INDUSTRIAL CONTROL COMMUNICATIONS, INC.

Mitsubishi MELSEC Client Driver Manual

October 30, 2014

TABLE OF CONTENTS

1 Mitsubishi MELSEC Client

1.1 Overview

The Mitsubishi MELSEC communication protocol (also known as MC protocol) is an application-level protocol implemented on top of the Ethernet TCP/IP and UDP/IP layers. It is typically used to access information on Mitsubishi PLCs and motion controllers which support the MELSEC server protocol using 3E and 1E frames. For applications using a Q PLC or FX PLC, please consult Mitsubishi for the separate applicable quick start guide.

The client driver also supports SLMP (Seamless Message Protocol) connections to communicate to CC-Link IE Field networks. SLMP is a unified protocol for achieving seamless communication between applications without awareness of network hierarchy or boundaries between the CC-Link family network and general-purpose Ethernet devices. SLMP is implemented on network hierarchies, such as TCP/IP, CC-Link IE and CC-Link, and makes communication possible among these devices. SLMP uses MELSEC 3E frames, and is therefore a fundamental subset of the MELSEC client driver. Both MELSEC and SLMP can be used simultaneously.

The driver uses a connection object element to target each server. The connection object defines a connection to a specific endpoint (IP address and port number) and specifies the frame type used for all underlying service objects. The service objects themselves issue read and write requests to the servers. A connection object can be thought of as a communication channel or “pipe” which is created between the driver and the server device, independent of the service objects that make use of that communication channel to transfer data requests.

Multiple connections can be established with a single physical server residing at a given IP address, provided the assigned port number in each connection object is unique. In this case, the connection objects will be accessing non-conflicting endpoints, and will therefore be independently managed within the driver.

Figure 1 provides a graphical representation of the driver concept, and demonstrates the associations among service objects, connection objects, physical servers and logical internal devices. This example system contains five service objects, four connection objects, and three MELSEC or SLMP servers. The “internal devices” in each of the servers can be any of the supported internal device types (data registers, internal relays, motion registers, etc.)

The blue connections in Figure 1 show a situation in which two service objects have been mapped to two internal device types on a single physical server. In this case, both service objects share a single connection object. The red connections in Figure 1 show a similar situation (two service objects mapped to two internal device types on a single physical server). In this case, however, two separate connection objects have been deployed for server #3 (each with a different destination port number). This allows each service object associated with server #3 to have its own dedicated connection, which can improve performance by allowing these service objects to be processed in parallel.

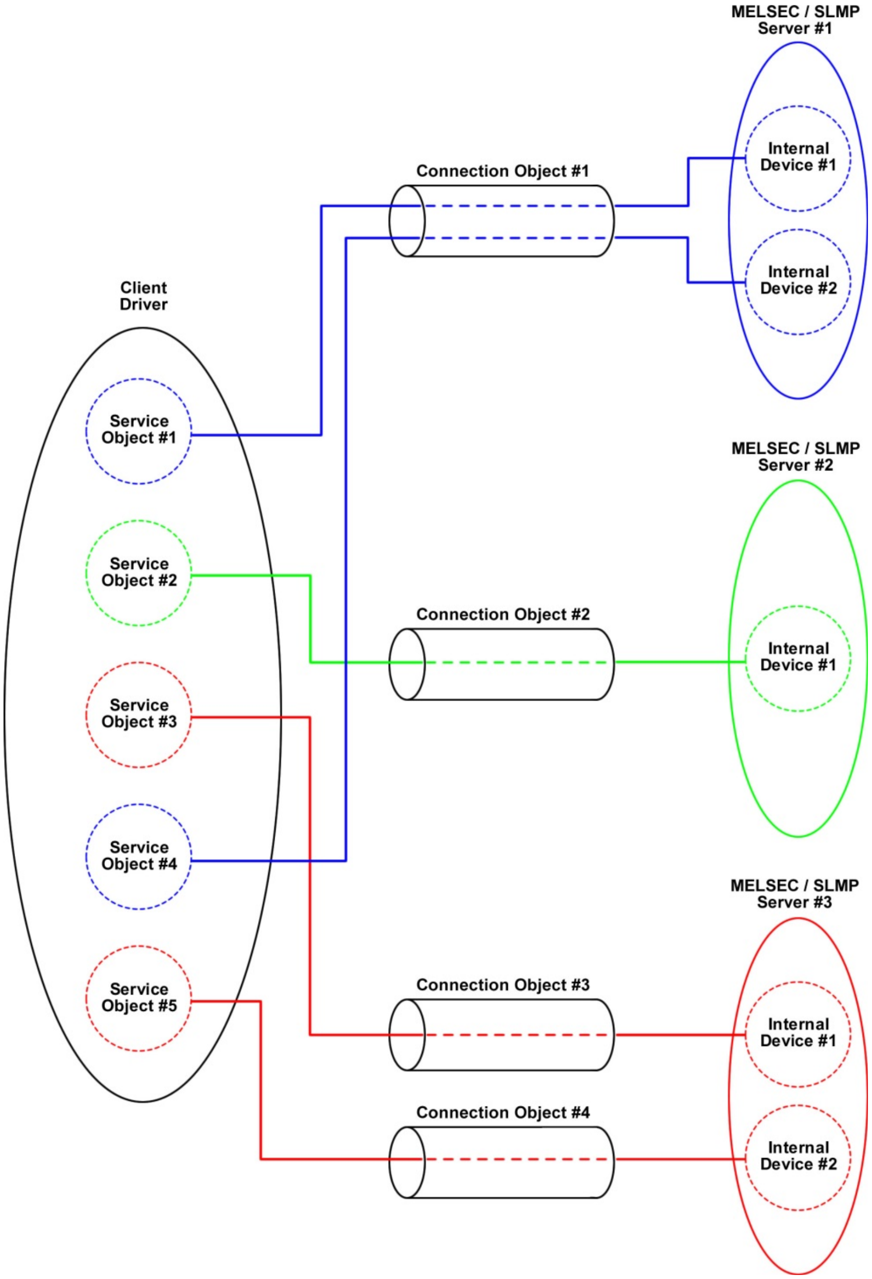

Figure 1: MELSEC / SLMP Client Concept

Some notes of interest are:

- All communication is conducted exclusively via TCP/IP.
- The driver supports frame type auto-detection: if a server supports both 3E and 1E frames, then 3E frames are given priority. Otherwise, the frame type can be manually selected (please refer to the server equipment user's manual or contact Mitsubishi for guidance in frame type selection.)
- For 3E frames, Device Memory Batch Read (0x0401) and Batch Write (0x1401) commands for both Bit and Word access are supported.
- For 1E frames, Device Memory Batch Read Bit (0x00), Batch Read Word (0x01), Batch Write Bit (0x02), and Batch Write Word (0x03) commands are supported.
- The driver supports up to 8 total connection objects (MELSEC and/or SLMP) and up to 64 total service objects.
- User-specified device codes allow the ability to target any device type, even if the device is not explicitly supported by the driver.
- Device data can be accessed without explicit configuration of service objects via EtherNet/IP explicit messaging. Refer to the separate EtherNet/IP client driver documentation for further details.

1.2 Client Settings

Scan Rate

This is the time in milliseconds the driver will wait between sending requests. This is a useful feature to reduce overall network utilization, or for certain devices or infrastructure components (such as radio modems) that may not be capable of sustaining the maximum packet rates that the driver is capable of producing. The start time for this delay is taken with respect to the moment at which the driver is capable of sending the next packet (due to either reception or timeout of the previous request). If no additional time is required, setting this field to 0 instructs the driver to send its next request packet as soon as possible.

1.3 Connection Object Settings

With the single exception of the ability to edit the "Frame Type" field, both MELSEC and SLMP connection object configuration is identical. To simplify the documentation, this section will therefore jointly summarize the settings for both types.

Name

A unique name used for identifying the connection object. Enter a string of up to 16 characters in length.

IP Address

Defines the IP address of the server device to be targeted by the connection object. All connection object endpoints (IP address and port settings) must be unique.

Port

Defines the TCP port number (1…65535) of the server device to be targeted by the connection object. All connection object endpoints (IP address and port settings) must be unique.

Frame Type

Fixed at "3E Frame" for SLMP connection objects. This field designates the frame type to use when communicating with the targeted server. If the required Frame Type is unknown, *Auto-Detect* can be selected to automatically determine the frame type when the driver establishes initial contact with the server. In this case, once the frame type has been successfully detected, the driver will then remember this type and use it as the initial preference for subsequent connection attempts. If both 3E and 1E frames are supported by the server, then 3E frames are given priority. Note that the auto-detection procedure may cause a communication error to be indicated on the server during the initial connection attempt, but this error can be cleared by power-cycling the server. To determine the required Frame Type for your equipment, please consult the server device documentation or contact Mitsubishi for assistance.

1.4 Service Object Settings

With the exception of the SLMP-specific *Network Number*, *PC/Station Number* and *Module IO Number* fields, both MELSEC and SLMP service object configuration is identical. To simplify the documentation, this section will therefore jointly summarize the settings for both types.

For each service object, the driver will continually read the designated device data from the targeted server, storing the value(s) in the database (if the read function is enabled). When data in the database changes where the service object is mapped, a write request is generated to the targeted server notifying it of the changed device data value(s) (if the write function is enabled).

Description

This 18-character (max) field is strictly for user reference: it is not used at any time by the driver.

Device Code

For user convenience, the driver provides native support for a variety of well-known device codes (refer to Table 1). However, this in no way limits the variety of device codes that the driver is able to target. Any device code that is supported by the server device can be specified by choosing "Other Device Code" from the drop-down box, and then entering the code in the *Code Value* field.

Code Value

This field is enabled only if "Device Code" is set to "Other Device Code". If the desired device code is not natively available in the *Device Code* drop-down, then choose "Other Device Code" and enter the appropriate code in this field. Note that this field expects a decimal value in the range of 0…65535. If the desired code is documented as a hexadecimal number, either enter it prepended with a "0x" (which will cause the Configuration Studio to interpret it as a hexadecimal number and automatically convert it to its decimal equivalent), or manually convert it to its decimal equivalent prior to entry.

Table 1: Natively-Supported Device Codes

Device	Type	3E Frame Device Code	1E Frame Device Code
Annunciator (F)	Bit	93$_H$	N/A
Counter Coil (CC)	Bit	C3$_H$	4343$_H$
Counter Contact (CS)	Bit	C4$_H$	4353$_H$
Counter Current Value (CN)	Word	C5$_H$	434E$_H$
Data Register (D)	Word	A8$_H$	4420$_H$
Direct Input (DX)	Bit	A2$_H$	N/A
Direct Output (DY)	Bit	A3$_H$	N/A
Edge Relay (V)	Bit	94$_H$	N/A
Index Register (Z)	Word	CC$_H$	N/A
Input (X)	Bit	9C$_H$	5820$_H$
Internal Relay (M)	Bit	90$_H$	4D20$_H$
Latch Relay (L)	Bit	92$_H$	N/A
Link Register (W)	Word	B4$_H$	5720$_H$
Link Relay (B)	Bit	A0$_H$	4220$_H$
Link Special Register (SW)	Word	B5$_H$	N/A
Link Special Relay (SB)	Bit	A1$_H$	N/A
Motion Register (#)	Word	E0$_H$	N/A
Output (Y)	Bit	9D$_H$	5920$_H$
Retentive Timer Coil (SC)	Bit	C6$_H$	N/A
Retentive Timer Contact (SS)	Bit	C7$_H$	N/A
Retentive Timer Current Value (SN)	Word	C8$_H$	N/A
Special Register (SD)	Word	A9$_H$	N/A
Special Relay (SM)	Bit	91$_H$	N/A
Step Relay (S)	Bit	98$_H$	N/A
Timer Coil (TC)	Bit	C0$_H$	5443$_H$
Timer Contact (TS)	Bit	C1$_H$	5453$_H$
Timer Current Value (TN)	Word	C2$_H$	544E$_H$

Starting Point

Specifies the initial point in a device's range of points. Enter a value from 0…16777215 (0x0…0xFFFFFF). If the connection uses 1E frames and the *Device Code* or *Code Value* targets a bit device, then the starting point should be a multiple of 16.

Number of Words

Specifies the number of words (1…719) from the *Starting Point* that are to be accessed. If the connection uses 1E frames, then the maximum value should be limited to 256 words: if a larger value is entered, the driver will internally limit it to 256. The targeted server may also impose additional device-specific limitations (please consult the server device’s documentation.)

Database Address

Defines the database address where the first point of this service object will be mapped. The Configuration Studio will not allow entry of a starting database address that will cause the service object to run past the end of the database. The highest valid database address, therefore, will depend on the number of items to be accessed.

Data Type

Fixed at “16-Bit Unsigned”.

Network Number

Applies to SLMP service objects only. Specifies the network number of the target station (0x0…0xEF).

PC/Station Number

Applies to SLMP service objects only. Specifies the PC/station number of the target station. Enter a value of 0x1…0x78, 0x7D, 0x7E or 0xFF.

Module IO Number

Applies to SLMP service objects only. Specifies the module I/O number (also known as the processor number) of the target station. Enter a value of 0x3D0…0x3D3, 0x3E0…0x3E3, or 0x3FF.

Read

Check this checkbox to enable reading (the service object will continuously read from the targeted server unless a pending write exists).

Write

Check this checkbox to enable writing (when values encompassed by this service object change in the database, these changes will be written down to the targeted server).

1.5 Diagnostics Object

Each service object can optionally include a diagnostics object for debugging and diagnostics.

Diagnostics Database Address

Enter the database address at which to store the diagnostics information.

ICC

INDUSTRIAL CONTROL COMMUNICATIONS, INC.

1600 Aspen Commons, Suite 210
Middleton, WI USA 53562-4720
Tel: [608] 831-1255   Fax: [608] 831-2045

http://www.iccdesigns.com

Printed in U.S.A