

Communication Exchange ControlLogix & M580

Prepared by

Behnaz Champey

Schneider Electric/Innovation & Technology/Network Competency Center



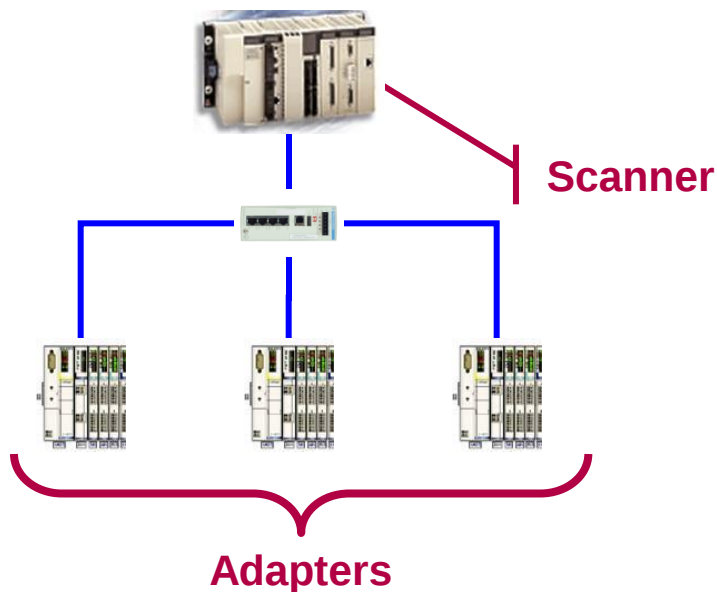
Device Classes

- **Scanner:**

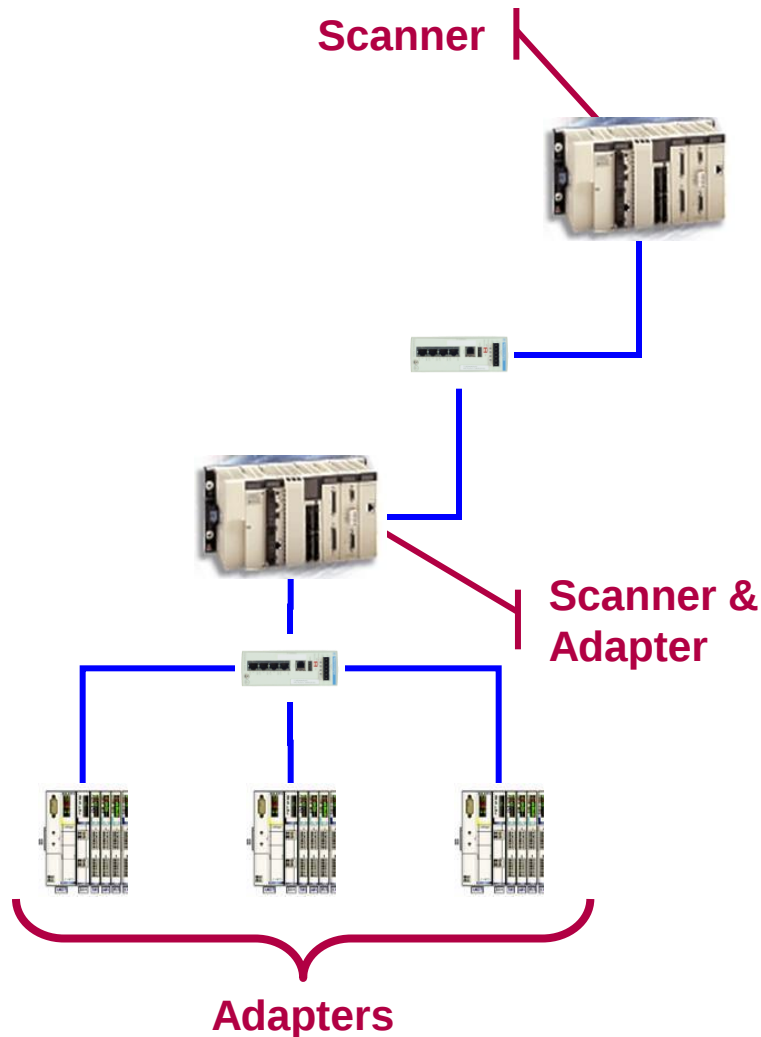
- Acts as a Client in I/O communication
 - Originates message requests
 - Consumes data

- **Adapter:**

- Acts as a Server in I/O communications
 - Target for message requests
 - Produces data



Device Classes



- **Scanner:**

- Acts as a Client in I/O communication
 - Originates message requests
 - Consumes data

- **Adapter:**

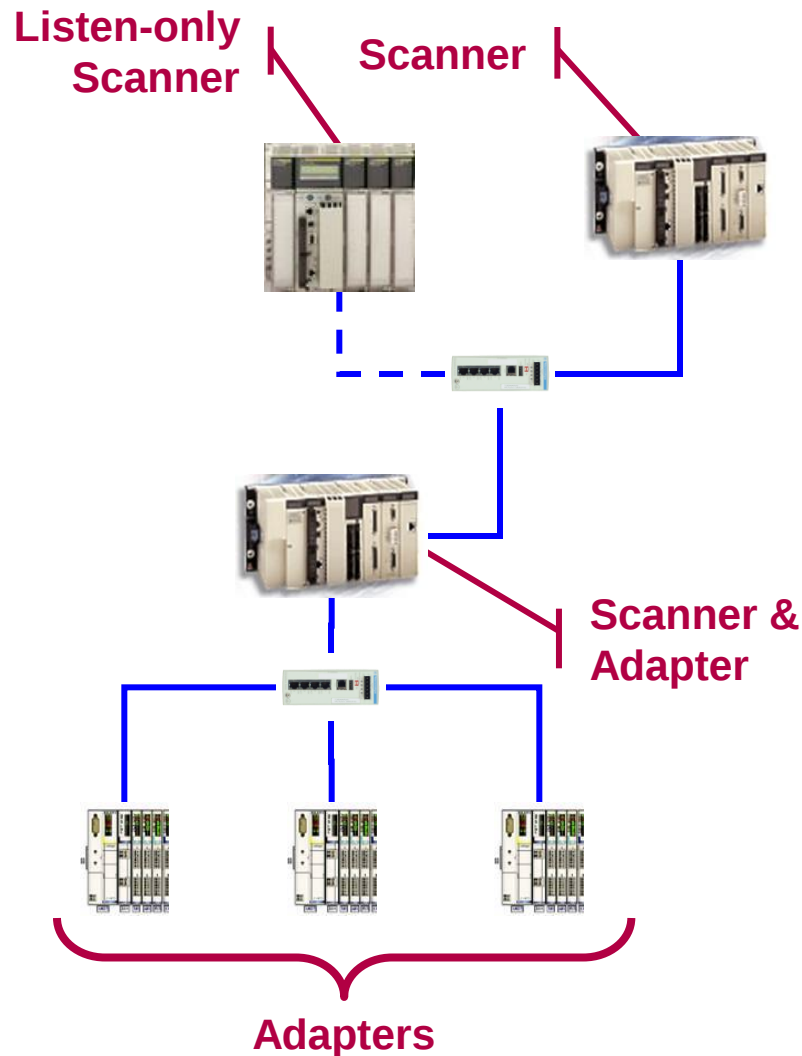
- Acts as a Server in I/O communications
 - Target for message requests
 - Produces data

- **Scanner as an Adapter**

- Scanner class device acting as an Adapter through Local slave function



Device Classes



- **Scanner:**

- Acts as a Client in I/O communication
 - Originates message requests
 - Consumes data

- **Adapter:**

- Acts as a Server in I/O communications
 - Target for message requests
 - Produces data

- **Scanner as an Adapter**

- Scanner class device acting as an Adapter through Local slave function

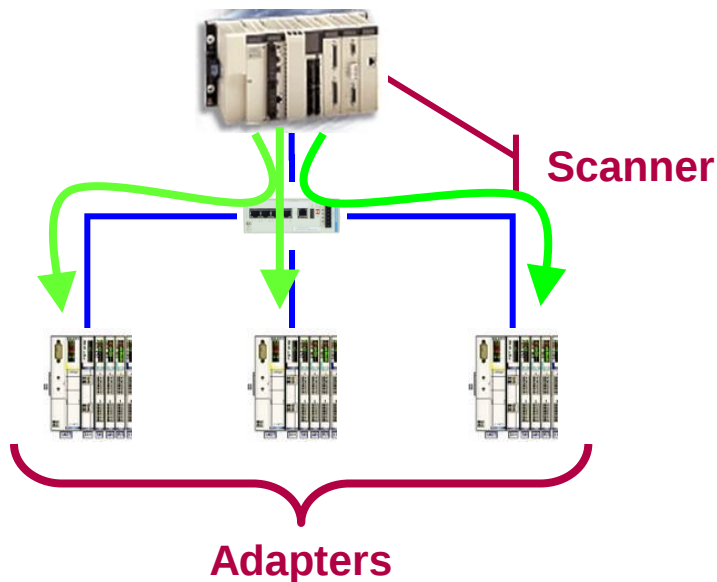
- **Listen-Only Scanner**

- Subscribes to another PLC's I/O communications



Messaging Types

Implicit Messaging (Class 1)



- **Scanner:**

- Initiates request (Originator)

- Establishes connections

- Dictates Request Packet Interval (RPI)

- Consumes data & writes outputs (if needed)

- **Adapters:**

- Acknowledge request (Target)

- Accept connections

- Collect and prepare data

- Produce data & accept output writes

- **Details:**

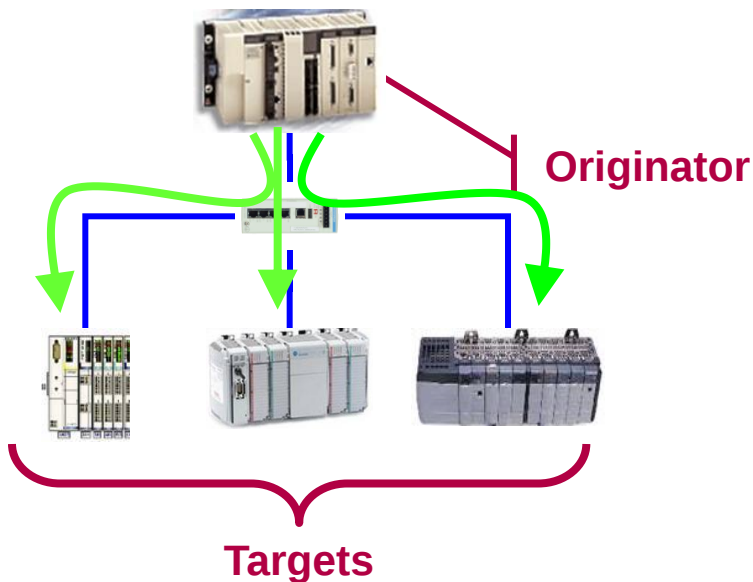
- Inputs: UDP Multicast messages

- Outputs: UDP Unicast messages



Messaging Types

Explicit Messaging (Class 3)

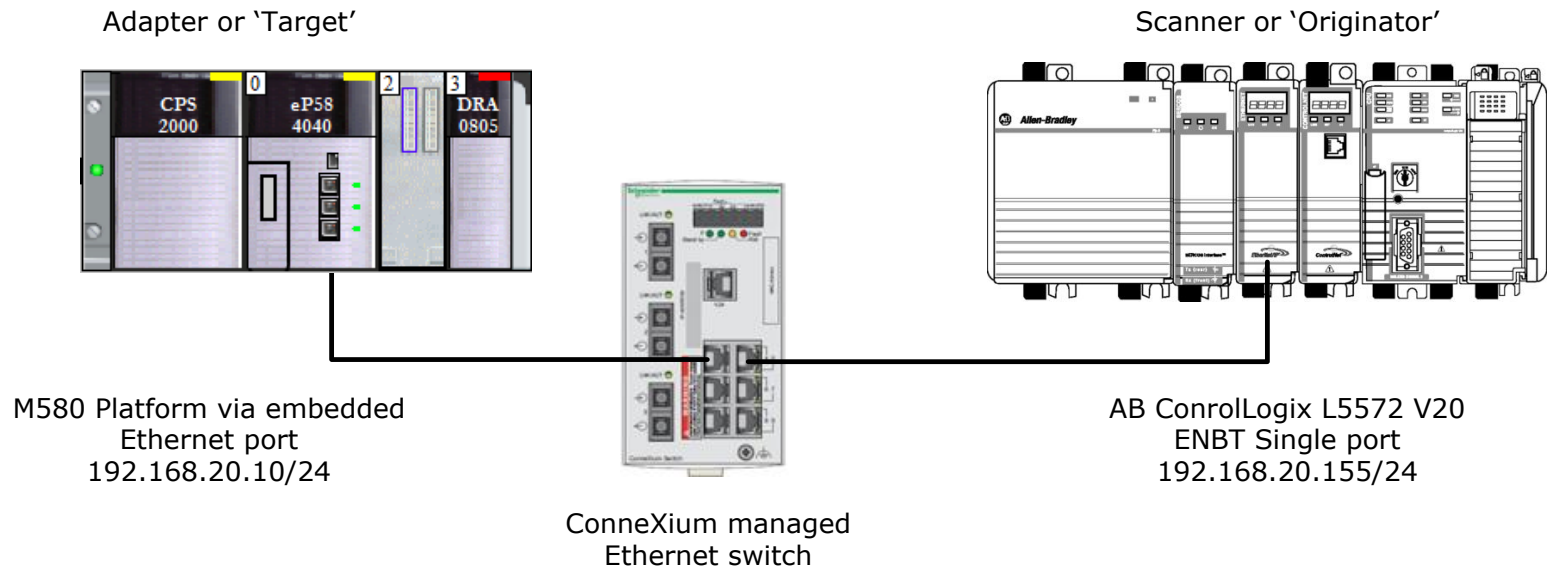


- **Originator:**
 - Initiates request
 - Establishes connections (as needed)
 - Acknowledges response
- **Target:**
 - Acknowledge request
 - Accept connections (if needed)
 - Collect and prepare data
 - Respond to request
- **Details:**
 - TCP-based
 - Unicast messages



Tested Architecture

EtherNet/IP Class 1 Communications

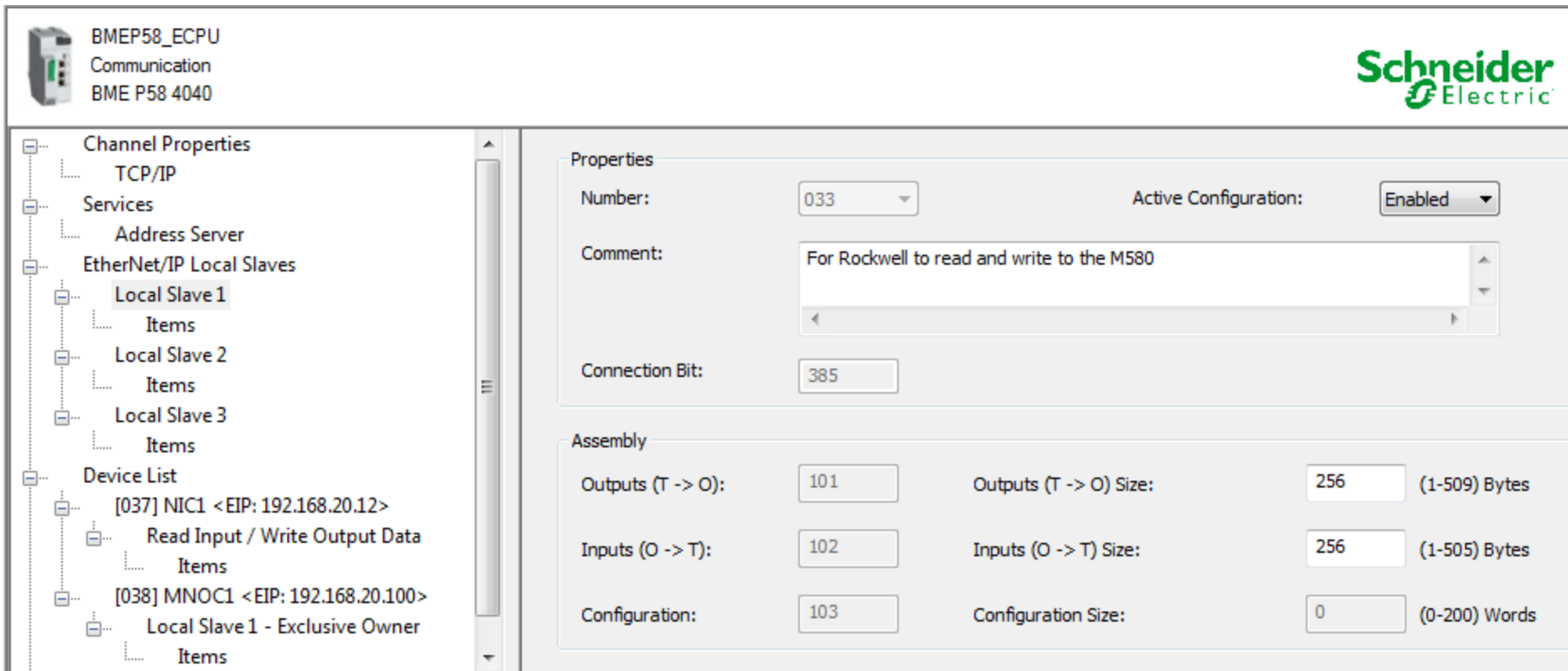


Assembly

Outputs (T -> O):	101	Outputs (T -> O) Size:	256	(1-509) Bytes
Inputs (O -> T):	102	Inputs (O -> T) Size:	256	(1-505) Bytes
Configuration:	103	Configuration Size:	0	(0-200) Words

EIP I/O connection information from M580 DTM config required for Logix versions < V20 (V20+ can use an EDS file)

M580 Adapter Configuration



The screenshot displays the Schneider Electric M580 Adapter Configuration software interface. The top left shows the device icon and name 'BMEP58_ECPU', with 'Communication' and 'BME P58 4040' listed below. The top right features the Schneider Electric logo. The left sidebar contains a tree view with 'Channel Properties' expanded, showing 'TCP/IP' and 'Services'. Under 'Services', 'EtherNet/IP Local Slaves' is expanded, showing 'Local Slave 1', 'Local Slave 2', and 'Local Slave 3', each with an 'Items' sub-entry. The 'Device List' section shows '[037] NIC1 <EIP: 192.168.20.12>' with 'Read Input / Write Output Data' and 'Items' sub-entries, and '[038] MNOC1 <EIP: 192.168.20.100>' with 'Local Slave 1 - Exclusive Owner' and 'Items' sub-entries. The main panel is divided into 'Properties' and 'Assembly' sections. The 'Properties' section includes 'Number' (033), 'Active Configuration' (Enabled), 'Comment' (For Rockwell to read and write to the M580), and 'Connection Bit' (385). The 'Assembly' section includes 'Outputs (T -> O):' (101), 'Outputs (T -> O) Size:' (256 (1-509) Bytes), 'Inputs (O -> T):' (102), 'Inputs (O -> T) Size:' (256 (1-505) Bytes), 'Configuration:' (103), and 'Configuration Size:' (0 (0-200) Words).

BMEP58_ECPU
Communication
BME P58 4040

Schneider Electric

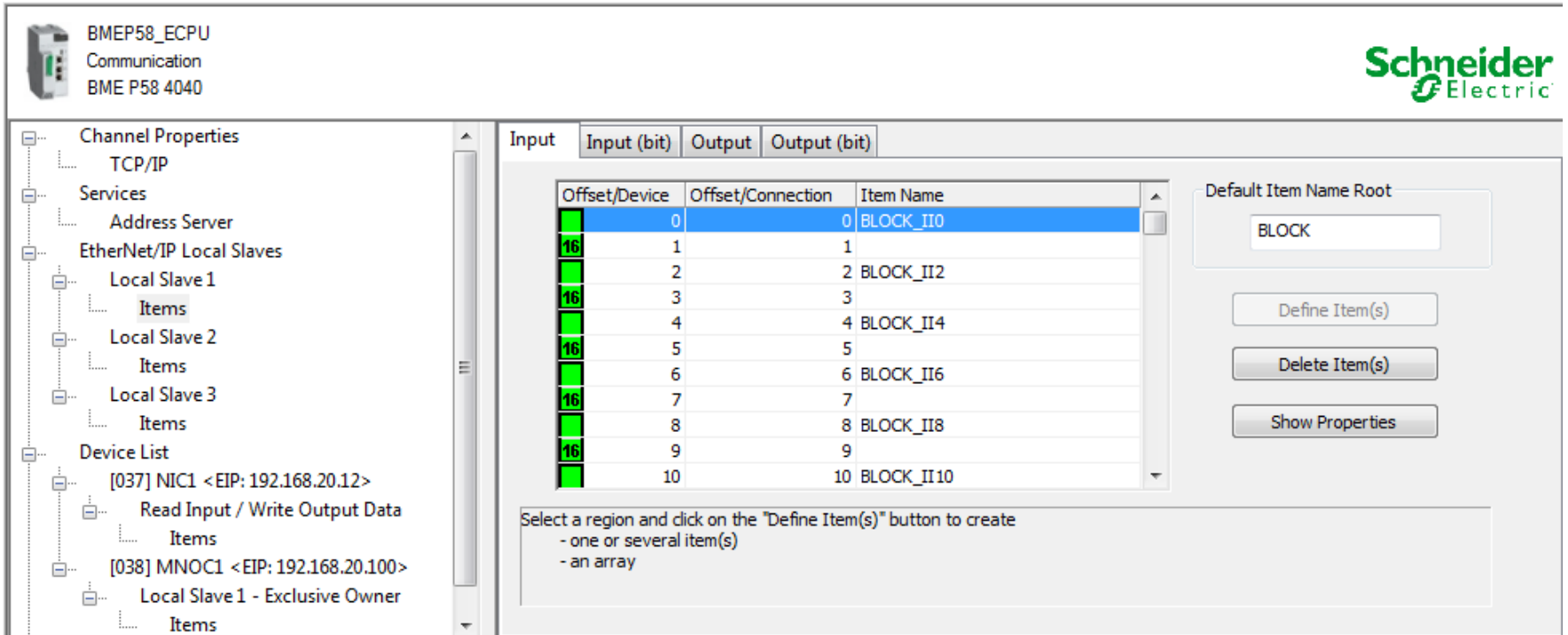
Channel Properties
TCP/IP
Services
Address Server
EtherNet/IP Local Slaves
Local Slave 1
Items
Local Slave 2
Items
Local Slave 3
Items
Device List
[037] NIC1 <EIP: 192.168.20.12>
Read Input / Write Output Data
Items
[038] MNOC1 <EIP: 192.168.20.100>
Local Slave 1 - Exclusive Owner
Items

Properties
Number: 033
Active Configuration: Enabled
Comment: For Rockwell to read and write to the M580
Connection Bit: 385

Assembly
Outputs (T -> O): 101
Outputs (T -> O) Size: 256 (1-509) Bytes
Inputs (O -> T): 102
Inputs (O -> T) Size: 256 (1-505) Bytes
Configuration: 103
Configuration Size: 0 (0-200) Words

- 256 bytes in/out translates to 128 16bit words in/out
- Total of 512 bytes transferred between devices

M580 Adapter Input/Output



The screenshot displays the Schneider Electric BMEP58_ECPU configuration interface. On the left, a tree view shows the device hierarchy: BMEP58_ECPU (Communication, BME P58 4040) with sub-items for Channel Properties (TCP/IP), Services (Address Server), EtherNet/IP Local Slaves (Local Slave 1, 2, 3, each with an Items sub-item), and Device List (NIC1, MNOC1, and Local Slave 1 - Exclusive Owner, each with an Items sub-item). The main window is titled 'Input' and contains a table with columns 'Offset/Device', 'Offset/Connection', and 'Item Name'. The table lists 11 items, all with a value of 16 in the 'Offset/Device' column, indicating 16-bit INTs. The 'Item Name' column shows 'BLOCK_II0' through 'BLOCK_II10'. To the right of the table is a 'Default Item Name Root' field set to 'BLOCK', with buttons for 'Define Item(s)', 'Delete Item(s)', and 'Show Properties'. Below the table, a text box instructs the user to 'Select a region and click on the "Define Item(s)" button to create' one or several items or an array.

Offset/Device	Offset/Connection	Item Name
0	0	BLOCK_II0
16	1	1
16	2	2 BLOCK_II2
16	3	3
16	4	4 BLOCK_II4
16	5	5
16	6	6 BLOCK_II6
16	7	7
16	8	8 BLOCK_II8
16	9	9
16	10	10 BLOCK_II10

- Set Input and Output items to 16bit INTs (Inputs shown as example)
 - Helps determines data type used in RSLogix5K
 - Not strictly necessary, but makes application easier to move INTs around instead of bytes

ControlLogix Scanner Config

The screenshot displays the ControlLogix Scanner Config interface. On the left is the 'Controller Organizer' tree, showing a hierarchy of components including 'Controller CommTest', 'Controller Tags', 'Controller Fault Handler', 'Power-Up Handler', 'Tasks', 'MainTask', 'MainProgram', 'Parameters and Local Tags', 'MainRoutine', 'Unscheduled Programs / Phases', 'Motion Groups', 'Ungrouped Axes', 'Add-On Instructions', 'Data Types', 'User-Defined', 'Strings', 'Add-On-Defined', 'Predefined', 'Module-Defined', 'Trends', 'Logical Model', 'I/O Configuration', '1756 Backplane, 1756-A4', '[0] 1756-L72 CommTest', '[1] 1756-ENBT/A Single_Port', 'Ethernet', 'ETHERNET-BRIDGE M580_System', 'CIP Bus', and '0 CIP-MODULE LS1'. The '1756-ENBT/A Single_Port' is highlighted.

The main window is titled 'Controller Tags - CommTest(controller)'. It has a 'Select Module Type' button circled in red. Below this is a 'Module Discovery' tab with a search bar and 'Clear Filters' and 'Hide Filters' buttons. The 'Module Type Category Filters' section shows 'Communication' checked. The 'Module Type Vendor Filters' section shows 'Allen-Bradley', 'Endress+Hauser', 'FANUC CORPORATION', 'FANUC Robotics America', and 'Mettler-Toledo' checked. Below these is a table of module types:

Catalog Number	Description	Vendor	Category
1788-ENBT	1788 10/100 Mbps Ethernet Bridge, Twisted-Pair Media	Allen-Bradley	Communication
1794-AENT	1794 10/100 Mbps Ethernet Adapter, Twisted-Pair Me...	Allen-Bradley	Communication
1794-AENTR	1794 10/100 Mbps Ethernet Adapter, 2-Port, Twisted...	Allen-Bradley	Communication
193-DNENCAT	Ethernet to DeviceNet Communications Auxiliary	Allen-Bradley	Communication
193-DNENCATR	Ethernet to DeviceNet Communications Auxiliary, 2-Port	Allen-Bradley	Communication
DriveLogix5730 Ethernet ...	10/100 Mbps Ethernet Port on DriveLogix5730	Allen-Bradley	Communication
E1 Plus	Electronic Overload Relay Communications Interface	Allen-Bradley	Communication
EtherNet/IP	SoftLogix5800 EtherNet/IP	Allen-Bradley	Communication
ETHERNET-BRIDGE	Generic EtherNet/IP CIP Bridge	Allen-Bradley	Communication
ETHERNET-MODULE	Generic Ethernet Module	Allen-Bradley	Communication
ILX34-AENWVG	1734 Wireless Ethernet Adapt...		
IND131 Ethernet/IP	Scale Terminal		
IND560 Ethernet/IP	Scale Terminal		

The 'ETHERNET-BRIDGE' module is circled in red. A 'Module Properties Report: Single_Port (ETHERNET-BRIDGE 1.1)' dialog box is open, showing the 'General' tab. The 'Type' is 'ETHERNET-BRIDGE Generic EtherNet/IP CIP Bridge'. The 'Parent' is 'Single_Port'. The 'Name' is 'M580_System'. The 'Description' is empty. The 'Address / Host Name' section has 'IP Address' selected with the value '192 . 168 . 20 . 10'. The 'Status' is 'Offline'. The 'OK', 'Cancel', 'Apply', and 'Help' buttons are at the bottom.

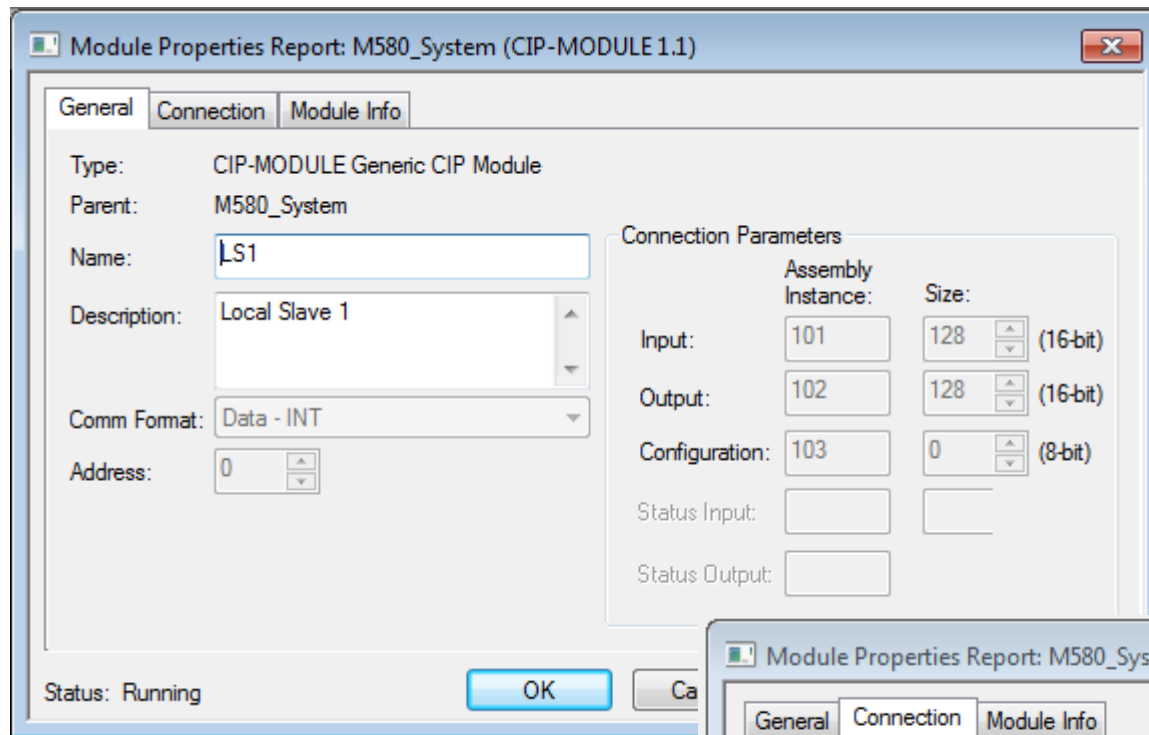
- By adding a CIP bridge it allows you to read multiple local slave instances from the M580's embedded Ethernet port.
- Enter its IP address

ControlLogix Scanner Config

The screenshot displays the 'ControlLogix Scanner Config' interface. On the left, the 'Controller Organizer' tree shows the project structure, with 'I/O Configuration' expanded to show '1756-ENBT/A Single_Port'. On the right, the 'Select Module Type' dialog is open, showing a list of module types. The 'ETHERNET-MODULE' row is highlighted with a red circle. The dialog also includes a search bar, filters, and a table of module details.

Catalog Number	Description	Vendor	Category
193-DNENCATR	Ethernet to DeviceNet Communications Auxiliary, 2-Port	Allen-Bradley	Communication
Drivelogix5730 Ethernet ...	10/100 Mbps Ethernet Port on DriveLogix5730	Allen-Bradley	Communication
E1 Plus	Electronic Overload Relay Communications Interface	Allen-Bradley	Communication
EtherNet/IP	SoftLogix5800 EtherNet/IP	Allen-Bradley	Communication
ETHERNET-BRIDGE	Generic EtherNet/IP CIP Bridge	Allen-Bradley	Communication
ETHERNET-MODULE	Generic Ethernet Module	Allen-Bradley	Communication
ILX34-AENWG	1734 Wireless Ethernet Adapter, Twisted-Pair Media	Prosoft Technol...	Communication
IND131 Ethernet/IP	Scale Terminal	Mettler-Toledo	Communication
IND560 Ethernet/IP	Scale Terminal	Mettler-Toledo	Communication
IND780 Ethernet/IP	Scale Terminal	Mettler-Toledo	Communication
PSSCENA	Ethernet Adapter, Twisted-Pair Media	Parker Hannifin ...	Communication
Stratix 8000	6 Port Managed Switch	Allen-Bradley	Communication
Stratix 8000	10 Port Managed Switch	Allen-Bradley	Communication
Stratix 8000	14 Port Managed Switch	Allen-Bradley	Communication
Stratix 8000	18 Port Managed Switch	Allen-Bradley	Communication
Stratix 8000	22 Port Managed Switch	Allen-Bradley	Communication
Stratix 8000	26 Port Managed Switch	Allen-Bradley	Communication

- At this point you can add multiple Generic Ethernet modules, one for every local slave instance.



Comm format comes from design decision to use 16bit INTs on M580

Instances are from M580 DTM configuration page or user manual

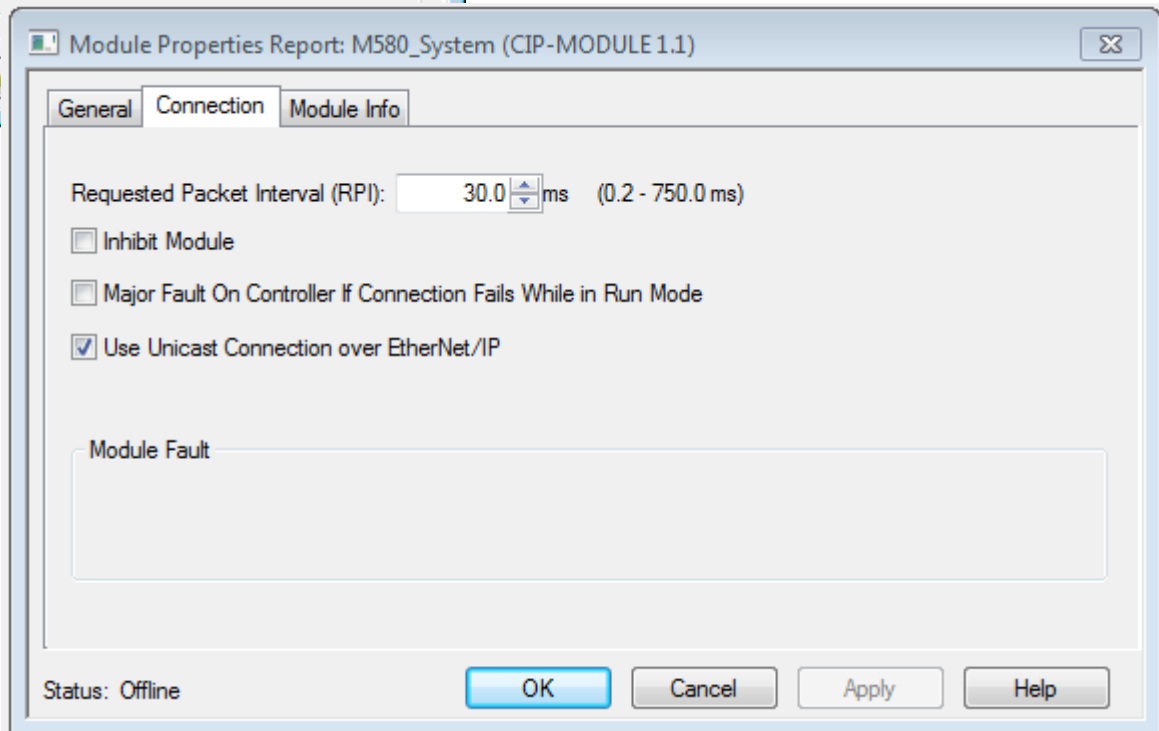
Size is from number of bytes selected in M580 DTM config (256 bytes each instance, so 128 16bit INTs)

M580 will publish data at 30ms (T→O) (typically multicast unless option is checked for unicast, i.e. this example)

ControlLogix will publish data at 30ms (O→T) (unicast)

Unicast is typically recommended unless multicast is required. Scanner determines what type to use.

ControlLogix does NOT have a slave adapter, so bidirectional Class1 communications have to be set up this way.



M580's Local Slave input/output area

BMEP58_ECPU_LS1		T_BMEP58_EC...	Local Slave 1
Freshness	1	BOOL	Global Freshness
Freshness_1	1	BOOL	Freshness of Object
Inputs		T_BMEP58_EC...	Input Variables
BLOCK_I10	11	INT	
BLOCK_I12	12	INT	
BLOCK_I14	13	INT	
BLOCK_I16	14	INT	
BLOCK_I18	15	INT	
BLOCK_I110	16	INT	
BLOCK_I112	17	INT	
BLOCK_I114	18	INT	
BLOCK_I116	19	INT	
BLOCK_I118	20	INT	
BLOCK_I120	21	INT	
BLOCK_I122	22	INT	
BLOCK_I124	23	INT	
BLOCK_I126	24	INT	
BLOCK_I128	25	INT	
BLOCK_I130	26	INT	
Free0		ARRAY[0..223] ...	Unused Variable

Local Slave input area is the area that ControlLogix writes to.

Originator to Target
CLogix to M580

BMEP58_ECPU_LS1		T_BMEP58_EC...	Local Slave 1
Freshness	1	BOOL	Global Freshness
Freshness_1	1	BOOL	Freshness of Object
Inputs		T_BMEP58_EC...	Input Variables
Outputs		T_BMEP58_EC...	Output Variables
BLOCK_Q10	100	INT	
BLOCK_Q12	101	INT	
BLOCK_Q14	102	INT	
BLOCK_Q16	103	INT	
BLOCK_Q18	105	INT	
BLOCK_Q110	106	INT	
Free1		ARRAY[0..243] ...	Unused Variable

Local Slave output area is the data that we publish or write to ControlLogix.

Target to Originator
M580 to ControlLogix

ControlLogix's input/output area

Controller Tags - CommTest(controller)

Scope: CommTest Show: All Tags

Name	Value	Force Mask	Style	Data Type
- M580_System:0:O.Data	{ ... }	{ ... }	Decimal	INT[128]
+ M580_System:0:O.Data[0]	11		Decimal	INT
+ M580_System:0:O.Data[1]	12		Decimal	INT
+ M580_System:0:O.Data[2]	13		Decimal	INT
+ M580_System:0:O.Data[3]	14		Decimal	INT
+ M580_System:0:O.Data[4]	15		Decimal	INT
+ M580_System:0:O.Data[5]	16		Decimal	INT
+ M580_System:0:O.Data[6]	17	16	Decimal	INT
+ M580_System:0:O.Data[7]	18		Decimal	INT
+ M580_System:0:O.Data[8]	19		Decimal	INT
+ M580_System:0:O.Data[9]	20		Decimal	INT
+ M580_System:0:O.Data[10]	21		Decimal	INT
+ M580_System:0:O.Data[11]	22		Decimal	INT
+ M580_System:0:O.Data[12]	23		Decimal	INT
+ M580_System:0:O.Data[13]	24		Decimal	INT

Output of originator written to input of target.

Input of originator being updated by Target's output

Controller Tags - CommTest(controller)

Scope: CommTest Show: All Tags

Name	Value	Force Mask	Style	Data Type
- M580_System:0:I.Data	{ ... }	{ ... }	Decimal	INT[128]
+ M580_System:0:I.Data[0]	100		Decimal	INT
+ M580_System:0:I.Data[1]	101		Decimal	INT
+ M580_System:0:I.Data[2]	102		Decimal	INT
+ M580_System:0:I.Data[3]	103		Decimal	INT
+ M580_System:0:I.Data[4]	105		Decimal	INT
+ M580_System:0:I.Data[5]	106		Decimal	INT
+ M580_System:0:I.Data[6]	0		Decimal	INT

Unity Diagnostics for Healthy Connection

 BMEP58_ECPU
Communication
BME P58 4040

 BMEP58_ECPU

-  Local Slave 1
-  [037] NIC1 <EIP: 192.168.20.12>
 -  Read Input / Write Output Data
-  [038] MNOC1 <EIP: 192.168.20.100>
 -  Local Slave 1 - Exclusive Owner

Logging

Local Slave Diagnostic I/O Values

	Output	0
	Counter	
	Frame Error	0
	Time-Out	0
	Refused	0
	Production	30517
	Consumption	30510
	Production Byte	7837952
	Consumption Byte	7993620
	Theoretical Packets per second	66
	Real Packets per second	70
	Diagnostic	
	Production Connection ID	0x46168E3A
	Consumption Connection ID	0x00BD1401
	O -> T API	30000
	T -> O API	30000
	O -> T RPI	30000

Unity Diagnostics for IO Values

The screenshot displays the Unity Diagnostics software interface. On the left, a tree view shows the device hierarchy: BMEP58_ECPU (Communication, BME P58 4040) is expanded, showing Local Slave 1, [037] NIC1 <EIP: 192.168.20.12>, Read Input / Write Output Data, [038] MNOC1 <EIP: 192.168.20.100>, and Local Slave 1 - Exclusive Owner. The main panel is titled 'Local Slave Diagnostic' and 'I/O Values'. It features a 'Refresh every 500 msec' checkbox. The 'Input' section shows a list of 16 rows of hex values (e.g., 0B 00 0C 00 0D 00 0E 00;) and a 'Length' of 256 bytes. The 'Output' section shows a list of 6 rows of hex values (e.g., 15 00 16 00 17 00 18 00;) and a 'Length' of 256 bytes. Both sections have a 'Status' of 0.

BMEP58_ECPU
Communication
BME P58 4040

BMEP58_ECPU
Local Slave 1
[037] NIC1 <EIP: 192.168.20.12>
Read Input / Write Output Data
[038] MNOC1 <EIP: 192.168.20.100>
Local Slave 1 - Exclusive Owner
Logging

Local Slave Diagnostic I/O Values

☒ Refresh every 500 msec

Input

0B 00 0C 00 0D 00 0E 00;
0F 00 10 00 11 00 12 00;
13 00 14 00 15 00 16 00;
17 00 18 00 19 00 1A 00;
1B 00 1C 00 1D 00 1E 00;
00 00 00 00 00 00 00 00;
00 00 00 00 00 00 00 00;
00 00 00 00 00 00 00 00;
00 00 00 00 00 00 00 00;
00 00 00 00 00 00 00 00;
00 00 00 00 00 00 00 00;

Length 256 bytes

Status 0

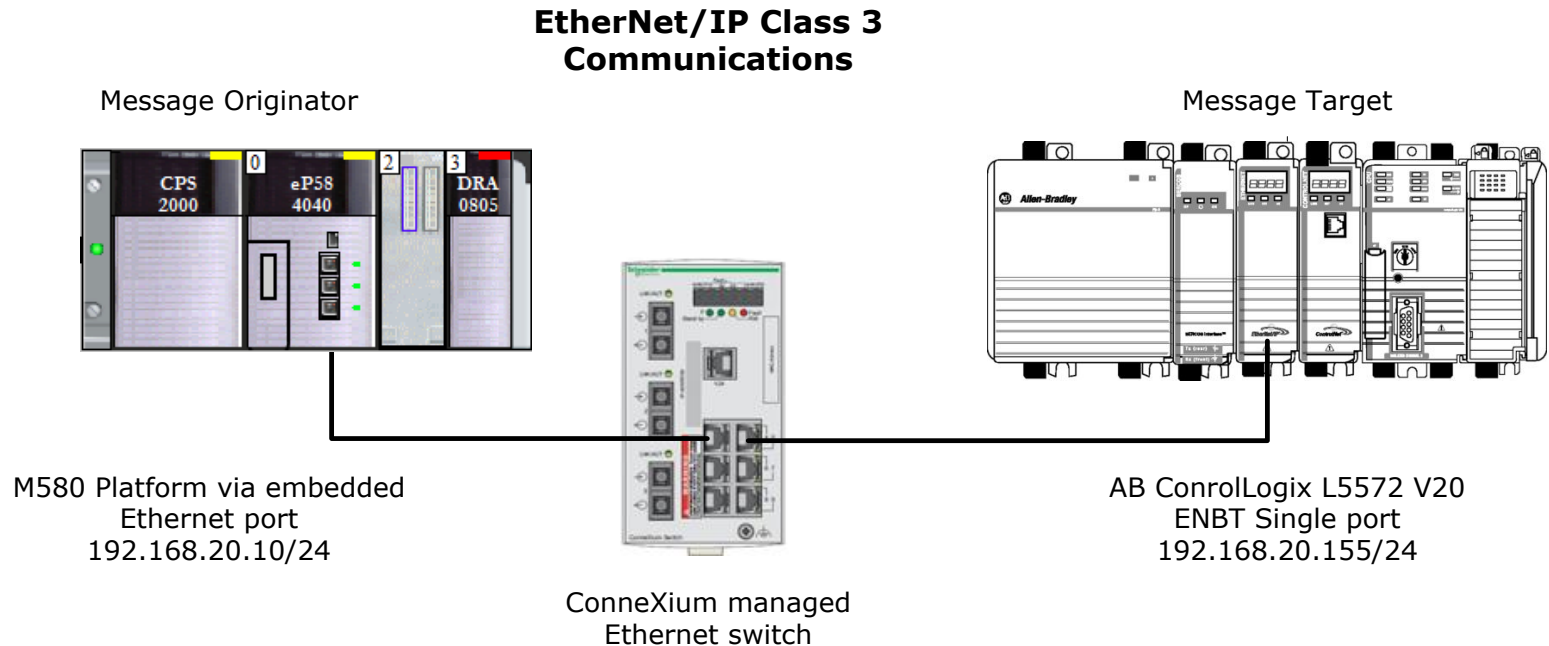
Output

15 00 16 00 17 00 18 00;
19 00 1A 00 00 00 00 00;
00 00 00 00 00 00 00 00;
00 00 00 00 00 00 00 00;
00 00 00 00 00 00 00 00;
00 00 00 00 00 00 00 00;

Length 256 bytes

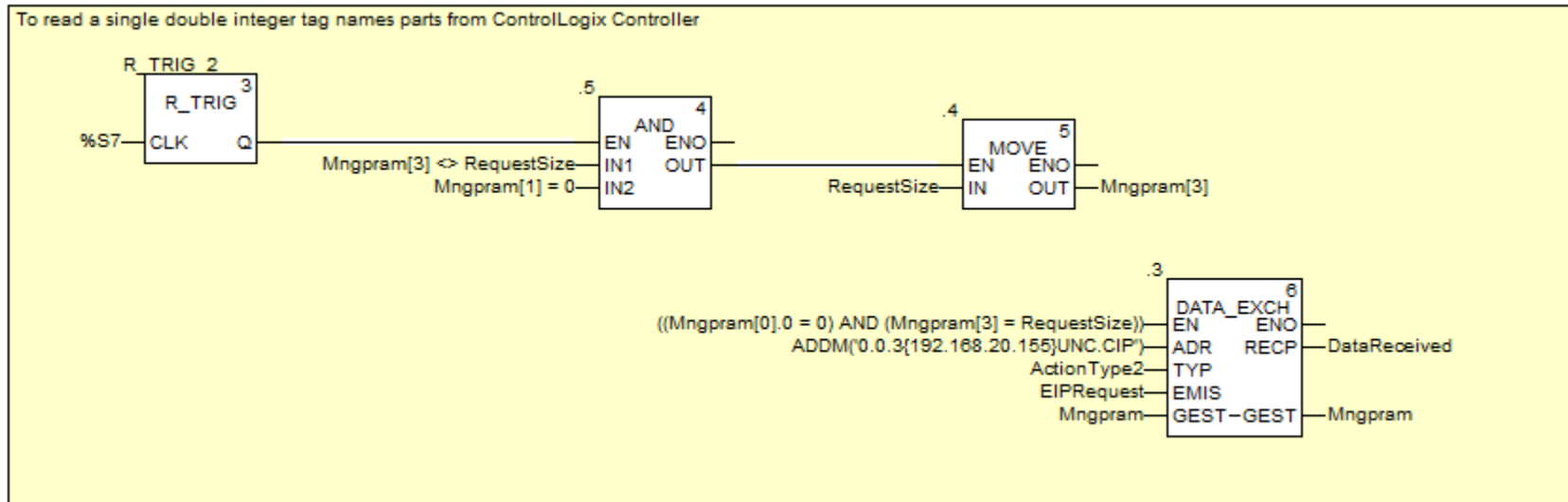
Status 0

EtherNet/IP Class 3 communication, Explicit messaging



- Explicit Messages can be originated from the M580 or the ControlLogix PACs. This example demonstrates M580 as the message originator.
- In M580 platform DATA_EXCH FB is used to read a tag from ControlLogix

Explicit Message – DATA_EXCH



The online help can be used for information on the DATA_EXCH function block. However the programmer must be familiar with the CIP request in its entirety. The EMIS is an array that contains the CIP request.

The next page outlines the request in detail. The request example is to read a single tag named *parts*.

CIP Request

Variable EIPRequest is assigned to EMIS pin on the DATA_EXCH function block. The request is divided to two parts the CIP header and CIP request. EIPRequest[0] through EIPRequest [2] are part of the CIP header.

Indicating service request is an unconnected, with the provided path to the connection manager.

```

Common Industrial Protocol
  Service: Unknown Service (0x52) (Request)
    0... .... = Request/Response: Request (0x00)
    .101 0010 = Service: Unknown (0x52)
    Request Path Size: 2 (words)
  Request Path: Connection Manager, Instance: 0x01
    Path Segment: 0x20 (8-Bit Class Segment)
    Path Segment: 0x24 (8-Bit Instance Segment)
  
```

EIPRequest	ARRAY[0....		
EIPRequest[0]	INT	16#0252	HiByte=02 (Request path size); LowByte=52 (Service request-unconnected send)
EIPRequest[1]	INT	16#0620	HiByte=06 (Class, request path to connection manager); LowByte=20 (Segment, path semgment)
EIPRequest[2]	INT	16#0124	HiByte=01 (Instance); LowByte=24 (Instance Segment)
EIPRequest[3]	INT	16#1007	HiByte=10 (Timeout ticks); LowByte=[Priority=0, Tick time=7]
EIPRequest[4]	INT	16#000C	Message request size
EIPRequest[5]	INT	16#044C	HiByte=04 (Request path size); LowByte=4C (Service request: reading by ASCII)
EIPRequest[6]	INT	16#0591	HiByte= 05 (Data size); LowByte=91 (Segment type: Data Segment (reading tag name))
EIPRequest[7]	INT	16#6170	symbol: parts
EIPRequest[8]	INT	16#7472	symbol: parts
EIPRequest[9]	INT	16#0073	symbol: parts
EIPRequest[10]	INT	16#0001	Data: 0001
EIPRequest[11]	INT	16#0001	HiByte = 00 (reserved); LowByte=01 (Route Path Size)
EIPRequest[12]	INT	16#0001	HiByte=00 (Route path address); LowByte=01 (Route path port)

CIP Header

CIP Request

CIP Request

```

CIP Connection Manager
├── Service: Unconnected Send (Request)
├── Command Specific Data
│   ├── ...0 .... = Priority: 0
│   ├── ... 0111 = Tick time: 7
│   ├── Time-out ticks: 16
│   ├── Actual Time Out: 2048ms
│   └── Message Request Size: 12
├── Message Request
│   └── Common Industrial Protocol
│       ├── Service: Unknown Service (0x4c) (Request)
│       │   ├── ... .. = Request/Response: Request (0x00)
│       │   ├── .100 1100 = Service: Unknown (0x4c)
│       │   └── Request Path Size: 4 (words)
│       ├── Request Path: parts
│       │   └── Path Segment: 0x91 (ANSI Extended Symbol Segment)
│       └── CIP Class Generic
│           ├── Route Path Size: 1 (words)
│           ├── Reserved: 0x00
│           └── Route Path: Port: 1, Address: 0
│               ├── Path Segment: 0x01 (Port Segment)
│               │   ├── 000. .... = Path Segment Type: Port Segment (0)
│               │   ├── ...0 .... = Extended Link Address: False
│               │   └── .... 0001 = Port: 1
│               └── Port Segment
│                   └── Link Address: 0

```

The CIP request is EIPRequest[3] through EIPRequest [12].

Provides information about the specific request, i.e. reading by ASCII, name of the tag, message size, timeout ...

EIPRequest	ARRAY[0....		
EIPRequest[0]	INT	16#0252	HiByte=02 (Request path size); LowByte=52 (Service request-unconnected send)
EIPRequest[1]	INT	16#0620	HiByte=06 (Class, request path to connection manager); LowByte=20 (Segment, path segment)
EIPRequest[2]	INT	16#0124	HiByte=01 (Instance); LowByte=24 (Instance Segment)
EIPRequest[3]	INT	16#1007	HiByte=10 (Timeout ticks); LowByte=[Priority=0, Tick time=7]
EIPRequest[4]	INT	16#000C	Message request size
EIPRequest[5]	INT	16#044C	HiByte=04 (Request path size); LowByte=4C (Service request: reading by ASCII)
EIPRequest[6]	INT	16#0591	HiByte= 05 (Data size); LowByte=91 (Segment type: Data Segment (reading tag name))
EIPRequest[7]	INT	16#6170	symbol: parts
EIPRequest[8]	INT	16#7472	symbol: parts
EIPRequest[9]	INT	16#0073	symbol: parts
EIPRequest[10]	INT	16#0001	Data: 0001
EIPRequest[11]	INT	16#0001	HiByte = 00 (reserved); LowByte=01 (Route Path Size)
EIPRequest[12]	INT	16#0001	HiByte=00 (Route path address); LowByte=01 (Route path port)

CIP Header

CIP Request