Industrial Protocols User's Guide

Second Edition, July 2011

www.moxa.com/product

Industrial Protocols User's Guide

The software described in this manual is furnished under a license agreement and may be used only in accordance with the terms of that agreement.

Copyright Notice

Trademarks

Disclaimer

Information in this document is subject to change without notice and does not represent a commitment on the part of Moxa.

Moxa provides this document as is, without warranty of any kind, either expressed or implied, including, but not limited to, its particular purpose. Moxa reserves the right to make improvements and/or changes to this manual, or to the products and/or the programs described in this manual, at any time.

Information provided in this manual is intended to be accurate and reliable. However, Moxa assumes no responsibility for its use, or for any infringements on the rights of third parties that may result from its use.

This product might include unintentional technical or typographical errors. Changes are periodically made to the information herein to correct such errors, and these changes are incorporated into new editions of the publication.

Technical Support Contact Information

www.moxa.com/support

Moxa Americas
Toll-free: 1-888-669-2872
Tel: +1-714-528-6777
Fax: +1-714-528-6778

Moxa Europe
Tel: +49-89-3 70 03 99-0
Fax: +49-89-3 70 03 99-99

Moxa China (Shanghai office)
Toll-free: 800-820-5036
Tel: +86-21-5258-9955
Fax: +86-21-5258-5505

Moxa Asia-Pacific
Tel: +886-2-8919-1230
Fax: +886-2-8919-1231

Table of Contents

1

MODBUS/TCP MAP

Introduction

MODBUS TCP is a protocol commonly used for the integration of a SCADA system. It is also a vendor-neutral communication protocol used to monitor and control industrial automation equipment such as PLCs, sensors, and meters. In order to be fully integrated into industrial systems, Moxa's switches support Modbus TCP/IP protocol for real-time monitoring in a SCADA system.

Data Format and Function Code

MODBUS TCP supports different types of data format for reading. The primary four types of them are:

Data Access Type		**Function Code**	**Function Name**	**Note**
Bit access	Physical Discrete Inputs	2	Read Discrete Inputs	
	Internal Bits or Physical Coils	1	Read Coils	
Word access (16-bit access)	Physical Input Registers	4	Read Input Registers	Moxa Support
	Physical Output Registers	3	Read Holding Registers	

Moxa switches support **Function Code 4** with **16-bit (2-word)** data access for **read-only** information.

MODBUS Data Map and Information Interpretation of Moxa Switches

The data map addresses of Moxa switches shown in the following table start from **MODBUS address 30001** for Function Code 4. For example, the address offset 0x0000 (hex) equals MODBUS address 30001, and the address offset 0x0010 (hex) equals MODBUS address 30017. Note that all the information read from Moxa switches are in hex mode. To interpret the information, refer to the ASCII table for the translation (e.g. 0x4D = 'M', 0x6F = 'o').

Address Offset	**Data Type**	**Interpretation**	**Description**
System Information			
0x0000	1 word	HEX	Vendor ID = 0x1393
0x0001	1 word		Unit ID (Ethernet = 1)
0x0002	1 word	HEX	Product Code = 0x0003
0x0010	20 words	ASCII	Vendor Name = "Moxa" Word 0 Hi byte = 'M' Word 0 Lo byte = 'o' Word 1 Hi byte = 'x' Word 1 Lo byte = 'a' Word 2 Hi byte = '\0' Word 2 Lo byte = '\0'
0x0030	20 words	ASCII	Product Name = "EDS-408A" Word 0 Hi byte = 'E' Word 0 Lo byte = 'D' Word 1 Hi byte = 'S' Word 1 Lo byte = '-' Word 2 Hi byte = '4'

			Word 2 Lo byte = ‘0’ Word 3 Hi byte = ‘8’ Word 3 Lo byte = ‘A’ Word 4 Hi byte = ‘\0’ Word 4 Lo byte = ‘\0’
0x0050	1 word		Product Serial Number
0x0051	2 words		Firmware Version Word 0 Hi byte = major (A) Word 0 Lo byte = minor (B) Word 1 Hi byte = release (C) Word 1 Lo byte = build (D)
0x0053	2 words	HEX	Firmware Release Date For example: Word 0 = 0 x 0609 Word 1 = 0 x 0705 Firmware was released on 2007-05-06 at 09 o’clock
0x0055	3 words	HEX	Ethernet MAC Address Ex: MAC = 00-01-02-03-04-05 Word 0 Hi byte = 0 x 00 Word 0 Lo byte = 0 x 01 Word 1 Hi byte = 0 x 02 Word 1 Lo byte = 0 x 03 Word 2 Hi byte = 0 x 04 Word 2 Lo byte = 0 x 05
0x0058	1 word	HEX	Power 1 0x0000: Off 0x0001: On
0x0059	1 word	HEX	Power 2 0x0000: Off 0x0001: On
0x005A	1 word	HEX	Fault LED Status 0x0000: No 0x0001: Yes
0x0082	1 word	HEX	DO1 0x0000: Off 0x0001: On
Port Information			
0x1000 to 0x1011	1 word	HEX	Port 1 to 8 Status 0x0000: Link down 0x0001: Link up 0x0002: Disable 0xFFFF: No port
0x1100 to 0x1111	1 word	HEX	Port 1 to 8 Speed 0x0000: 10M-Half 0x0001: 10M-Full 0x0002: 100M-Half 0x0003: 100M-Full 0xFFFF: No port
0x1200 to 0x1211	1 word	HEX	Port 1 to 8 Flow Ctrl 0x0000:Off 0x0001:On 0xFFFF:No port
0x1300 to 0x1311	1 word	HEX	Port 1 to 8 MDI/MDIX 0x0000: MDI 0x0001: MDIX 0xFFFF: No port
0x1400 to 0x1413 (Port 1) 0x1414 to 0x1427 (Port 2)	20 words	ASCII	Port 1 to 8 Description Port Description = “100TX,RJ45.” Word 0 Hi byte = ‘1’ Word 0 Lo byte = ‘0’ Word 1 Hi byte = ‘0’ Word 1 Lo byte = ‘T’ … Word 4 Hi byte = ‘4’ Word 4 Lo byte = ‘5’ Word 5 Hi byte = ‘.’ Word 5 Lo byte = ‘\0’

Packets Information			
0x2000 to 0x2023	2 words	HEX	Port 1 to 8 Tx Packets Ex: port 1 Tx Packet Amount = 44332211 Received MODBUS response: 0x44332211 Word 0 = 4433 Word 1 = 2211
0x2100 to 0x2123	2 words	HEX	Port 1 to 8 Rx Packets Ex: port 1 Rx Packet Amount = 44332211 Received MODBUS response: 0x44332211 Word 0 = 4433 Word 1 = 2211
0x2200 to 0x2223	2 words	HEX	port 1 to 8 Tx Error Packets Ex: port 1 Tx Error Packet Amount = 44332211 Received MODBUS response: 0x44332211 Word 0 = 4433 Word 1 = 2211
0x2300 to 0x2323	2 words	HEX	port 1 to 8 Rx Error Packets Ex: port 1 Rx Error Packet Amount = 44332211 Received MODBUS response: 0x44332211 Word 0 = 4433 Word 1 = 2211
Redundancy Information			
0x3000	1 word	HEX	Redundancy Protocol 0x0000: None 0x0001: RSTP 0x0002:Turbo Ring 0x0003:Turbo Ring V2 0x0004:Turbo Chain
0x3100	1 word	HEX	RSTP Root 0x0000: Not Root 0x0001: Root 0xFFFF: RSTP Not Enable
0x3200 to 0x3211	1 word	HEX	RSTP Port 1 to 8 Status 0x0000: Port Disabled 0x0001: Not RSTP Port 0x0002: Link Down 0x0003: Blocked 0x0004: Learning 0x0005: Forwarding 0xFFFF: RSTP Not Enable
0x3300	1 word	HEX	TurboRing Master/Slave 0x0000: Slave 0x0001: Master 0xFFFF: Turbo Ring Not Enable
0x3301	1 word	HEX	TurboRing 1st Port status 0x0000: Port Disabled 0x0001: Not Redundant Port 0x0002: Link Down 0x0003: Blocked 0x0004: Learning 0x0005: Forwarding
0x3302	1 word	HEX	TurboRing 2nd Port status 0x0000: Port Disabled 0x0001: Not Redundant Port 0x0002: Link Down 0x0003: Blocked 0x0004: Learning 0x0005:Forwarding
0x3303	1 word	HEX	TurboRing Coupling 0x0000: Off 0x0001: On 0xFFFF: Turbo Ring is Not Enabled
0x3304	1 word	HEX	TurboRing Coupling Port Status 0x0000: Port Disabled 0x0001: Not Coupling Port 0x0002: Link Down 0x0003: Blocked 0x0005: Forwarding 0xFFFF: Turbo Ring is Not Enabled

0x3305	1 word	HEX	TurboRing Coupling Control Port Status 0x0000: Port Disabled 0x0001: Not Coupling Port 0x0002: Link Down 0x0003: Blocked 0x0005: Forwarding 0x0006: Inactive 0x0007:Active 0xFFFF:Turbo Ring is Not Enabled
0x3500	1 word	HEX	TurboRing V2 Coupling Mode 0x0000: None 0x0001: Dual Homing 0x0002: Coupling Backup 0x0003: Coupling Primary 0xFFFF:Turbo Ring V2 is Not Enabled
0x3501	1 word	HEX	TurboRing V2 Coupling Port Primary Status (Used in Dual Homing, Coupling Backup, and Coupling Primary) 0x0000:Port Disabled 0x0001: Not Coupling Port 0x0002: Link Down 0x0003: Blocked 0x0004: Learning 0x0005: Forwarding 0xFFFF: Turbo Ring V2 is Not Enabled
0x3502	1 word	HEX	TurboRing V2 Coupling Port Backup Status (Only using in Dual Homing) 0x0000: Port Disabled 0x0001: Not Coupling Port 0x0002: Link Down 0x0003: Blocked 0x0004: Learning 0x0005: Forwarding 0xFFFF: Turbo Ring V2 Not Enable
0x3600	1 word	HEX	TurboRing V2 Ring 1 status 0x0000: Healthy 0x0001: Break 0xFFFF:Turbo Ring V2 Not Enable
0x3601	1 word	HEX	TurboRing V2 Ring 1 Master/Slave 0x0000: Slave 0x0001: Master 0xFFFF: Turbo Ring V2 Ring 1 Not Enable
0x3602	1 word	HEX	TurboRing V2 Ring 1 1st Port Status 0x0000: Port Disabled 0x0001: Not Redundant Port 0x0002: Link Down 0x0003: Blocked 0x0004:Learning 0x0005:Forwarding 0xFFFF:Turbo Ring V2 Ring 1 is Not Enabled
0x3603	1 word	HEX	TurboRing V2 Ring 1's 2nd Port Status 0x0000: Port Disabled 0x0001: Not Redundant Port 0x0002: Link Down 0x0003: Blocked 0x0004: Learning 0x0005: Forwarding 0xFFFF: Turbo Ring V2 Ring 1 is Not Enabled
0x3680	1 word	HEX	TurboRing V2 Ring 2 Status 0x0000: Healthy 0x0001: Break 0xFFFF: Turbo Ring V2 Ring 2 is Not Enabled
0x3681	1 word	HEX	TurboRing V2 Ring 2 Master/Slave 0x0000: Slave 0x0001: Master 0xFFFF: Turbo Ring V2 Ring 2 is Not Enabled

0x3682	1 word	HEX	TurboRing V2 Ring 2’s 1st Port Status 0x0000: Port Disabled 0x0001: Not Redundant 0x0002: Link Down 0x0003: Blocked 0x0004: Learning 0x0005: Forwarding 0xFFFF: Turbo Ring V2 Ring 2 is Not Enabled
0x3683	1 word	HEX	TurboRing V2 Ring 2’s 2nd Port Status 0x0000: Port Disabled 0x0001: Not Redundant 0x0002: Link Down 0x0003: Blocked 0x0004: Learning 0x0005: Forwarding 0xFFFF: Turbo Ring V2 Ring 2 is Not Enabled
0x3700	1 word	HEX	Turbo Chain Switch Roles 0x0000: Head 0x0001: Member 0x0002: Tail 0xFFFF: Turbo Chain is Not Enabled
0x3701	1 word	HEX	Turbo Chain 1st Port status 0x0000: Link Down 0x0001: Blocking 0x0002: Blocked 0x0003: Forwarding 0xFFFF: Turbo Ring V2 Ring 2 Not Enable
0x3702	1 word	HEX	Turbo Chain 2nd Port status 0x0000: Link Down 0x0001: Blocking 0x0002: Blocked 0x0003: Forwarding 0xFFFF: Turbo Ring V2 Ring 2 Not Enable

2

EtherNet/IP

Introduction

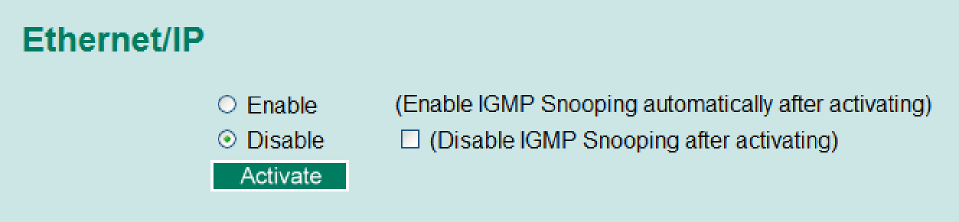
EtherNet/IP is an Industrial Ethernet Protocol defined by the ODVA association. The protocol is open to the public and vendors can implement EtherNet/IP into their industrial devices without incurring a license fee. Many vendors have adopted this protocol as the standard communication protocol between devices. For example, Rockwell Automation uses EtherNet/IP as the standard protocol for their Logix controllers over Ethernet networks.

To allow complete integration with a Rockwell system, Moxa switches not only provide a full-functioning of industrial network infrastructure, but also enable the SCADA system to monitor the status of the switches as well as that of the PLCs, .making the switches part of a Rockwell system.

Messaging Types

EtherNet/IP supports two types of communication methods for EtherNet/IP devices: Explicit Messaging and Implicit Messaging. Explicit Messaging is unscheduled and is used for a request/response communication procedure (or client/server procedure). Explicit Messaging uses TCP/IP over Ethernet. Implicit Messaging is scheduled and is used for a producer/consumer communication with UDP over Ethernet. Implicit Messaging is also called I/O Messaging.

Configuring EtherNet/IP on Moxa Switches



Ethernet/IP

○ Enable (Enable IGMP Snooping automatically after activating)

● Disable ☐ (Disable IGMP Snooping after activating)

Check the **Enable** checkbox to enable EtherNet/IP. With EtherNet/IP enabled, IGMP Snooping and IGMP Query functions will be enabled automatically to be properly integrated in Rockwell systems for multicast Implicit (I/O) Messaging.

CIP Objects of EtherNet/IP

Several communication objects are defined in CIP (Common Industrial Protocol). Moxa switches support the following objects for PLCs and SCADA systems to monitor:

- Identity Object
- TCP/IP Interface Object
- Ethernet Link Object
- Assembly Object

- Message Router Object
- Connection Manager Object
- Port Object
- Moxa Networking Object (Vendor Specific)

The supported attributes and services of the above objects are introduced in the table below, including the access rules for each attribute. To understand the details of each attribute of the standard objects, refer to the official documents of CIP introduction (Vol. 1) and the EtherNet/IP Adaptation of CIP (Vol. 2).

Identity Object

The Class code of Identity object is **0x01** (Defined in CIP Vol1, 5-2).

There is **one** instance of this object in our product. It stores the information of the production and the device. The following tables summarize the class attributes and the instance attributes.

Class Attribute List

Attr ID	Access Rule	Name	Data Type	Description
1	Get	Revision	UINT (16)	Revision of this object
2	Get	Max Instance	UINT (16)	Maximum instance number of an object currently created in this class level of the device
3	Get	Number of Instances	UINT (16)	Number of object instances currently created in this class level of the device.
6	Get	Maximum ID Number Class Attributes	UINT (16)	The attribute ID number of the last class attribute of the class definition implemented in the device
7	Get	Maximum ID Number Instance Attributes	UINT (16)	The attribute ID number of the last instance attribute of the class definition implemented in the device

Instance Attribute List

Attr ID	Access Rule	Name	(Struct.)	Data Type	Description
1	Get	Vendor ID		UINT (16)	991, the vendor ID of Moxa.
2	Get	Device Type		UINT (16)	0 x 307, "Managed Ethernet Switch".
3	Get	Product Code		UINT (16)	Please refer to Product Code Table.
4	Get	Revision		(Struct.)	The version of the Identity object
			Major	USINT (8)	The structure member, major
			Minor	USINT (8)	The structure member, minor.
5	Get	Status		WORD (16)	Not used
6	Get	Serial Number		UDINT (32)	The serial number of each device
7	Get	Product Name		SHORT_ STRING	The product name in human-readable format
15	Get/Set	Assigned Name		STRINGI	The assigned switch name For example: "Managed Redundant Switch xxxxx". (xxxxx is series number.)
17	Get/Set	Geographic Location		STRINGI	The assigned switch location The default string is "Switch Location".

The Identity Object Instance supports the following CIP Common services:

Common Service List

Service Code	Implementation		Service Name	Description
	Class	Instance		
0x01		✓	Get_Attributes_All	Returns the contents of all attributes of the class
0x0E	✓	✓	Get_Attribute_Single	Used to read an object instance attribute.
0X10		✓	Set_Attribute_Single	Used to write an object instance attribute

Product Code Table

Product Code	Model Name	Product Code	Model Name	Product Code	Model Name
0x0001	n/a	0x0012	EDS-G509	0x0023	TN-5510-PoE
0x0002	n/a	0x0013	EDS-P510	0x0024	TN-5508-PoE
0x0003	EDS-726	0x0014	EDS-516A-MM-M12	0x0025	n/a
0x0004	n/a	0x0015	IKS6526SB	0x0026	IKS-6524
0x0005	EDS-518A	0x0016	EDS-608	0x0027	n/a
0x0006	EDS-405A	0x0017	IKS-6726-PoE	0x0028	n/a
0x0007	EDS-408A	0x0018	EDS-611	0x0029	EDS-P506A
0x0008	EDS-505A	0x0019	EDS-616	0x002A	PT-7728-PTP
0x0009	EDS-508A	0x001A	EDS-619	0x002B	PT-510
0x000A	EDS-510A	0x001B	TN-5518	0x002C	PT-508
0x000B	EDS-516A	0x001C	TN-5516	0x002D	IKS-G6848
0x000C	EDS-728	0x001D	TN-5510	0x002E	IKS-G6852
0x000D	PT-7728	0x001E	TN-5508	0x002F	IKS-G6724
0x000E	EDS-828	0x001F	EOM-104	0x0030	IKS-G6728
0x000F	PT-7828	0x0020	PT-G7509	0x0031	IKS-G6824
0x0010	PT-7710	0x0021	TN-5518-PoE	0x0032	IKSG6828
0x0011	IKS-6726 or PT7728S_old	0x0022	TN-5516-PoE	0x0033	n/a

TCP/IP Interface Object

The Class code of TCP/IP Interface object is **0xf5** (Defined in CIP Vol2, 5-3).

There is **one** instance of this object.

The following tables summarize the attributes of this object.

Class Attribute List

Attr ID	Access Rule	Name	Data Type	Description
1	Get	Revision	UINT (16)	Revision of this object.
2	Get	Max Instance	UINT (16)	Maximum instance number of an object currently created in this class level of the device
3	Get	Number of Instances	UINT (16)	Number of object instances currently created at this class level of the device
6	Get	Maximum ID Number Class Attributes	UINT (16)	The attribute ID number of the last class attribute of the class definition implemented in the device
7	Get	Maximum ID Number Instance Attributes	UINT (16)	The attribute ID number of the last instance attribute of the class definition implemented in the device

Instance Attribute List

Attr ID	Access Rule	Name	(Struct.)	Data Type	Description
1	Get	Status		DWORD (32)	Interface status 0 = The Interface Configuration attribute has not been configured. 1 = The Interface Configuration attribute contains valid configuration obtained from BOOTP, DHCP or nonvolatile storage. *2 = The Interface Configuration attribute contains valid configuration, obtained from hardware settings (e.g.: pushwheel, thumbwheel, etc.)
2	Get	Configuration Capability		DWORD (32)	Interface capability flags Bit map of capability flags: Bit 0: BOOTP Client Bit 1: DNS Client Bit 2: DHCP Client Bit 3: DHCP-DNS Update Bit 4: Configuration Settable

3	Get/Set	Configuration Control		DWORD (32)	Interface control flags Bit map of control flags: Bit 0 to 3: Startup Configuration 0 = The device shall use the interface configuration values previously stored (for example, in non-volatile memory or via hardware witches). 1 = The device shall obtain its interface configuration values via BOOTP. 2 = The device shall obtain its interface configuration values via DHCP upon start-up. 3 to15 = Reserved. *Bit 4: DNS Enable If True, the device shall resolve host names by querying a DNS server.
4	Get	Physical Link Object		(Struct.)	Path to physical link object
			Path Size	UINT (16)	Size of Path
			Path	Padded EPATH	Logical segments identifying the physical link object
5	Get/Set	Interface Configuration		(Struct.)	TCP/IP network interface configuration
			IP Address	UDINT (32)	The device's IP address
			Network Mask	UDINT (32)	The device's network mask
			Gateway Address	UDINT (32)	Default gateway address
			Name Server	UDINT (32)	Primary name server
			Name Server2	UDINT (32)	Secondary name server
			Domain Name	STRING	Default domain name
6	Get/Set	Host Name		STRING	Host name

The TCP/IP Object Instance supports the following CIP Common services:

Common Service List

Service Code	Implementation		Service Name	Description
	Class	Instance		
0 x 01		✓	Get_Attributes_All	Returns the contents of all attributes of the class
0 x 0E	✓	✓	Get_Attribute_Single	Used to read an object instance attribute
0 x 10		✓	Set_Attribute_Single	Used to modify an object instance attribute

Ethernet Link Object

The Class code of Ethernet Link object is **0xf6** (Defined in CIP Vol2, 5-4).

For each switch port, there is an instance of this class.

The following table shows the mapping of instance number and the switch port number.

Instance Number	Mapping to
0	Ethernet Link class
1	1st switch port
2	2nd switch port
3	3rd switch port
–	–

The following tables summarize the attributes of the Ethernet Link object.

There are some vendor specific attributes in the table (Starting from attribute Id 100).

Class Attribute List

Attr ID	Access Rule	Name	Data Type	Description
1	Get	Revision	UINT (16)	Revision of this object
2	Get	Max Instance	UINT (16)	Maximum instance number of an object currently created in this class level of the device
3	Get	Number of Instances	UINT (16)	Number of object instances currently created in this class level of the device
6	Get	Maximum ID Number Class Attributes	UINT (16)	The attribute ID number of the last class attribute of the class definition implemented in the device
7	Get	Maximum ID Number Instance Attributes	UINT (16)	The attribute ID number of the last instance attribute of the class definition implemented in the device
100	Get	Moxa-specific Revision	UINT (16)	Revision of Moxa specific attributes and services

Instance attribute list

Attr ID	Access Rule	Name	(Struct.)	Data Type	Description
1	Get	Interface Speed		UDINT (32)	Interface speed currently in use (Speed in Mbps, e.g., 0, 10, 100, 1000, etc.)
2	Get	Interface Flags		DWORD (32)	Refer to the Interface Flags table.
3	Get	Physical Address		ARRAY of 6 USINT(8)	MAC layer address (The System MAC address).
4	Get	Interface Counters		(Struct.)	Counters relevant to the receipt of packets.
			In Octets	UDINT (32)	Octets received on the interface.
			In Ucast Packets	UDINT (32)	Unicast packets received on the interface.
			In NUcast Packets	UDINT (32)	Non-unicast packets received on the interface.
			In Discards	UDINT (32)	Inbound packets received on the interface but are discarded.
			In Errors	UDINT (32)	Inbound packets that contain Errors (does not include In Discards).
			*In Unknown Protos	UDINT (32)	Inbound packets with unknown protocols (not support).
			Out Octets	UDINT (32)	Octets sent on the interface.
			Out Ucast Packets	UDINT (32)	Unicast packets sent on the interface.
			Out NUcast Packets	UDINT (32)	Non-unicast packets sent on the interface.
			Out Discards	UDINT (32)	Discarded outbound packets.
			Out Errors	UDINT (32)	Outbound packets that contain errors.
5	Get	Media Counters		(Struct.)	
			Alignment Errors	UDINT (32)	Received frames that are not an integral number of octets in length.
			FCS Errors	UDINT (32)	Received frames that do not pass the FCS check.
			Single Collisions	UDINT (32)	Successfully transmitted frames which experienced exactly one collision.
			Multiple Collisions	UDINT (32)	Successfully transmitted frames which experienced more than one collision.
			SQE Test Errors	UDINT (32)	Number of times the SQE test error message is generated.

			Deferred Transmissions	UDINT (32)	Frames for which first transmission attempt is delayed because the medium is busy.
			Late Collisions	UDINT (32)	Number of times a collision is detected later than 512 bit times into the transmission of a packet.
			Excessive Collisions	UDINT (32)	Frames for which transmission fails due to excessive collisions.
			MAC Transmit Errors	UDINT (32)	Frames for which transmission fails due to an internal MAC sublayer transmit error.
			Carrier Sense Errors	UDINT (32)	Times that the carrier sense condition was lost or never asserted when attempting to transmit a frame.
			Frame Too Long	UDINT (32)	Received frames that exceed the maximum permitted frame size.
			MAC Receive Errors	UDINT (32)	Frames for which reception on an interface fails due to an internal MAC sublayer receive error.
6	Get/Set	Interface Control		(Struct.)	Configuration for physical interface.
			Control Bits	WORD (16)	Bit 0: Auto-Negotiate Value 0: Force Value 1: Auto-Nego Bit 1: Half/Full Duplex Value 0: half duplex Value 1: full duplex Bit 2 to 15: Reserved, all zero
			Forced Interface Speed	UINT (16)	Speed at which the interface shall be forced to operate.
100	Get	Interface Port Index		UDINT (32)	Port index.
101	Get	Interface Port Description		STRING	Port description.
102	Get/Set	Broadcast Storm Protection		USINT (8)	Value 0: Disabled Broadcast Storm Protection. Value 1: Enable Broadcast Storm Protection. (not all production support)
103	Get	Interface Utilization		USINT (8)	RX interface utilization in percentage
104	Get/Set	Utilization Alarm Upper Threshold		USINT (8)	RX interface utilization upper limit in percentage
105	Get/Set	Utilization Alarm Lower Threshold		USINT (8)	RX interface utilization lower limit in percentage(not support)
106	Get/Set	Port Link Alarm		USINT (8)	Value 0: Ignore Value 1: On (Relay 1) Value 2: On (Relay 2) Value 3: Off (Relay 1) Value 4: Off (Relay 2)
107	Get/Set	Port Traffic-Overload Alarm		USINT (8)	Value 0: Disable Value 1: Enable(Relay 1) Value 2: Enable(Relay 2)

Interface Flags

Bit(s)	Called	Definition
0	Link Status	0 indicates an inactive link; 1 indicates an active link.
1	Half/Full Duplex	0 indicates half duplex; 1 indicates full duplex.
2 to 32	Reserved	Not used

The Ethernet Link Object Instance supports the following CIP Common services:

Common Service List

Service Code	Implementation		Service Name	Description
	Class	Instance		
0x0E	✓	✓	Get_Attribute_Single	Used to read an object instance attribute
0x10		✓	Set_Attribute_Single	Used to modify an object instance attribute

Assembly Object

The Moxa switch support **static** assembly object for CIP I/O messaging.

The Class code is **0x04** (Defined in CIP Vol 1, 5-5).

There are three instances of this object as the following.

	Instance Number	Size (byte)
Input	2	[TBD]
Output	1	[TBD]
Configuration	3	0

The **Input** means the data is produced by switch which includes the information and status report to the originator for monitoring. The **Output** means the data is generated by the originator (remote host) and is consumed by switch.

Class Attribute List

Attr ID	Access Rule	Name	Data Type	Description
1	Get	Revision	UINT (16)	Revision of this object

Instance Attribute List

Attr ID	Access Rule	Name	(Struct.)	Data Type	Description
3	Get/Set	Data		Array of BYTE	The implicit messaging content
4	Get	Size		UINT (16)	Number of bytes in Attr. 3

Common Service List

Service Code	Implementation		Service Name	Description
	Class	Instance		
0x0E	✓	✓	Get_Attribute_Single	Used to read an object instance attribute
0x10		✓	Set_Attribute_Single	Used to modify an object instance attribute

For the definition of the I/O messaging, see the following table for details.

I/O Messaging Content

Direction	I/O data	Size	Value & Description
Input	Switch Fault Status	UDINT (32)	Please refer to Moxa Networking Object Attr ID 2.
	Port Exist	ULINT (64)	Please refer to Moxa Networking Object Attr ID 4.
	Port Link Status	ULINT (64)	Please refer to Moxa Networking Object Attr ID 6.
Output	Port Enable	ULINT (64)	Please refer to Moxa Networking Object Attr ID 5.

Message Router Object

The object within a node that distributes messaging requests to the appropriate application objects.

The supported messaging connections are as the following:

- Explicit Messaging
- Unconnected Messaging
- Implicit messaging

When using the UCMM to establish an explicit messaging connection, the target application object is the Message Router object (Class Code **2**).

There is no class attribute for Message Router object.

Instance Attribute List

Attr ID	Access Rule	Name	(Struct.)	Data Type	Description
1	Get	Object_list		(Struct.)	A list of supported objects
			Number	UINT (16)	Number of supported classes in the classes array
			Classes	Array of UINT (16)	List of supported class codes
2	Get	Number Available		UINT (16)	Maximum number of connections supported
3	Get	Number Active		UINT (16)	Number of connections currently used by system components
4	Get	Active Connections		Array of UINT (16)	A list of the connection IDs of the currently active connections

Common Service List

Service Code	Implementation		Service Name	Description
	Class	Instance		
0x0E		✓	Get_Attribute_Single	Used to read an object instance attribute

Connection Manager Object

The Connection Manager Class allocates and manages the internal resources associated with both I/O and Explicit Messaging connections.

The class code is **0x06**. There is one instance of this object.

The supported connection trigger type is ***cyclic*** and ***change of state***.

The instance attribute list is introduced as the following.

Class Attribute List

Attr ID	Access Rule	Name	Data Type	Description
1	Get	Revision	UINT (16)	Revision of this object

Class Attribute List

Attr ID	Access Rule	Name	Data Type	Description
1	Get/Set	Open Requests	UINT(16)	Number of Forward Open service requests received

Common Service List

Service Code	Implementation		Service Name	Description
	Class	Instance		
0x0e	✓	✓	Get_Attribute_Single	Returns the contents of the specified attribute
0x10		✓	Set_Attribute_Single	Used to modify an object instance attribute
0x4E		✓	Forward_Close	Closes a connection
0x54		✓	Forward_Open	Opens a connection

Port Object

The port object represents the underlying interface of CIP which is EtherNet/IP.

The class code is **0xf4**. There is one instance of this object.

The instance attribute “**Port Type**” identifies the CIP adaptation.

Class Attribute List

Attr ID	Access Rule	Name	(Struct.)	Data Type	Description
1	Get	Revision		UINT (16)	Revision of this object
2	Get	Max Instance		UINT (16)	Maximum instance number of an object currently created in this class level of the device

3	Get	Number of Instances		UINT (16)	Number of object instances currently created at this class level of the device.
8	Get	Entry Port		UINT (16)	The attribute ID number of the last class attribute of the class definition implemented in the device
9	Get	Port Instance Info		(Array of Struct.)	
			Port Type	UINT (16)	Enumerates the type of port
			Port Number	UINT (16)	CIP port number associated with this port

Instance Attribute List

Attr ID	Access Rule	Name	(Struct.)	Data Type	Description
1	Get	Port Type		UINT (16)	Enumerates the type of port. 4 = EtherNet/IP.
2	Get	Port Number		UINT (16)	CIP port number associated with this port.
3	Get	Link Object		(Struct.)	
			Path Length	UINT (16)	Number of 16 bit words in the following path.
			Link Path	Padded EPATH	Logical path segments that identify the object for this port.
4	Get	Port Name		SHORT_STRING	String which names the physical network port. The maximum number of characters in the string is 64.
5	Get	Port Type Name		SHORT_STRING	String which names the port type. The maximum number of characters in the string is 64.
6	Get/Set	Port Description		SHORT_STRING	String which describes the port. The maximum number of characters in the string is 64.
7	Get	Node Address		Padded EPATH	Node number of this device on port. The range within this data type is restricted to a Port Segment.
9	Get	Port Key		Packed EPATH	Electronic key of network/chassis this port is attached to. This attribute shall be limited to format 4 of the Logical Electronic Key segment.

Common Service List

Service Code	Implementation		Service Name	Description
	Class	Instance		
0x0E	✓	✓	Get_Attribute_Single	Used to read an object instance attribute
0x10		✓	Set_Attribute_Single	Used to modify an object instance attribute

Moxa Networking Object (Vendor Specific)

The Moxa Networking object includes system information and status.

It can also be used to do the device diagnostic & configuration through explicit messaging.

The class code is **0x**404.

Class Attribute List

Attr ID	Access Rule	Name	Data Type	Description
1	Get	Revision	UINT (16)	Revision of this object

Instance Attribute List

Attr ID	Access Rule	Name	Data Type	Description
1	Get	Firmware Version	UDINT (32)	Switch firmware version
2	Get	System Fault Status	UDINT (32)	Switch fault status Bit 0: Reserved Value 0: Ok Value 1: Fail Bit 1: Reserved Value 0: Ok Value 1: Fail Bit 2: Port utilization alarm Value 0: No alarm

				Value 1: alarm Bit 3: Port link up Value 0: No alarm Value 1: Alarm Bit 4: Port link down Value 0: No alarm Value 1: Alarm Bit 5: Turbo ring break(Ring Master only) Value 0: No alarm Value 1: Alarm Bit 6: Power Input 1 fail Value 0: No alarm Value 1: Alarm Bit 7: Power Input 2 fail Value 0: No alarm Value 1: Alarm Bit 8:DI 1(off) Value 0: No alarm Value 1: Alarm Bit 9: DI 1(on) Value 0: No alarm Value 1: Alarm Bit 10: DI 2(off) Value 0: No alarm Value 1: Alarm Bit 11: DI 2(on) Value 0: No alarm Value 1: Alarm Bit 12 to 31: Reserved
3	Get	Switch Port Number	USINT (8)	Switch max port number
4	Get	Port Exist	ULINT (64)	switch per port exist Bit mask, the LSB indicates the first port. Value 0: Not exist Value 1: Exist
5	Get/Set	Port Enable	ULINT (64)	Switch per port enable Bit mask, the LSB indicates the first port. Value 0: Enable Value 1: Disable
6	Get	Port Link Status	ULINT (64)	Switch per port link status Bit mask, the LSB indicates the first port. Value 0: Link down Value 1: Link up
7	Get/Set	IGMP Snooping Enable	USINT (8)	IGMP snooping enable: Value 0: Disable Value 1: Enable
8	Get/Set	Query Interval	UDINT (32)	Query interval range from 20 to 600 secs
9	Get/Set	IGMP Enhanced Mode	USINT (8)	IGMP enhanced mode 0: Enable(default) 1: Disable
14	Get/Set	Relay 1	USINT (8)	Override relay warning setting 0: Disable(default) 1: Enable
15	Get/Set	Relay 2	USINT (8)	Override relay warning setting 0: Disable (default) 1: Enable
16	Get/Set	Power 1 Relay Warning	USINT (8)	Power input 1 failure (on->off) 0: Disable (default) 1: Enable (relay 1) 2: Enable (relay 2)
17	Get/Set	Power 2 Relay Warning	USINT (8)	Power input 2 failure (on->off) 0: Disable (default) 1: Enable (relay 1) 2: Enable (relay 2)
18	Get/Set	DI 1 (0ff) Relay Warning	USINT (8)	DI 1 (0ff) 0: Disable (default) 1: Enable (relay 1) 2: Enable (relay 2)
19	Get/Set	DI 1 (on) Relay Warning	USINT (8)	DI 1 (0n) 0: Disable (default) 1: Enable (relay 1) 2: Enable (relay 2)
20	Get/Set	DI 2 (0ff) Relay Warning	USINT (8)	DI 2 (0ff) 0: Disable (default)

				1: Enable (relay 1) 2: Enable (relay 2)
21	Get/Set	DI 2 (on) Relay Warning	USINT (8)	DI 2 (0n) 0: Disable (default) 1: Enable (relay 1) 2: Enable (relay 2)
22	Get/Set	Turbo Ring Break Relay Warning	USINT (8)	Turbo ring break (Ring Master only) 0: Disable (default) 1: Enable (relay 1) 2: Enable (relay 2)
23	Get	CPU Usage	USINT (8)	Percent of usage (0 to100)

Common Service List

Service Code	Implementation		Service Name	Description
	Class	Instance		
0x0E	✓	✓	Get_Attribute_Single	Used to read an object instance attribute
0x10		✓	Set_Attribute_Single	Used to modify an object instance attribute

EDS File

The EDS (Electronic Data Sheet) file contains electronic descriptions of all relevant communication parameters and objects of an EtherNet/IP device. It is required for RSLogix 5000 to recognize Moxa switch and its CIP capability.

The list includes the sections which are described in our EDS file.

- [File]
- [Device]
- [Device Classification]
- [Port]

Icon should be 32 * 32 in pixel.

Add-On Instructions (AOI)

The AOI encapsulates Moxa switch supported EtherNet/IP functions in a common interface logic component. In RSLogix 5000 programming, users could use the AOI to communicate with Moxa switches and need not know the internal logic.

Our AOI would provide logic of Moxa switch configuration and monitoring by using EtherNet/IP in explicit messaging and implicit messaging. The AOI also provides some tags for RSLogix 5000/SCADA programming.

AOI test program

A test program of RSLogix 5000 to demonstrate AOI usage is provided. This program would demonstrate the following procedures.

1. Get an attribute of the Moxa switch.
2. Set an attribute of the Moxa switch.
3. Monitor an I/O data of the Moxa switch.