AVEVA™ Communication Drivers Pack – Mitsubishi – MELSEC Driver

User Guide

Publication date: Tuesday, May 9, 2023

Publication ID: 868864

Contact Information

AVEVA Group Limited
High Cross
Madingley Road
Cambridge
CB3 0HB. UK

https://sw.aveva.com/

For information on how to contact sales and customer training, see https://sw.aveva.com/contact.

For information on how to contact technical support, see https://sw.aveva.com/support.

To access the AVEVA Knowledge and Support center, visit https://softwaresupport.aveva.com.

Contents

Chapter 1

Getting Started with MELSEC Communication Driver

About the MELSEC Communication Driver

MELSEC Communication Driver is the driver for Ethernet and Serial communication with Mitsubishi Q, QnA, L, and iQ-R Series devices using MELSEC Protocol.

Supported Hardware and Software

The MELSEC Communication Driver enables Ethernet and serial communication with Q, QnA, L, and the iQ-R-series PLCs from Mitsubishi Electric, using the MELSEC protocol.

To program these PLCs, you need to use the GX Developer, GX Works2, or GX Works3 programming software from Mitsubishi Electric.

Device Configuration

The support PLCs communicate over both UDP and TCP ports.

The Communication Method for:

- Q-series PLCs is the MC protocol (MELSEC Communication Protocol)
- iQ-R series PLCs is the SLMP protocol

Notes: It is necessary to configure the Destination IP Address when communicating over UDP/IP.

Conformance

The following hardware and software was used for conformance testing of this Communication Driver.

Configuration:Q Series PLC (Ethernet)

- Device: Q00JCPU with QJ71E71-100 module
- Cable: Ethernet
- UDP Port: 5000

- TCP Port: 5002

Configuration: Q Series PLC (Serial)

- Device: Q00JCPU with QJ71C24 module
- Cable: Null Modem Cable with Gender Adapter
- Serial Port: CH1 (RS232)
- Comm Settings: 9600, 8, 1, Odd

Configuration: iQ-R Series PLC (Ethernet)

- Device: R16SFCPU with Ethernet port embedded on the CPU
- Cable: Ethernet
- UDP Port: 3001
- TCP Port: 5002

Chapter 2

Configuring the MELSEC Communication Driver

Working with the MELSEC Communication Driver Configuration

Each server instance has its own hierarchy of objects, and each object has parameters that you need to configure in order to establish communication between the Communication Driver and individual devices on the network. You can view server configuration hierarchy of a MELSEC Communication Driver instance under its **Configuration** node.

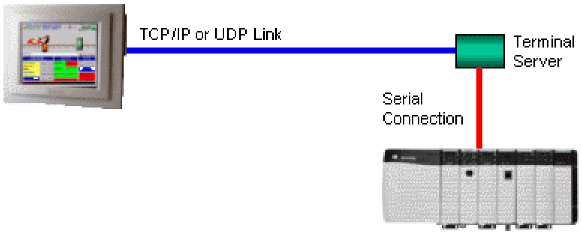
This section only describes how to configure object parameters for a MELSEC server instance. For more general information about adding and configuring objects, see "Configuring Your Communication Driver" in the Communication Drivers Pack Help.

Direct Serial Communication and Serial Encapsulation

This Communication Driver supports direct serial communication with the target device, as well as serial encapsulation over a TCP/IP or UDP/IP network link.

Direct serial communication requires that the target device be directly connected to the computer that is running this Communication Driver. If you plan to use direct serial communication, you need to know the serial communication settings (e.g., baud rate, parity, etc.) that have been configured on the device, because you must configure this Communication Driver to match those settings.

Serial encapsulation enables serial communication with a target device that is connected to a terminal server on your TCP/IP or UDP/IP network. The terminal server is like a virtual serial port. It converts TCP or UDP messages to serial data and then relays them to the connected device. If you plan to use serial encapsulation, the target device should already be connected to the terminal server, and you need to know the IP address of the terminal server and the port number assigned to the device.



Configuring the Communication Settings of a Channel

Configure the communication settings for a selected channel to ensure uninterrupted communication with the device network.

Communication Type

The specific transport protocol to be used. Select one of the following options from the list:

Option	**Description**
UDP	UDP/IP Ethernet communication.
TCP	TCP/IP Ethernet communication.
Serial	Direct serial communication.

Setting the Station ID of a Device

Set the station ID for a selected device so that the Communication Driver can identify and communicate with it on the network.

Syntax

There are 2 possible syntax options of valid station IDs:

- For UDP/IP or TCP/IP Ethernet communication, the station ID for a target device must use the following syntax:

  ```
     <IP address>:<port number>
  ```

  where,

  IP address:The IP address of the device on the Ethernet network

  port number: The port number of the MELSEC protocol

  - For a device part of the iQ-R series, the syntax is:

    ```
          iqr:<IP address>:<port number>
    ```

    where,

iqr: Identifies that the device is part of the Mitsubishi iQ-R series PLCs

- For direct serial communication, the station ID for a target device must use the following syntax:

  ```
  <station number>
  ```

  where,

  station number: The station number of the PLC

Examples

Examples of valid station IDs:

- For Ethernet: 10.168.23.73:5001
  - For an iQ-R series device: iqr:10.168.25.74:5002
- Using the station number: 2

Adding and Configuring Channel Selector Object

The server-specific configuration portion of the MELSEC Communication Driver hierarchy tree under the OI Server Manager starts at the Channel Selector object. This object lets you set server parameters for communication with agents (devices) in the hierarchy tree.

Adding Channel Selector Objects

To add a Channel Selector object to your MELSEC hierarchy

- In the console tree, right-click **Configuration** and then click **Add Channel Selector Connection**. The "New_ChannelSelector_000" object appears in the hierarchy.

Edit the object name to appropriately describe components of your specific hardware environment. If you do not rename the object at this time, a numeric sequencing system is applied. You can rename the hierarchy entry later.

Configuring Channel Selector Objects

All of the serial communication settings are included in the channel parameters of the Communication Driver.

To configure the serial communication settings for a channel:

1. In the Operations Integration Server Manager, navigate to the ChannelSelector object that you want to configure:
   a. Expand the Operations Integration Server Manager, expand the node group, expand **Local**, and then expand Operations Integration Supervisory Servers.
   b. Locate and expand **Mitsubishi Electric - MELSEC**, and then expand its **Configuration** node.
   c. Select the ChannelSelector object that you want to configure. The channel parameters are displayed in the details pane on the right.
2. In the **Serial Encapsulation** box, select the encapsulation mode:
   a. If you select **None**, configure the direct serial communication settings:
      i. In the **COM** box, select the COM port to which the target device is connected.

   ii. In the **Baud Rate**, **Data Bits**, **Stop Bits**, and **Parity** boxes, configure the serial communication settings to match the settings that have already been configured on the target device.

b. If you select either **TCP/IP** or **UDP/IP**, configure the serial encapsulation settings:

   i. In the **IP Address** box, type the IP address of the terminal server.

   ii. In the **Port Number** box, type the port number on the terminal server that has been assigned to the target device.

   iii. If you want to make the target device responsible for establishing communication with this Communication Driver, select **Server Mode**.

   Note: This option is not available for UDP/IP.

3. In the **Block Size (Words)** field, enter the maximum number of bytes that will be requested from the device in a single read operation.

   Note: You can type any value from 1 to 15. The default value is 7.

4. In the **Write word bits (0=No/1=Yes)** field**:**

   - Enter 1, to allow writing to the individual bits of a Word.
   - Enter 0, to not allow writing to individual bits of a Word.

5. In the **Signed or Unsigned Value** dropdown list, select the type of the register value.

6. Use **Advanced** option to modify additional communication settings. For more information see Advanced.

Advanced

Click this button to open the **Advanced Settings** dialog box, which provides access to additional communication settings such as timeouts, retries, and buffer sizes. You might need to change these settings if the Communication Driver behaves unexpectedly during run time, but the default settings should work for most network configurations. For more information about these settings, see "Advanced Settings" in the Communication Drivers Pack Help

Adding and Configuring Device Selector Objects

The MELSEC Communication Driver can connect to PLCs. These connections are modeled in the hierarchy by means of Device Selector objects, each of which models the end-point of the communications path.

From the **ChannelSelector** branch of the Communication Driver hierarchy, create the new **DeviceSelector** object.

To add a Device Selector connection to your MELSEC hierarchy

1. In the console tree, right-click the **ChannelSelector** object, and then click **Add DeviceSelector Connection**. The **New_DeviceSelector_000** object and associated **Parameters** configuration view appear.

2. Rename the object as needed to reflect the connection.

3. Configure the **Station** field.

   Note: This field is mandatory, if you select **TCP** or **UDP** option under the **Serial/Ethernet** drop down list in the **Channel Selector** screen. You should leave this field blank if you select **Serial (NITP)** or **Serial (TBP)** option under the **Serial/Ethernet** drop down list in the **Channel Selector** screen.

For **TCP** or **UDP** options, use the following syntax:
`<IP Address>:<Port Number>`

Parameter	Default Value	Description
IP Address	none	The IP network address of the target device. **Required**.
Port Number	none	The port number of the Mitsubishi controller. It can be either a TCP or UDP port.

Example: 198.2.65.1:5002

For more information see Setting the Station ID of a Device.

4. Configure **Device Groups** and **Device Items**.

Device Group Definitions

Use the **Device Groups** configuration view, to create, add, delete, and define device groups. You can also configure default update intervals for the objects and edit update intervals in this dialog box. To open the Device Groups dialog box, in the Device Selector configuration editor, click the **Device Groups** tab.

Note: When you select another part of the Communication Driver tree hierarchy, you are prompted to save the modifications to the configuration set.

To create or add device groups

1. Right-click anywhere in the table, and then click **Add**. A device group is added with a default name and update interval.
2. Enter a unique name up to 32 characters long for the device group.

To delete device groups

1. Right-click the device group to be deleted, and then click **Delete**.
2. Read the warning, and then click **Yes**.

To edit device groups

Use the **Edit** option from the **Device Groups** tab only for configuring the Communication Driver's unsolicited message handling.

To configure default update intervals

To configure a default update interval for the object, right-click in the **Device Groups** box and then click **Config Default Update Interval**.

To edit update intervals

To edit the update interval for an object, double-click its value in the **Update Interval** column and make the edits.

or

Right-click its value in the **Update Interval** column and then click **Modify Update** Interval.

The update interval is the frequency, in milliseconds, that the MELSEC Communication Driver acquires data from the topics associated with that device group.

Different topics can be polled at different rates from a PLC by defining multiple device group names for the same PLC and setting a different update interval for each device group.

Device Item Definitions

The device item name is an is an “alias” or a label for the data in the device. It is an alternative name for the item reference, and ca be used instead of the item reference when you create the client application. Device item configuration is optional, but is strongly recommended.

To create or add device items

1. Right-click anywhere in the table, and then click **Add**.
2. In the **Name** column, type a unique item name. The maximum is 32 characters.
3. In the corresponding line, double-click the **Item Reference** column and enter the correlated item reference for the name you created.

To rename device items

Right-click the device item to be renamed and click **Rename**. Make the changes.

To delete device items

Right-click the device item to be deleted from the list and click **Delete**.

To clear all device items

Right-click in the **Device Items** box and click **Clear All**. All the device items listed are cleared after you confirm their deletion.

NOTE: You can import a .csv file containing your item definitions to help streamline configuration. See "Exporting and Importing CSV Files" in the Communication Drivers Pack.

Chapter 3

MELSEC Communication Driver Reference

Item Reference Syntax

Use item references to access data stored in memory registers in connected devices, as well as to access standard system items in the Communication Driver itself.

This section only describes the item reference syntax and options for the MELSEC server. For more general information about item references, see "Managing Device Items" and "Item Reference Descriptions" in the Communication Drivers Pack Help.

Item references in this Communication Driver use the following syntax.

- For Discrete (1-bit) memory registers, use the following syntax:
  ```
     <register type><address>
  ```
  where,

  register type: The type of memory register on the connected device, such as Input Relay (X), Output Relay (Y), File Register (R), and so on.

  address: The address of the memory register on the connected device. This value can be either decimal or hexadecimal, depending on the register type.

  For example: L5, B40

- For Word (2-byte) memory registers, use the following syntax:
  ```
     <register type><address>@[data type]
  ```
  where,

  register type: The type of memory register on the connected device, such as Input Relay (X), Output Relay (Y), File Register (R), and so on.

address: The address of the memory register on the connected device. This value can be either decimal or hexadecimal, depending on the register type.

data type: The formatted data type of the value read from the memory register. This is optional; if the data type is not specified, the value will be formatted as Word (i.e., unsigned,16-bit decimal) by default.

For example: TN5@F

- For bit-accessible memory registers and strings — i.e., register types DSH, WSH, RSH — use the following syntax:

```
   <register type><address>.<bit number or string length>
```

where,

register type: The type of memory register on the connected device, such as Input Relay (X), Output Relay (Y), File Register (R), and so on.

address: The address of the memory register on the connected device. This value can be either decimal or hexadecimal, depending on the register type.

bit number or string length: The number of the bit to read/write. Bit 00 is the low bit. Bit 15 is the high bit. For string registers like DSH, WSH and RSH, the length of the string is specified here.

For example: DSH1.2

Address Descriptions

The address descriptions consist of the register type, its item name and the allowable range of values, the default data type, allowable suffixes (if any), and allowable access methods.

Register Type	**Item Reference**		**Length**	**Data Types**	**Access**
	Item Name	**Range**			
Input	X	0000 to 3FFF (hex)	1 bit	-	Read/Write
Direct Input	DX	0000 to 3FFF (hex)	1 bit	-	Read/Write
Output	Y	0000 to 3FFF (hex)	1 bit	-	Read/Write
Direct Output	DY	0000 to 3FFF (hex)	1 bit	-	Read/Write
Link Relay	B	0000 to 3FFF (hex)	1 bit	-	Read/Write
Special Link Relay	SB	0000 to 07FF (hex)	1 bit	-	Read/Write
Internal Relay	M	0 to 16383	1 bit	-	Read/Write
Special Internal Relay	SM	0 to 2047	1 bit	-	Read/Write
Latch Relay	L	0 to 16383	1 bit	-	Read/Write
Step Relay	S	0 to 16383	1 bit	-	Read/Write

Edge Relay	V	0 to 2047	1 bit	-	Read/Write
Annunciator Relay	F	0 to 2047	1 bit	-	Read/Write
Timer Contacts	TS	0 to 2047	1 bit	-	Read/Write
Timer Coil	TC	0 to 2047	1 bit	-	Read/Write
Timer Value	TN	0 to 2047	2 bytes	Word, BCD	Read/Write
Integrating Timer Contact	SS	0 to 2047	1 bit	-	Read/Write
Integrating Timer Coil	SC	0 to 2047	1 bit	-	Read/Write
Integrating Timer Value	SN	0 to 2047	2 bytes	Word	Read/Write
Counter Contact	CS	0 to 1023	1 bit	-	Read/Write
Counter Coil	CC	0 to 1023	1 bit		Read/Write
Counter Value	CN	0 to 1023	2 bytes	Word, BCD	Read/Write
Data Register	D	0 to 12287	2 bytes	Word, BCD	Read/Write
		0 to 12286			Read/Write
Data Register, bit access	D	0.0 to 12287.15	2 bytes	DWord, LBCD, Float	Read/Write
Data Register, string access, HiLo byte ordering	DSH	0.2 to 12286.2	String	String	Read/Write
Special Data Register	SD	0.128 to 12223.128	2 bytes	Word, BCD	Read/Write
		0 to 2047		DWord, LBCD, Float	Read/Write
		0 to 2046			Read/Write
Link Register	W	0000 to 3FFF (hex)	2 bytes	Word, BCD	Read/Write
		0000 to 3FFE (hex)		DWord, LBCD, Float	Read/Write
Link Register, bit access	W	0000.00 to 3FFF.15 (hex)	2 bytes	-	Read/Write
Link Register,	WSH	0000.002 to 3FFE.002	String	String	Read/Write

string access, HiLo byte ordering		(hex)			
Special Link Register	SW	0000.128 to 3FBF.128 (hex)	2 bytes	Word, BCD	Read/Write
		0000 to 07FF (hex)		DWord, LBCD, Float	Read/Write
Index Register	Z	0000 to 15	2 bytes	Word, BCD	Read/Write
		0000 to 14	2 bytes	DWord, LBCD, Float	Read/Write
Index Register, bit access	Z	00.00 to 15.15	4 bytes	-	Read/Write
File Register	R	00.00 to 14.31	2 bytes	Word, BCD	Read/Write
		0000 to 32767		DWord, LBCD, Float	Read/Write
		0000 to 32766			Read/Write
File Register, bit access	R	0000.00 to 32767.15	2 bytes		Read/Write
		0000.002 to 32766.002			Read/Write
File Register, strings access, HiLo byte ordering	RSH	0000.128 to 32703.128	String	String	Read/Write

Available Registers in iQ-R Series Devices

The following table lists all the registers, their item references, allowable suffixes (if any) that are available for devices part of the iQ-R series.

Register Type	**Item Name**	**Suffix**	**Access**
Input	X	-	Read/Write
Direct Input	DX	-	Read/Write
Output	Y	-	Read/Write
Direct Output	DY	-	Read/Write

Link Relay	B	-	Read/Write
Special Link Relay	SB	-	Read/Write
Internal Relay	M	-	Read/Write
Special Internal Relay	SM	-	Read/Write
Latch Relay	L	-	Read/Write
Edge Relay	V	-	Read/Write
Annunciator Relay	F	-	Read/Write
Timer Contacts	TS	-	Read/Write
Timer Coil	TC	-	Read/Write
Timer Value	TN	Word, BCD	Read/Write
Counter Contact	CS	-	Read/Write
Counter Coil	CC		Read/Write
Counter Value	CN	Word, BCD	Read/Write
Data Register	D	Word, BCD	Read/Write
Data Register, bit access	D	DWord, LBCD, Float	Read/Write
Special Data Register	SD	Word, BCD, DWord, LBCD, Float	Read/Write
Link Register	W	Word, BCD,DWord, LBCD, Float	Read/Write
Link Register, bit access	W	-	Read/Write
Special Link Register	SW	Word, BCD, DWord, LBCD, Float	Read/Write
Index Register	Z	Word, BCD, DWord, LBCD, Float	Read/Write
Index Register, bit access	Z	-	Read/Write
File Register	R	Word, BCD, DWord, LBCD, Float	Read/Write
File Register, bit access	R		Read/Write

Long Counter Contact	LCS		Read/Write
Long Counter Coil	LCC		Read/Write
Long Counter Current Value	LCN	Word, BCD	Read/Write
Long Timer Contact	LTS		Read/Write
Long Timer Coil	LTC		Read/Write
Long Timer Current Value	LTN	Word, BCD	Read only
Retentive Timer Contact	STS		Read/Write
Retentive Timer Coil	STC		Read/Write
Retentive Timer Current Value	STN	Word	Read/Write
Long Retentive Timer Contact	LSTS		Read/Write
Long Retentive Timer Coil	LSTC		Read/Write
Long Retentive Timer Current Value	LSTN	Word	Read only
Long Index Register	LZ	Word, BCD, DWord, LBCD, Float	Read/Write
Refresh Data Register	RD		Read/Write
CPU buffer memory	G		Read/Write
Fixed cycle area of the CPU buffer memory	HG		Read/Write

Supported Data Types

The data type is specified as a suffix in the item syntax. This Communication Driver supports the following data types.

DataType /Suffix	**Description**	**Range of Values**

SHORT	Signed, 16-bit decimal value	-32768 to 32767
WORD	Unsigned,16-bit decimal value	0 to 65535
BCD	16-bit binary coded decimal (BCD) with byte swap	0 to 9999
LONG DWORD	Signed, 32-bit decimal value Unsigned, 32-bit decimal value	-2147483648 to 2147483647 0 to 4294967295
LBCD	32-bit binary coded decimal (BCD) with byte swap	0 to 99999999
FLOAT	32-bit floating point value	N/A

Examples of Item References

These are examples of valid item references for this Communication Driver. For more information about the referenced addresses, see the manufacturer's documentation for your device.

Register Type	**Address on the Device**	**Item Reference**
Data Register	D0000-Decimal	D0
Link Register	W0007-Hexadecimal	W7
File Register	R0010-Decimal	R10
Timer Value Timer Contact	TN0017-Decimal TS0000-Decimal	TN17 TS0
Timer Coil Counter Current Value	TC0007-Decimal CN1000-Decimal	TC7 CN1000
Counter Contact	CS0007-Decimal	CS7
Counter Coil Input Relay	CC1400-Decimal X0007-Hexadecimal	CC1400 X7
Output Relay Internal Relay	Y7772-Hexadecimal M0007-Decimal	Y7772 M7
Link Relay	B0000-Hexadecimal	B0

Latch Relay	L0010-Decimal	L10
Annunciator	F0007-Decimal	F7
Special Link Relay	SB0000-Hexadecimal	SB0
Edge Relay	V0009-Decimal	V9
Step Relay	S0000-Decimal	S0
Special Link Register	SW0007-Hexadecimal	SW7
Retentive Timer Contact	SS0000-Decimal	SS0
Retentive Timer Coil	SC0010-Decimal	SC10
Retentive Timer Current Value	SN0001-Decimal	SN1
Direct Input	DX0000-Hexadecimal	DX0
Direct Output	DY0005-Hexadecimal	DY5
Special Relay	SM0002-Decimal	SM2
Special Register	SD0001-Decimal	SD1
Index Register	Z0001-Decimal	Z1
File Register	ZR0001-Hexadecimal	ZR1

Chapter 4

Troubleshooting the MELSEC Communication Driver

MELSEC Communication Driver Error Codes

The following tables describe the additional error codes that you might receive when poll/poke requests and operations fail.

Code	Description	Possible Causes	Solution
1	Error Connect	IP address or port number is invalid	Check the IP Address and port number.
2	PLC Error or Invalid Memory Address	You may have tried to Read a memory from the PLC (Device) that is not present on that CPU	Check if the address (Device) configured does exist in the PLC
3	Check sum Error	The check sum byte received is not equal to the expected byte	Contact your technical support representative.
4	Protocol Error	The driver received an unexpected message from the device	Contact your technical support representative
0	OK	Communicating without error.	None required.
-15	Timeout waiting for message to start	• Disconnected cables. • PLC is turned off, in stop mode, or in error mode. • Wrong station number. • Wrong parity (for serial communication). • Wrong RTS/CTS configuration (for serial communication).	• Check cable wiring. • Check the PLC mode — it must be RUN. • Check the station number. • Increase the timeout in the driver's advanced settings. • Check the RTS/CTS configuration (for serial communication).

-17	Timeout between rx characters.	• PLC in stop or error mode • Wrong station number • Wrong parity • Wrong RTS/CTS configuration settings	• Check the cable wiring • Check the PLC stat e (it must be RUN) • Check the station number • Check the configuration. Review the Communication Parameters section for valid RTS/CTS configurations.
-37	INVALID HEADER	The header field specified in the driver worksheet is invalid	Check the supported header fields and use one of the supported headers
-39	INVALID BLOCK SIZE	Driver sheet is configured with range of addresses greater than the maximum block size. (This can be a single address specified outside of the supported range.)	Check the maximum block size and configure the driver sheet correctly. Specifically, check that the addresses are within the supported range.