FieldServer Protocol Driver Sheet
EtherNet/IP Driver

FS-8704-14 Version: 1.12 / Rev. 1.D

1 DESCRIPTION

The EtherNet/IP driver allows the FieldServer to transfer data to and from devices over Ethernet using the EtherNet/IP protocol. The FieldServer can emulate either a Server or Client. The EtherNet/IP driver uses port 44818 by default.

EtherNet/IP uses CIP (Control and Information Protocol), the common network, transport and application layers also shared by ControlNet and DeviceNet. EtherNet/IP then makes use of standard Ethernet and TCP/IP technology to transport CIP communications packets. The result is a common, open application layer on top of open and highly popular Ethernet and TCP/IP protocols.

The Driver is able to read/write using the Data Table structure employed by all Logix Series PLC's.

PCCC support is also provided for legacy devices that do not fully support CIP encapsulation. EIP PCCC Encapsulation was tested at the FST factory using a PLC5 I785 ENET card. The following data types were tested:

- N
- F
- S

The Driver also supports PCCC communication on SLC and MicroLogix (tested on MicroLogix 1400 Device).

Fragmented Services (0x52) is supported for data_table read and write operations.

1.1 Connection Facts

FieldServer Mode	Nodes	Comments
Client	1	Only 1 Client Node allowed.
Server	32	32 Server Nodes allowed.

2 FORMAL DRIVER TYPE

Ethernet

Client or Server

3 COMPATIBILITY

FieldServer Model	Compatible
FS-B35 Series	Yes
ProtoNode/ProtoAir	Yes
QuickServer FS-QS-10xx	No
QuickServer FS-QS-12xx	Yes
QuickServer FS-QS-20xx	No
QuickServer FS-QS-22xx	Yes

4 CONNECTION INFORMATION

Connection Type: Ethernet
Ethernet Speeds Supported: 10Base-T, 100Base-T

5 DEVICES TESTED

Device	Tested (FACTORY, SITE)
ODVA Conformance Tool (ENetCT Ver A3.5)	Factory
FlexLogix PLC/ 1788 - Enet Ethernet Card	Factory
Wago 750-841 Programmable Fieldbus Coupler	Factory
ControlLogix 5561 with 1756-ENBT A Ethernet card	Factory
Micrologix 1400	Site

6 SUPPORTED COMMUNICATION FUNCTIONS

EtherNet/IP is an object orientated protocol. The object oriented structure therefore allows for classes, instances, attributes and services. The 'data types' listed below are to be considered as the objects supported in the protocol. Each of these has attributes that have been supported to differing degrees.

6.1 Data Types Supported

FieldServer Data Type	Description (or Device Data Type)
Identity – Class Code 0x01	Attributes Supported: *One instance supported (0x01)* Attributes List: Vendor ID Device Type Product Code Device Revision Status Serial Number Device Description (text) Services Supported: Get_Attribute_All Get_Attribute_Single
Message Router – Class Code 0x02	Attributes Supported: *One instance supported (0x01)* Attribute List: Max Connections Services Supported: Get_Attribute_Single
Assembly – Class Code 0x04	Attributes Supported: *Class Instance Support (0x00)* Class Attributes: 0x02 (Max Instance) *Two instances supported (0x0100 and 0x0101)* Attribute List: Member List Not Supported Data Services Supported: Get_Attribute_Single
Connection Manager – Class Code 0x06	Forward Open Service Forward Close Service
Register – Class Code 0x07	Attributes Supported: *Class Instance Support (0x00)* Class Attributes: 0x02 (Max Instance) *Two instances supported (0x01 and 0x02)* Attribute List: Status Flag Direction (read/write) Size of Data (bits) Services Supported: Get_Attribute_Single
Discrete Input Point – Class Code 0x08	No visible interface currently
Discrete Output Point – Class Code 0x09	No visible interface currently
Analog Input Point – Class Code 0x0A	Attributes Supported: *Class Instance Support (0x00)* Class Attributes: 0x02 (Max Instance) *Two instances supported (0x01 and 0x02)* Attribute List: Number of Attributes Not Supported Analog value (UINT16) not supported Vendor ID Services Supported: Get_Attribute_Single

FieldServer Data Type	Description (or Device Data Type)
Analog Output Point – Class Code 0x0B	Attributes Supported: *Class Instance Support (0x00)* Class Attributes: 0x02 (Max Instance) *Two instances supported (0x01 and 0x02)* Attribute List: Number of Attributes Not Supported Analog value (UINT16) not supported Vendor ID Services Supported: Set_Attribute_Single Get_Attribute_Single
TCP/IP Interface Object – Class Code 0xF5	Attributes Supported: *One instance supported (0x01)* Attribute List: Status Configuration Capability Configuration Control Physical Link Object Interface Configuration Host Name Services Supported: Get_Attribute_Single
EtherNet Link Object – Class Code 0xF6	Attributes Supported: *One instance supported (0x01)* Attribute List: Interface Speed Interface Flags Physical Address Interface Counters Media Counters Services Supported: Get_Attribute_Single
Data Table Object – Private Object	Attributes Supported: This object does not support instances or attributes but uses the data table structure, and associated tags, in Logix5000 PLC's. Services Supported: CIP Read Data

6.2 Connection Types Supported

Connection Type	Support Details
Unconnected Messages	Unconnected messages are supported to objects mentioned above.
Explicit Messages	Both client and server support Explicit Messages to all supported objects.
Implicit Messages Using EDS[1] File	Implicit Messages are supported.

6.3 Read Operations Supported

The functions below are supported to varying degrees by the objects above. The exact support for functions is mentioned in the table above.

FieldServer as a Client (Scanner)	FieldServer as a Server (Adapter)
Get_Attribute_Single – Service Code 0x0E	Get_Attribute_Single – Service Code 0x0E
Data_Table_Read – Service Code 0x4C	Get_Attribute_All – Service Code 0x01
	Data_Table_Read – Service Code 0x4C

6.4 Write (Control) Options Supported

FieldServer as a Client (Scanner)	FieldServer as a Server (Adapter)
Set_Attribute_Single – Service Code 0x10	Set_Attribute_Single – Service Code 0x10
Data_Table_Write – Service Code 0x4D	Data_Table_Write – service code 0x4D

[1] Electronic Data Sheets (EDS) are simply ASCII files that describe how a device can be used on an EtherNet/IP network. It describes the objects, attributes and services available in the device.

6.5 Unsupported Functions and Data Types

Function	Reason
Programming messages	FieldServer is a data transfer device, and as such, programming messages are not required.
All Group Functions. (e.g. Analog Input Group Object)	Possibility of later support.
All Application Specific Data Objects (e.g. AC/DC Drive Object)	Possibility of later support.

MSA Safety
1991 Tarob Court, Milpitas, California 95305 USA
O. +1 408 262-6611 TF: +1 800 72 SIERRA (74377) E. SMC-insidesales@msasafety.com
www.sierramonitor.com