Using EtherNet/IP Explicit Messaging

EtherNet/IP explicit messaging allows the originator (PLC or HMI) to request individual services from the target device (RMC). These requests are made explicitly rather than being scheduled cyclically like I/O. Explicit messaging is much more flexible than I/O in terms of what data or services are accessed in the target device, since I/O connections must pre-configure the I/O data to be exchanged.

In most cases, RMC users use EtherNet/IP explicit messaging to read and write registers in the RMC. The RMC provides two methods for doing so. Also, some advanced EtherNet/IP users may want to access standard CIP services and attributes. The following sections describe how to use EtherNet/IP explicit messaging in each of these cases.

Reading and Writing RMC Registers using the Allen-Bradley PCCC/DF1 Services

When using an Allen-Bradley PLC with EtherNet/IP, the PLC's MSG (message) block is used to read and write registers in the RMC. See Using Allen-Bradley Controllers via Message Block for details. Requests made using this method use Allen-Bradley file addressing, which is described in the DF1 Addressing topic.

Other PLCs or HMIs that can read or write from Allen-Bradley file addresses (e.g. F7:0, L8:16) may also be able to use this method. The underlying object in the RMC is the Allen-Bradley PCCC/DF1 Object. This object provides services that encapsulate the PCCC/DF1 commands and functions. The RMC implements the following PCCC function codes:

- Diagnostic Status (06 03)
- Echo (06 00)
- SLC Protected Typed Logical Read with 2 Address Fields (0F A1)
- SLC Protected Typed Logical Read with 3 Address Fields (0F A2)
- SLC Protected Typed Logical Write with 2 Address Fields (0F A9)
- SLC Protected Typed Logical Write with 3 Address Fields (0F AA)
- PLC5 Typed Read (0F 68)
- PLC5 Typed Write (0F 67)
- PLC5 Word Range Read (0F 01)
- PLC5 Word Range Write (0F 00)

Reading and Writing RMC Registers using the Data Table Object

For PLCs that support explicit messaging, but do not support the PCCC/DF1 services above, the RMC's Data Table Object can be used to read and write registers. The Data Table Object provides relatively simple services for reading and writing registers. The PLC must build and issue a CIP Service Request and then extract the results from the CIP Service Response. The method for issuing and receiving CIP service requests and responses are PLC-specific.

To invoke a CIP service, you must know the object class, object instance, service ID, and format of the request and response data. For all services in the Data Table Object, the object class and instance will be as follows:

Field	Value
Object Class	0xC0 (192)
Object Instance	0x01 (1)

The Data Table Object supports the following services:

Service ID	Service Name	Description
0x4B (75)	Read (LSB First)	Read one or more registers from the RMC. All multi-byte values are encoded least-significant byte (LSB) first.
0x4C (76)	Write (LSB First)	Writes one or more registers to the RMC. All multi-byte values are encoded least-significant byte (LSB) first.
0x4D (77)	Read (MSB First)	Read one or more registers from the RMC. All multi-byte values are encoded most-significant byte (MSB) first.
0x4E (78)	Write (MSB First)	Writes one or more registers to the RMC. All multi-byte values are encoded most-significant byte (MSB) first.

Notice that there are really only two services: read and write. However, because there are two standards for encoding multi-byte data, both LSB-first and MSB-first versions of each service are provided.

The following sections describe the format of the CIP requests and responses for the service types. Notice that the first six bytes of the requests and the first four bytes of the response in the charts that follow make up the standard CIP message routing header. Therefore your PLC or HMI may already build this part of the packet for you.

Read (LSB or MSB First) Request:

Offset	Type1	Size	Field Name	Description
0	USINT	1	Service	Must be 0x4B (LSB First) or 0x4D (MSB First).
1	USINT	1	Path Size	Must be 0x02.
2	USINT[4]	4	Path	Must be 0x20 0xC0 0x24 0x01.
6	UINT	2	File	Register file in the RMC to read.
8	UINT	2	Element	First element to read from the specified file.
10	UINT	2	Count	Number of DINTs or REALs to read.

1 The byte order of multi-byte fields is determined by the Read service selected. For service 0x4B (Read LSB First), the least-significant byte must be sent first. For service 0x4D (Read MSB First), the most-significant byte must be sent first.

Read (LSB or MSB First) Response:

Offset	Type1	Size	Field Name	Description
0	USINT	1	Reply Service	Will be 0xCB (LSB First) or 0xCD (MSB First).
1	USINT	1	Reserved	Will be 0
2	USINT	1	General Status	See below. Zero (0) means success.
3	USINT	1	Additional Status Size	Will be 0
4	DINT or REAL	4	Register0	Value of first register
8	DINT or	4	Register1	Value of second register

	REAL			
:	:	:	:	:
4+4xN	DINT or REAL	4	RegisterN	Value of Nth register

1 The byte order of multi-byte fields is determined by the Read service selected. For service 0x4B (Read LSB First), the least-significant byte will be sent first. For service 0x4D (Read MSB First), the most-significant byte will be sent first.

Write (LSB or MSB First) Request:

Offset	Type1	Size	Field Name	Description
0	USINT	1	Service	Must be 0x4C (LSB First) or 0x4E (MSB First).
1	USINT	1	Path Size	Must be 0x02.
2	USINT[4]	4	Path	Must be 0x20 0xC0 0x24 0x01.
6	UINT	2	File	Register file in the RMC to write to.
8	UINT	2	Element	First element in the specified file to write to.
10	UINT	2	Count	Number of DINTs or REALs to write.
12	DINT or REAL	4	Register0	Value of first register
16	DINT or REAL	4	Register1	Value of second register
:	:	:	:	:
12+4xN	DINT or REAL	4	RegisterN	Value of Nth register

1 The byte order of multi-byte fields is determined by the Write service selected. For service 0x4C (Write LSB First), the least-significant byte must be sent first. For service 0x4E (Write MSB First), the most-significant byte must be sent first.

Write (LSB or MSB First) Response:

Offset	Type	Size	Field Name	Description
0	USINT	1	Reply Service	Will be 0xCC (LSB First) or 0xCE (MSB First).
1	USINT	1	Reserved	Will be 0
2	USINT	1	General Status	See below. Zero (0) means success.
3	USINT	1	Additional Status Size	Will be 0

In the CIP service response, the General Status field can hold one of several values:

Gen Status	Description
0x00 (0)	Success. The request succeeded.

	Invalid service data.
0x03 (3)	This error will occur if the File or Element values are invalid, or for a write, if the length of the request does not match the expected length based on the Count value.
	Reply data too large.
0x11 (17)	This error will occur if the response generated is too large to be sent. If you receive this error, you must decrease the Count value for a Read request. The maximum value varies based on the PLC, but is generally around 120 registers.
	Request length too small.
0x13 (19)	This error will occur on a Read or Write request if the length of the request is smaller than expected.
	Request length too large.
0x15 (21)	This error will occur on a Read request if the length of the request is larger than expected.

Accessing Standard CIP Services and Attributes

The RMC supports a number of standard CIP objects. These objects have various services and attributes that can be accessed through CIP Service requests. The RMC75E and RMC150E Statement of Conformance documents—available for download on Delta's web site (www.deltamotion.com)—list the objects, services, and attributes supported in the these controllers. Advanced EtherNet/IP users can use these documents, together with the EtherNet/IP specification and PLC documentation to access services in the supported objects.

See Also

EtherNet/IP Overview

Send comments on this topic.