humanoid robots may disconnect if the RGB Eyes cable is loose or faulty.

- **Eliminate noise sources:**
  Keep signal wires short and away from high-current components.

---

4. ARC Project

Finally, review your ARC project configuration.
Excessive scripting or rapid communication with the EZB can overload the communication channel.

- **Reduce script load:**
  Avoid multiple scripts continuously looping and sending commands.
- **Throttle communication:**
  Add delays where appropriate to prevent flooding the EZB with requests.
- **Test incrementally:**
  Disable parts of the project to isolate the cause of the issue.

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.

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

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 doesn't have a camera light, you'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 > CameraStart > Settings > Privacy > Microphone
- In Windows 11, go to: Start > Settings > Privacy & security > CameraStart > 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 you're using a device assigned to you by your workplace or if you've added a work account to your device. If that's the case, you'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.

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

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

EZB Audio Playback Issues

Synthiam ARC supports real-time audio streaming to compatible EZ-B controllers, allowing sound generated on your computer to be played directly through the EZ-B’s onboard audio hardware. When configured correctly, this delivers smooth, responsive audio for speech, sound effects, and interactive robot behaviors.

However, if you
experience audio skipping, looping, or stuttering, the cause is almost always related to how digital audio is streamed
and how wireless interference affects that stream.

How ARC Audio Streaming Works

To understand why audio can skip, it helps to know how ARC and the EZ-B handle audio internally.

Audio generated by Synthiam ARC is transmitted to the EZ-B as a continuous stream of digital packets. These packets
arrive over the active connection (typically Wi-Fi) and are written to a continuous Direct Memory Access (DMA)
buffer inside the EZ-B. The EZ-B's digital-to-analog converter (DAC) continuously reads from this buffer and converts
the digital samples into analog audio output.

Under ideal conditions, new audio data arrives fast enough to keep the buffer filled with fresh samples. The DAC
reads smoothly from the buffer, resulting in uninterrupted audio playback.

Problems occur when the flow of audio packets is interrupted.

Why Audio Skips or Loops

When Wi-Fi interference, congestion, or packet loss occurs, audio packets may arrive late or not at all. The DAC does
not stop when this happens—it continues reading from the DMA buffer at a fixed rate. If new audio data has not yet
replaced the old data in the buffer, the DAC loops over the existing samples.

This behavior causes the most common audio symptoms:

- Repeated words or sounds
- Short looping audio fragments
- Stuttering or choppy playback
- Audio that briefly freezes before continuing

Importantly, this is not a software bug in ARC or a hardware failure in the EZ-B. It is the expected behavior of a
real-time audio system that depends on uninterrupted packet delivery.

Common Causes of Audio Interference

Several factors can interfere with reliable audio streaming:

- Busy or congested Wi-Fi channels

- Nearby routers using overlapping channels
- Microwaves, cordless phones, or other 2.4 GHz devices
- Long distance or weak signal strength
- Multiple devices connecting to the EZ-B’s access point
- Accidentally modified EZ-B firmware settings

Any of these conditions can introduce enough packet delay or loss to cause audible skipping.

Recommended Solutions

1. Use a Wired Connection When Available

If your EZ-B model supports a hardwired USB or UART connection, this is the most reliable solution. A wired connection eliminates Wi-Fi interference and ensures consistent packet delivery for audio streaming.

Wired connections are especially recommended for:

- Robots used in classrooms or public spaces
- Environments with many Wi-Fi networks
- Projects that rely heavily on speech or continuous audio

2. Scan for a Clear Wi-Fi Channel

If you are using Wi-Fi, scan for available channels using a Wi-Fi analyzer on your computer or mobile device.
Look for a channel with minimal overlap and traffic.

Some EZB controllers are connected via WiFi, such as the ESP32, EZ-Robot IoTiny, and EZB v4.
While WiFi is convenient because it doesn’t require wires, it is the second most common cause of issues, next to brownouts due to lower power.
The WiFi connection should operate on a channel that is not saturated, providing the greatest 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 TOOL] to check and switch to a less saturated channel. If possible, consider hardwiring the EZB to the PC.

[Download the WiFi Scan Tool] (WifiInfo.exe)

Once a clear channel is identified, update the EZ-B’s Wi-Fi settings to use that channel. Reducing channel congestion
can dramatically improve audio reliability.

3. Connect the EZ-B to Your Local Network

Instead of running the EZ-B as a standalone hotspot, consider connecting it to your

existing local Wi-Fi network.
When the EZ-B is part of your main network:

- Signal strength is often better
- Your router manages channel selection
- Interference is reduced compared to ad-hoc hotspots

This configuration is strongly recommended for home labs, classrooms, and workshops.

4. Secure the EZ-B Wi-Fi Network

Always set a password on the EZ-B’s Wi-Fi network. Open networks attract nearby devices and people searching for free internet access. Even though the EZ-B does not provide internet routing, it still must handle connection attempts and network traffic from every client that connects.

Excessive connection attempts consume processing time and wireless bandwidth, increasing the likelihood of dropped audio packets.

A password-protected network ensures that only your ARC instance is communicating with the EZ-B.

5. Reset the EZ-B to Factory Settings

Firmware settings can sometimes be changed unintentionally during testing or experimentation. If audio issues suddenly appear and no obvious environmental cause is apparent, resetting the EZ-B to its factory configuration is a good troubleshooting step.

Pressing the RESET button on the EZ-B restores default firmware settings, clears accidental misconfigurations, and often resolves unexplained behavior.

Additional Best Practices

- Keep the EZ-B within a reasonable range of the Wi-Fi source
- Avoid placing the EZ-B near large metal objects or enclosures
- Minimize other high-bandwidth Wi-Fi activity during audio playback
- Ensure your computer running ARC has a stable network connection

Real-time audio streaming is inherently sensitive to network conditions. By reducing interference and ensuring consistent packet delivery, you can achieve smooth, reliable audio playback from your EZ-B.