

Mitsubishi Serial Driver

© 2021 PTC Inc. All Rights Reserved.

Table of Contents

| | |
|--|-----------|
| Mitsubishi Serial Driver | 1 |
| Table of Contents | 2 |
| Mitsubishi Serial Driver | 3 |
| Overview | 3 |
| Setup | 4 |
| Channel Properties — General | 4 |
| Channel Properties — Serial Communications | 5 |
| Channel Properties — Write Optimizations | 8 |
| Channel Properties — Advanced | 9 |
| Device Properties — General | 9 |
| Device Properties — Scan Mode | 10 |
| Device Properties — Timing | 11 |
| Device Properties — Auto-Demotion | 12 |
| Device Properties — Bridging | 13 |
| Device Properties — 32-Bit Data | 13 |
| Device Properties — Redundancy | 14 |
| Multi-Level Networks | 15 |
| Data Types Description | 16 |
| Address Descriptions | 17 |
| Mitsubishi A Series Address Descriptions | 17 |
| Mitsubishi Q Series Address Descriptions | 21 |
| Event Log Messages | 27 |
| Device reported an invalid address or error. Block deactivated. Block address range = '<address>' to '<address>'. | 27 |
| Received block length does not match expected length. Received block length = <number> (bytes), Expected block length = <number> (bytes). | 27 |
| Error code received from device. Error code = <code>h. | 27 |
| Possible network number problem with device. | 27 |
| Possible PC number problem with device. | 28 |
| Error Mask Definitions | 28 |
| Appendix: PLC Setup | 29 |
| A1SJ71C24-R2 Settings and Diagrams | 30 |
| A1SJ71C24-R4 Settings and Diagrams | 31 |
| AJ71C24-S8 Setting and Diagrams | 32 |
| QJ71C24N Configuration | 33 |
| Index | 37 |

Mitsubishi Serial Driver

Help version [1.042](#)

CONTENTS

[Overview](#)

What is the Mitsubishi Serial Driver?

[Setup](#)

How do I configure a device for use with this driver?

[Data Types Description](#)

What data types does this driver support?

[Address Descriptions](#)

How do I address a data location on a Mitsubishi Serial device?

[Event Log Messages](#)

What error messages does the Mitsubishi Serial Driver produce?

[Appendix: PLC Setup](#)

Overview

The Mitsubishi Serial Driver provides a reliable way to connect Mitsubishi Serial devices to client applications; including HMI, SCADA, Historian, MES, ERP, and countless custom applications. It is intended for use with Mitsubishi A and Q Series devices communicating via the AJ71C24 and QJ71C24N communications cards (among other communication cards that support the specified communication protocols). *For more information, refer to [Setup](#).*

Setup

Supported Devices

A-Series PLCs
Q-Series PLCs

Channel and Device Limits

The maximum number of channels supported by this driver is 100. The maximum number of devices supported by this driver is 32 per channel.

Communication Protocol

Format 1 with Checksum Enabled for A Series
Format 5 with Checksum Enabled for Q Series

Supported Communication Parameters

Baud Rate: 300, 600, 1200, 2400, 9600, 19200 or 38400
Parity: None, Even or Odd
Data Bits: 5, 6, 7 or 8
Stop Bits: 1 or 2

Ethernet Encapsulation

This driver supports Ethernet Encapsulation, which allows the driver to communicate with serial devices attached to an Ethernet network using a serial-to-Ethernet terminal server. Ethernet Encapsulation can be enable through the channel properties. For more information, refer to the server help file.

• See Also: [Appendix: PLC Setup](#)

Channel Properties — General

This server supports the use of multiple simultaneous communications drivers. Each protocol or driver used in a server project is called a channel. A server project may consist of many channels with the same communications driver or with unique communications drivers. A channel acts as the basic building block of an OPC link. This group is used to specify general channel properties, such as the identification attributes and operating mode.

| | | |
|---------------------|--|---------|
| Property Groups | <input type="checkbox"/> Identification | |
| General | Name | |
| Write Optimizations | Description | |
| Advanced | Driver | |
| | <input type="checkbox"/> Diagnostics | |
| | Diagnostics Capture | Disable |

Identification

Name: Specify the user-defined identity of this channel. In each server project, each channel name must be unique. Although names can be up to 256 characters, some client applications have a limited display window when browsing the OPC server's tag space. The channel name is part of the OPC browser information. The property is required for creating a channel.

• For information on reserved characters, refer to "How To... Properly Name a Channel, Device, Tag, and Tag Group" in the server help.

Description: Specify user-defined information about this channel.

Many of these properties, including Description, have an associated system tag.

Driver: Specify the protocol / driver for this channel. This property specifies the device driver that was selected during channel creation. It is a disabled setting in the channel properties. The property is required for creating a channel.

Note: With the server's online full-time operation, these properties can be changed at any time. This includes changing the channel name to prevent clients from registering data with the server. If a client has already acquired an item from the server before the channel name is changed, the items are unaffected. If, after the channel name has been changed, the client application releases the item and attempts to re-acquire using the old channel name, the item is not accepted. Changes to the properties should not be made once a large client application has been developed. Utilize proper user role and privilege management to prevent operators from changing properties or accessing server features.

Diagnostics

Diagnostics Capture: When enabled, this option makes the channel's diagnostic information available to OPC applications allows the usage of statistics tags that provide feedback to client applications regarding the operation of the channel. Because the server's diagnostic features require a minimal amount of overhead processing, it is recommended that they be utilized when needed and disabled when not. The default is disabled.

Note: This property is not available if the driver does not support diagnostics.

For more information, refer to "Communication Diagnostics" and "Statistics Tags" in the server help.

Channel Properties — Serial Communications

Serial communication properties are available to serial drivers and vary depending on the driver, connection type, and options selected. Below is a superset of the possible properties.

Click to jump to one of the sections: [Connection Type](#), [Serial Port Settings](#) or [Ethernet Settings](#), and [Operational Behavior](#).

Note: With the server's online full-time operation, these properties can be changed at any time. Utilize proper user role and privilege management to prevent operators from changing properties or accessing server features.

| Property Groups | | |
|------------------------------|--|------------|
| General | | |
| Serial Communications | | |
| Write Optimizations | | |
| Advanced | | |
| | <input type="checkbox"/> Connection Type | |
| | Physical Medium | COM Port |
| | <input type="checkbox"/> Serial Port Settings | |
| | COM ID | 39 |
| | Baud Rate | 19200 |
| | Data Bits | 8 |
| | Parity | None |
| | Stop Bits | 1 |
| | Flow Control | RTS Always |
| | <input type="checkbox"/> Operational Behavior | |
| | Report Communication Errors | Enable |
| | Close Idle Connection | Enable |
| | Idle Time to Close (s) | 15 |

Connection Type

Physical Medium: Choose the type of hardware device for data communications. Options include COM Port, None, Modem, and Ethernet Encapsulation. The default is COM Port.

- **None:** Select None to indicate there is no physical connection, which displays the [Operation with no Communications](#) section.
- **COM Port:** Select Com Port to display and configure the [Serial Port Settings](#) section.
- **Modem:** Select Modem if phone lines are used for communications, which are configured in the [Modem Settings](#) section.
- **Ethernet Encap.:** Select if Ethernet Encapsulation is used for communications, which displays the [Ethernet Settings](#) section.
- **Shared:** Verify the connection is correctly identified as sharing the current configuration with another channel. This is a read-only property.

Serial Port Settings

COM ID: Specify the Communications ID to be used when communicating with devices assigned to the channel. The valid range is 1 to 9991 to 16. The default is 1.

Baud Rate: Specify the baud rate to be used to configure the selected communications port.


Data Bits: Specify the number of data bits per data word. Options include 5, 6, 7, or 8.

Parity: Specify the type of parity for the data. Options include Odd, Even, or None.

Stop Bits: Specify the number of stop bits per data word. Options include 1 or 2.

Flow Control: Select how the RTS and DTR control lines are utilized. Flow control is required to communicate with some serial devices. Options are:

- **None:** This option does not toggle or assert control lines.
- **DTR:** This option asserts the DTR line when the communications port is opened and remains on.
- **RTS:** This option specifies that the RTS line is high if bytes are available for transmission. After all buffered bytes have been sent, the RTS line is low. This is normally used with RS232/RS485 converter hardware.
- **RTS, DTR:** This option is a combination of DTR and RTS.
- **RTS Always:** This option asserts the RTS line when the communication port is opened and remains on.
- **RTS Manual:** This option asserts the RTS line based on the timing properties entered for RTS Line Control. It is only available when the driver supports manual RTS line control (or when the properties are shared and at least one of the channels belongs to a driver that provides this support).
RTS Manual adds an **RTS Line Control** property with options as follows:
 - **Raise:** This property specifies the amount of time that the RTS line is raised prior to data transmission. The valid range is 0 to 9999 milliseconds. The default is 10 milliseconds.
 - **Drop:** This property specifies the amount of time that the RTS line remains high after data transmission. The valid range is 0 to 9999 milliseconds. The default is 10 milliseconds.
 - **Poll Delay:** This property specifies the amount of time that polling for communications is delayed. The valid range is 0 to 9999. The default is 10 milliseconds.

 **Tip:** When using two-wire RS-485, "echoes" may occur on the communication lines. Since this communication does not support echo suppression, it is recommended that echoes be disabled or a RS-485 converter be used.

Operational Behavior

- **Report Communication Errors:** Enable or disable reporting of low-level communications errors. When enabled, low-level errors are posted to the Event Log as they occur. When disabled, these same errors are not posted even though normal request failures are. The default is Enable.
- **Close Idle Connection:** Choose to close the connection when there are no longer any tags being referenced by a client on the channel. The default is Enable.
- **Idle Time to Close:** Specify the amount of time that the server waits once all tags have been removed before closing the COM port. The default is 15 seconds.

Ethernet Settings

🔴 **Note:** Not all serial drivers support Ethernet Encapsulation. If this group does not appear, the functionality is not supported.

Ethernet Encapsulation provides communication with serial devices connected to terminal servers on the Ethernet network. A terminal server is essentially a virtual serial port that converts TCP/IP messages on the Ethernet network to serial data. Once the message has been converted, users can connect standard devices that support serial communications to the terminal server. The terminal server's serial port must be properly configured to match the requirements of the serial device to which it is attached. *For more information, refer to "Using Ethernet Encapsulation" in the server help.*

- **Network Adapter:** Indicate a network adapter to bind for Ethernet devices in this channel. Choose a network adapter to bind to or allow the OS to select the default.
 - *Specific drivers may display additional Ethernet Encapsulation properties. For more information, refer to [Channel Properties — Ethernet Encapsulation](#).*

Modem Settings

- **Modem:** Specify the installed modem to be used for communications.
- **Connect Timeout:** Specify the amount of time to wait for connections to be established before failing a read or write. The default is 60 seconds.
- **Modem Properties:** Configure the modem hardware. When clicked, it opens vendor-specific modem properties.
- **Auto-Dial:** Enables the automatic dialing of entries in the Phonebook. The default is Disable. *For more information, refer to "Modem Auto-Dial" in the server help.*
- **Report Communication Errors:** Enable or disable reporting of low-level communications errors. When enabled, low-level errors are posted to the Event Log as they occur. When disabled, these same errors are not posted even though normal request failures are. The default is Enable.
- **Close Idle Connection:** Choose to close the modem connection when there are no longer any tags being referenced by a client on the channel. The default is Enable.
- **Idle Time to Close:** Specify the amount of time that the server waits once all tags have been removed before closing the modem connection. The default is 15 seconds.

Operation with no Communications

- **Read Processing:** Select the action to be taken when an explicit device read is requested. Options include Ignore and Fail. Ignore does nothing; Fail provides the client with an update that indicates failure. The default setting is Ignore.

Channel Properties — Write Optimizations

The server must ensure that the data written from the client application gets to the device on time. Given this goal, the server provides optimization properties to meet specific needs or improve application responsiveness.

| | | |
|----------------------------|---|--------------------------------------|
| Property Groups | <input type="checkbox"/> Write Optimizations | |
| General | Optimization Method | Write Only Latest Value for All Tags |
| Write Optimizations | Duty Cycle | 10 |
| | | |

Write Optimizations

Optimization Method: Controls how write data is passed to the underlying communications driver. The options are:

- **Write All Values for All Tags:** This option forces the server to attempt to write every value to the controller. In this mode, the server continues to gather write requests and add them to the server's internal write queue. The server processes the write queue and attempts to empty it by writing data to the device as quickly as possible. This mode ensures that everything written from the client applications is sent to the target device. This mode should be selected if the write operation order or the write item's content must uniquely be seen at the target device.
- **Write Only Latest Value for Non-Boolean Tags:** Many consecutive writes to the same value can accumulate in the write queue due to the time required to actually send the data to the device. If the server updates a write value that has already been placed in the write queue, far fewer writes are needed to reach the same final output value. In this way, no extra writes accumulate in the server's queue. When the user stops moving the slide switch, the value in the device is at the correct value at virtually the same time. As the mode states, any value that is not a Boolean value is updated in the server's internal write queue and sent to the device at the next possible opportunity. This can greatly improve the application performance.
 - **Note:** This option does not attempt to optimize writes to Boolean values. It allows users to optimize the operation of HMI data without causing problems with Boolean operations, such as a momentary push button.
- **Write Only Latest Value for All Tags:** This option takes the theory behind the second optimization mode and applies it to all tags. It is especially useful if the application only needs to send the latest value to the device. This mode optimizes all writes by updating the tags currently in the write queue before they are sent. This is the default mode.

Duty Cycle: is used to control the ratio of write to read operations. The ratio is always based on one read for every one to ten writes. The duty cycle is set to ten by default, meaning that ten writes occur for each read operation. Although the application is performing a large number of continuous writes, it must be ensured that read data is still given time to process. A setting of one results in one read operation for every write operation. If there are no write operations to perform, reads are processed continuously. This allows optimization for applications with continuous writes versus a more balanced back and forth data flow.

● **Note:** It is recommended that the application be characterized for compatibility with the write optimization enhancements before being used in a production environment.

Channel Properties — Advanced

This group is used to specify advanced channel properties. Not all drivers support all properties; so the Advanced group does not appear for those devices.

| | | |
|---------------------|---|-------------------|
| Property Groups | <input type="checkbox"/> Non-Normalized Float Handling | |
| General | Floating-Point Values | Replace with Zero |
| Write Optimizations | <input type="checkbox"/> Inter-Device Delay | |
| Advanced | Inter-Device Delay (ms) | 0 |

Non-Normalized Float Handling: A non-normalized value is defined as Infinity, Not-a-Number (NaN), or as a Denormalized Number. The default is Replace with Zero. Drivers that have native float handling may default to Unmodified. Non-normalized float handling allows users to specify how a driver handles non-normalized IEEE-754 floating point data. Descriptions of the options are as follows:

- **Replace with Zero:** This option allows a driver to replace non-normalized IEEE-754 floating point values with zero before being transferred to clients.
- **Unmodified:** This option allows a driver to transfer IEEE-754 denormalized, normalized, non-number, and infinity values to clients without any conversion or changes.

● **Note:** This property is not available if the driver does not support floating-point values or if it only supports the option that is displayed. According to the channel's float normalization setting, only real-time driver tags (such as values and arrays) are subject to float normalization. For example, EFM data is not affected by this setting.

● *For more information on the floating-point values, refer to "How To ... Work with Non-Normalized Floating-Point Values" in the server help.*

Inter-Device Delay: Specify the amount of time the communications channel waits to send new requests to the next device after data is received from the current device on the same channel. Zero (0) disables the delay.

● **Note:** This property is not available for all drivers, models, and dependent settings.

Device Properties — General

| | | |
|-----------------|--|-------------------|
| Property Groups | <input type="checkbox"/> Identification | |
| General | Name | A Series |
| Scan Mode | Description | |
| Timing | Channel Assignment | Mitsubishi Serial |
| Auto-Demotion | Driver | Mitsubishi Serial |
| Bridging | Model | A Series |
| 32-Bit Data | ID Format | Decimal |
| Redundancy | ID | 0 |
| | <input type="checkbox"/> Operating Mode | |
| | Data Collection | Enable |
| | Simulated | No |

Identification

Name: User-defined identity of this device.

Description: User-defined information about this device.

Channel Assignment: User-defined name of the channel to which this device currently belongs.

Driver: Selected protocol driver for this device.

Model: The specific version of the device. Properties can vary slightly based on the device product model.

ID Format: Select how the device identity is formatted. Options include Decimal, Octal, and Hex.

ID: the station number of the device for communication with the driver. This may be the ultimate destination device for read and write commands or act as the bridge device to another network. The ID number can range from 0 to 31.

Operating Mode

Data Collection: This property controls the device's active state. Although device communications are enabled by default, this property can be used to disable a physical device. Communications are not attempted when a device is disabled. From a client standpoint, the data is marked as invalid and write operations are not accepted. This property can be changed at any time through this property or the device system tags.

Simulated: This option places the device into Simulation Mode. In this mode, the driver does not attempt to communicate with the physical device, but the server continues to return valid OPC data. Simulated stops physical communications with the device, but allows OPC data to be returned to the OPC client as valid data. While in Simulation Mode, the server treats all device data as reflective: whatever is written to the simulated device is read back and each OPC item is treated individually. The item's memory map is based on the group Update Rate. The data is not saved if the server removes the item (such as when the server is reinitialized). The default is No.

Notes:

1. This System tag (_Simulated) is read only and cannot be written to for runtime protection. The System tag allows this property to be monitored from the client.
2. In Simulation mode, the item's memory map is based on client update rate(s) (Group Update Rate for OPC clients or Scan Rate for native and DDE interfaces). This means that two clients that reference the same item with different update rates return different data.

Simulation Mode is for test and simulation purposes only. It should never be used in a production environment.

Device Properties — Scan Mode

The Scan Mode specifies the subscribed-client requested scan rate for tags that require device communications. Synchronous and asynchronous device reads and writes are processed as soon as possible; unaffected by the Scan Mode properties.

| | | |
|------------------|----------------------------|--------------------------------------|
| Property Groups | ☐ Scan Mode | |
| General | Scan Mode | Respect Client-Specified Scan Rate ▼ |
| Scan Mode | Initial Updates from Cache | Disable |
| | | |

Scan Mode: Specify how tags in the device are scanned for updates sent to subscribing clients. Descriptions of the options are:

- **Respect Client-Specified Scan Rate:** This mode uses the scan rate requested by the client.
- **Request Data No Faster than Scan Rate:** This mode specifies the value set as the maximum scan rate. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.
 - **Note:** When the server has an active client and items for the device and the scan rate value is increased, the changes take effect immediately. When the scan rate value is decreased, the changes do not take effect until all client applications have been disconnected.
- **Request All Data at Scan Rate:** This mode forces tags to be scanned at the specified rate for subscribed clients. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.
- **Do Not Scan, Demand Poll Only:** This mode does not periodically poll tags that belong to the device nor perform a read to get an item's initial value once it becomes active. It is the OPC client's responsibility to poll for updates, either by writing to the `_DemandPoll` tag or by issuing explicit device reads for individual items. *For more information, refer to "Device Demand Poll" in server help.*
- **Respect Tag-Specified Scan Rate:** This mode forces static tags to be scanned at the rate specified in their static configuration tag properties. Dynamic tags are scanned at the client-specified scan rate.

Initial Updates from Cache: When enabled, this option allows the server to provide the first updates for newly activated tag references from stored (cached) data. Cache updates can only be provided when the new item reference shares the same address, scan rate, data type, client access, and scaling properties. A device read is used for the initial update for the first client reference only. The default is disabled; any time a client activates a tag reference the server attempts to read the initial value from the device.

Device Properties — Timing

The device Timing properties allow the driver's response to error conditions to be tailored to fit the application's needs. In many cases, the environment requires changes to these properties for optimum performance. Factors such as electrically generated noise, modem delays, and poor physical connections can influence how many errors or timeouts a communications driver encounters. Timing properties are specific to each configured device.

| | | |
|-----------------|--|------|
| Property Groups | <input type="checkbox"/> Communication Timeouts | |
| General | Connect Timeout (s) | 3 |
| Scan Mode | Request Timeout (ms) | 1000 |
| Timing | Attempts Before Timeout | 3 |
| Redundancy | <input type="checkbox"/> Timing | |
| | Inter-Request Delay (ms) | 0 |

Communications Timeouts

Connect Timeout: This property (which is used primarily by Ethernet based drivers) controls the amount of time required to establish a socket connection to a remote device. The device's connection time often takes longer than normal communications requests to that same device. The valid range is 1 to 30 seconds. The default is typically 3 seconds, but can vary depending on the driver's specific nature. If this setting is not supported by the driver, it is disabled.

● **Note:** Due to the nature of UDP connections, the connection timeout setting is not applicable when communicating via UDP.

Request Timeout: Specify an interval used by all drivers to determine how long the driver waits for a response from the target device to complete. The valid range is 50 to 9,999,999 milliseconds (167.6667

minutes). The default is usually 1000 milliseconds, but can vary depending on the driver. The default timeout for most serial drivers is based on a baud rate of 9600 baud or better. When using a driver at lower baud rates, increase the timeout to compensate for the increased time required to acquire data.

Attempts Before Timeout: Specify how many times the driver issues a communications request before considering the request to have failed and the device to be in error. The valid range is 1 to 10. The default is typically 3, but can vary depending on the driver's specific nature. The number of attempts configured for an application depends largely on the communications environment. This property applies to both connection attempts and request attempts.

Timing

Inter-Request Delay: Specify how long the driver waits before sending the next request to the target device. It overrides the normal polling frequency of tags associated with the device, as well as one-time reads and writes. This delay can be useful when dealing with devices with slow turnaround times and in cases where network load is a concern. Configuring a delay for a device affects communications with all other devices on the channel. It is recommended that users separate any device that requires an inter-request delay to a separate channel if possible. Other communications properties (such as communication serialization) can extend this delay. The valid range is 0 to 300,000 milliseconds; however, some drivers may limit the maximum value due to a function of their particular design. The default is 0, which indicates no delay between requests with the target device.

● **Note:** Not all drivers support Inter-Request Delay. This setting does not appear if it is not available.

Device Properties — Auto-Demotion

The Auto-Demotion properties can temporarily place a device off-scan in the event that a device is not responding. By placing a non-responsive device offline for a specific time period, the driver can continue to optimize its communications with other devices on the same channel. After the time period has been reached, the driver re-attempts to communicate with the non-responsive device. If the device is responsive, the device is placed on-scan; otherwise, it restarts its off-scan time period.

| | | |
|----------------------|-------------------------------|---------|
| Property Groups | Auto-Demotion | |
| General | Demote on Failure | Enable |
| Scan Mode | Timeouts to Demote | 3 |
| Timing | Demotion Period (ms) | 10000 |
| Auto-Demotion | Discard Requests when Demoted | Disable |

Demote on Failure: When enabled, the device is automatically taken off-scan until it is responding again.

● **Tip:** Determine when a device is off-scan by monitoring its demoted state using the `_AutoDemoted` system tag.

Timeouts to Demote: Specify how many successive cycles of request timeouts and retries occur before the device is placed off-scan. The valid range is 1 to 30 successive failures. The default is 3.

Demotion Period: Indicate how long the device should be placed off-scan when the timeouts value is reached. During this period, no read requests are sent to the device and all data associated with the read requests are set to bad quality. When this period expires, the driver places the device on-scan and allows for another attempt at communications. The valid range is 100 to 3600000 milliseconds. The default is 10000 milliseconds.

Discard Requests when Demoted: Select whether or not write requests should be attempted during the off-scan period. Disable to always send write requests regardless of the demotion period. Enable to discard

writes; the server automatically fails any write request received from a client and does not post a message to the Event Log.

Device Properties — Bridging

The Bridging properties are used to specify the desired routing path.

| | | |
|-----------------|--|-----|
| Property Groups | <input type="checkbox"/> Bridging | |
| General | Network Number | 0 |
| Scan Mode | PC Number | 255 |
| Timing | | |
| Auto-Demotion | | |
| Bridging | | |
| 32-Bit Data | | |
| Redundancy | | |

Network Number: This parameter specifies the Network Number of the target device. The valid range is 0 to 255, with 0 being used for local connections. The default setting is 0.

● **Note:** This setting only applies to the Q Series model.

PC Number: This parameter specifies the PC Number of the target device. The valid range is 0 to 64, with 255 being used for local connection. The default setting is 255.

● *If this device communicates with devices on remote networks, refer to [Appendix: Multi-Level Networks](#).*

Device Properties — 32-Bit Data

| | | |
|--------------------|---|--------|
| Property Groups | <input type="checkbox"/> 32-Bit Data | |
| General | First Word Low | Enable |
| Bridging | | |
| 32-Bit Data | | |

In a Mitsubishi A-Series device, the addresses of two consecutive registers are used for 32-bit data types.

First Word Low: Set to Enable for the driver to assume the first word is low for the 32-bit value. Set to Disable for the driver to assume that the first word is high for the 32-bit value. The default is Enable.

● **Note:** This property can't be changed while there are active references on the device.

Device Properties — Redundancy

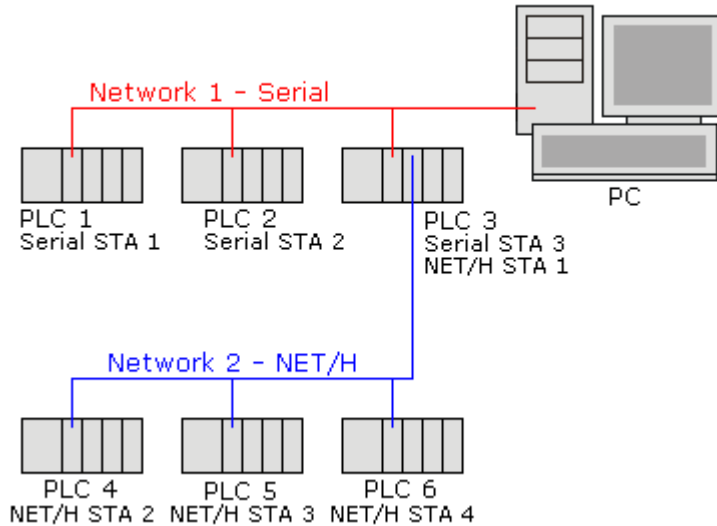
| | | |
|---------------------|------------------------|---------------------|
| Property Groups | [-] Redundancy | |
| General | Secondary Path | Channel.Device1 ... |
| Scan Mode | Operating Mode | Switch On Failure |
| Timing | Monitor Item | |
| Auto-Demotion | Monitor Interval (s) | 300 |
| Tag Generation | Return to Primary ASAP | Yes |
| Tag Import Settings | | |
| Redundancy | | |

Redundancy is available with the Media-Level Redundancy Plug-In.

• Consult the website, a sales representative, or the [user manual](#) for more information.

Multi-Level Networks

The Q Series model can be used to communicate with devices on remote networks. In the example shown below, PLC 1, PLC 2, and PLC 3 are on the local serial network. PLC 4, PLC 5, and PLC 6 are on a remote NET/H network. PLC 3 serves as a relay device connecting the two networks.



• For more information, refer to [Setup](#).

For example, imagine that PLC 1, PLC 2, and PLC 3 have QJ71C24N serial communication modules configured as serial stations 1, 2, and 3 respectively. In addition to the serial module, PLC 3 also has a QJ71BR11 NET/H module configured as NET/H station 1. PLC 4, PLC 5, and PLC 6 have NET/H modules configured as NET/H stations 2, 3, and 4 respectively.

To communicate with all 6 PLCs, 6 devices must be created in the server project. The device IDs and network settings should be as shown below.

| PLC | Device ID | Network Number | PC Number | Comment |
|-----|-----------|----------------|-----------|---|
| 1 | 1 | 0 | 255 | Local network, local PC |
| 2 | 2 | 0 | 255 | Local network, local PC |
| 3 | 3 | 0 | 255 | Local network, local PC |
| 4 | 3 | 2 | 2 | Network 2, PC 2 (NET/H STA 2), via PLC 3 (serial STA 3) |
| 5 | 3 | 2 | 3 | Network 2, PC 3 (NET/H STA 3), via PLC 3 (serial STA 3) |
| 6 | 3 | 2 | 4 | Network 2, PC 4 (NET/H STA 4), via PLC 3 (serial STA 3) |

• For more information, refer to [Bridging](#).

• **Note:** A relay device may take 5 or more seconds to report a failed Read and Write to a remote device. It is recommended that the request timeout for remote devices be set accordingly. For more information, refer to [Device Setup](#).

Data Types Description

The Mitsubishi Serial Driver supports the following data types.

| Data Type | Description |
|-----------|---|
| Boolean | Single bit |
| Word | Unsigned 16-bit value bit 0 is the low bit bit 15 is the high bit |
| Short | Signed 16-bit value bit 0 is the low bit bit 14 is the high bit bit 15 is the sign bit |
| BCD | Two byte packed BCD Value range is 0-9999. Behavior is undefined for values beyond this range. |
| DWord | Unsigned 32-bit value bit 0 is the low bit bit 31 is the high bit |
| Long | Signed 32-bit value bit 0 is the low bit bit 30 is the high bit bit 31 is the sign bit |
| Float | 32-bit floating point value |
| String | Null terminated ASCII string Support, includes HiLo LoHi byte order selection and string lengths up to 128 bytes. |
| LBCD | Four byte packed BCD Value range is 0 - 99999999. Behavior is undefined for values beyond this range. |

Address Descriptions

Address specifications vary depending on the model in use. Select a link from the following list to obtain specific address information for the model of interest.

[A Series](#)

[Q Series](#)

Mitsubishi A Series Address Descriptions

Address specifications vary depending on the model in use. The default data types for dynamically defined tags are shown in **bold**.

| Device Type | Range | Data Type | Access |
|-----------------------|-----------------------|-------------------|------------|
| Inputs* | X0000-X1FFF (hex) | Boolean | Read/Write |
| | X0000-X1FF0 (hex) | Short, Word, BCD | |
| | X0000-X1FE0 (hex) | Long, DWord, LBCD | |
| Outputs* | Y0000-Y1FFF (hex) | Boolean | Read/Write |
| | Y0000-Y1FF0 (hex) | Short, Word, BCD | |
| | Y0000-Y1FE0 (hex) | Long, DWord, LBCD | |
| Link Relays* | B0000-B1FFF (hex) | Boolean | Read/Write |
| | B0000-B1FF0 (hex) | Short, Word, BCD | |
| | BCD B0000-B1FE0 (hex) | Long, DWord, LBCD | |
| Internal Relays* | M0000-M8191 | Boolean | Read/Write |
| | M0000-M8176 | Short, Word, BCD | |
| | M0000-M8160 | Long, DWord, LBCD | |
| Special Int. Relays * | M9000-M9255 | Boolean | Read Only |
| | M9000-M9240 | Short, Word, BCD | |
| | M9000-M9224 | Long, DWord, LBCD | |
| Latch Relays* | L0000-L8191 | Boolean | Read/Write |
| | L0000-L8176 | Short, Word, BCD | |
| | L0000-L8160 | Long, DWord, LBCD | |
| Annunciator Relays* | F0000-F2047 | Boolean | Read/Write |
| | F0000-F2032 | Short, Word, BCD | |

| Device Type | Range | Data Type | Access |
|-------------------|---------------|-------------------|------------|
| | F0000-F2016 | Long, DWord, LBCD | |
| Step Relays* | S0000-S8191 | Boolean | Read/Write |
| | S0000-S8176 | Short, Word, BCD | |
| | S0000-S8160 | Long, DWord, LBCD | |
| Timer Contacts* | TS0000-TS2047 | Boolean | Read/Write |
| | TS0000-TS2032 | Short, Word, BCD | |
| | TS0000-TS2016 | Long, DWord, LBCD | |
| Timer Coils* | TC0000-TC2047 | Boolean | Read/Write |
| | TC0000-TC2032 | Short, Word, BCD | |
| | TC0000-TC2016 | Long, DWord, LBCD | |
| Counter Contacts* | CS0000-CS1023 | Boolean | Read/Write |
| | CS0000-CS1008 | Short, Word, BCD | |
| | CS0000-CS0992 | Long, DWord, LBCD | |
| Counter Coils* | CC0000-CC1023 | Boolean | Read/Write |
| | CC0000-CC1008 | Short, Word, BCD | |
| | CC0000-CC0992 | Long, DWord, LBCD | |

* Users can specify a Long data type by appending a space and an "L" to the address. For example, "D00000" would be entered as "D00000 L". This does not apply to arrays or bit accessed registers.

● **Notes:**

1. In a typical A-series PLC memory configuration, L, S and M addresses all map to the same M memory in the PLC.
2. All Boolean device types can be accessed as Short, Word, BCD, Long, DWord and LBCD. However, the device must be addressed on a 16-bit boundary.

| Device Type | Range | Data Type | Access |
|---------------|---------------|--------------------------|------------|
| Timer Value | TN0000-TN2047 | Boolean | Read/Write |
| | TN0000-TN2047 | Short , Word, BCD | |
| Counter Value | CN0000-CN1023 | Boolean | Read/Write |
| | CN0000-CN1023 | Short, Word , BCD | |

| Device Type | Range | Data Type | Access |
|--|--|-----------------------------|------------|
| Data Registers*** | D0000-D8191 | Short, Word, BCD | Read/Write |
| | D0000-D8190 | Long, DWord, LBCD, Float | |
| Data Register Bit Access | D0000.00-D8191.15* | Short, Word, BCD, Boolean** | Read/Write |
| | D0000.00-D8190.31* | Long, DWord, LBCD | |
| Data Registers String Access HiLo Byte Ordering | DSH00000.002-DSH08190.002 DSH00000.128-DSH08127.128 The string length may also be specified using a colon. The string length must be between 2-128 bytes and even. | String | Read/Write |
| Data Registers String Access LoHi Byte Ordering | DSL00000.002-DSL08190.002 DSL00000.128-DSL08127.128 The string length may also be specified using a colon. The string length must be between 2-128 bytes and even. | String | Read/Write |
| Special Data Registers*** | D90000-D9255 | Short, Word, BCD | Read Only |
| | D9000-D9254 | Long, DWord, LBCD, Float | |
| Data Register Bit Access | D90000.00-D9255.15* | Short, Word, BCD, Boolean** | Read Only |
| | D90000.00-D9254.31* | Long, DWord, LBCD | |
| Link Registers*** | W0000-W1FFF (hex) | Short, Word, BCD | Read/Write |
| | W0000-W1FFE (hex) | Long, DWord, Float, LBCD | |
| Link Register Bit Access | W0000.00-W1FFF.15* | Short, Word, BCD, Boolean** | Read/Write |
| | W0000.00-W1FFE.31* | Long, DWord, LBCD | |
| Link Registers String Access HiLo Byte Ordering | WSH0000.002-WSH1FFE.002 WSH0000.128-WSH1FBF.128 The string length may also be specified using a colon. The | String | Read/Write |

| Device Type | Range | Data Type | Access |
|--|--|--|------------|
| | string length must be between 2-128 bytes and even. | | |
| Link Registers String Access LoHi Byte Ordering | WSL0000.002-WSL1FFE.002 WSL0000.128-WSL1FBF.128 The string length may also be specified using a colon. The string length must be between 2-128 bytes and even. | String | Read/Write |
| File Register*** | R0000-R8191 R0000-R8190 | Short , Word, BCD Long, DWord, Float, LBCD | Read/Write |
| File Register Bit Access | R0000.00-R8191.15* R0000.00-R8190.31* | Short , Word, BCD, Boolean** Long, DWord, LBCD | Read/Write |
| File Registers String Access HiLo Byte Ordering | RSH00000.002-RSH08190.002 RSH00000.128-RSH08127.128 The string length may also be specified using a colon. The string length must be between 2-128 bytes and even. | String | Read/Write |
| File Registers String Access LoHi Byte Ordering | RSL00000.002-RSL08190.002 RSL00000.128-RSL08127.128 The string length may also be specified using a colon. The string length must be between 2-128 bytes and even. | String | Read/Write |

* For register memory, the data types Short, Word, BCD, DWord, Long, Boolean and LBCD can append an optional ".bb" (dot bit) or a ":bb" (colon bit) to the address to reference a bit in a particular value. The valid ranges for the optional bit is 0-15 for Short, Word, BCD, and Boolean; and 0-31 for Long, DWord and LBCD. Strings use the bit number to specify length. The valid length of a string in D memory is 2 to 128 bytes. The string length must be even numbers. Float types do not support bit operations. The bit number is always in decimal notation.

** When accessing register memory as Boolean, a bit number is required.

*** Users can specify a Long data type by appending a space and an "L" to the address. For example, "D00000" would be entered as "D00000 L". This does not apply to arrays or bit accessed registers.

Array Access

Device types can be accessed as arrays of Short, Word, BCD, Long, DWord, LBCD or Float. The default setting is Word. The maximum allowed array size depends on the data type and device type. The maximum array size for register device types is 32 for Long, DWord, Float and LBCD data type and 64 for Word, Short, and BCD data types. The maximum array size for bit device types except TC, TS, CC, CS is 16 for Long, DWord and LBCD data types and 32 for Word, Short, and BCD data types. For TC, TS, CC, CS device types the

maximum array size is 15 for Word data types and 7 for DWord data types. Arrays can be either 1 dimension or 2. Regardless of the dimensions, the array size must not exceed the limits already stated.

☀ Due to protocol limitation, the largest bit device type array that can be successfully written to is 10 Words (5 DWords).

Array Examples

1. D100 [4] Single dimension includes the following register addresses: D100, D101, D102, D103.
2. M016 [3][4] Two Dimensions includes the following device addresses as words: M016, M032, M048, M064, M080, M096, M112, M128, M144, M160, M176, M192 3 rows x 4 columns = 12 words 12 x 16 (word) = 192 total bits.

Additional Device Examples

1. Access X device memory as Word : X??? where the ??? is a hex number on 16-bit boundaries such as 010, 020, 030 and so forth.
2. Access M device memory as Long : M???? where the ??? is a decimal number on 16-bit boundaries such as 0, 16, 32, 48 and so forth.

Mitsubishi Q Series Address Descriptions

Address specifications vary depending on the model in use. The default data types for dynamically defined tags are shown in **bold**.

| Device Type | Range | Data Type | Access |
|-----------------|---------------------|-------------------|------------|
| Inputs* | X0000-X3FFF (Hex) | Boolean | Read/Write |
| | X0000-X3FF0 (Hex) | Short, Word, BCD | |
| | X0000-X3FE0 (Hex) | Long, DWord, LBCD | |
| Direct Inputs* | DX0000-DX3FFF (Hex) | Boolean | Read/Write |
| | DX0000-DX3FF0 (Hex) | Short, Word, BCD | |
| | DX0000-DX3FE0 (Hex) | Long, DWord, LBCD | |
| Outputs | Y0000-Y3FFF (Hex) | Boolean | Read/Write |
| | Y0000-Y3FFF (Hex) | Short, Word, BCD | |
| | Y0000-Y3FF0 (Hex) | | |
| Direct Outputs* | DY0000-DY3FFF (Hex) | Boolean | Read/Write |
| | DY0000-DY3FF0 (Hex) | Short, Word, BCD | |
| | DY0000-DY3FE0 (Hex) | Long, DWord, LBCD | |
| Link Relays* | B0000-B3FFF (Hex) | Boolean | Read/Write |

| Device Type | Range | Data Type | Access |
|----------------------|---------------------|-------------------|------------|
| | B0000-B3FF0 (Hex) | Short, Word, BCD | |
| | B0000-B3FE0 (Hex) | Long, DWord, LBCD | |
| Special Link Relays | SB0000-SB07FF (Hex) | Boolean | Read/Write |
| | SB0000-SB07F0 (Hex) | Short, Word, BCD | |
| | SB0000-SB07E0 (Hex) | Short, Word, BCD | |
| Internal Relays* | M0000-M16383 | Boolean | Read/Write |
| | M0000-M16368 | Short, Word, BCD | |
| | M0000-M16352 | Long, DWord, LBCD | |
| Special Int. Relays* | SM0000-SM2047 | Boolean | Read/Write |
| | SM0000-SM2032 | Short, Word, BCD | |
| | SM0000-SM2016 | Long, DWord, LBCD | |
| Latch Relays* | L0000-L16383 | Boolean | Read/Write |
| | L0000-L16368 | Short, Word, BCD | |
| | L0000-L16352 | Long, DWord, LBCD | |
| Annunciator Relays* | F0000-F2047 | Boolean | Read/Write |
| | F0000-F2032 | Short, Word, BCD | |
| | F0000-F2016 | Long, DWord, LBCD | |
| Edge Relays* | V0000-V2047 | Boolean | Read/Write |
| | V0000-V2032 | Short, Word, BCD | |
| | V0000-V2016 | Long, DWord, LBCD | |
| Step Relays* | S0000-S16383 | Boolean | Read/Write |
| | S0000-S16368 | Short, Word, BCD | |
| | S0000-S16352 | Long, DWord, LBCD | |
| Timer Contacts* | TS0000-TS2047 | Boolean | Read/Write |
| | TS0000-TS2032 | Short, Word, BCD | |
| | TS0000-TS2016 | Long, DWord, LBCD | |
| Timer Coils* | TC0000-TC2047 | Boolean | Read/Write |
| | TC0000-TC2032 | Short, Word, BCD | |

| Device Type | Range | Data Type | Access |
|-----------------------------|---------------|-------------------|------------|
| | TC0000-TC2016 | Long, DWord, LBCD | |
| Integrating Timer Contacts* | SS0000-SS2047 | Boolean | Read/Write |
| | SS0000-SS2032 | Short, Word, BCD | |
| | SS0000-SS2016 | Long, DWord, LBCD | |
| Integrating Timer Coils* | SC0000-SC2047 | Boolean | Read/Write |
| | SC0000-SC2032 | Short, Word, BCD | |
| | SC0000-SC2016 | Long, DWord, LBCD | |
| Counter Contacts* | CS0000-CS1023 | Boolean | Read/Write |
| | CS0000-CS1008 | Short, Word, BCD | |
| | CS0000-CS0992 | Long, DWord, LBCD | |
| Counter Coils* | CC0000-CC1023 | Boolean | Read/Write |
| | CC0000-CC1008 | Short, Word, BCD | |
| | CC0000-CC0992 | Long, DWord, LBCD | |

* Users can specify a Long data type by appending a space and an "L" to the address. For example, "D00000" would be entered as "D00000 L". This does not apply to arrays or bit accessed registers.

● **Note:** All Boolean device types can be accessed as Short, Word, BCD, Long, DWord, and LBCD. However, the device must be addressed on a 16-bit boundary.

| Device Type | Range | Data Type | Access |
|--------------------------|--|--|------------|
| Timer Value | TN0000-TN2047 | Short , Word, BCD | Read/Write |
| Integrating Timer Value | SN0000-SN2047 | Short , Word, BCD | Read/Write |
| Counter Value | CN0000-CN1023 | Short, Word , BCD | Read/Write |
| Data Registers*** | D00000-D12287 | Short , Word, BCD | Read/Write |
| | D00000-D12286 | Long, DWord, Float, LBCD | |
| Data Register Bit Access | D00000.00 D12287.15* D00000.00 D12286.31* | Short , Word, BCD, Boolean** Long, DWord, LBCD | Read/Write |
| Data Registers | DSH00000.002-DSH12286.002 | String | Read/Write |

| | | | |
|--|--|---|------------|
| String Access HiLo Byte Ordering | DSH00000.128-DSH12223.128 The string length may also be specified using a colon. The string length must be between 2-128 bytes and even. | | |
| Data Registers String Access LoHi Byte Ordering | DSL00000.002-DSL12286.002 DSL00000.128-DSL12223.128 The string length may also be specified using a colon. The string length must be between 2-128 bytes and even. | String | Read/Write |
| Special Data Registers*** | SD0000-SD2047 SD0000-SD2046 | Short , Word, BCD Long, DWord, Float, LBCD | Read/Write |
| Data Register Bit Access | SD0000.00-SD2047.15* SD0000.00-SD2046.31* | Short , Word, BCD, Boolean** Long, DWord, LBCD | Read/Write |
| Link Registers*** | W0000-W3FFF (Hex) W0000-W3FFE (Hex) | Short , Word, BCD Long, DWord, Float, LBCD | Read/Write |
| Link Register Bit Access | W0000.00-W3FFF.15* W0000.00-W3FFE.31* | Short , Word, BCD, Boolean** Long, DWord, LBCD | Read/Write |
| Link Registers String Access HiLo Byte Ordering | WSH0000.002-WSH3FFE.002 WSH0000.128-WSH3FBF.128 The string length may also be specified using a colon. The string length must be between 2-128 bytes and even. | String | Read/Write |
| Link Registers String Access LoHi Byte Ordering | WSL0000.002-WSL3FFE.002 WSL0000.128-WSL3FBF.128 The string length may also be specified using a colon. The string length must be between 2-128 bytes and even. | String | Read/Write |
| Special Link Registers*** | SW0000-SW07FF (Hex) SW0000-SW07FE (Hex) | Short , Word, BCD Long, DWord, Float, LBCD | Read/Write |
| Link Register Bit Access | SW0000.00-SW07FF.15* | Short , Word, BCD, | Read/Write |

| | | | |
|--|--|---|------------|
| | SW0000.00-SW07FE.31* | Boolean** Long, DWord, LBCD | |
| File Register*** | R00000-R32767 R00000-R32766 | Short, Word, BCD Long, DWord, Float, LBCD | Read/Write |
| File Register Bit Access | R00000.00-R32767.15* R00000.00-R32766.31* | Short, Word, BCD, Boolean** Long, DWord, LBCD | Read/Write |
| File Registers String Access HiLo Byte Ordering | RSH00000.002-RSH32766.002 RSH00000.128-RSH32703.128 The string length may also be specified using a colon. The string length must be between 2-128 bytes and even. | String | Read/Write |
| File Registers String Access LoHi Byte Ordering | RSL00000.002-RSL32766.002 RSL00000.128-RSL32703.128 The string length may also be specified using a colon. The string length must be between 2-128 bytes and even. | String | Read/Write |
| Index Registers*** | Z00-Z15 Z00-Z14 | Short, Word, BCD Long, DWord, Float, LBCD | Read/Write |
| Index Register Bit Access | Z00.00-Z15.15* Z00.00-Z14.31* | Short, Word, BCD, Boolean** Long, DWord, LBCD | Read/Write |

* For register memory, the data types Short, Word, BCD, DWord, Long, Boolean and LBCD can append an optional ".bb" (dot bit) or ":bb" (colon bit) to the address to reference a bit in a particular value. The valid ranges for the optional bit is 0-15 for Short, Word, Boolean and BCD; and 0-31 for Long, DWord and LBCD. Strings use the bit number to specify length. The valid length of a string in D memory is 2 to 128 bytes. The string length must also be even numbers. Float types do not support bit operations. The bit number is always in decimal notation.

** When accessing register memory as Boolean, a bit number is required.

*** Users can specify a Long data type by appending a space and an "L" to the address. For example, "D00000" would be entered as "D00000 L". This does not apply to arrays or bit accessed registers.

Array Access

All device types can be accessed in arrays of Short, Word, BCD, Long, DWord, Float or LBCD format. The default data type is Word. The size of the array depends on the data type and device type. All Register device types can access up to 64 elements for Short, Word and BCD; and 32 elements for Long, DWord, Float, and LBCD. All Bit memory types can be accessed with up to 32 elements for Short, Word, and BCD; and 16 elements for Long, DWord, Float and LBCD. Arrays can either 1 dimension or 2. Regardless of the dimensions, the array size must not exceed the limits already stated. Appending array notation onto a normal device reference enters arrays.

Array Examples

1. D100 [4] Single dimension includes the following register addresses: D100, D101, D102, D103.
2. M016 [3][4] Two Dimensions includes the following device addresses as words: M016, M032, M048, M064, M080, M096, M112, M128, M144, M160, M176, M192 3 rows x 4 columns = 12 words 12 x 16 (word) = 192 total bits.

Additional Device Examples

1. Access X device memory as Word : X??? where the ??? is a hex number on 16-bit boundaries such as 010, 020, 030, and so forth.
2. Access M device memory as Long : M???? where the ???? is a decimal number on 16-bit boundaries such as 0, 16, 32, 48, and so forth.

Event Log Messages

The following information concerns messages posted to the Event Log pane in the main user interface. Consult the OPC server help on filtering and sorting the Event Log detail view. Server help contains many common messages, so should also be searched. Generally, the type of message (informational, warning) and troubleshooting information is provided whenever possible.

Device reported an invalid address or error. Block deactivated. | Block address range = '<address>' to '<address>'.

Error Type:

Error

Possible Cause:

An attempt has been made to reference a non-existent location in the specified device.

Possible Solution:

Verify the tags assigned to addresses in the specified range on the device and eliminate any that reference invalid locations.

Received block length does not match expected length. | Received block length = <number> (bytes), Expected block length = <number> (bytes).

Error Type:

Warning

Possible Cause:

The data type maximum length or the length dictated in the address definition set a range that the results do not fit.

Possible Solution:

Verify the data type is correct and check the address definition for a length definition and correct or update.

Error code received from device. | Error code = <code>h.

Error Type:

Warning

Possible Cause:

Communication with the device succeeded, but the device reported a problem.

Possible Solution:

Consult the device documentation for information about the error code provided.

Possible network number problem with device.

Error Type:

Warning

Possible Cause:

The bridge device could not route a message to the destination network specified.

Possible Solution:

Verify the network number setting and hardware configuration. Use zero (0) to indicate a local connection.

See Also:

1. PLC Setup
2. Bridging

Possible PC number problem with device.

Error Type:

Warning

Possible Cause:

The bridge device could not route a message to the destination PC specified.

Possible Solution:

Verify the PC number setting and hardware configuration. Use 255 to indicate a local connection.

See Also:

1. PLC Setup
2. Bridging

Error Mask Definitions

B = Hardware break detected

F = Framing error

E = I/O error

O = Character buffer overrun

R = RX buffer overrun

P = Received byte parity error

T = TX buffer full

Appendix: PLC Setup

Provided for reference only. Please refer to the manufacturer documentation.

For more information on a specific PLC, click a link from the list below.

[A1SJ71C24-R2](#)

[A1SJ71C24-R4](#)

[AJ71C24-S8](#)

[QJ71C24N](#)

● **Note:** The AJ71C24 and QJ71C24N family of communications modules occupy ranges of X and Y memory. Writing to this memory with the Mitsubishi Serial Driver may disable the module causing a loss of communications. For more information, refer to the communications module manual.

A1SJ71C24-R2 Settings and Diagrams

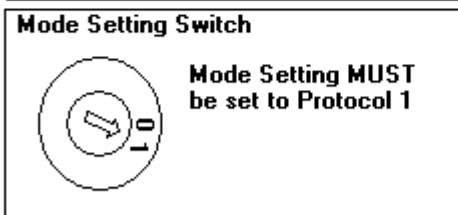
Provided for reference only. Please refer to the manufacturer documentation.

Switch Settings for the A1SJ71C24-R2

The switch positions shown match the driver defaults.

| SW | ON ← | SW Item | ON | OFF |
|----|--------------------------|-----------------------|-----------------|---------|
| 03 | <input type="checkbox"/> | 03 Unused | -- | -- |
| 04 | <input type="checkbox"/> | 04 Write During Run | Enable | Disable |
| 05 | <input type="checkbox"/> | 05 Transmission | See Table Below | |
| 06 | <input type="checkbox"/> | 06 Speed Setting | | |
| 07 | <input type="checkbox"/> | 07 | | |
| 08 | <input type="checkbox"/> | 08 Data bit setting | 8 | 7 |
| 09 | <input type="checkbox"/> | 09 Parity bit setting | Set | Not Set |
| 10 | <input type="checkbox"/> | 10 Parity Even/Odd | Even | Odd |
| 11 | <input type="checkbox"/> | 11 Stop bit setting | 2 Bits | 1 Bit |
| 12 | <input type="checkbox"/> | 12 Sum Check setting | Set | Not Set |

| Baud Rate | 300 | 600 | 1200 | 2400 | 4800 | 9600 | 19200 |
|-----------|-----|-----|------|------|------|------|-------|
| SW05 | OFF | ON | OFF | ON | OFF | ON | OFF |
| SW06 | OFF | OFF | ON | ON | OFF | OFF | ON |
| SW07 | OFF | OFF | OFF | OFF | ON | ON | ON |



For models of the AJ71C24 which support multidrop operation, the Station Number must be set between 0 and 31. The A1SJ71C24-R2 uses a set station number of 0.

A1S71C24-R4 Settings and Diagrams

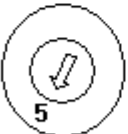
Provided for reference only. Please refer to the manufacturer documentation.

Switch Settings for the AJ71C24-R4
 The switch positions shown match the driver defaults.

| SW | ON ← | SW Item | ON | OFF |
|----|--------------------------|------------------------|-----------------|---------|
| 01 | <input type="checkbox"/> | SW1 Unused | ---- | ---- |
| 02 | <input type="checkbox"/> | SW2 ComputerLink/Multi | CompL | Multi |
| 03 | <input type="checkbox"/> | SW3 Unused | ---- | ---- |
| 04 | <input type="checkbox"/> | SW4 Write During Run | Enable | Disable |
| 05 | <input type="checkbox"/> | SW5 Transmission | See Table Below | |
| 06 | <input type="checkbox"/> | SW6 Speed Setting | | |
| 07 | <input type="checkbox"/> | SW7 | | |
| 08 | <input type="checkbox"/> | SW8 Data bit setting | 8 | 7 |
| 09 | <input type="checkbox"/> | SW9 Parity bit setting | Set | Not Set |
| 10 | <input type="checkbox"/> | SW10 Parity Even/Odd | Even | Odd |
| 11 | <input type="checkbox"/> | SW11 Stop bit setting | 2 Bits | 1 Bit |
| 12 | <input type="checkbox"/> | SW12 Sum Check setting | Set | Not Set |

| Baud Rate | 300 | 600 | 1200 | 2400 | 4800 | 9600 | 19200 |
|-----------|-----|-----|------|------|------|------|-------|
| SW05 | OFF | ON | OFF | ON | OFF | ON | OFF |
| SW06 | OFF | OFF | ON | ON | OFF | OFF | ON |
| SW07 | OFF | OFF | OFF | OFF | ON | ON | ON |

Mode Setting Switch



The AJ71C24R4 **MUST** be set for Protocol 1. This means the Mode switch must be set to position 5.

For models of the AJ71C24 which support multidrop operation, the Station Number must be set between 0 and 31.

The Dip switch positions shown here are for Computer Link operation with Writes enabled, Protocol 1, 19200, 7,E,2, and check sum enabled.

AJ71C24-S8 Setting and Diagrams

Provided for reference only. Please refer to the manufacturer documentation.


Switch Settings for the AJ71C24-S8

The switch positions shown match the driver defaults.

| SW | ON ← | SW Item | ON | OFF |
|----|-------------------------------------|-------------------------------------|-----------------|---------|
| 11 | <input checked="" type="checkbox"/> | 11 Main Channel | RS-422 | RS-232 |
| 12 | <input checked="" type="checkbox"/> | 12 Data bit setting | 8 | 7 |
| 13 | <input checked="" type="checkbox"/> | 13 Transmission 14 Speed Setting | See Table Below | |
| 14 | <input checked="" type="checkbox"/> | | | |
| 15 | <input checked="" type="checkbox"/> | | | |
| 16 | <input checked="" type="checkbox"/> | 16 Parity bit setting | Set | Not Set |
| 17 | <input checked="" type="checkbox"/> | 17 Parity Even/Odd | Even | Odd |
| 18 | <input checked="" type="checkbox"/> | 18 Stop bit setting | 2 Bits | 1 Bit |
| | ON ← | 21 Sum Check setting | Set | Not Set |
| 21 | <input checked="" type="checkbox"/> | 22 Write During Run | Enable | Disable |
| 22 | <input checked="" type="checkbox"/> | 23 Send Terminator | Present | Absent |
| 23 | <input checked="" type="checkbox"/> | 24 Receive Terminator | Present | Absent |
| 24 | <input checked="" type="checkbox"/> | | | |

| Baud Rate | 300 | 600 | 1200 | 2400 | 4800 | 9600 | 19200 |
|-----------|-----|-----|------|------|------|------|-------|
| SW13 | OFF | ON | OFF | ON | OFF | ON | OFF |
| SW14 | OFF | OFF | ON | ON | OFF | OFF | ON |
| SW15 | OFF | OFF | OFF | OFF | ON | ON | ON |

Mode Setting Switch



The AJ71C24 **MUST** be set for Protocol 1. This means the mode switch can be set to Mode 1, 5, and A.

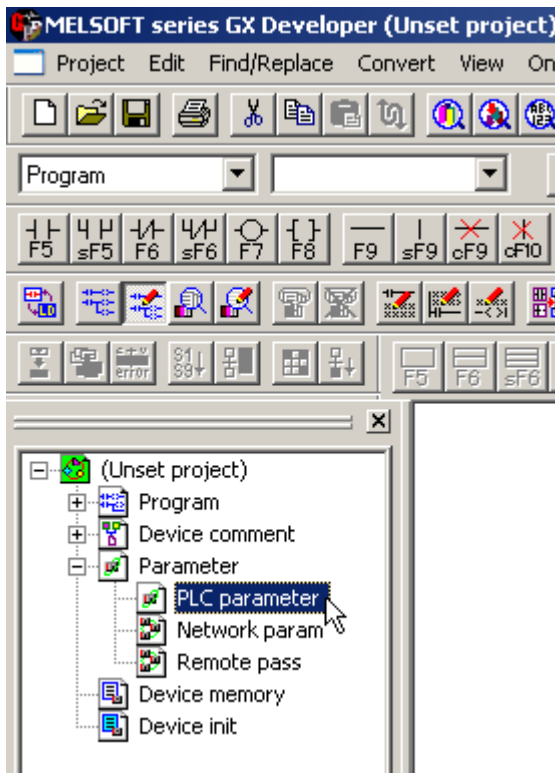
For models of the AJ71C24 which support multidrop operation, the Station Number must be set between 0 and 31.

QJ1C24N Configuration

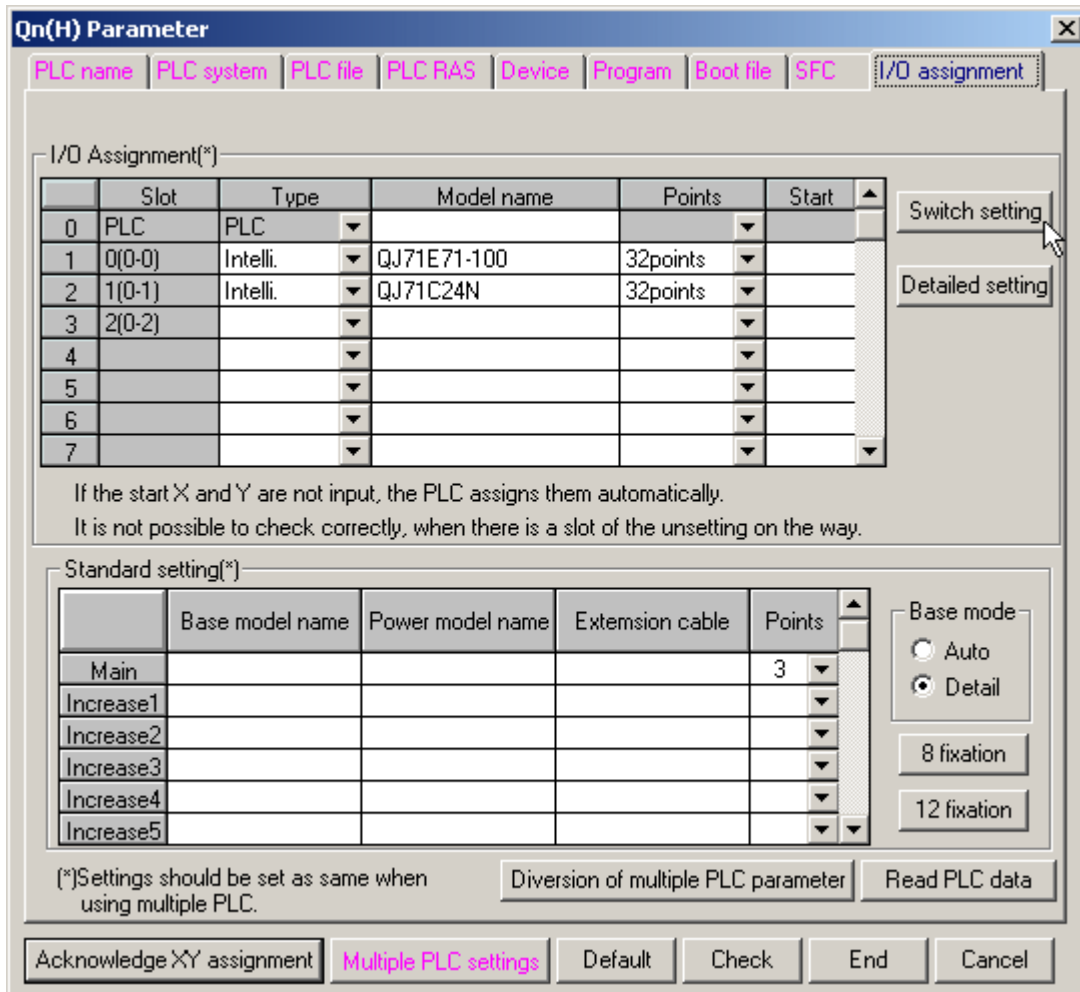
Provided for reference only. Please refer to the manufacturer documentation.

The QJ1C24N communications module is configured with the GX Developer programming software, which is available from Mitsubishi PLC dealers.

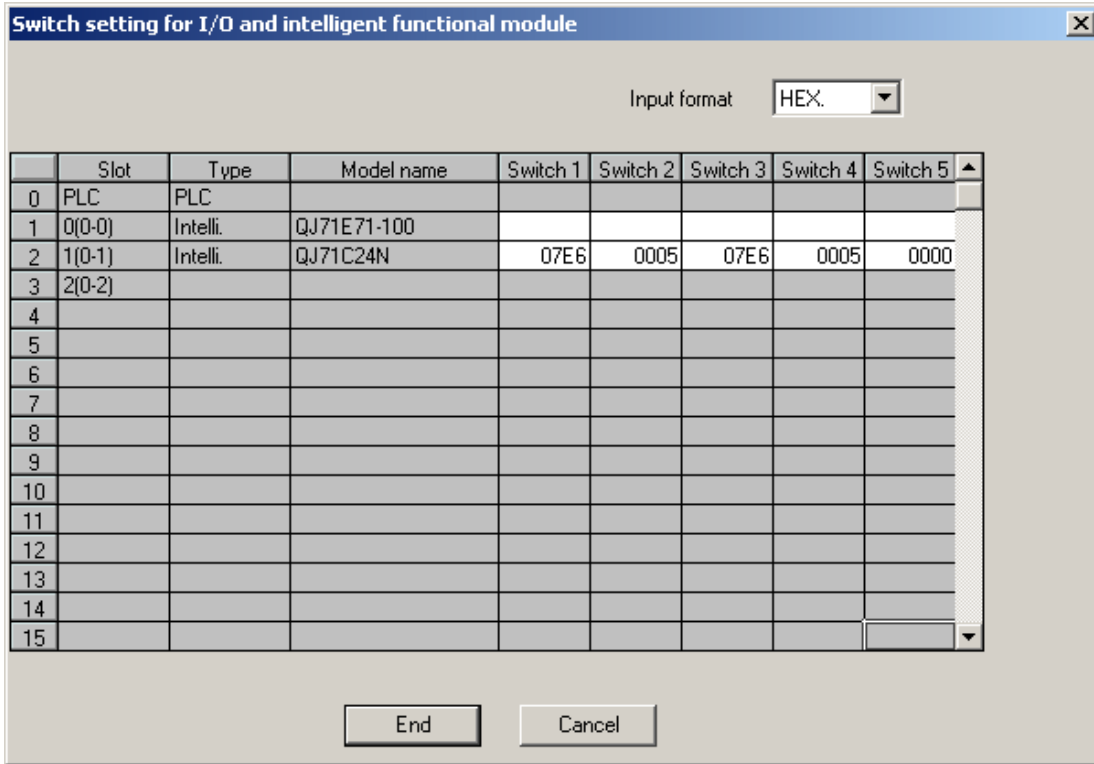
1. To begin configuring the PLC to work with this driver, open an existing **GX Developer** project. Alternatively, read the current **PLC Parameter** configuration from the device.
2. Next, edit the **PLC Parameter** settings.



3. Open the **I/O Assignment** tab and then click **Switch Setting** to configure the QJ1C24N module. In the example shown below, the PLC has a QJ1E71-100 Ethernet communications module in slot 0 and a QJ1C24N serial communications module in slot 1.



● **Note:** The recommended software switch settings are shown below.



4. Entries for switch settings can be calculated from the following tables. Switches 1 and 2 are for channel 1 and switches 3 and 4 are for channel 2. For more information, refer to the tables and examples below.

Switch 1/ Switch 3 Transmission Setting

| Bit | Description | OFF (0) | ON (1) | Remark |
|-----|-----------------------|-------------|---------|-------------------------------------|
| 0 | Operational | Independent | Link | Must be OFF on channel 1 |
| 1 | Data bits | 7 | 8 | |
| 2 | Parity bit | No | Yes | |
| 3 | Even/Odd Parity | Odd | Even | |
| 4 | Stop bit | 1 | 2 | |
| 5 | Check Sum | No | Yes | Must be ON for use with this driver |
| 6 | Write during RUN | Prohibited | Allowed | ON to write data externally |
| 7 | Setting modifications | Prohibited | Allowed | |

Note: To use this driver with its default communications settings (8 data bits, odd parity, 1 stop bit, check sum) and to allow writes during RUN and setting modification, set bits 0 through 7 to 01100111 (0xE6).

Switch 1/ Switch 3 Communications Setting

| Baud | Bits 8 to 15 |
|------|--------------|
| 300 | 0x00 |
| 600 | 0x01 |
| 1200 | 0x02 |
| 0x03 | 2400 |
| 0x04 | 4800 |

| Baud | Bits 8 to 15 |
|--------|--------------|
| 9600 | 0x05 |
| 14400 | 0x06 |
| 19200 | 0x07 |
| 28800 | 0x08 |
| 38400 | 0x09 |
| 57600 | 0x0A |
| 115200 | 0x0B |

● **Note:** To use this driver with its default baud rate of 19200, set bits 8 through 15 to 0x07.

Switch 2/ Switch 4 Protocol Setting

| Setting | Description |
|-------------|----------------------|
| 0x00 | GX Developer |
| 0x01 | Mode 1* |
| 0x02 | Mode 2 |
| 0x03 | Mode 3 |
| 0x04 | Mode 4 |
| 0x05 | Mode 5** |
| 0x06 | Non-procedure |
| 0x07 | Bidirectional |
| 0x08 | For linked operation |
| 0x09 - 0x0D | Prohibited |
| 0x0E | ROM/RAM/switch test |
| 0x0F | Loop back test |

* Must be used with A Series driver model.

** Must be used with Q Series driver model.

● **Note:** To use this driver with the Q Series model selected, set switch 2 (or 4) to 0x05. This driver can also be used with the A Series model selected if switch 2 (or 4) is set to 0x01.

Switch 5 Channel Setting

Set communication objects as channels between 0 and 31 when multi-drop connection is selected. Set to 0 if 1:1 connection is selected.

● **Note:** Set switch 5 to 0x0000 to assign the station number 0 to the PLC.

5. Once the switch settings are entered, write the PLC Parameters back to the device.

6. Then, cycle the power on the PLC to make the new settings active.

Index

A

- A1S71C24-R2 Settings and Diagrams 30
- A1S71C24-R4 Settings and Diagrams 31
- Address Descriptions 17
- A71C24-S8 Setting and Diagrams 32
- A71C24 and Q71C24N communications cards 3
- Attempts Before Timeout 12
- Auto-Demotion 12

B

- Baud Rate 4
- BCD 16
- Boolean 16
- Bridging 13

C

- Channel Assignment 10
- Communication Protocol 4
- Communications Timeouts 11-12
- Connect Timeout 11

D

- Data Bits 4
- Data Collection 10
- Data Types Description 16
- Demote on Failure 12
- Demotion Period 12
- Device reported an invalid address or error. Block deactivated. | Block address range = '<address>' to '<address>'. 27
- Discard Requests when Demoted 13
- Do Not Scan, Demand Poll Only 11
- Driver 10

DWord 16

E

Error code received from device. | Error code = <code>h. 27

Error Mask Definitions 28

Ethernet Encapsulation 4

Event Log Messages 27

F

First Word Low 13

Float 16

Framing 28

H

Hardware 28

I

I/O error 28

ID 10

ID Format 10

Identification 9

Initial Updates from Cache 11

Inter-Request Delay 12

L

LBCD 16

Long 16

M

Mitsubishi A Series Address Descriptions 17

Mitsubishi Q Series Address Descriptions 21

Mitsubishi Serial devices 3

Model 10

Multi-Level Networks 15

N

Network Number 13

O

Overflow 28

Overview 3

P

Parity 4, 28

PC Number 13

PLC Setup 29

Possible network number problem with device. 27

Possible PC number problem with device. 28

Q

QJ71C24N 33

R

Received block length does not match expected length. | Received block length = <number> (bytes),
Expected block length = <number> (bytes). 27

Redundancy 14

Request Timeout 12

Respect Tag-Specified Scan Rate 11

RX buffer overrun 28

S

Scan Mode 11

Setup 4

Short 16

Simulated 10
Station number 10
Stop Bits 4
String 16
Supported Devices 4

T

Timeouts to Demote 12
TXbuffer full 28

W

Word 16