

# **NETWORK SYSTEM**

## **Devicenet Network**

Devicenet Introduction

Configure Devicenet Network

Control devices via Devicenet

Back up on DeviceNet

## **Ethernet Network**

Ethernet Introduction

Configure Ethernet Network

Control devices via Ethernet

## **ControlNet Network**

ControlNet Overview

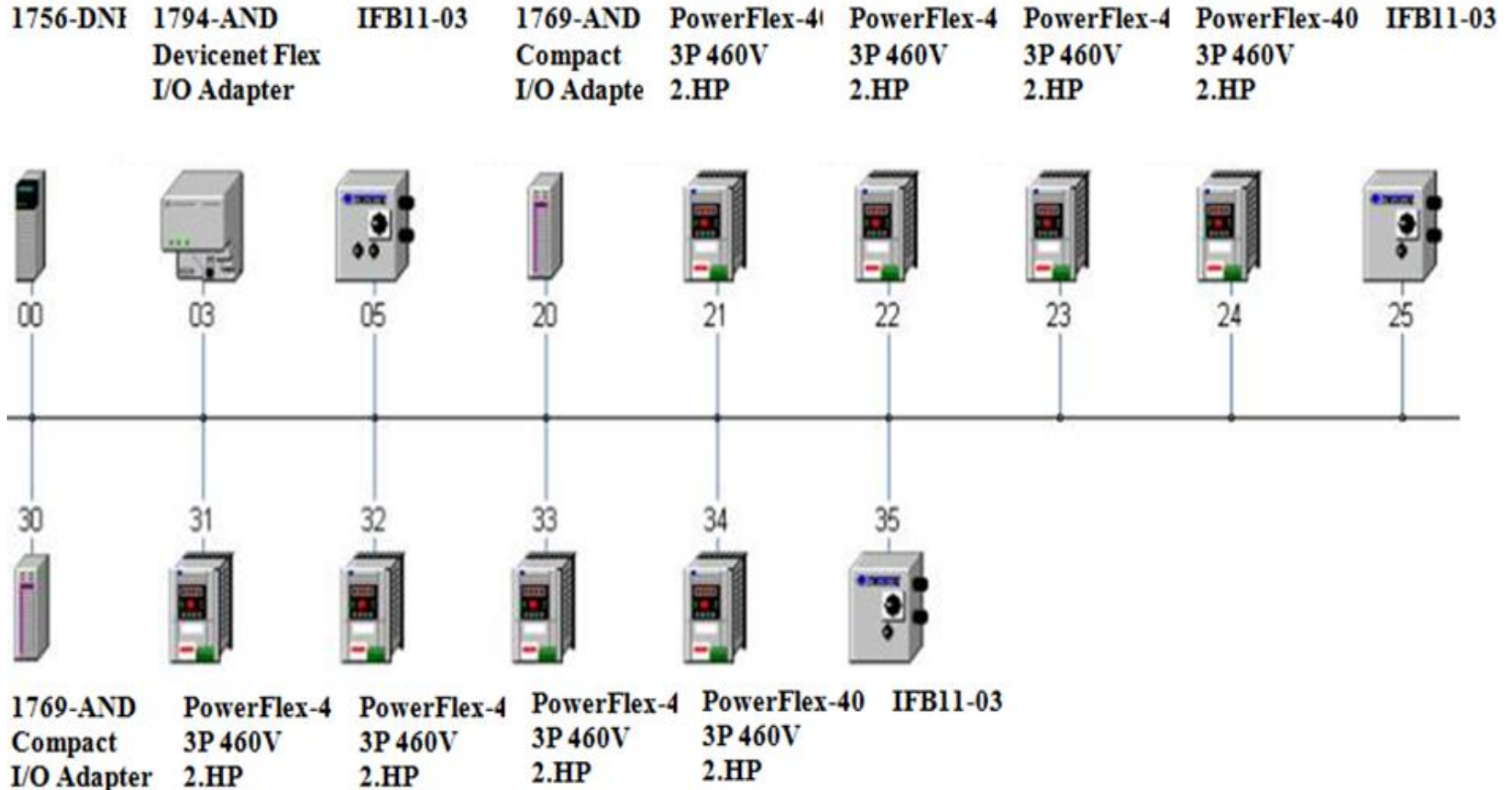
Configure ControlNet Network

Control devices via ControlNet

**HCM.UTE 3-2014**

# DEVICENET NETWORK

## DeviceNet Overview

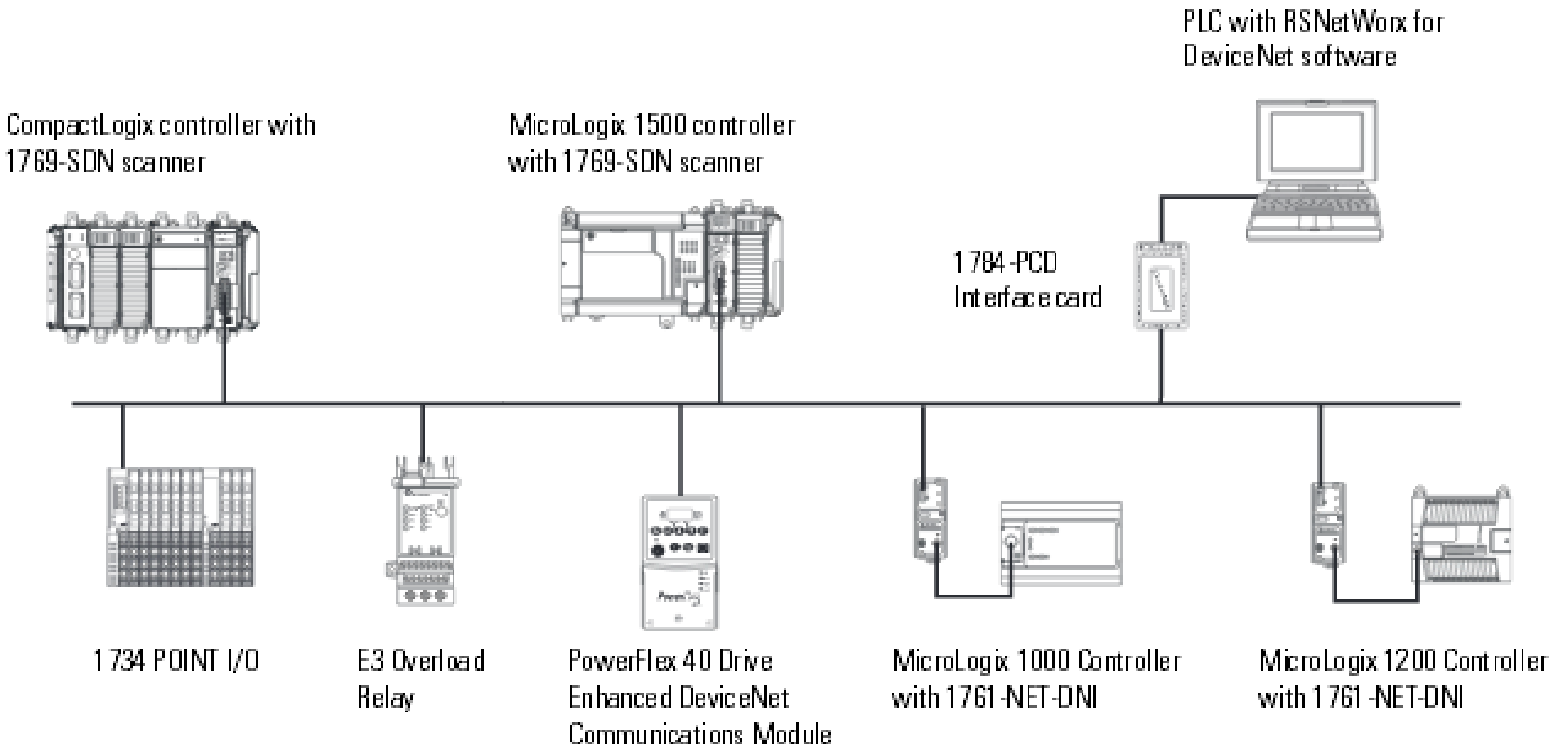


*Up to 64 nodes(0-63) in DeviceNet Network*

*There are one Master and Slaves*

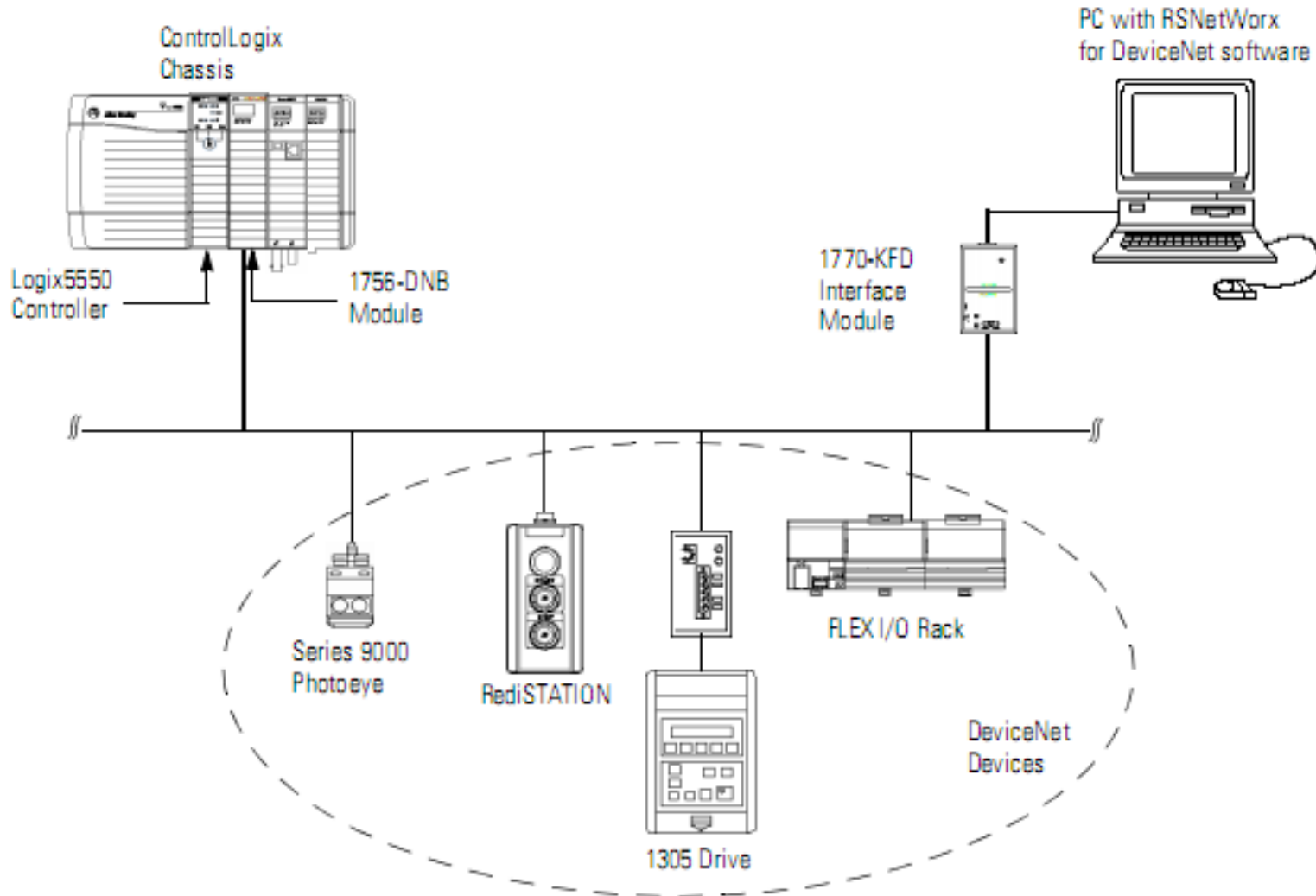
# DEVICENET NETWORK

## Communication between DeviceNet Devices and CompactLogix via 1769 SDN Module



# DEVICENET NETWORK

## Communication between DeviceNet Devices and CompactLogix via 1756 DNB Module



# DEVICENET NETWORK

## DeviceNet Features and Functionality

Network size	Up to 64 nodes
Network Length	125Kbps: 500m 250Kbps: 250m 500Kbps: 100m
Data Packets	0-8 byte
Bus Topology	Linear (trunkline/dropline); power and signal on the same network cable
Bus Addressing	Peer to Peer, MultiMaster hoặc Master/Slave
System Features	Removal and replacement of devices from the network under power

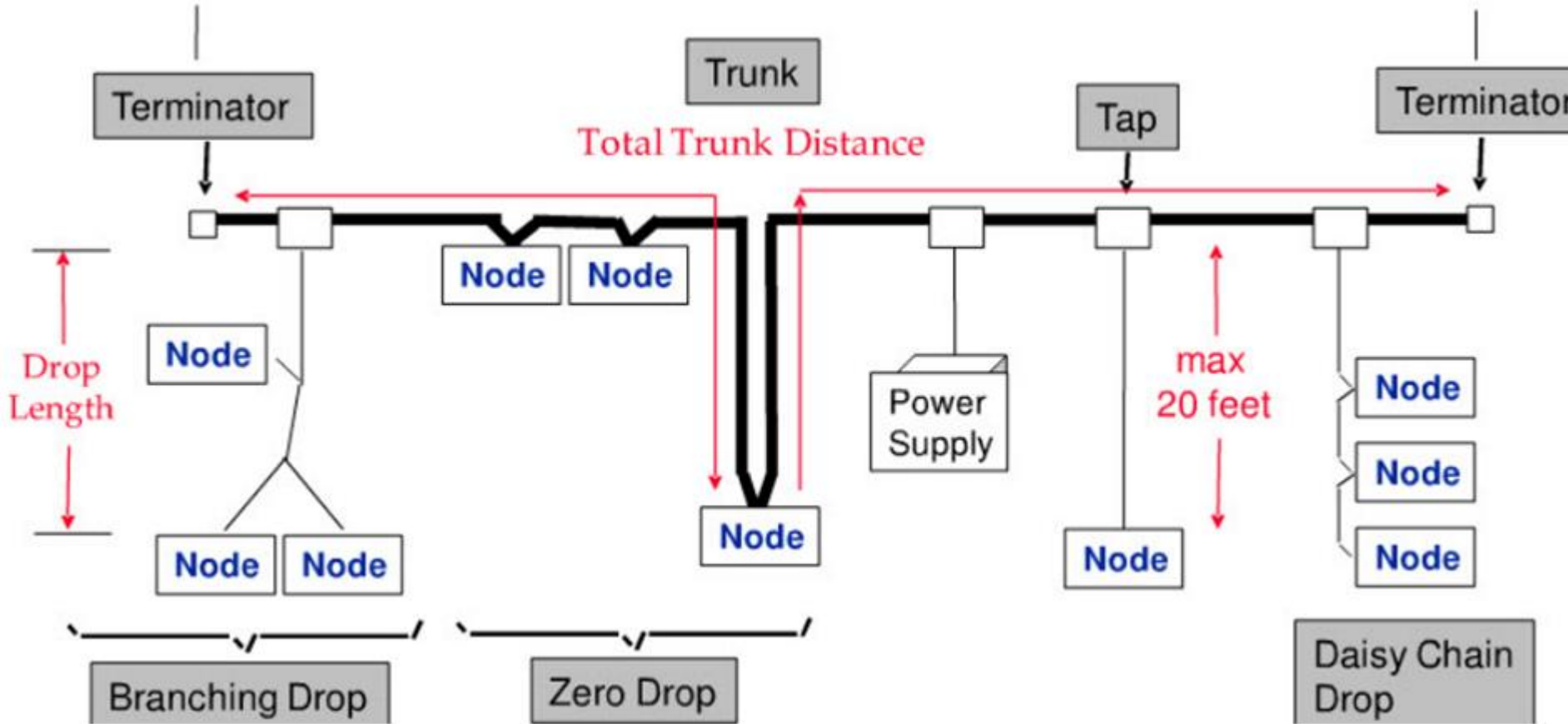
# DEVICENET NETWORK

## Assign an address to each Devices

Give this device	This address	Notes
Scanner	0	If you have multiple scanners, give them the lowest addresses in sequence (0, 1...).
Any device on the network except the scanner	1...61	<ul style="list-style-type: none"><li>• Give the lower addresses to devices with 15 bytes or more of input or output data.</li><li>• Gaps between addresses are OK and have no effect on system performance. If you are uncertain of the final lay-out of your system, leave gaps between addresses. This gives you some flexibility as you develop your system.</li></ul>
Computer interface to the network	62	If you connect a computer directly to the DeviceNet network, use address 62 for the computer. <ul style="list-style-type: none"><li>• Many computer interface devices use this address as their default.</li><li>• The 1784-U2DN device can connect a computer directly to a DeviceNet network.</li></ul>
No device	63	Always leave address 63 open. Out of the box, most DeviceNet devices are preset for address 63. <ul style="list-style-type: none"><li>• Some devices have no switches or push button to set the address. They require software, such as RSNetWorx for DeviceNet software to change the address. This means that you must first place it on the network at its preset address of 63 before you can change the address.</li><li>• If another device is already using address 63, there will be an address conflict and you won't be able to communicate with the newly connected device.</li><li>• Leaving address 63 open makes it possible to configure a new device.</li><li>• The auto-address recovery feature also requires address 63 to be open.</li></ul>

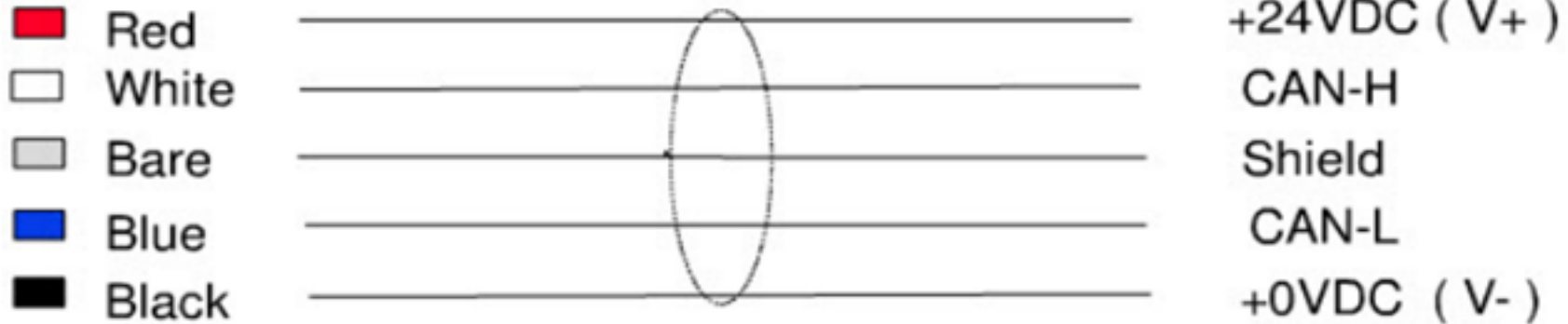
# DEVICENET NETWORK

## Bus Topology



# DEVICENET NETWORK

## DeviceNet Cable



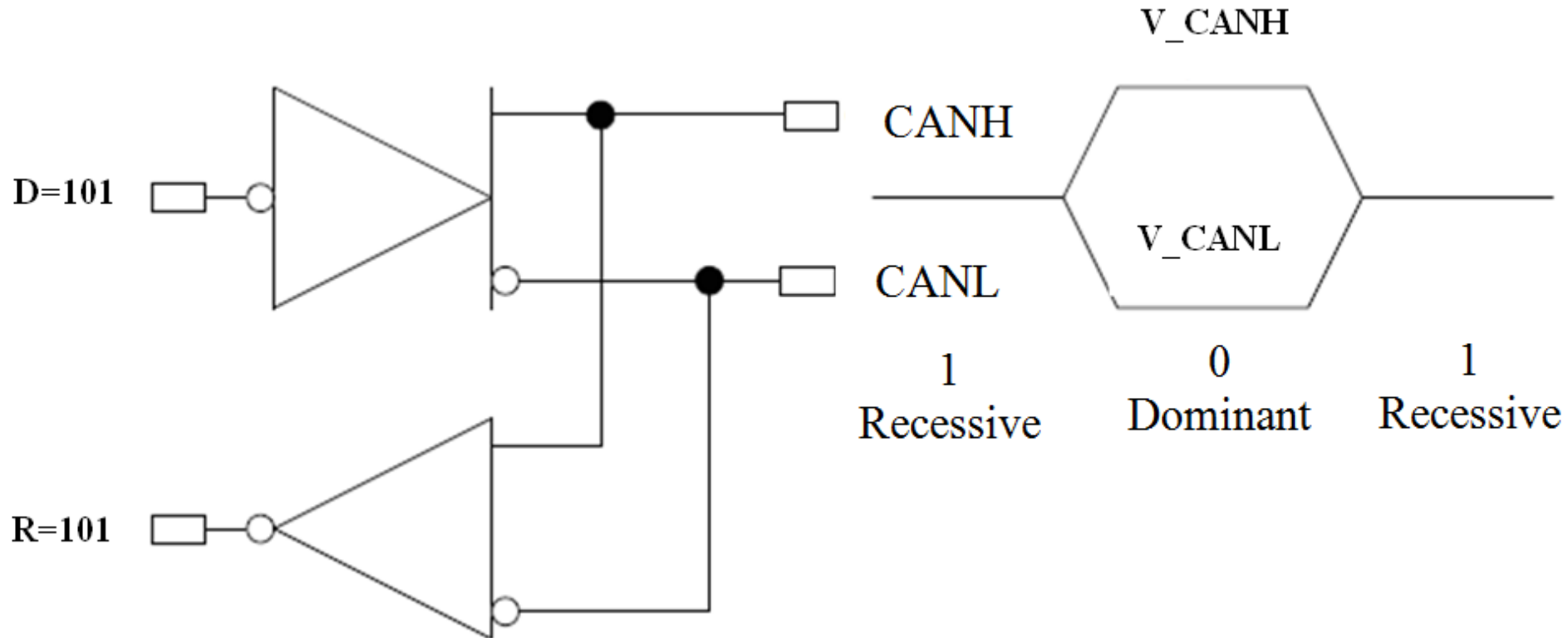
## DeviceNet Cable specifications

- 5 conductors, 1 pair of 24volts DC Power, 1 pair for can communication, one shield.
- Maximum current for 24 volts power DC is 3 amps.



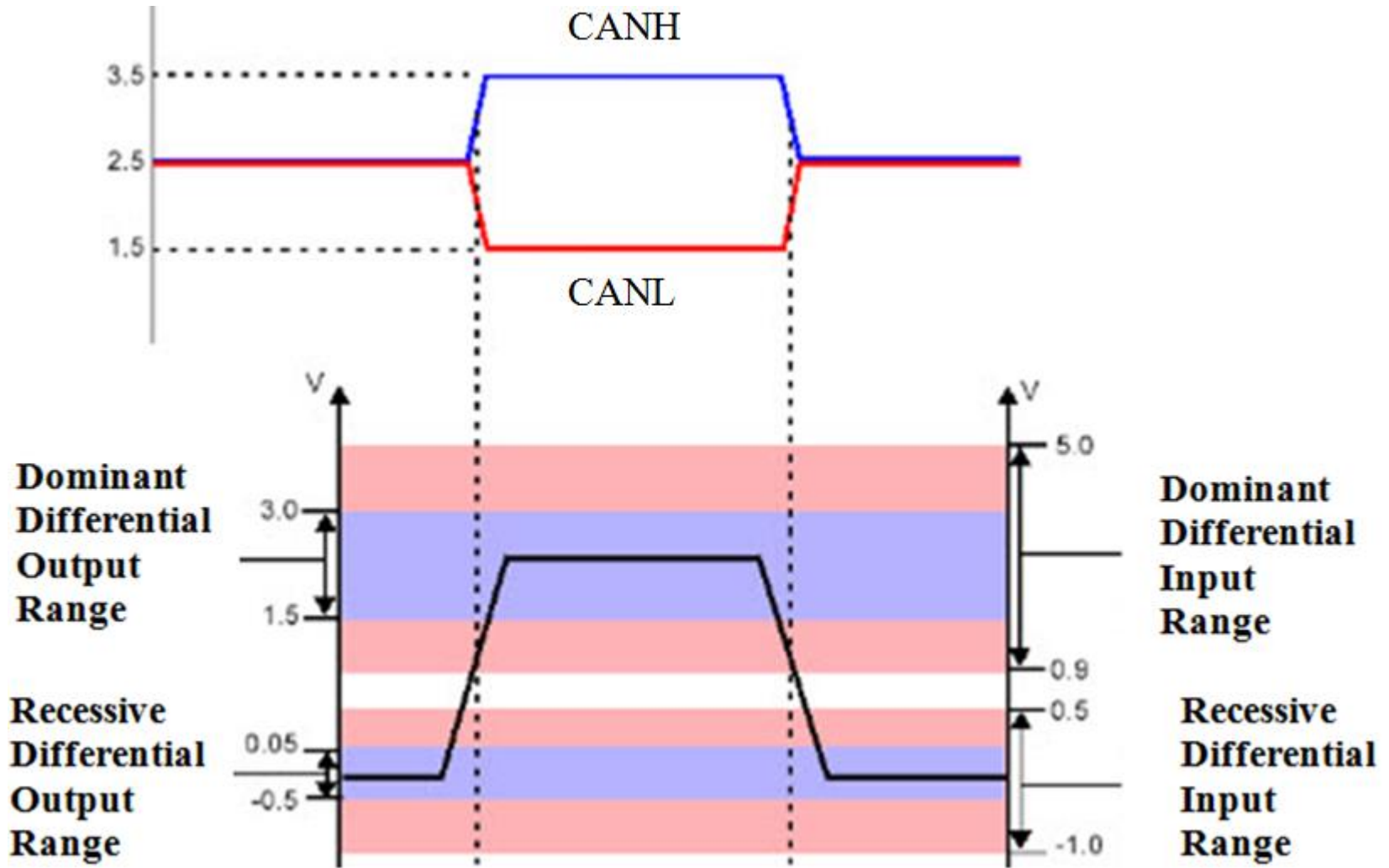
# DEVICENET NETWORK

## The Inverted Logic of DeviceNet Bus



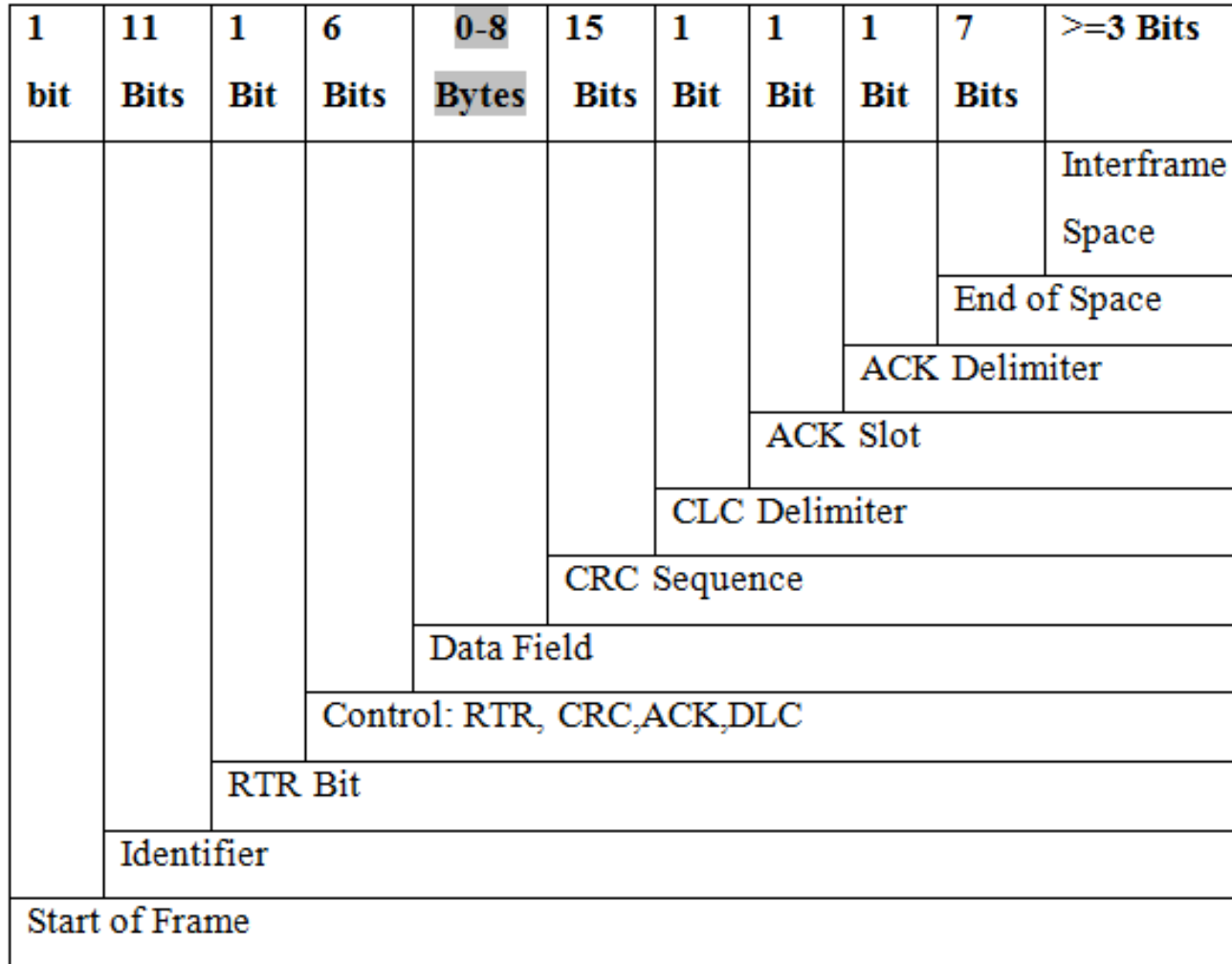
# DEVICENET NETWORK

## The Inverted Logic of a DeviceNet Bus



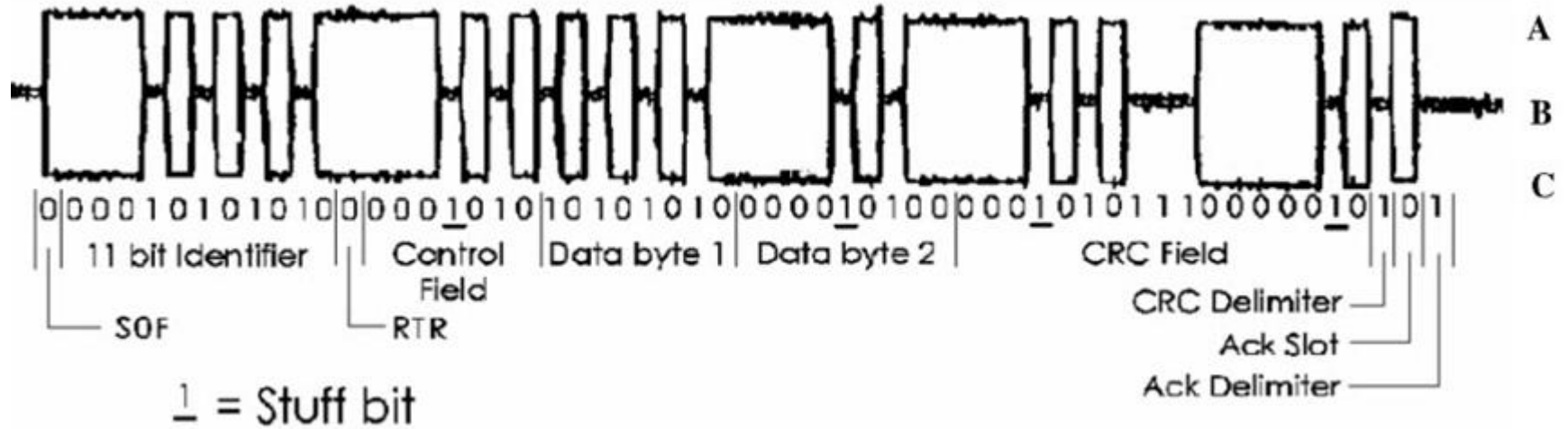
# DEVICENET NETWORK

## DeviceNet Data Frame



# DEVICENET NETWORK

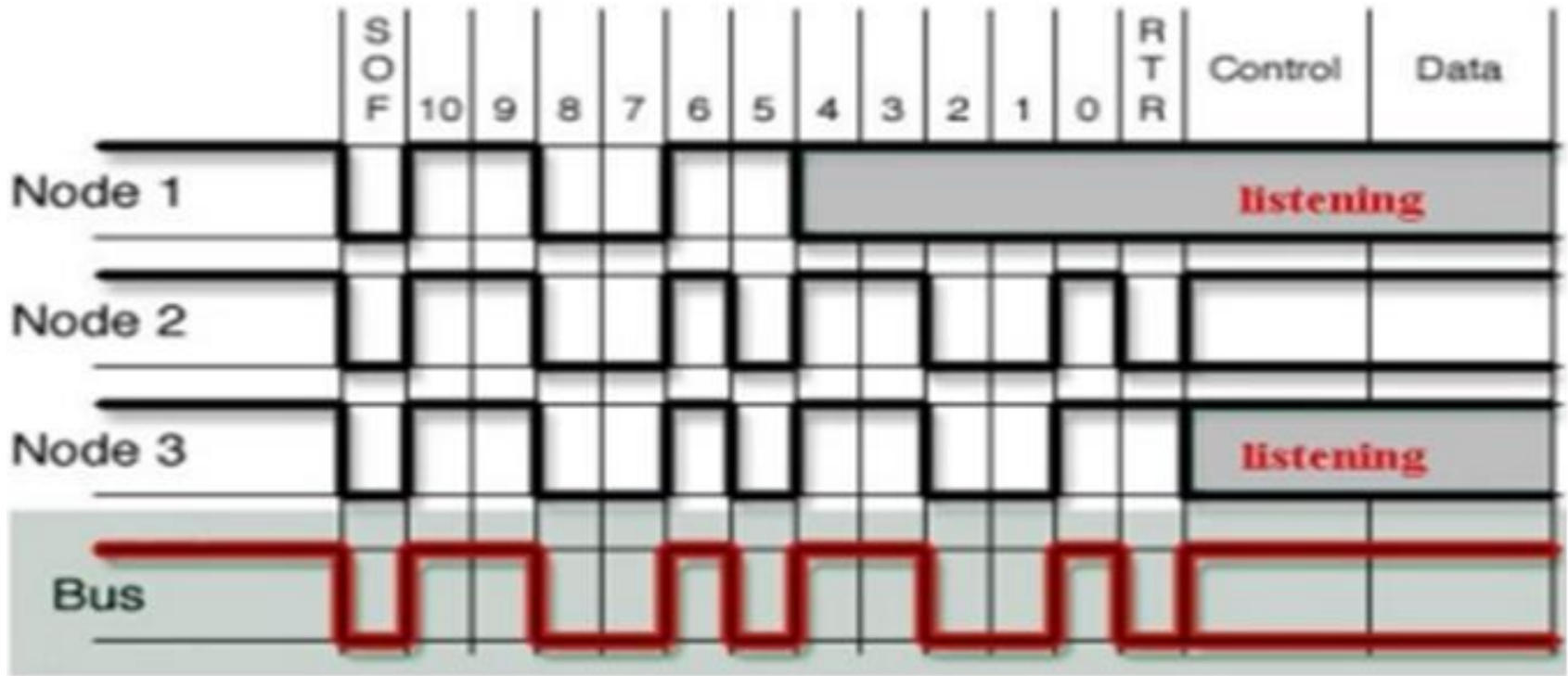
## Logic State of Data Frame



*Identifier field and RTR bit are used to identify which node has right to access bus*

# DEVICENET NETWORK

## Arbitration of DeviceNet Bus



If a node transmitting a recessive bit receives a dominant bit while sending **the arbitration field**, it stops transmitting

The winner of all nodes transmitting simultaneously is the one with the **lowest-numbered 11-bit identifier**

# DEVICENET NETWORK MESSAGES

DeviceNet supports two types of messaging: **Explicit Messaging** and **I/O Messaging**

**Explicit Messages:** Used between two devices for configuration and diagnostic data transfer. They are low priority and not time-critical.

**I/O Messages** is time-critical and have high priority

## Basic Format of Explicit Messages

Destination	Service	Class	Instance	Attribute	Data
node address	code	ID	ID	ID	

**Destination Node Address:** The node address of the Unit that is sending the explicit messages (commands)

**Service Code, Class ID, Instance ID, Attribute ID:** The parameters used for specifying the command, processing object, and processing content.

# EXPLICIT MESSAGES

## Service Code, Class ID, Instance ID, Attribute ID of Devices

<b>Service Code</b>	<b>Class ID</b>	<b>Instance ID</b>	<b>Attribute ID</b>	<b>Description</b>
<b>0E Hex</b>	01	01	01	Get Vendor ID
<b>0E Hex</b>	01	01	02	Get Device Type
<b>0E Hex</b>	01	01	03	Get Product Code
<b>0E Hex</b>	01	01	04	Get Revision
<b>0E Hex</b>	01	01	05	Get Status
<b>0E Hex</b>	01	01	06	Get Serial Number
<b>0E Hex</b>	01	01	07	Get Product Name
<b>0E Hex</b>	03	01	01	Get MAC ID
<b>0E Hex</b>	03	01	02	Get Baudrate
<b>10 Hex</b>				Set MAC ID
<b>10 Hex</b>				Set Baudrate
<b>10 Hex</b>	<b>90</b>	<b>01</b>	<b>12</b>	<b>Set Heartbeat to a device</b>
<b>0E Hex</b>	<b>90</b>	<b>01</b>	<b>12</b>	<b>Read Heartbeat from a device</b>
<b>05 Hex</b>				Reset Device

If we know Service, Class, Instance and Attribute of Devices, we can access any data of devices.

# EXPLICIT MESSAGES

Setting heartbeat for Device using Service Code, Class Code, Instance Code and Attribute Code



**Message Configuration - Set\_Heartbeat**

Configuration | Communication | Tag

Message Type: CIP Generic

Service Type: Set Attribute Single      Source Element: Configure\_Heartbeat

Source Length: 2 (Bytes)

Service Code: 10 (Hex)      Class: 90 (Hex)      Destination: [ ]

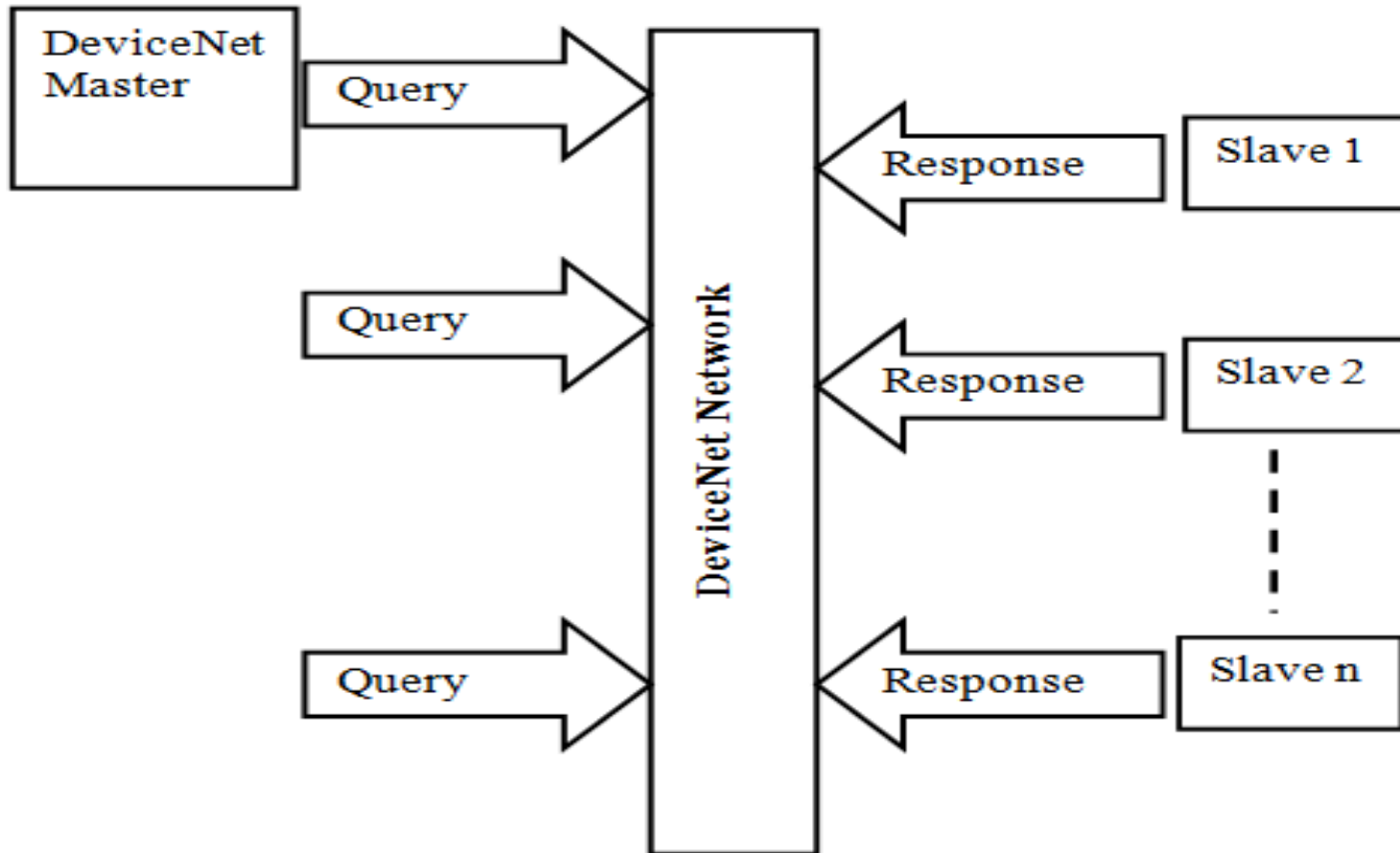
Instance: 1      Attribute: 12 (Hex)

New Tag...



# I/O MESSAGES

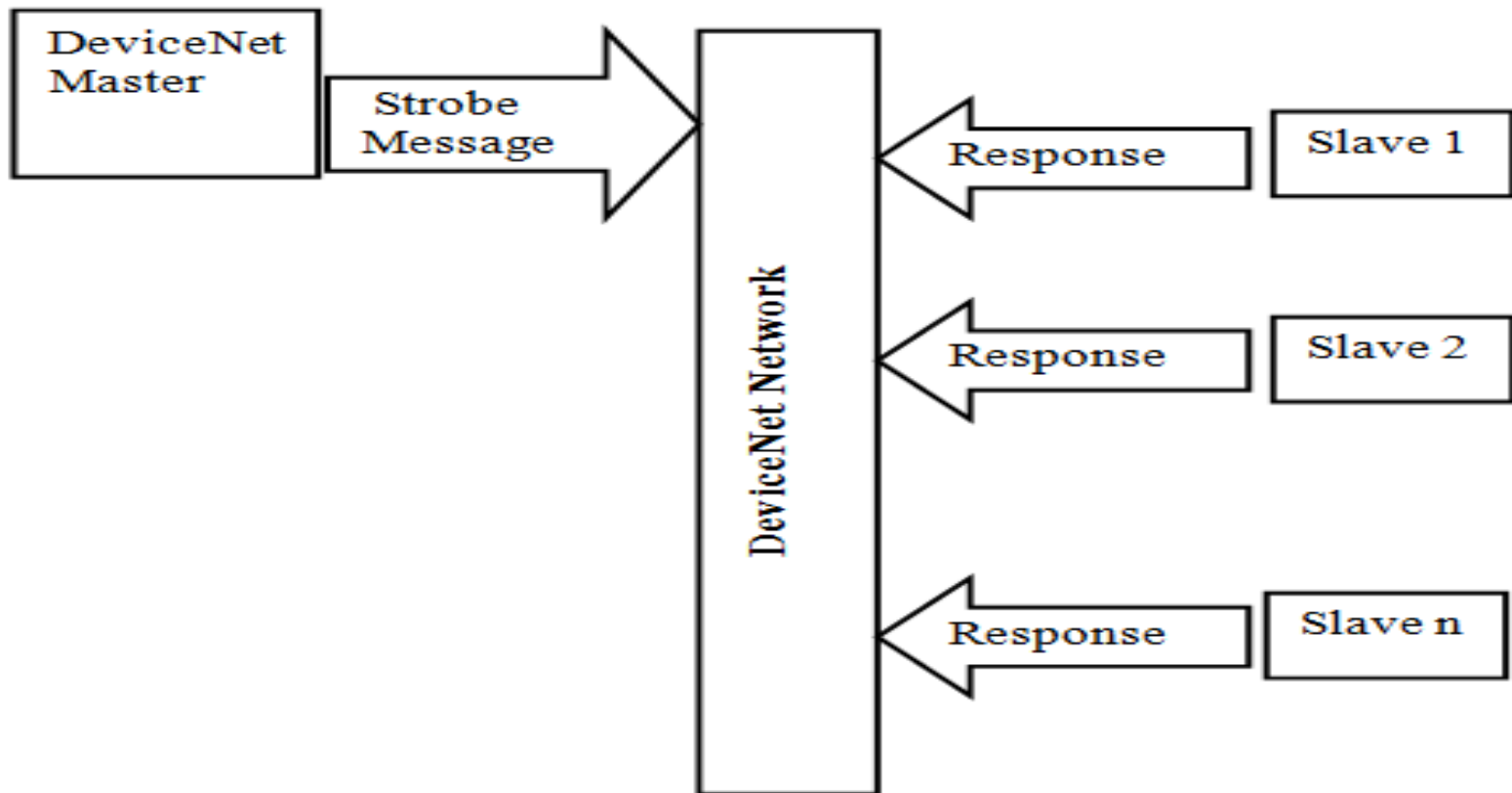
**Polled I/O Messages:** Master Scanner automatically sends a message containing outputs to each slave with a connection configured for polling. The slave sends back a response containing input data. Polling therefore requires 2 messages to update the I/O data for each polled device



*Polling is the most accurate but least efficient method of updating I/O data*

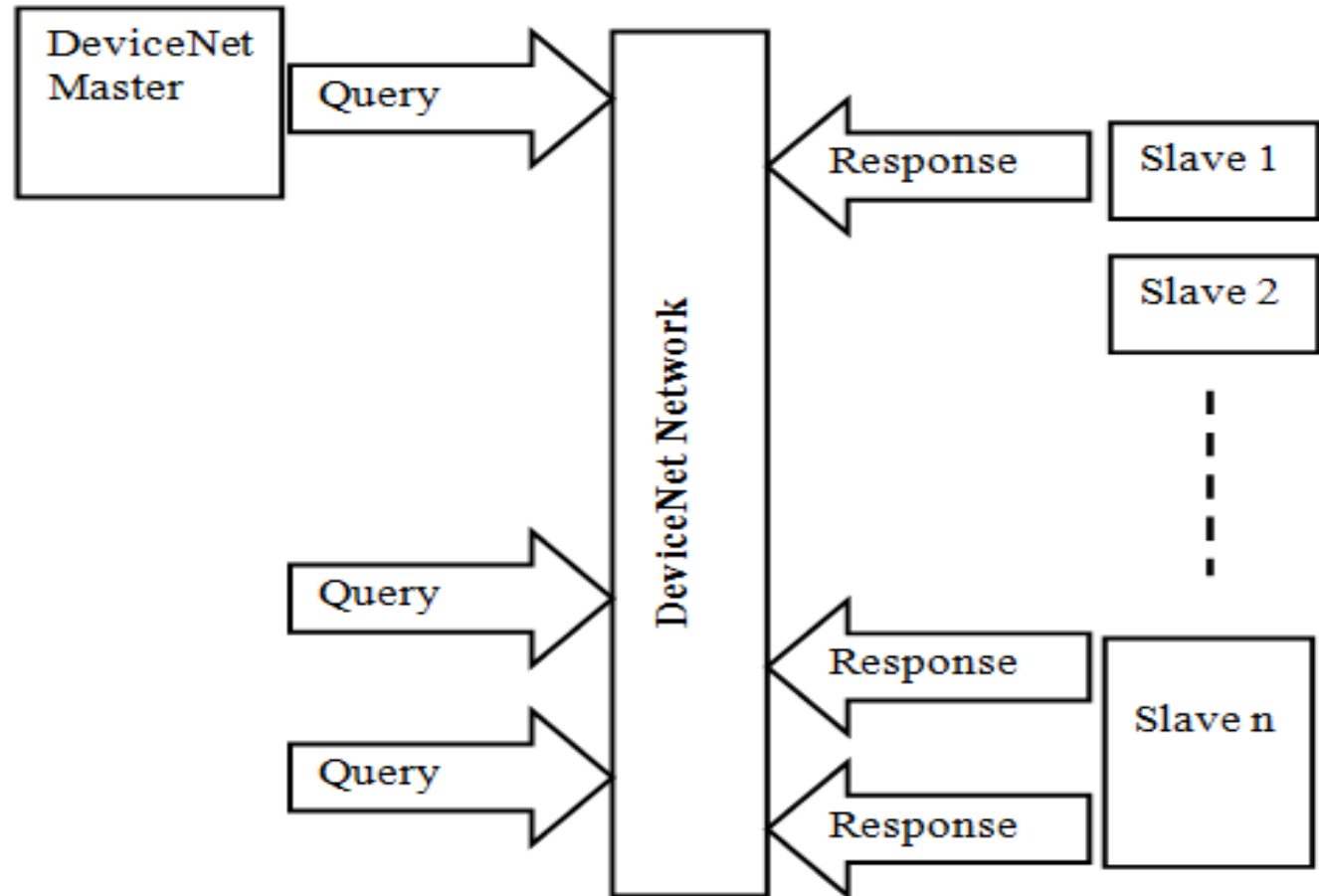
# I/O MESSAGE

**Strobed I/O Messages:** The master produces a single Strobe request message that is consumed by all devices with a connection configured for strobing, requesting their current status. This occurs at the rate selected using the Scan Interval parameter of the DeviceNet Master Module



# I/O MESSAGES

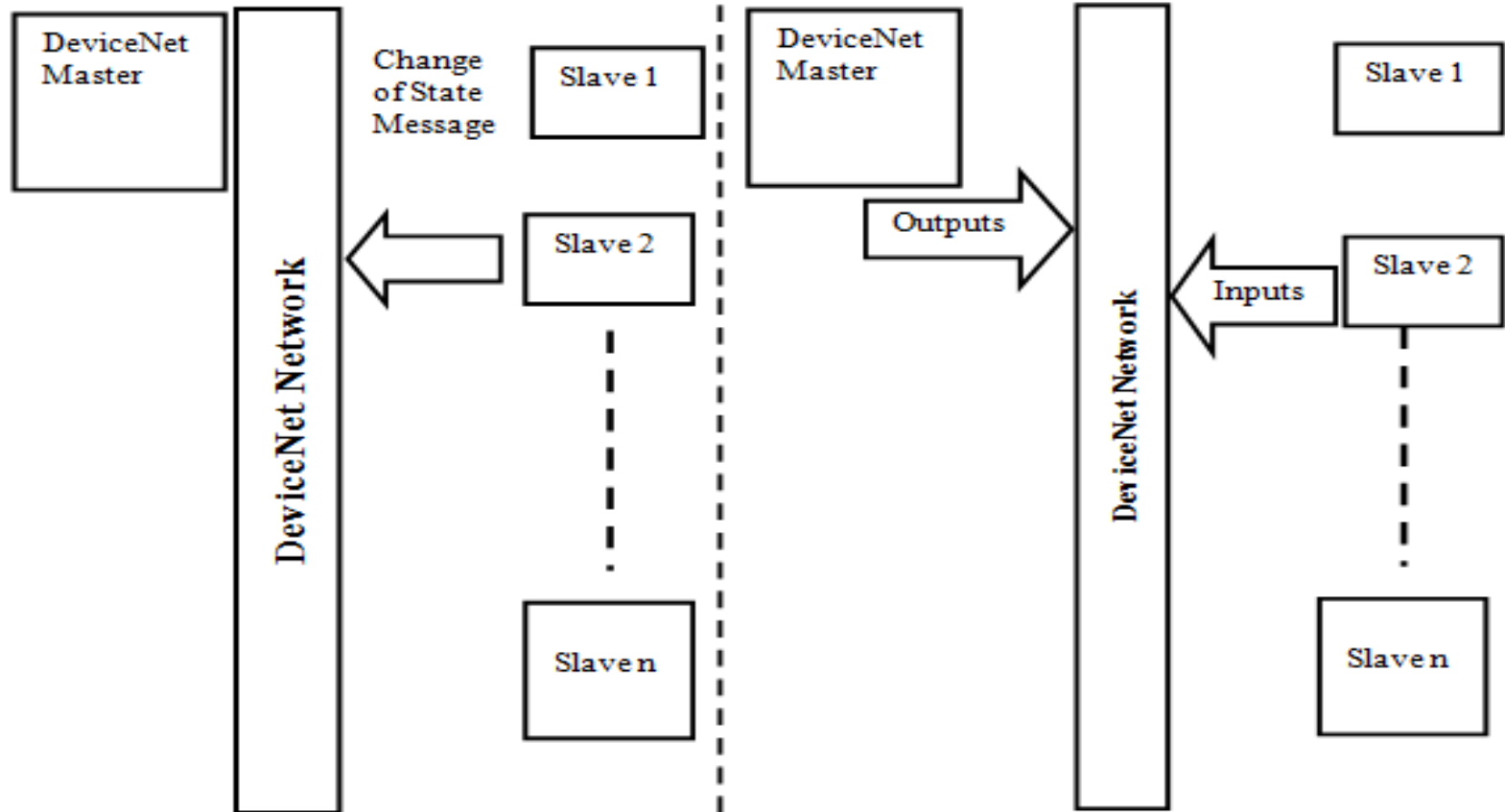
**Cyclic I/O Messages:** The Master Scanner automatically sends a message containing outputs to a device with a connection configured for Cyclic update. The device sends back a response containing input data



Unlike Polled messaging, Cyclic messaging can use a different interval as configured for each slave

# I/O MESSAGE

**COS I/O Messages:** A connection configured for **Change-of-State** (COS) I/O Messaging is activated only when the device sends a message to the master, reporting a change of status. The master then sends an output message to the device and the device responds with its input data



Devices Produce Change of State Message

Master sends Outputs, Devices Replies

# I/O MESSAGE

## Selecting I/O message to transfer between Master and Slave

The diagram shows three modules connected to a common bus:

- 1769-SDN Scanner Module (ID 00)
- PowerFlex 700S 2 208V 4.8A (ID 03)
- 1794-ADN DeviceNet Flex I/O Adapter (ID 15)

The software interface shows the configuration for the selected device (03, PowerFlex 700S 2 208V 4.8A):

**1769-SDN Scanner Module**

Available Devices: [Empty]

Scanlist:

- 03, PowerFlex 700S 2 20...
- 15, 1794-ADN DeviceNet...

**Edit I/O Parameters : 03, PowerFlex 700S 2 208V 4.8A**

Strobed:

- Input Size: 0 Bytes
- Use Output Bit:

Polled:

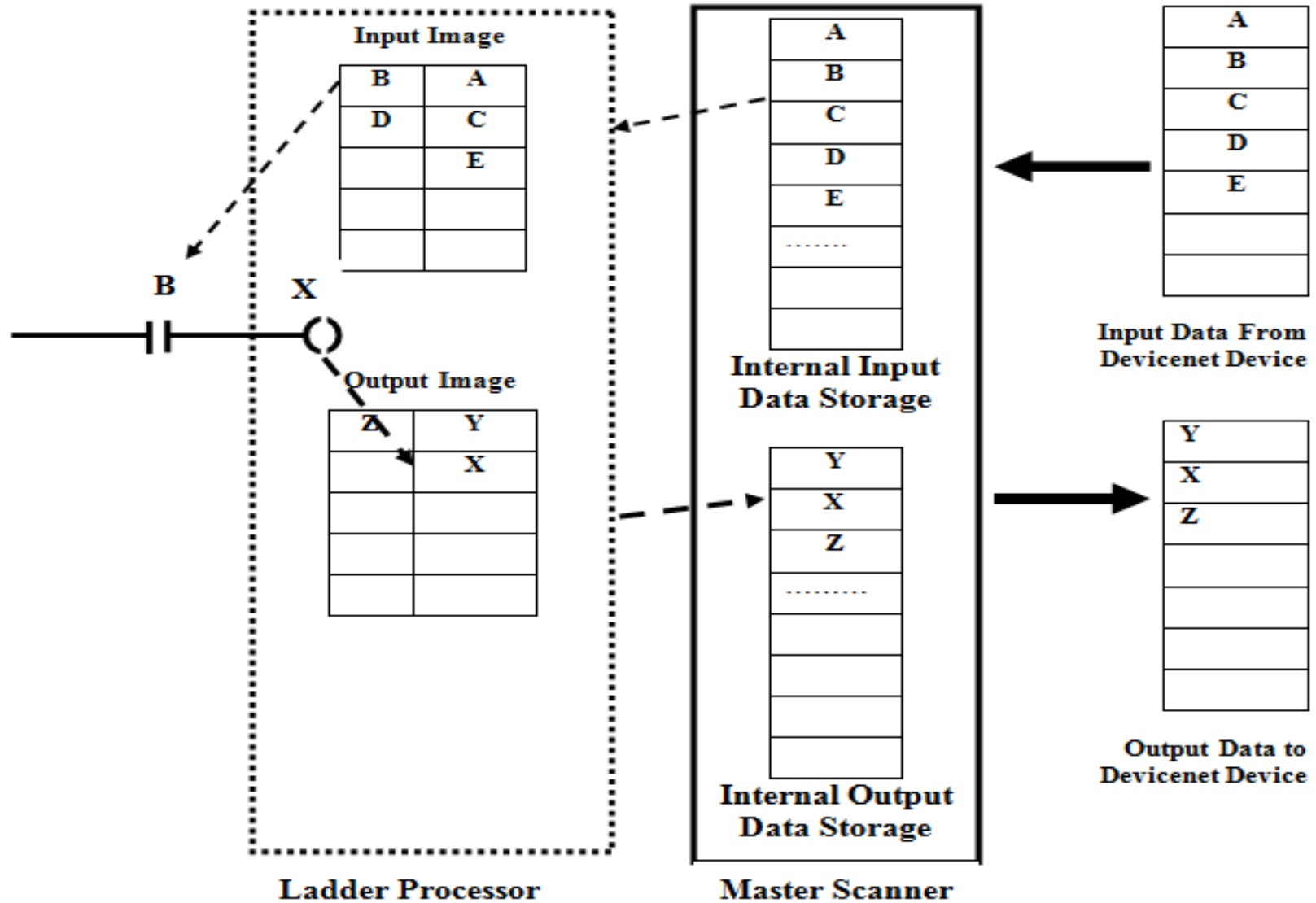
- Input Size: 8 Bytes
- Output Size: 8 Bytes
- Poll Rate: Every Scan

Change of State / Cyclic:

- Change of State  Cyclic
- Input Size: 8 Bytes
- Output Size: 0 Bytes
- Heartbeat Rate: 250 msec

Buttons: OK, Cancel, Restore I/O Sizes, Advanced...

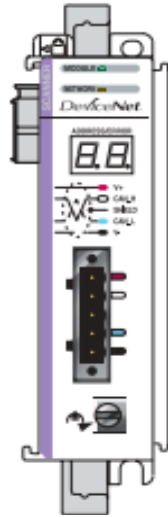
# MASTER SCANNER DATA FLOW



# MASTER SCANNER

CompactLogix™ Scanner 1769-SDN

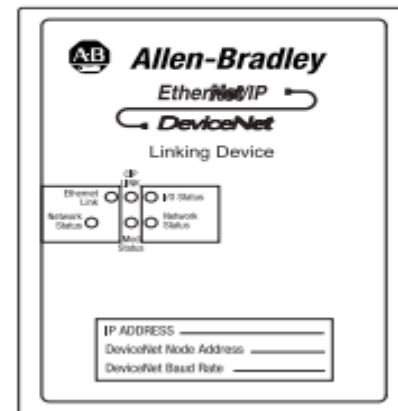
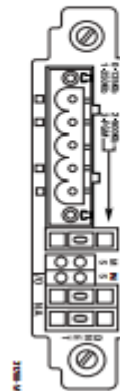
ControlLogix® Scanner 1756-DNB



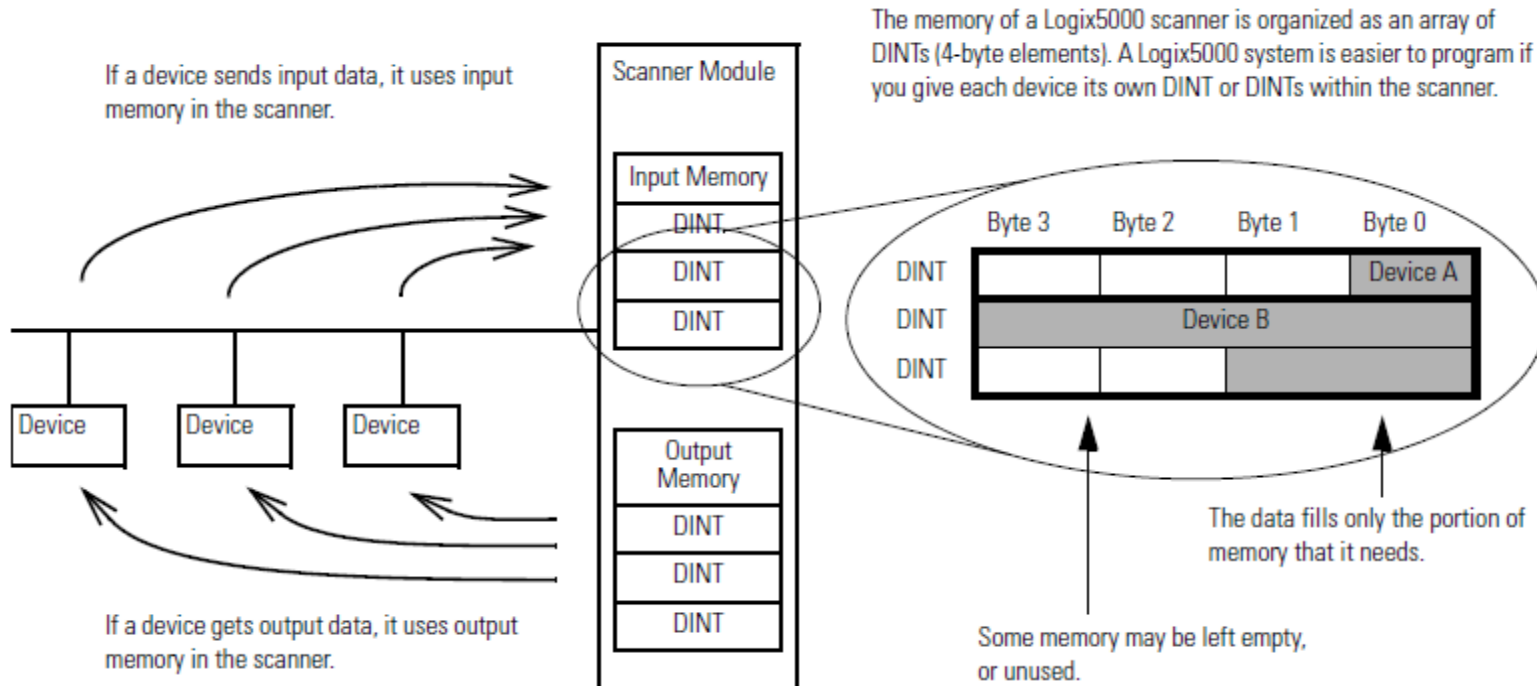
DriveLogix™ and FlexLogix™  
Communication Card 1788-DNB0



EtherNet/IP to DeviceNet Linking  
Device 1788-EN2DN



# SCANNER MEMORY



## Check the I/O Limits of the Scanner

Once you tally the input and output data for the network, make sure it is within the limits of the scanner. If they exceed the limits, use multiple scanners.

Scanner	Maximum input data (DINTs)	Maximum output data (DINTs)
1756-DNB	124	123
1769-SDN	90	90
1788-CN2DN	124	123
1788-EN2DN	124	123
1788-DNBO	124	123



# DEVICENET SLAVES

**POINT™ I/O Interface**  
1734-PDN



**POINT™ I/O Adapter**  
1734-ADN and 1734-ADNX



**POINT™ Block I/O Module**  
1734D



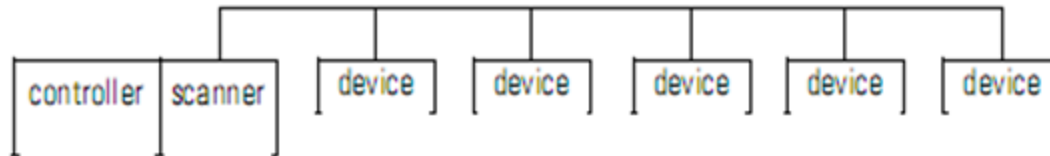
**FLEX™ I/O Adapter**  
1794-ADN



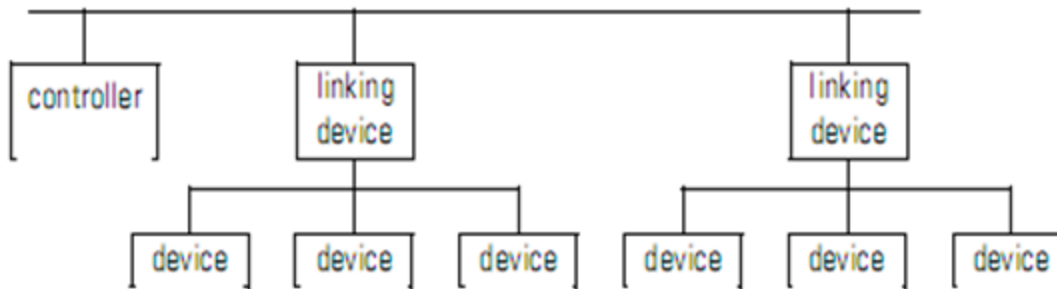
# DEVICENET NETWORK CONFIGURATION

- 1. Choose whether to use a single network or several distributed networks.

- single network



- several smaller distributed networks (subnets)



- 2. Choose a scanner.

**If you are using:**

**And the main network is:**

**Use this scanner:**

single network



Corresponding scanner for your controller

subnets

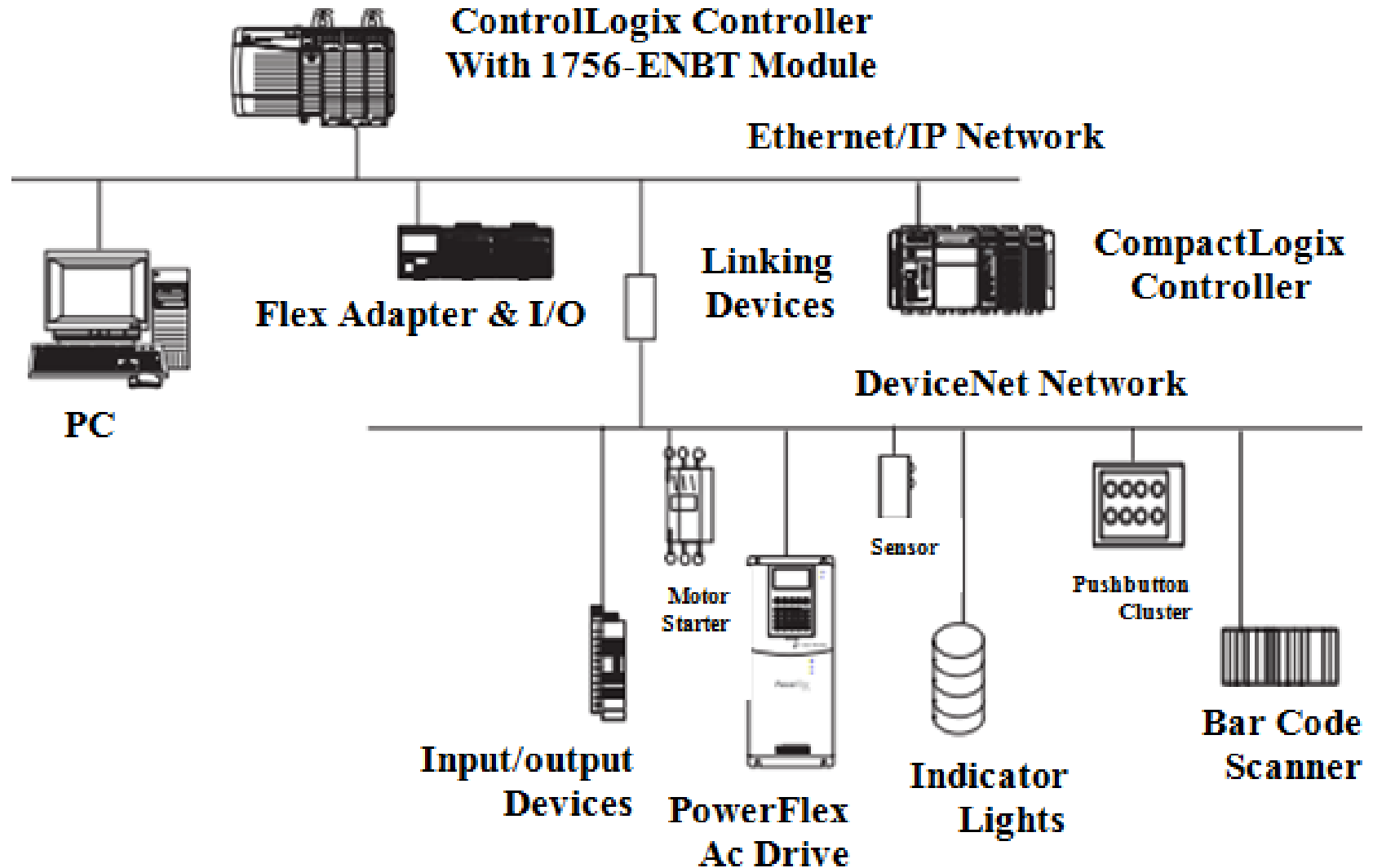
EtherNet/IP

EtherNet/IP to DeviceNet Linking Device 1788-EN2DN

ControlNet™

ControlNet to DeviceNet Linking Device 1788-CN2DN

# DEVICENET NETWORK CONFIGURATION



# DEVICENET NETWORK CONFIGURATION

Configure I/O modules for Slaves by RSnetwork for Devicenet

The screenshot shows the RSNetWorx for DeviceNet software interface. The main window displays a rack configuration with two modules: a 1769-SDN Scanner Module in slot 00 and a 1794-ADN DeviceNet Flex I/O Adapter in slot 02. An inset window titled "1794-ADN DeviceNet Flex I/O Adapter" is open, showing the "Module Configuration" tab. This window allows users to select and configure the adapter and its associated modules. The "Hardware" list on the left includes various modules, and the "Properties" table on the right shows the configuration for each slot. A black arrow points from the "1794-ADN DeviceNet Flex I/O Adapter" in the main window to the "1794-ADN/B" module in slot 00 of the inset window.

Slot	Module Type
n/a	1794-ADN/B
00	1794-IB32/A
01	1794-IB16D/A
02	1794-OB16/A
03	1794-OB16D/A
04	1794-IA16/A
05	1794-OA16/A
06	
07	

# DEVICENET NETWORK CONFIGURATION

Configure I/O address for modules in Master Scanner

The image displays two screenshots of the 1769-SDN Scanner Module configuration software, illustrating the process of configuring I/O addresses for a 1794-ADN DeviceNet Flex I/O Adapter. The top part of the image shows a network diagram with a 1769-SDN Scanner Module (address 00) and a 1794-ADN DeviceNet Flex I/O Adapter (address 02) connected to a network.

The left screenshot shows the configuration window for the 1769-SDN Scanner Module. The 'Scanlist' tab is active, showing a table of nodes:

Node	Type	Size	Map
02, 1794-...	Polled	16	1:I.Data[0].0
[n/a] 17...		2 By...	
[00] 1794-IB32/A.Data			
[01] 179...		2 By...	
[02] 179...		2 By...	
[03] 179...		2 By...	

The 'Memory' dropdown is set to 'Discrete' and the 'Start DWord' is 0. The 'I.Data' table shows the mapping of I/O addresses:

Bits 31 - 0	
1:I.Data[0]	02, 1794-ADN DeviceNet Flex I/O Adapter
1:I.Data[1]	02, 1794-ADN DeviceNet Flex I/O Adapter
1:I.Data[2]	02, 1794-ADN DeviceNet Flex I/O Adapter
1:I.Data[3]	02, 1794-ADN DeviceNet Flex I/O Adapter
1:I.Data[4]	
1:I.Data[5]	
1:I.Data[6]	
1:I.Data[7]	
1:I.Data[8]	

The right screenshot shows the same configuration window, but with the 'ADR' tab active. The 'Node' table shows the mapping of addresses:

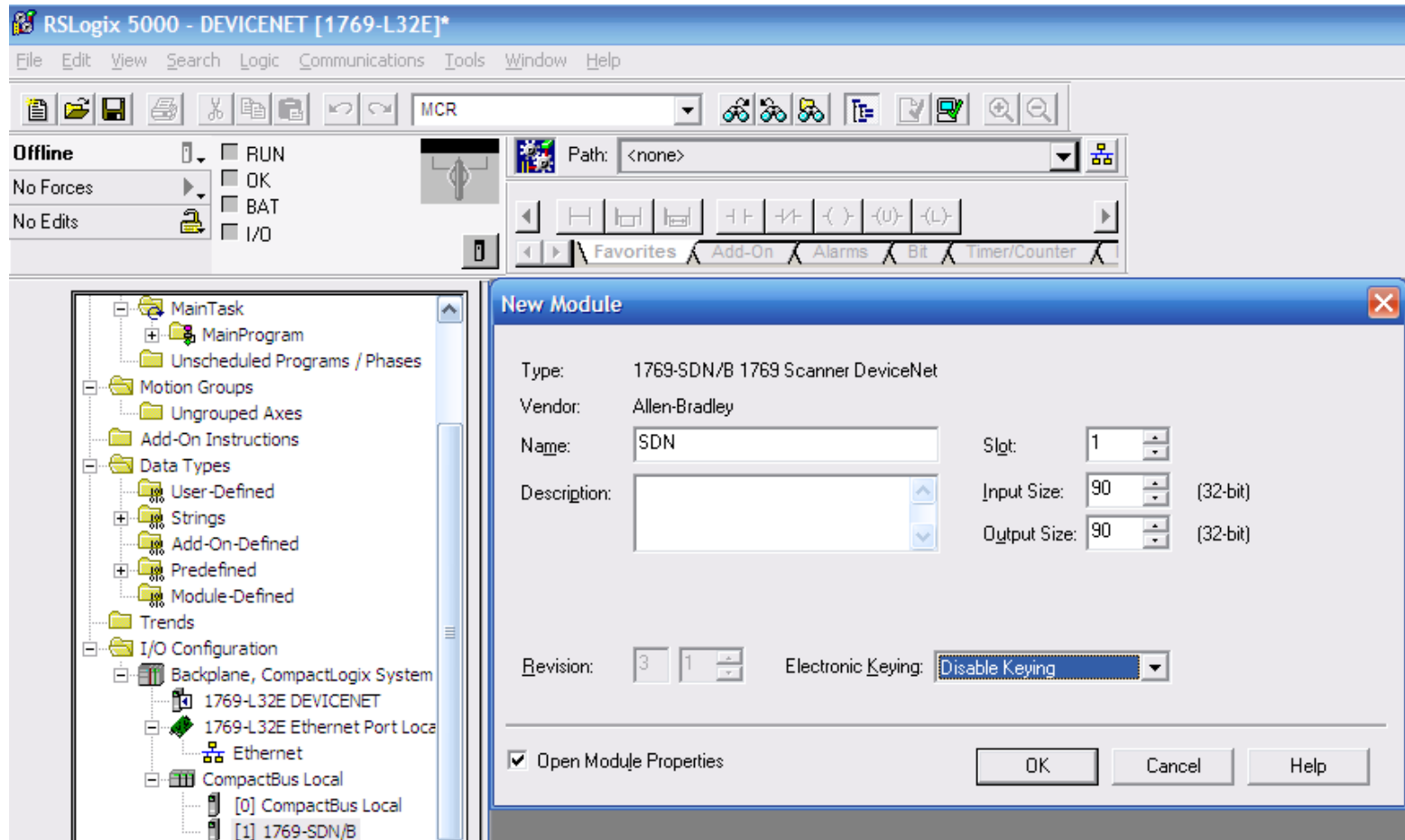
Node	Type	Size	Map
02, 179...	Polled	10	1:O.Data[0].0
[01] 17...		2 By...	
[02] 17...		2 By...	
[03] 1794-OB16D/A.Data			
[04] 17...		2 By...	
[05] 17...		2 By...	

The 'Memory' dropdown is set to 'Discrete' and the 'Start DWord' is 0. The 'O.Data' table shows the mapping of addresses:

Bits 31 - 0	
1:O.Data[0]	02, 1794-ADN DeviceNet Flex I/O Adapter
1:O.Data[1]	02, 1794-ADN DeviceNet Flex I/O Adapter
1:O.Data[2]	02, 1794-ADN Device...
1:O.Data[3]	
1:O.Data[4]	
1:O.Data[5]	
1:O.Data[6]	
1:O.Data[7]	
1:O.Data[8]	

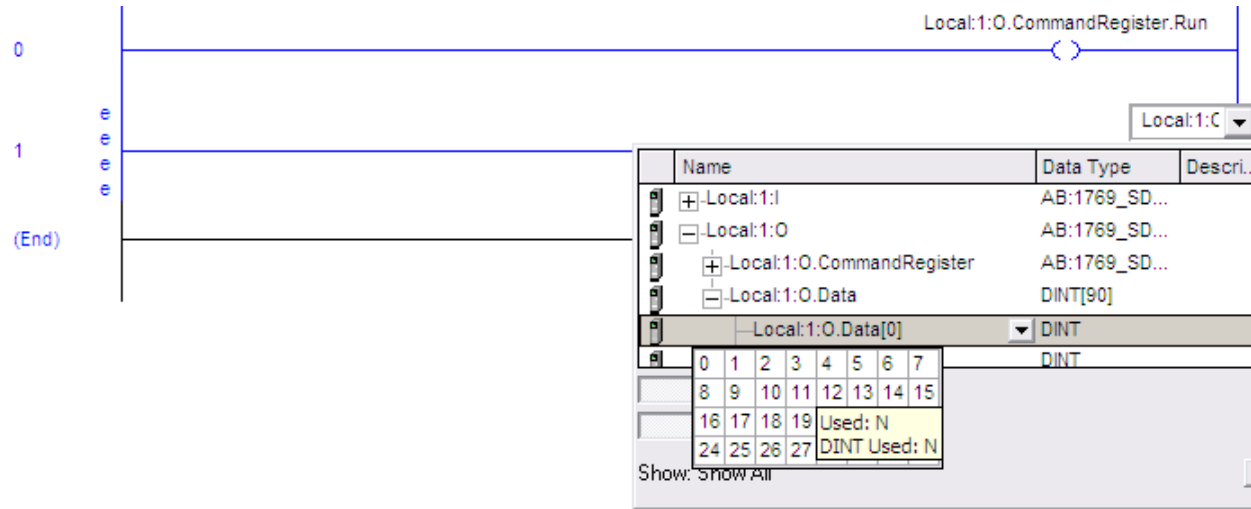
# DEVICENET NETWORK CONFIGURATION

## Insert Master Scanner into Project using RSLogix5000



# DEVICENET NETWORK CONFIGURATION

Creating logic to exchange I/O data over Devicenet Network

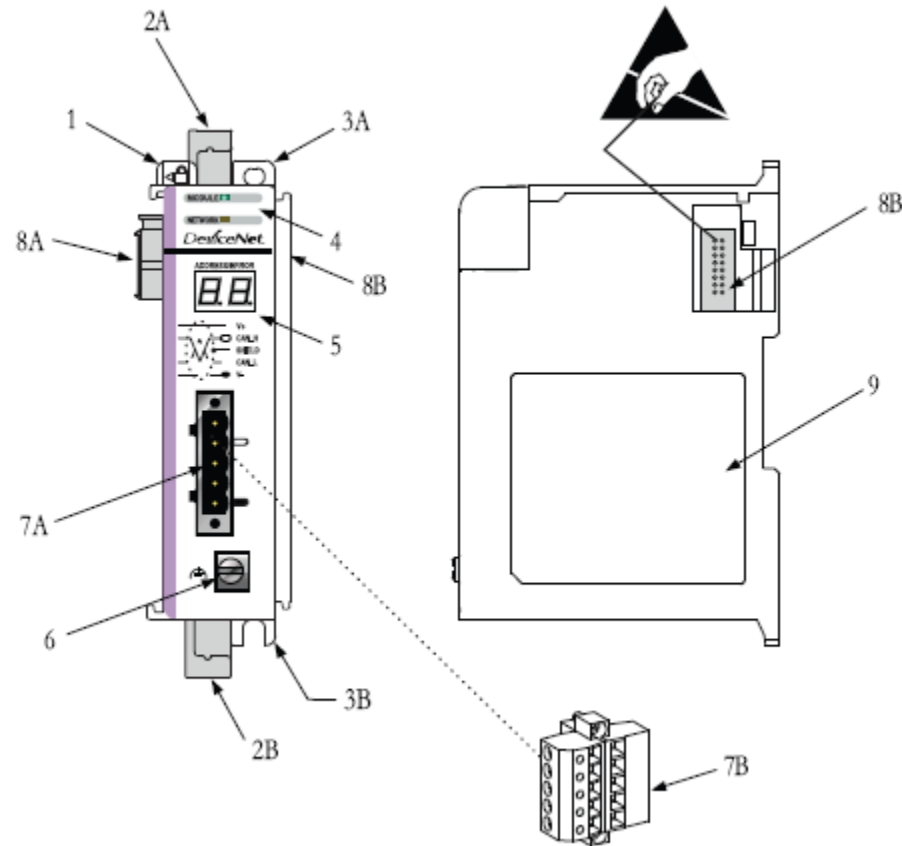


# COMPACTLOGIX DEVICENET SCANNER

## 1769 SDN(Devicenet Master)

### Module Features

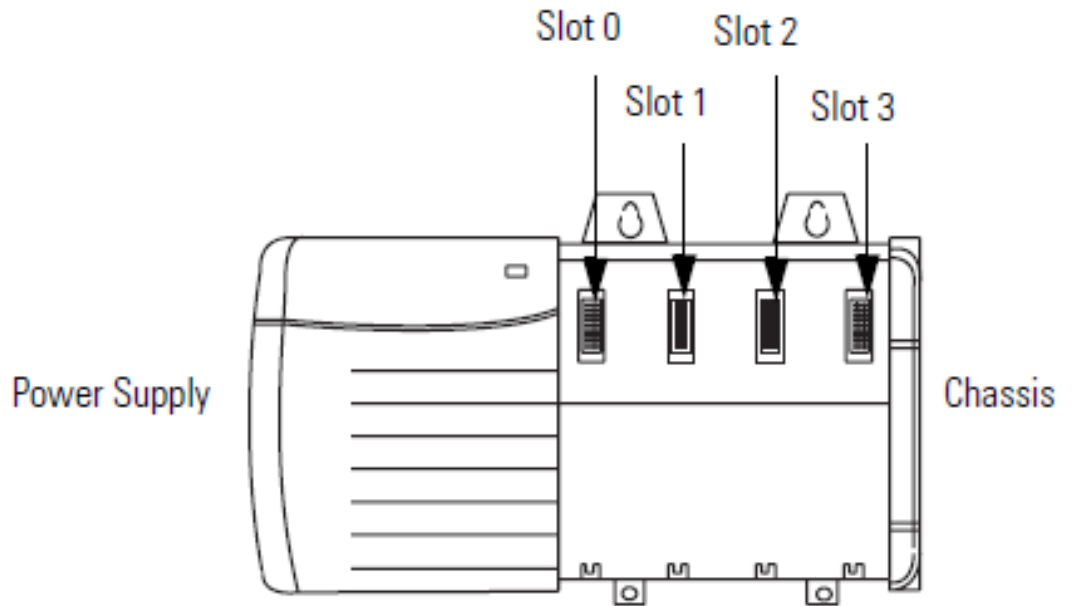
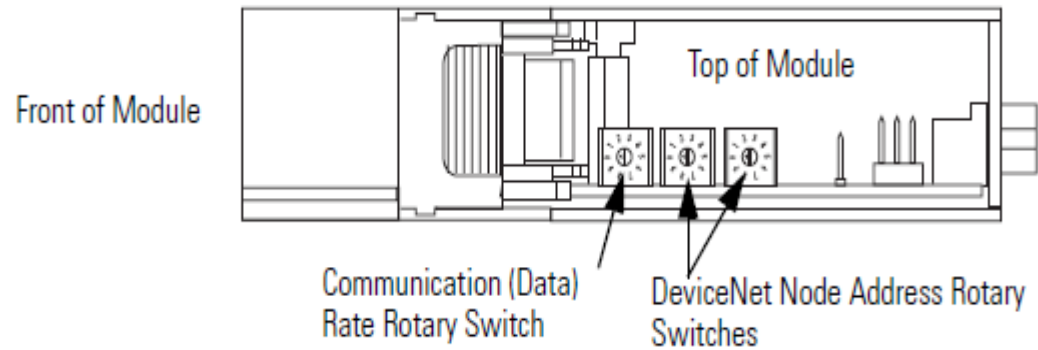
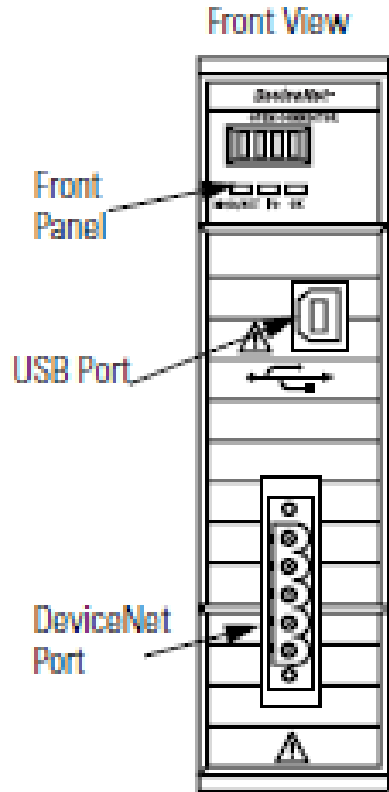
Item	Description
1	Bus lever (with locking function)
2A	Upper DIN rail latch
2B	Lower DIN rail latch
3A	Upper panel mounting tab
3B	Lower panel mounting tab
4	Module and Network status LEDs
5	Address and Error numeric display
6	Grounding screw
7A	DeviceNet mating male receptacle
7B	Removable DeviceNet female connector
8A	Movable bus connector with female pins
8B	Bus connector with male pins
9	Nameplate label





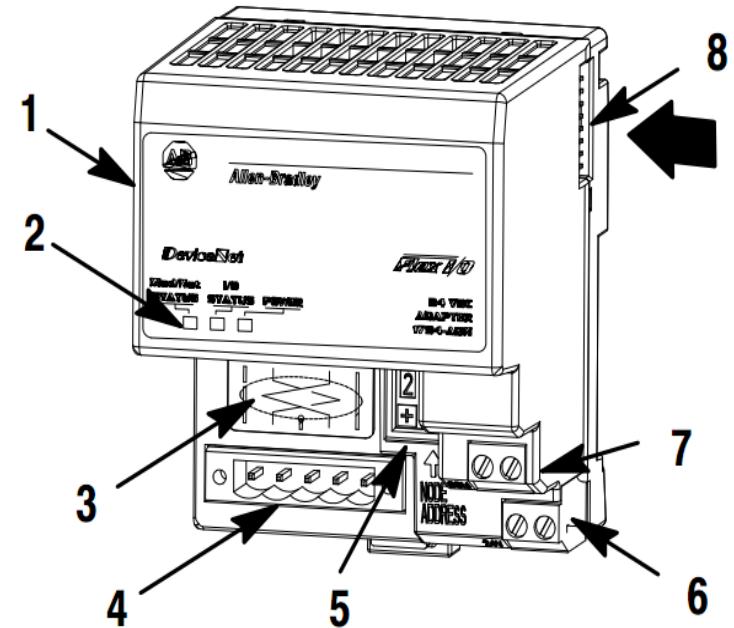
# CONTROLNET DEVICENET SCANNER

## 1756 DNB (Devicenet Master)



# DEVICENET SLAVE

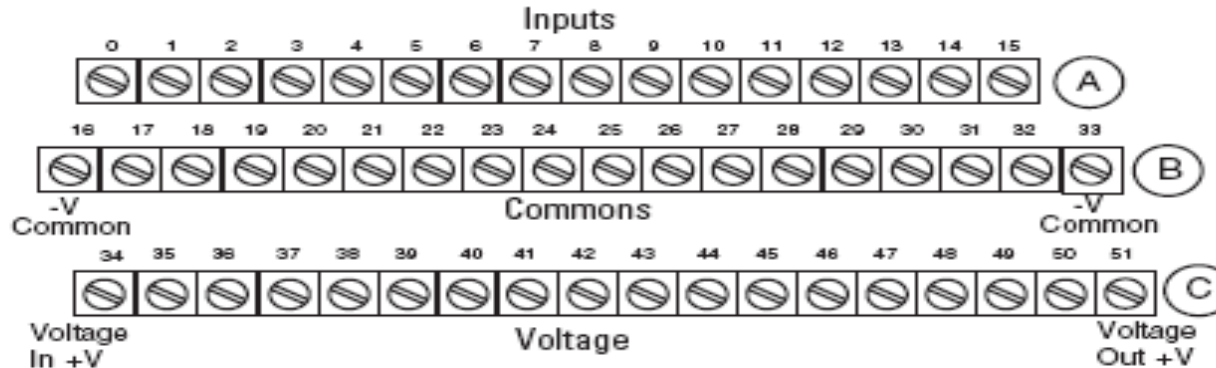
## 1794 ADN(DeviceNet Slave)



# DEVICENET I/O MODULES

## Flex I/O 1794-IB8, IB16.

### 1794-TB3 and -TB3S Terminal Base Wiring for 1794-IB8, -IB16, and -IB16K



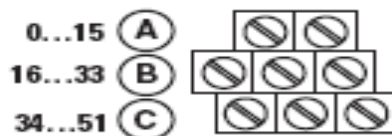
Connect V common to terminal B-16.

Connect +V to terminal C-34.

Use B-33 and C-51 to daisy-chain to the next terminal base unit.

(1794-TB3 shown)

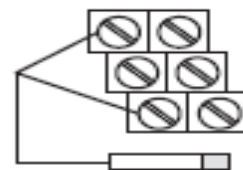
### 2- and 3-Wire Input Wiring for 1794-IB8, -IB16, and -IB16K



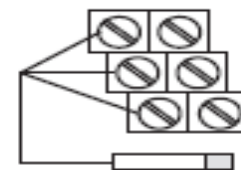
(A) = Sink input

(B) = Common

(C) = +V DC



2-Wire Device  
(Sourcing Output)

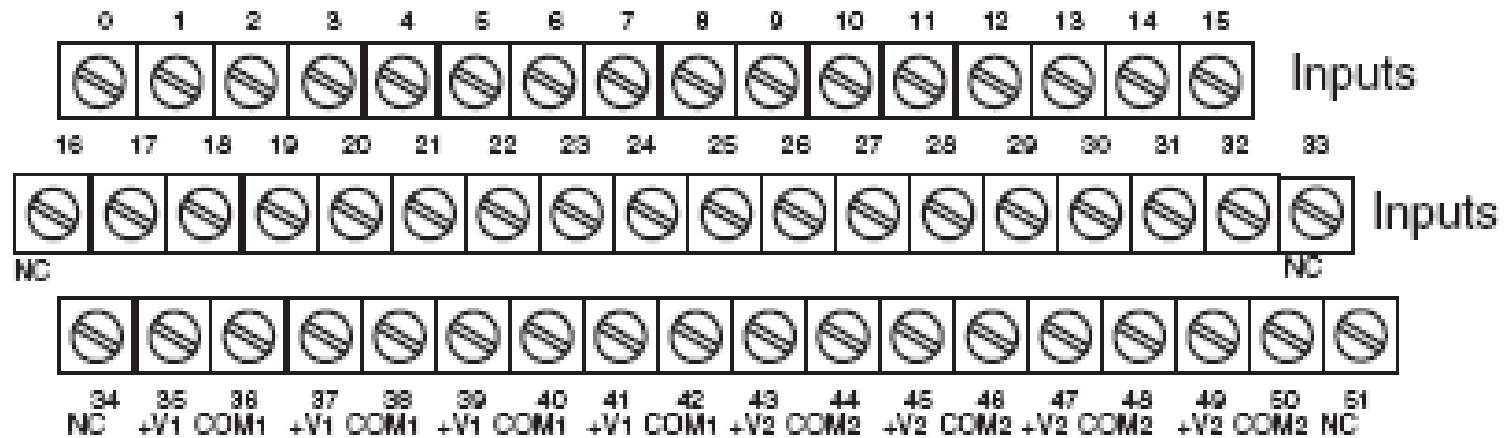


3-Wire Device  
(Sourcing Output)

# DEVICENET I/O MODULES

## Flex I/O 1794-IB32

### 1794-TB32 or -TB32S Terminal Base Wiring for the 1794-IB32



+V1 = Terminals 35, 37, 39, and 41 (1794-TB32 shown)

+V2 = Terminals 43, 45, 47, and 49

COM1 = Terminals 36, 38, 40, and 42

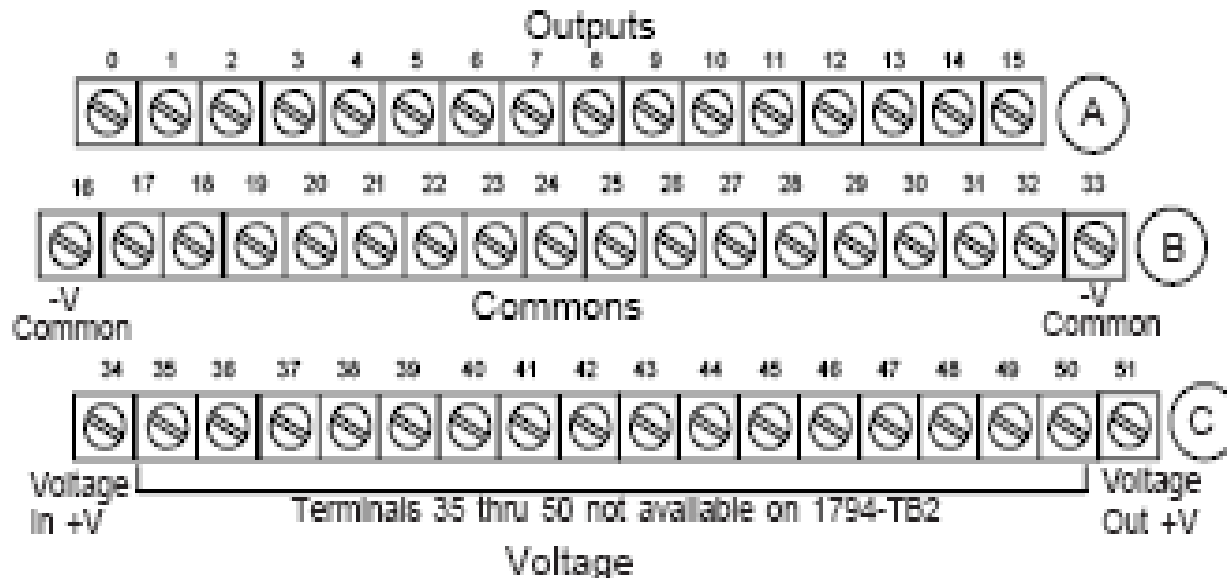
COM2 = Terminals 44, 46, 48, and 50

NC = No connections (terminals 16, 33, 34, and 51)

# DEVICENET I/O MODULES

Flex I/O 1794-OB18,OB16.

1794-TB2, -TB3 and -TB3S Terminal Base Wiring for 1794-OB8, -OB8EP, -OB8EPK, -OB16, -OB16P and -OB16PK



Connect -V (Supply Common) to terminal B-16

(1794-TB3 shown)

Connect +V (Supply +Voltage) to terminal C-34

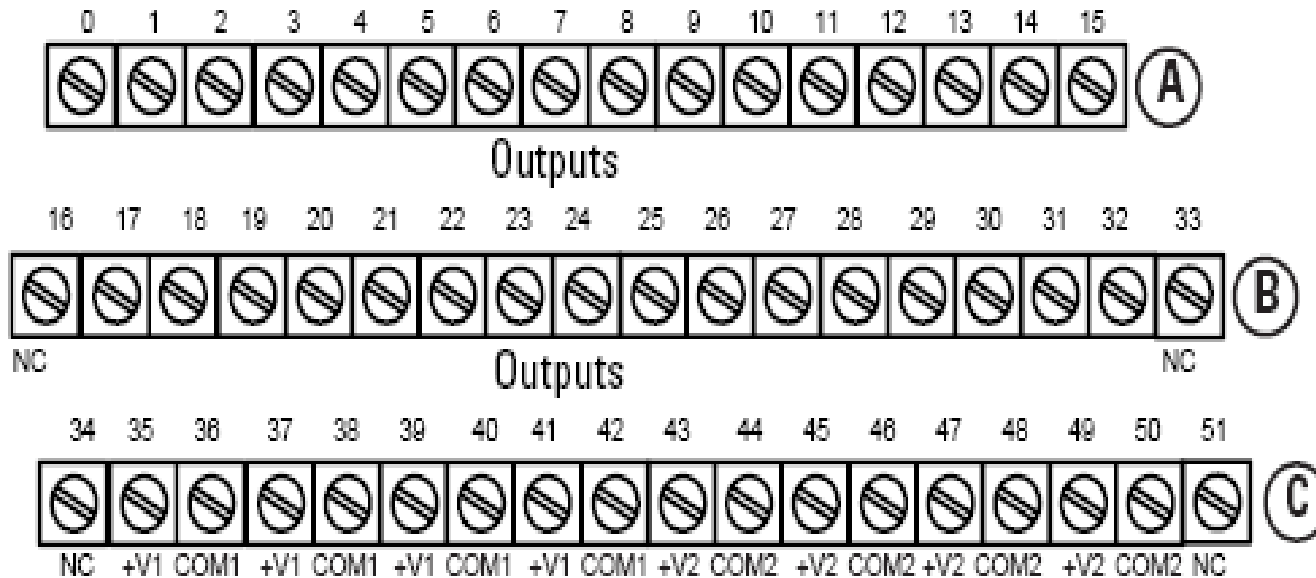
(Use B-33 and C-51 for daisy-chaining to next terminal base unit.)

Total current draw through the terminal base is limited to 10A. Separate power connections to each terminal base may be necessary.

# DEVICENET I/O MODULES

Flex I/O 1794-OB32.

## 1794-TB32 and -TB32S Terminal Base Wiring for 1794-OB32P



+V1 = Terminals 35, 37, 39 and 41

(1794-TB32 shown)

+V2 = Terminals 43, 45, 47 and 49

COM1 = Terminals 36, 38, 40 and 42

COM2 = Terminals 44, 46, 48 and 50

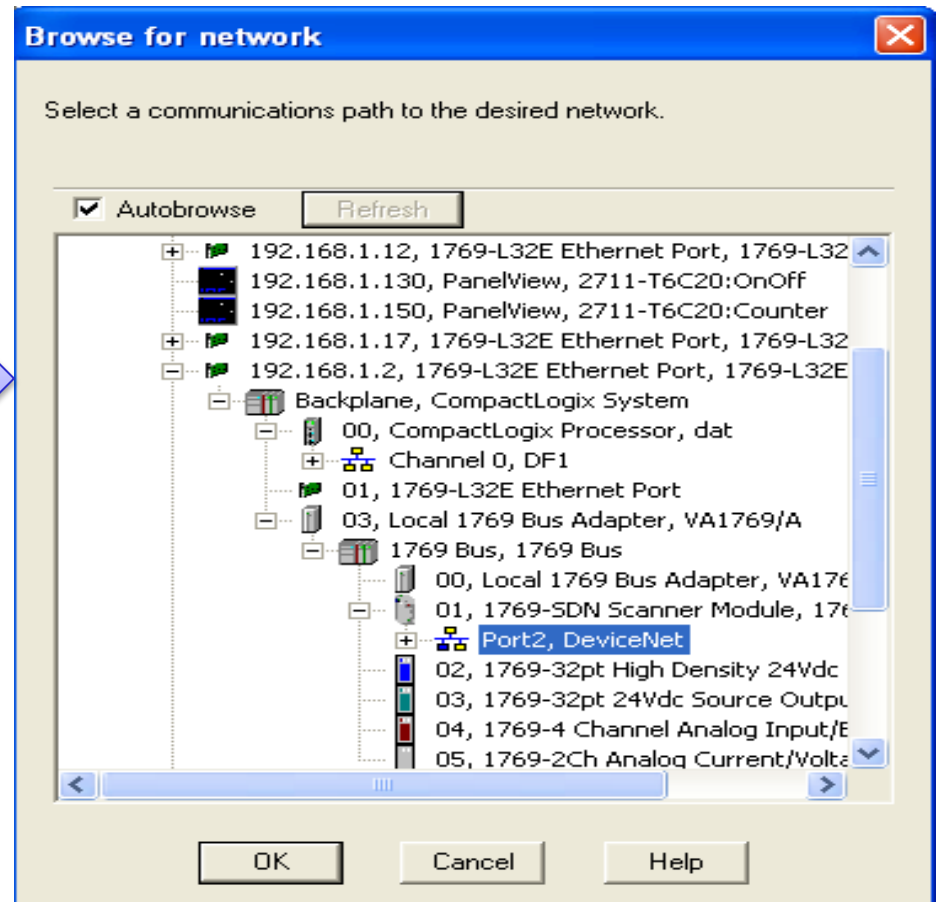
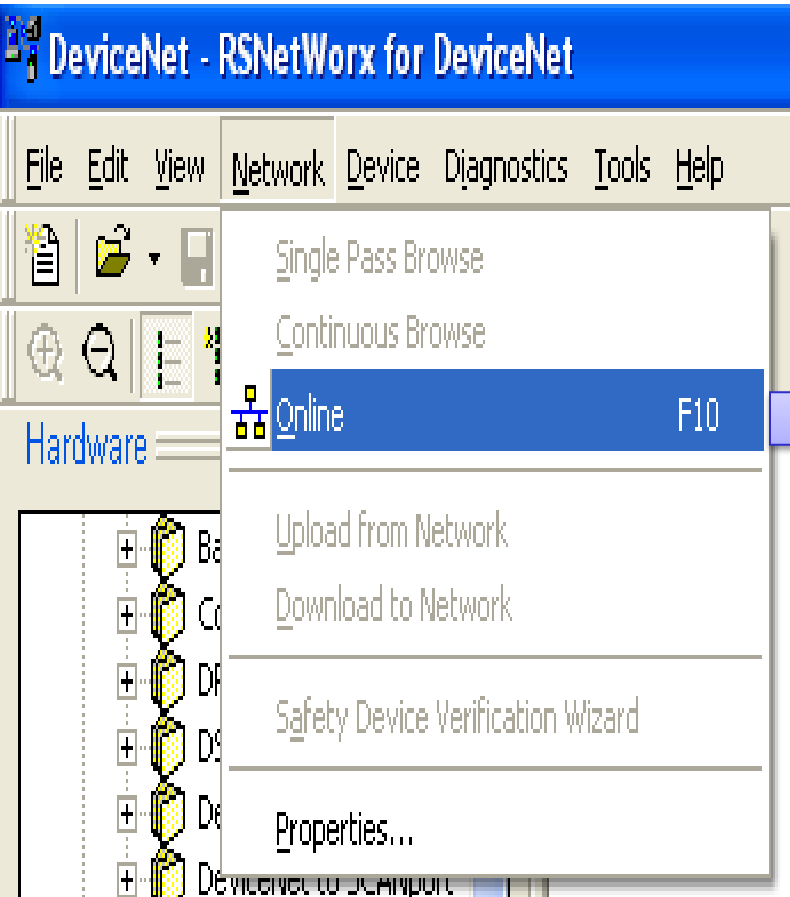
NC = No connections (terminals 16, 33, 34 and 51)

# CONTROL DEVICES VIA D\_NET

1. Connecting Devices to DeviceNet Network
2. Setting the address of Scanner and other devices
3. Creating Scanlist file by RSnetwork for devicenet
4. Configuring I/O memory area of modules
5. Downloading the configuration to Scanner
6. Creating a Project in RSlogix5000
7. Writing logic program
8. Downloading the program to PLC

# CONTROL DEVICES VIA D\_NET

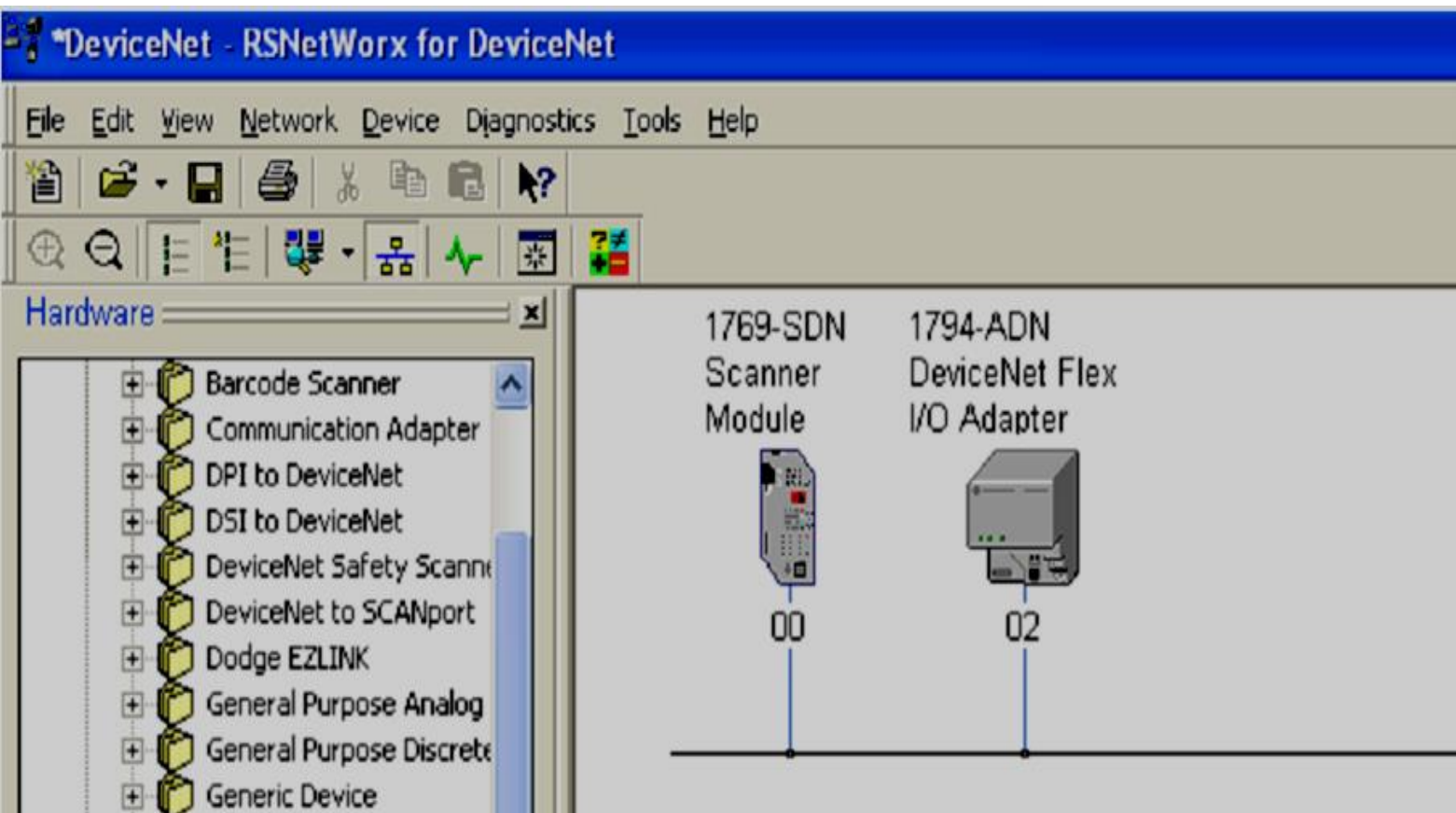
- Creating Scanlist file by Rsnetwork for Dnet





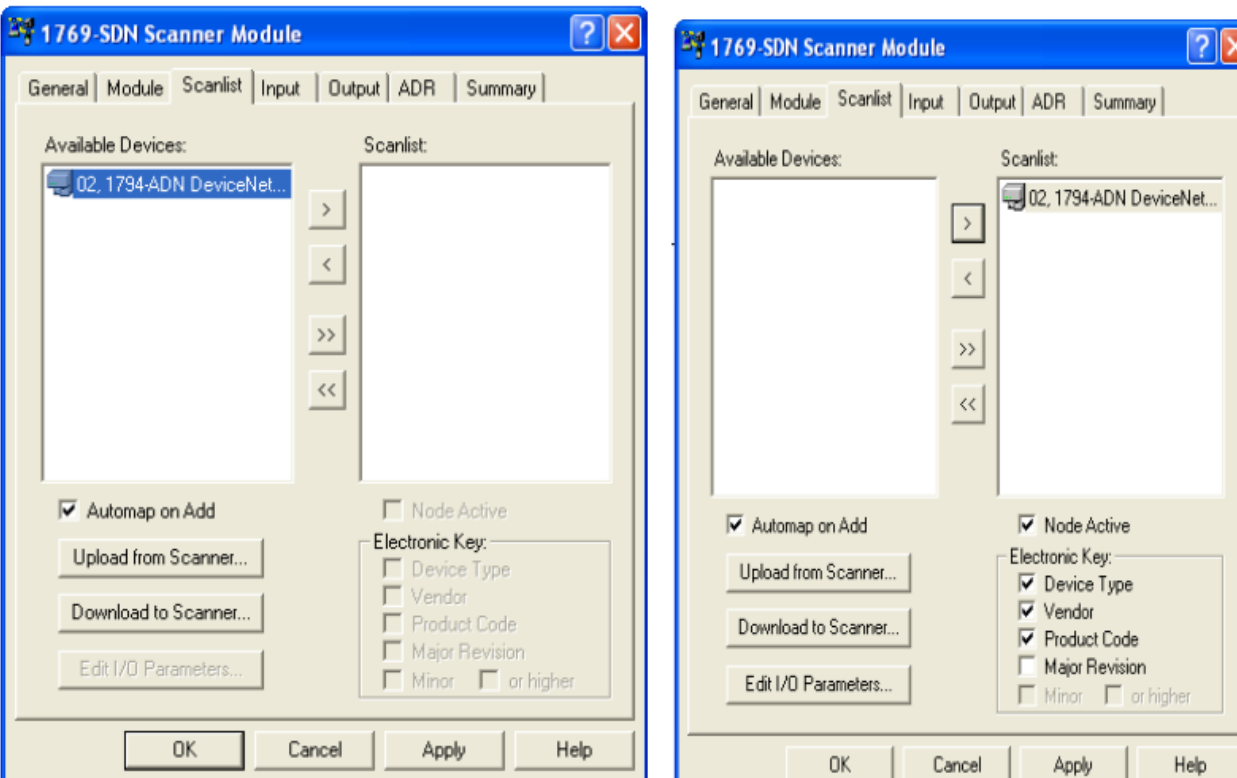
# CONTROL DEVICES VIA D\_NET

Depend on system hardware, Scanlist file is different



# CONTROL DEVICES VIA D\_NET

## Add devices to Scanlist



*Select Scanner module/upload/ Click Scanlist tab/ choose devices to add to Scanlist*

# CONTROL DEVICES VIA D\_NET

Specify I/O memory areas of modules at Input and Output tab then download to Scanner module.

1769-SDN Scanner Module  
1794-ADN DeviceNet Flex I/O Adapter



1769-SDN Scanner Module  
1794-ADN DeviceNet Flex I/O Adapter



1769-SDN Scanner Module

General | Module | Scanlist | **Input** | Output | ADR | Summary

Node	Type	Size	Map
02, 1794-...	Polled	16	1:I.Data[0].0
[n/a] 17...		2 By...	
[00] 1794-IB32/A,Data		2 By...	
[01] 179...		2 By...	
[02] 179...		2 By...	
[03] 179...		2 By...	

Memory: Discrete Start DWord: 0

Bits 31 - 0	
1:I.Data[0]	02, 1794-ADN DeviceNet Flex I/O Adapter
1:I.Data[1]	02, 1794-ADN DeviceNet Flex I/O Adapter
1:I.Data[2]	02, 1794-ADN DeviceNet Flex I/O Adapter
1:I.Data[3]	02, 1794-ADN DeviceNet Flex I/O Adapter
1:I.Data[4]	
1:I.Data[5]	
1:I.Data[6]	
1:I.Data[7]	
1:I.Data[8]	

1769-SDN Scanner Module

General | Module | Scanlist | Input | **Output** | ADR | Summary

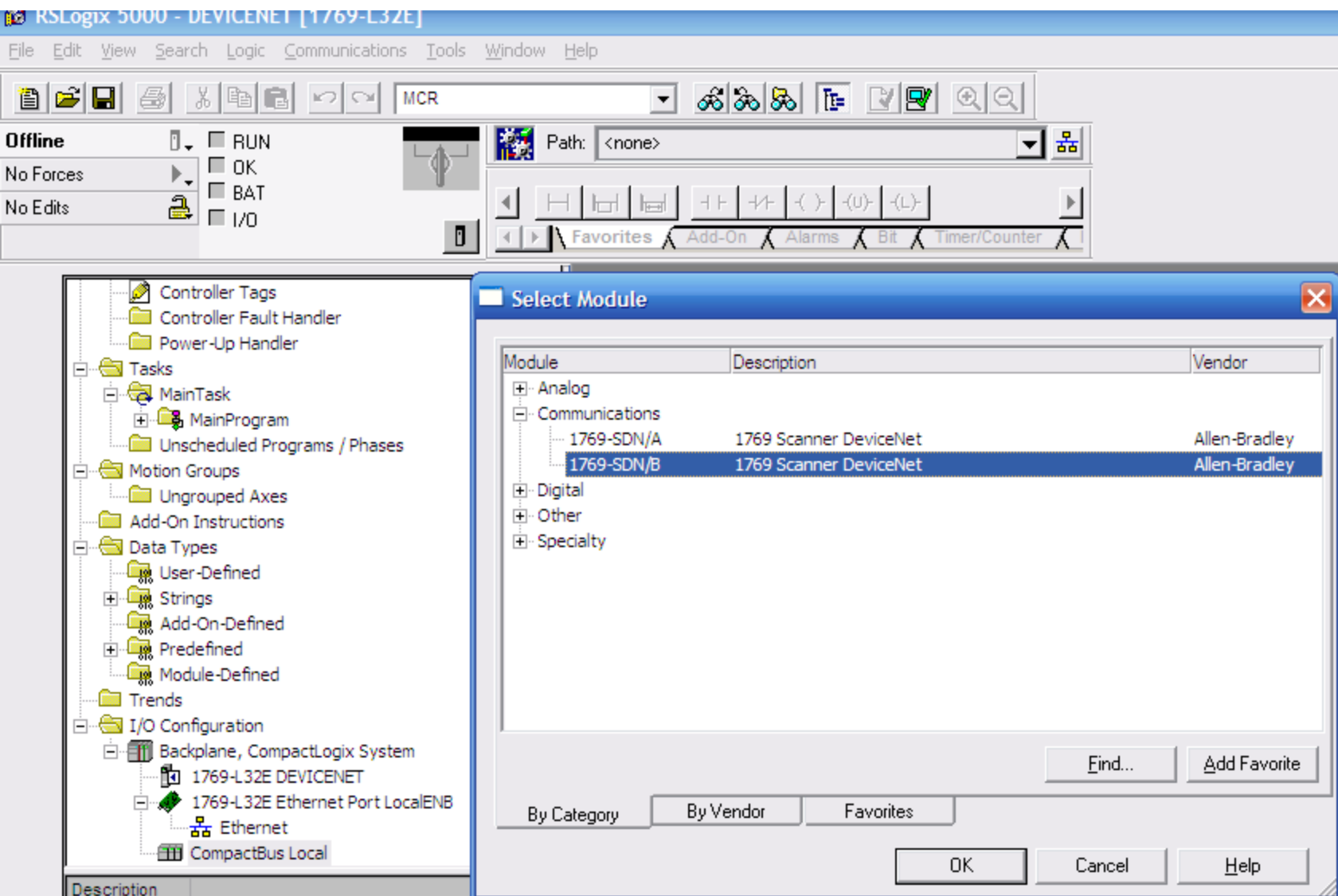
Node	Type	Size	Map
02, 179...	Polled	10	1:O.Data[0].0
[01] 17...		2 By...	
[02] 17...		2 By...	
[03] 1794-OB16D/A,Data		2 By...	
[04] 17...		2 By...	
[05] 17...		2 By...	

Memory: Discrete Start DWord: 0

Bits 31 - 0	
1:O.Data[0]	02, 1794-ADN DeviceNet Flex I/O Adapter
1:O.Data[1]	02, 1794-ADN DeviceNet Flex I/O Adapter
1:O.Data[2]	
1:O.Data[3]	02, 1794-ADN Device...
1:O.Data[4]	
1:O.Data[5]	
1:O.Data[6]	
1:O.Data[7]	
1:O.Data[8]	

# CONTROL DEVICES VIA D\_NET

- Creating a project in RSLogix5000 and add a Scanner



# CONTROL DEVICES VIA D\_NET

## Configuring the Scanner Module

The screenshot displays the RSLogix 5000 - DEVICENET [1769-L32E]\* interface. The main window shows a project tree on the left with folders like MainTask, Motion Groups, and I/O Configuration. The I/O Configuration folder is expanded, showing a 1769-SDN/B module. A 'New Module' dialog box is open, showing the configuration for a 1769-SDN/B 1769 Scanner DeviceNet module. The configuration includes:

- Type: 1769-SDN/B 1769 Scanner DeviceNet
- Vendor: Allen-Bradley
- Name: SDN
- Slot: 1
- Description: (empty)
- Input Size: 90 (32-bit)
- Output Size: 90 (32-bit)
- Revision: 3 1
- Electronic Keying: Disable Keying

The 'Open Module Properties' checkbox is checked. The dialog box has OK, Cancel, and Help buttons.

# CONTROL DEVICES VIA D\_NET

## Writing a simple program in Rslogix 5000.

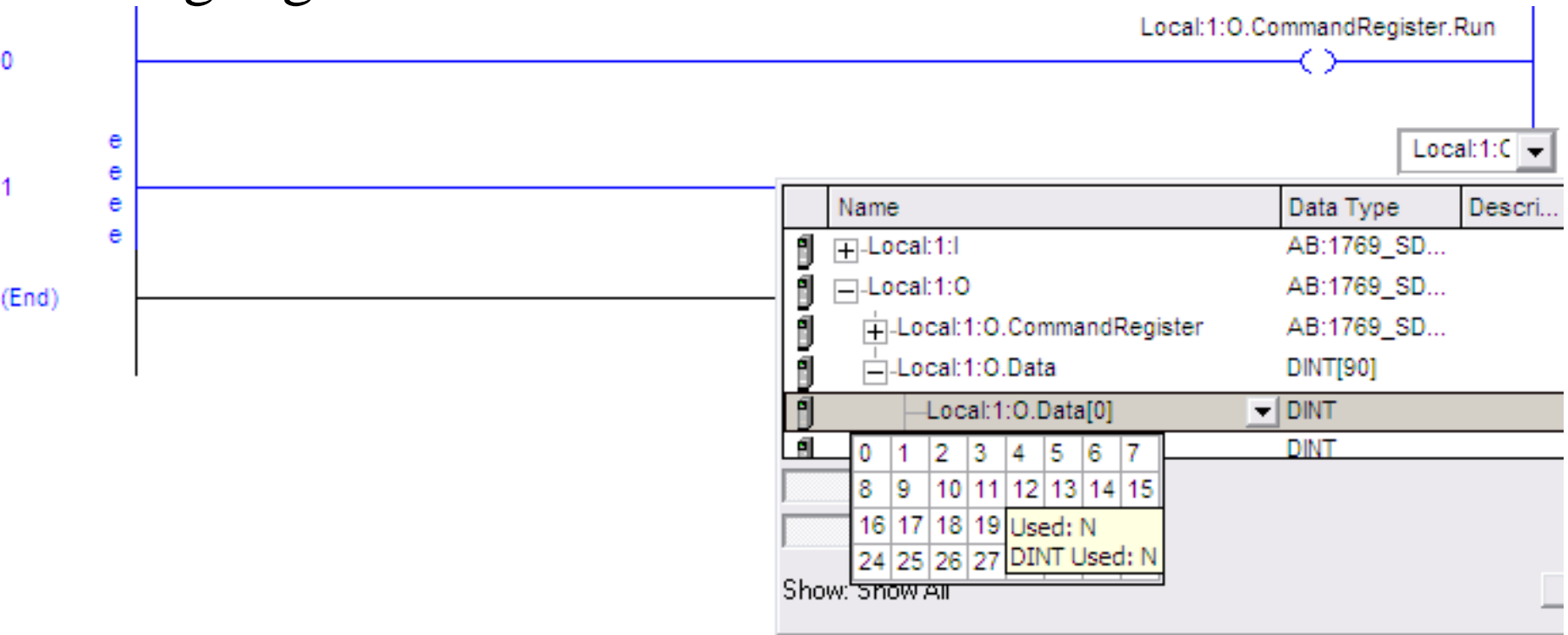
Name	Data Type	Descri...
+Local:1:I	AB:1769_SD...	
-Local:1:O	AB:1769_SD...	
-Local:1:O.CommandRegister	AB:1769_SD...	
Local:1:O.CommandRegister.Run	BOOL	

Controller  
Program  
Show: Show All

Notice: **CommandResister.Run** instruction must be executed to run devicenet network

# CONTROL DEVICES VIA D\_NET

Writing logic to access I/O modules.



# DEVICENET NETWORK TROUBLE SHOOTING

Status code (decimal)	Description	Action
65	The AutoScan option is on and the device is in idle mode.	None.
70	The address of the device is already in use by another device on the network.	Change the address of the device to an unused address.
71	Illegal data in scan list.	Reconfigure the scan list and remove any illegal data.
72	No communication with the device.	Inspect the device and verify connections.
73	Device's identity information does not match electronic key in scanner	<ul style="list-style-type: none"><li>• Make sure that the correct device is at this address.</li><li>• Make sure that the device matches the specified electronic key (vendor, product code, product type).</li></ul>
74	Data overrun on port detected.	<ul style="list-style-type: none"><li>• Modify your configuration and check for invalid data.</li><li>• Check network communication traffic.</li></ul>
75	Either or both of the following: <ul style="list-style-type: none"><li>• The device does <i>not</i> have a scan list.</li><li>• The device has <i>not</i> received communication from any other device</li></ul>	Check that the device has: <ul style="list-style-type: none"><li>• scan list</li><li>• properly wired connection to the network</li></ul>
76	No direct network traffic for scanner.	None. The scanner hears other network communication but does <i>not</i> hear any directed to it.
77	During initialization, the data size expected by the device does <i>not</i> match the scan list entry.	Check the device and the scan list for the correct input and output size for the device.



# DEVICENET NETWORK TROUBLE SHOOTING

78	Device is <i>not</i> communicating or communication is intermittent.	<ul style="list-style-type: none"><li>• Check that the device has a properly wired connection to the network.</li><li>• Check that the device has power.</li><li>• If the device is polled, make sure the interscan delay is long enough for the device to return its data.</li></ul>
79	Scanner has failed to transmit a message.	<ul style="list-style-type: none"><li>• Make sure that your scanner is connected to a valid network.</li><li>• Check for disconnected cables.</li></ul>
80	Scanner is in idle mode.	To run the network: <ol style="list-style-type: none"><li>1. Put controller in run/remote run mode.</li><li>2. Turn on the following member of command register for the scanner:</li></ol>
81	Controller has set the scanner to the faulted mode.	See if the following bit of the command register for the scanner is on: ...O.CommandRegister.Fault
82	Error detected in sequence of fragmented I/O messages from device.	<ul style="list-style-type: none"><li>• Check scan list device to make sure that its input and output data sizes are correct.</li><li>• Check the configuration of the device.</li></ul>
83	Device returns error responses when the scanner attempts to communicate with it.	<ul style="list-style-type: none"><li>• Check the accuracy of the scan list.</li><li>• Check the configuration of the device. The device may be in another scanner's scan list.</li><li>• Cycle power to the device.</li></ul>
84	Scanner is initializing the DeviceNet network.	None. This code clears itself once the scanner attempts to initialize all the devices on the network.

# DEVICENET NETWORK TROUBLE SHOOTING

85	During runtime, the device is sending the wrong size of data.	Contact Rockwell Automation support. See the back of this publication.
86	Device is in idle state/mode ( <i>not</i> producing data) while the scanner is in run mode.	<ul style="list-style-type: none"><li>• Check the configuration and status of the device.</li><li>• If you set up an interlock between 2 scanners (controllers), make sure both scanners are in run mode.</li></ul>
88	In shared inputs, the I/O parameters (polled, strobed, etc.) <i>do not</i> match between the scanners.	Use the same I/O parameters for the device in both scanners.
89	Scanner failed to configure a device using the Automatic Device Recovery (ADR) parameters	<ul style="list-style-type: none"><li>• Make sure that you installed a compatible device.</li><li>• The offline configuration for the device does not match the actual (online) configuration of the device.</li></ul>
90	Controller has set the scanner to the disabled mode.	See if the following bit of the command register for the scanner is on: ...0.CommandRegister.DisableNetwork
91	Bus-off condition (communication problem)	<ul style="list-style-type: none"><li>• Cycle power to the device.</li><li>• Make sure all devices are at the same baud rate.</li><li>• Make sure there is <i>no</i> short circuit between a CAN line (blue or white) and a power or shield line (black, red, shield).</li><li>• Check for any of the following sources of noise:<ul style="list-style-type: none"><li>• Close proximity to a high voltage power cable</li><li>• Improper or no termination resistor</li><li>• Improper grounding</li></ul></li><li>• Check for a device that is producing noise or inappropriate data on the network.</li></ul>

# DEVICENET NETWORK TROUBLE SHOOTING

92	The DeviceNet cable is <i>not</i> supplying power to the communication port.	<ul style="list-style-type: none"><li>• Make sure the network has 24V dc power.</li><li>• Check the connection to the device.</li></ul>
95	The firmware of the device is currently being updated.	None. Do not disconnect the device while the update is in progress. You will lose any existing data in the device's memory.
96	Communication port is in test mode.	None.
97	Controller has set the scanner to the halted mode.	<ol style="list-style-type: none"><li>1. See if the following bit of the command register for the scanner is on: ...0.CommandRegister.HaltScanner</li><li>2. Cycle power to the scanner.</li></ol>
98	General firmware error.	Replace device.
99	System failure.	Replace device.

# CONTROL DEVICES VIA D\_NET

## EX\_1

Set up a devicenet network includes two nodes.

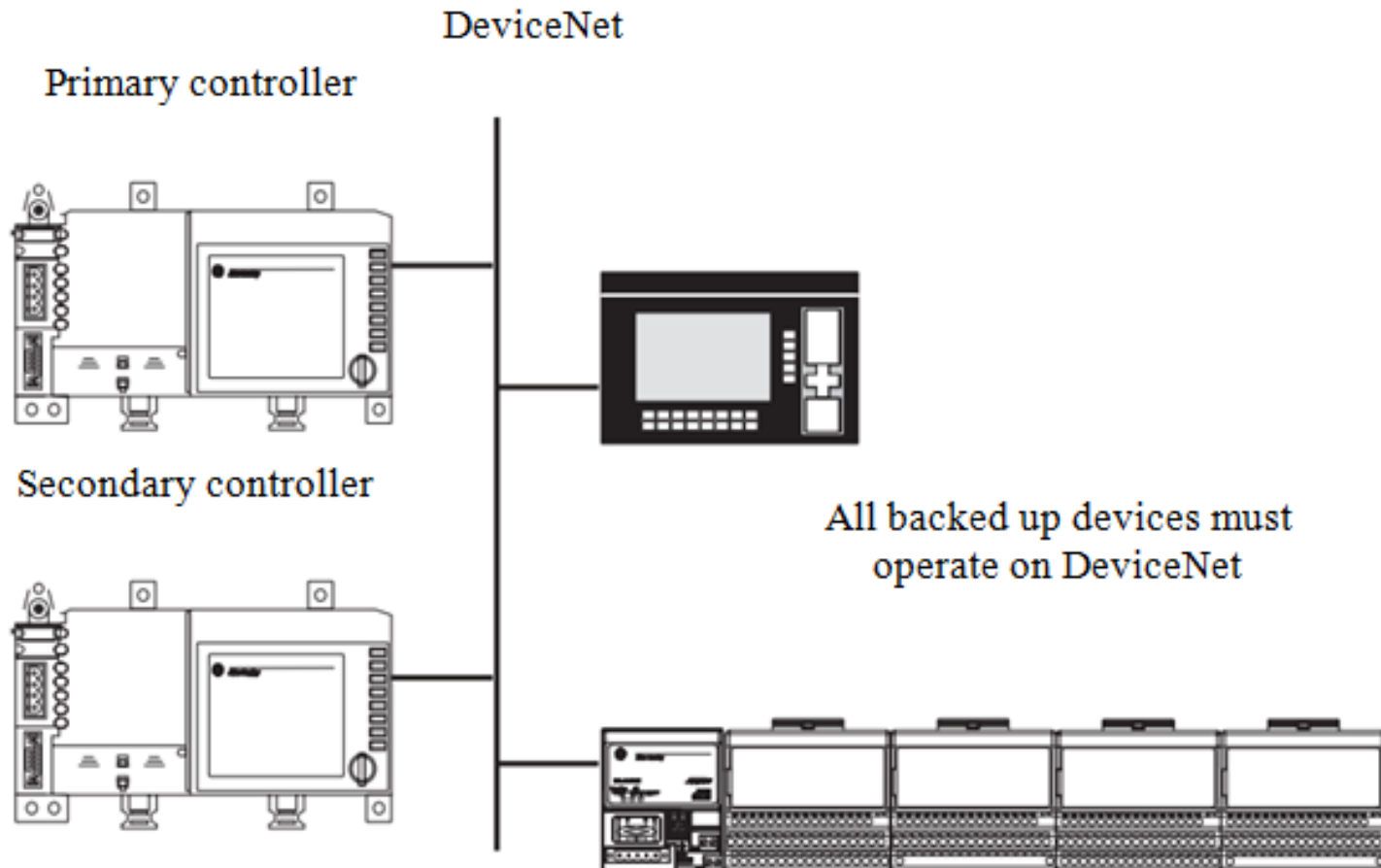
- Scanner has Address 0
- Slave has Address 4 and I/O modules
- Two buttons and one Motor are connected to I/O devicenet modules to Start and Stop the Motor.

## EX\_2

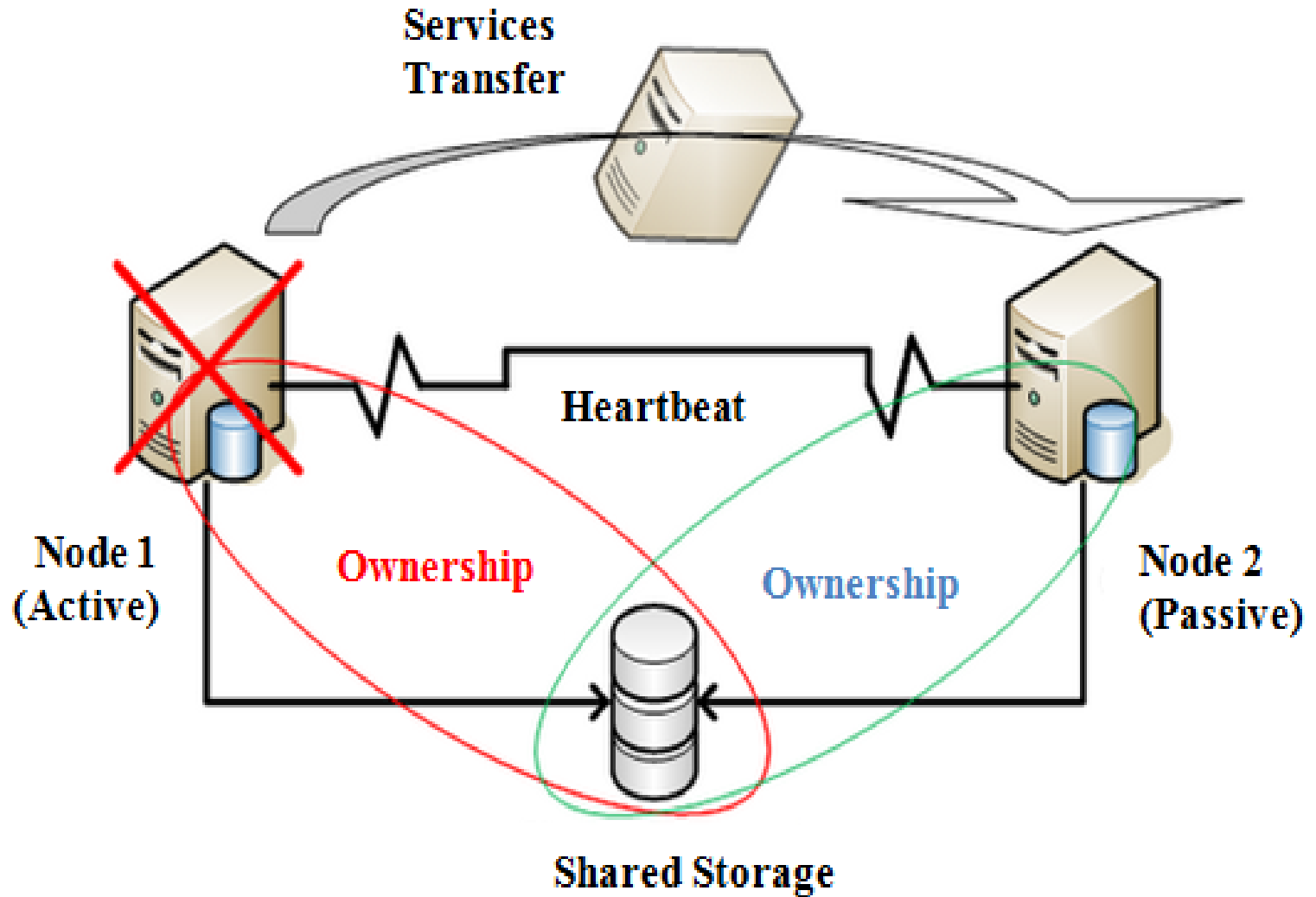
Set up a devicenet network includes two nodes.

- Scanner has Address 0
- Inverter(PowerFlex) is slave with address 4
- Start, Stop motor and increase,decrease motor speed from controller

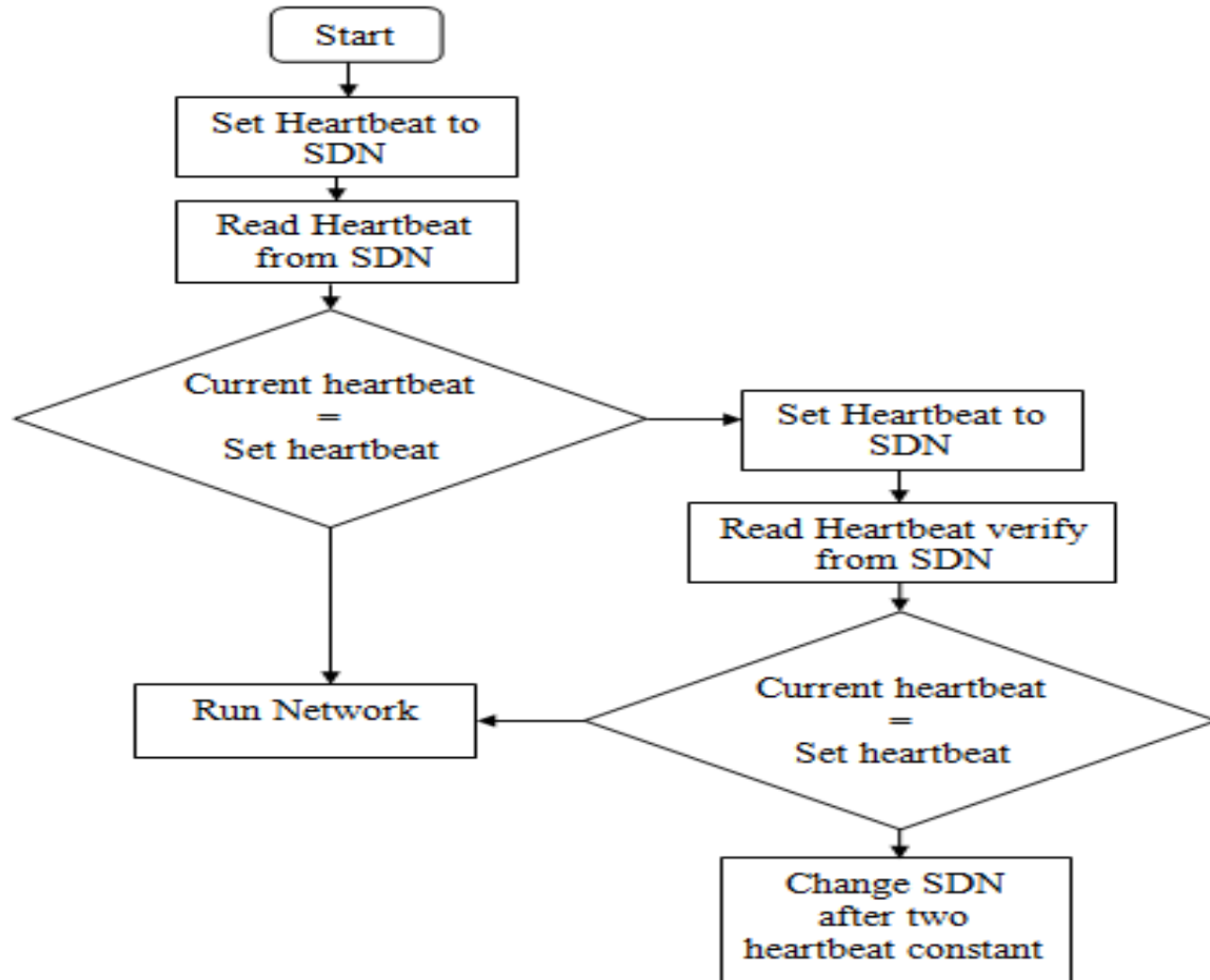
# BACK UP ON DEVICENET NETWORK



# BACK UP ON DEVICENET NETWORK



# BACK UP ON DEVICENET NETWORK



# BACK UP ON DEVICENET NETWORK

**Configure the Back up system.** *Follow these steps to configure a CompactLogix backup system on the DeviceNet network.*

1. Install all I/O and operator interfaces that you need to back up on the DeviceNet network.
2. Connect a CompactLogix controller with a 1769-SDN scanner module to the DeviceNet network.
3. Set the scanner module node address to 0 (*or the lower of the two node addresses reserved for the CompactLogix controller backup system*).
4. Apply power to the controller and the network.
5. Use RSNetWorx for DeviceNet software to download the network's scanlist to the 1769-SDN scanner module  
(*You can either use a scanlist from a new configuration or from a previously-used configuration. If the scanlist is a new configuration, we recommend that you save it to a new project for later use*)



# BACK UP ON DEVICENET NETWORK

6. Use RSLogix 5000 programming software to download the appropriate user program to the CompactLogix controller  
*(The program should contain the explicit messages that enable the backup feature for this controller and scanner module: Next section).*
7. Put the controller into Run mode.
8. Either disable power to the controller or disconnect the scanner module from the DeviceNet network  
*(This controller will be the secondary controller)*
9. Connect the other CompactLogix controller with a 1769-SDN scanner module on the network.
10. Set the node address to 0
11. Apply power to the controller and scanner module

# BACK UP ON DEVICENET NETWORK

12. Use RSNetWorx for DeviceNet software to download the same scanlist used in step 5

*(It may be necessary to browse the network again before downloading the scanlist. This second browsing of the network allows RSNetWorx for DeviceNet software to establish communication to the new scanner module at the same node number as the previous scanner)*

13. Use RSLogix 5000 programming software to download the user program to the second CompactLogix controller as you did in **step 6**. *(Typically, you download the same user program to the second CompactLogix controller as you did to the first. However, unlike the scanlists, the user programs in the controllers do not have to be identical).*

# BACK UP ON DEVICENET NETWORK

14. Put the controller into Run mode.

*(This controller is now ready to use and is the primary controller).*

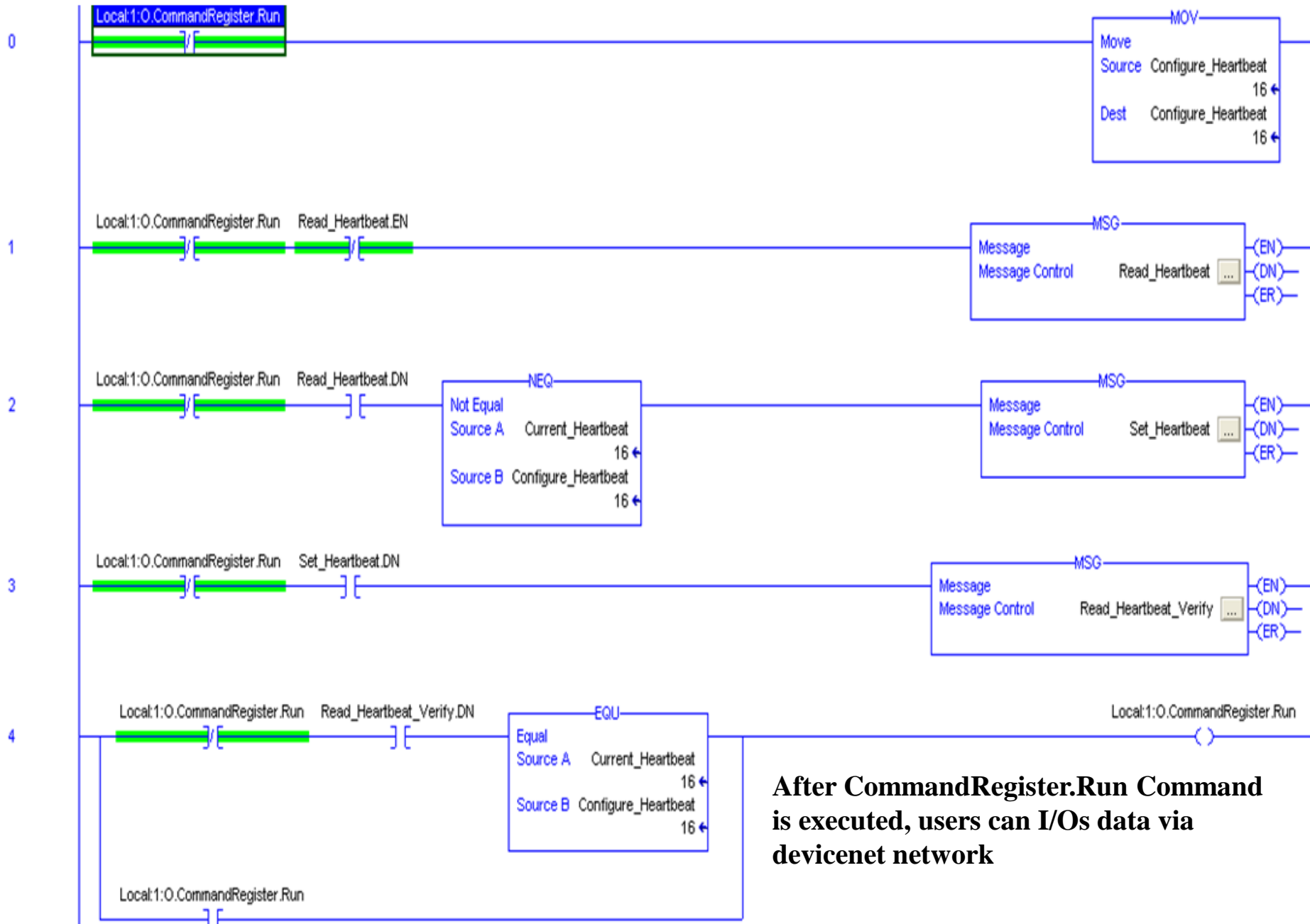
15. Reapply power to the secondary controller and/or reconnect the secondary scanner module to the DeviceNet subnet

# CREATING TAGS FOR BACKUP

Scope: test Show... Show All

Name	Alias For	Base Tag	Data Type	Style	Description
+Configure_Heartbeat			INT	Decimal	
+Current_Heartbeat			INT	Decimal	
+heartbeat			INT	Decimal	
+Local:1:I			AB:1769_SDN_1...		
+Local:1:O			AB:1769_SDN_3...		
+Read_Heartbeat			MESSAGE		
+Read_Heartbeat_Verify			MESSAGE		
+Set_Heartbeat			MESSAGE		

# WIRTING LOGIC FOR BACKUP



# CONFIGURING MSG TO GET HEARBEAT

Local:1:O.CommandRegister.Run Read\_Heartbeat.EN

Message Control Read\_Heartbeat

MSG

(EN)  
(DN)  
(ER)

### Message Configuration - Read\_Heartbeat

Configuration | Communication | Tag

Message Type: CIP Generic

Service Type: Get Attribute Single Source Element: [ ]

Source Length: 0 (Bytes)

Service Code: e (Hex) Class: 90 (Hex) Destination: Current\_Heartbeat

Instance: 1 Attribute: 12 (Hex)

New Tag...

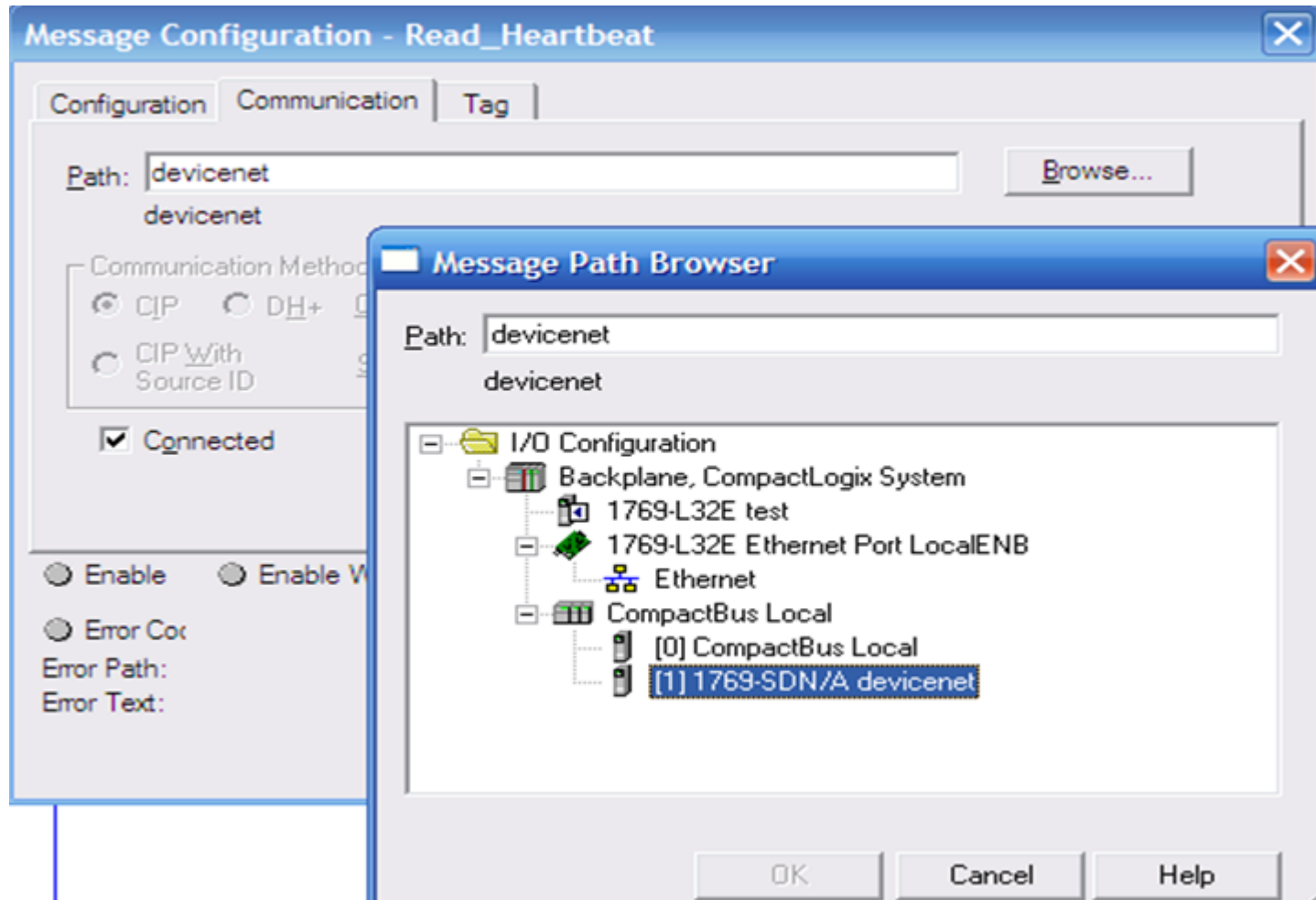
Enable  Enable Waiting  Start  Done Done Length: 0

Error Code: Extended Error Code:  Timed Out

Error Path:  
Error Text:

OK Cancel Apply Help

# SELECTING SDN TO GET HEARTBEAT



# CONFIGURING MSG TO SET HEARTBEAT



**Message Configuration - Set\_Heartbeat**

Configuration | Communication | Tag

Message Type:

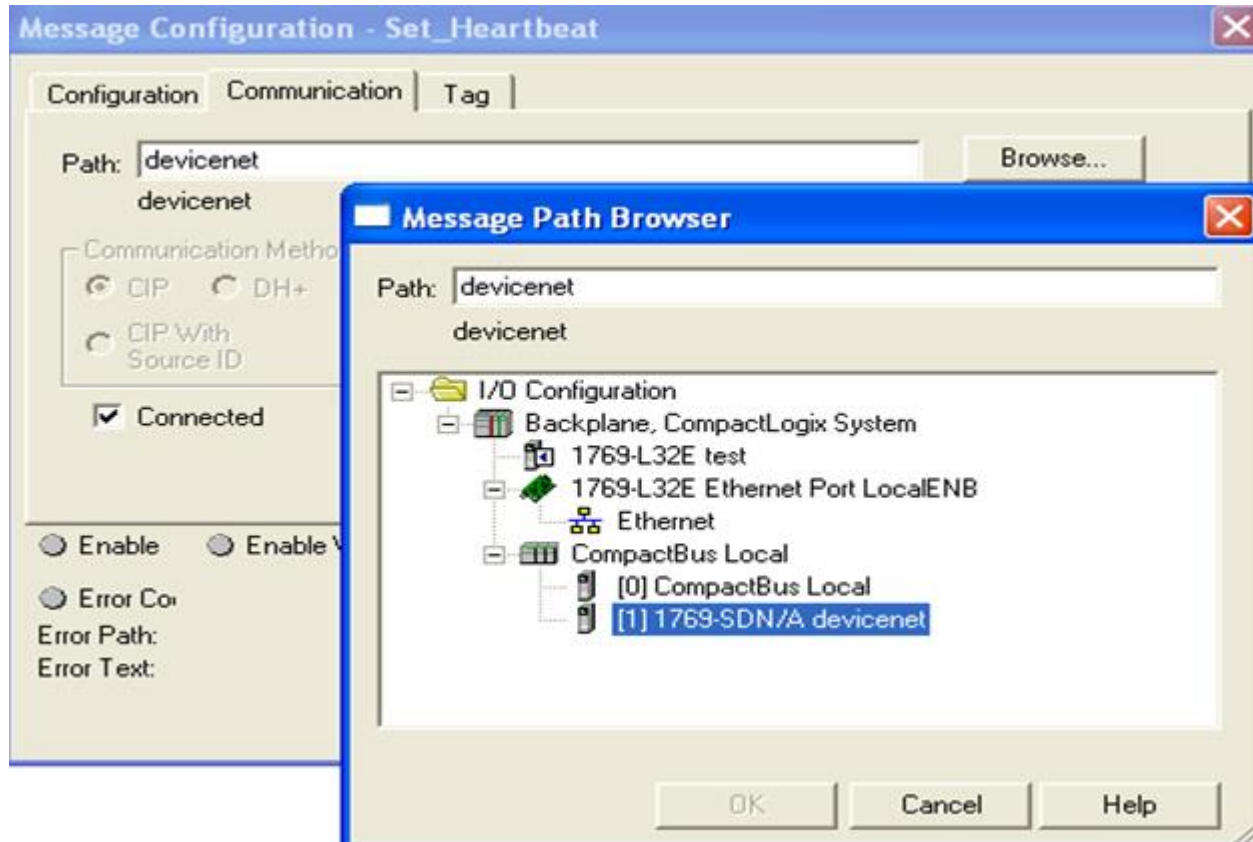
Service Type:  Source Element:

Service Code:  (Hex) Class:  (Hex) Source Length:  (Bytes)

Instance:  Attribute:  (Hex) Destination:



# SELECTING SDM TO SET HEARTBEAT



# DATA TRANSFER BETWEEN CONTROLLERS

## Transfer data via Ethernet network

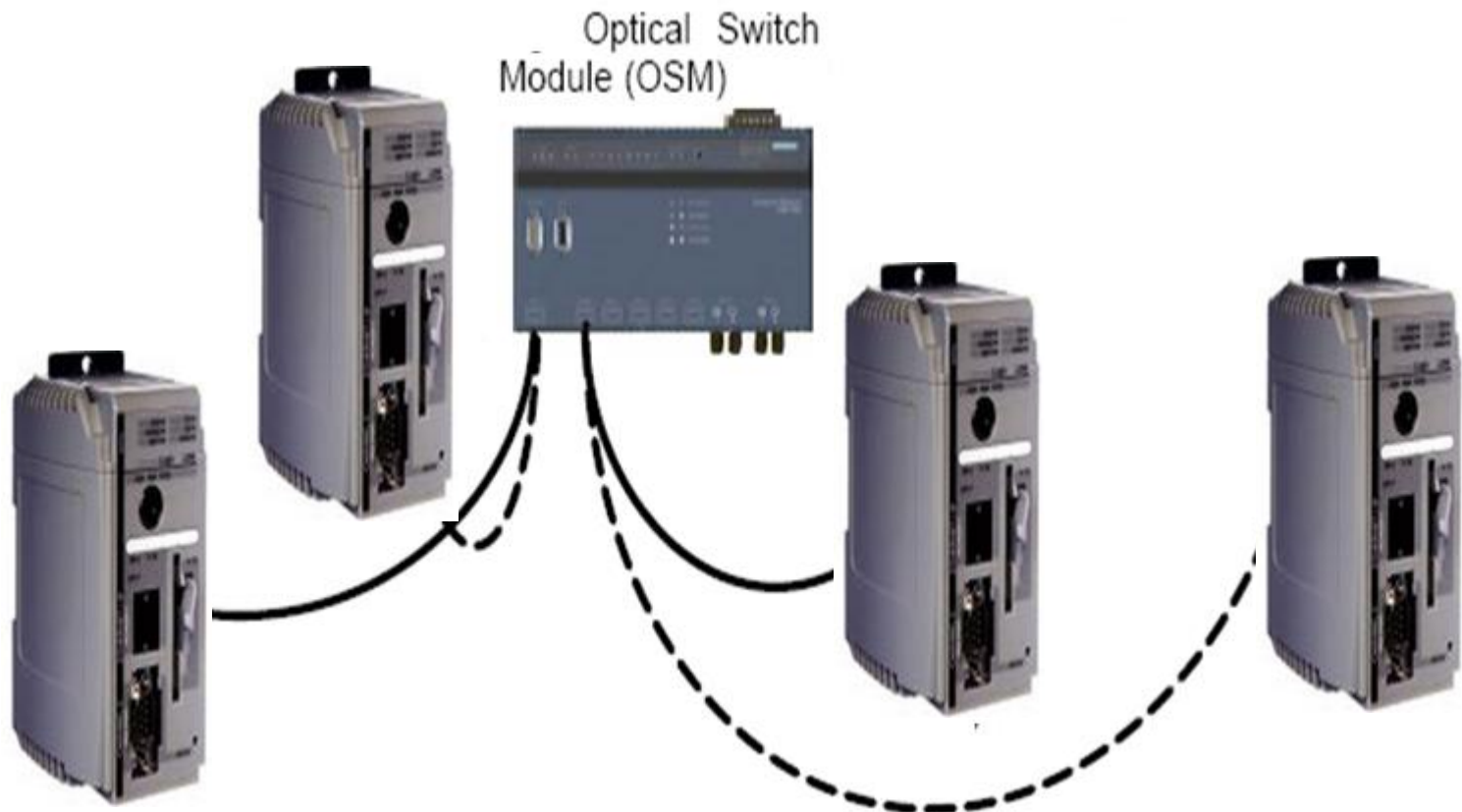
*Each device must has a different address*

192.168.1.20  
255.255.255.0

192.168.1.21  
255.255.255.0

192.168.1.24  
255.255.255.0

192.168.1.25  
255.255.255.0



# DATA TRANSFER BETWEEN CONTROLLERS

Transfer data via Ethernet network by produced and consumed tag

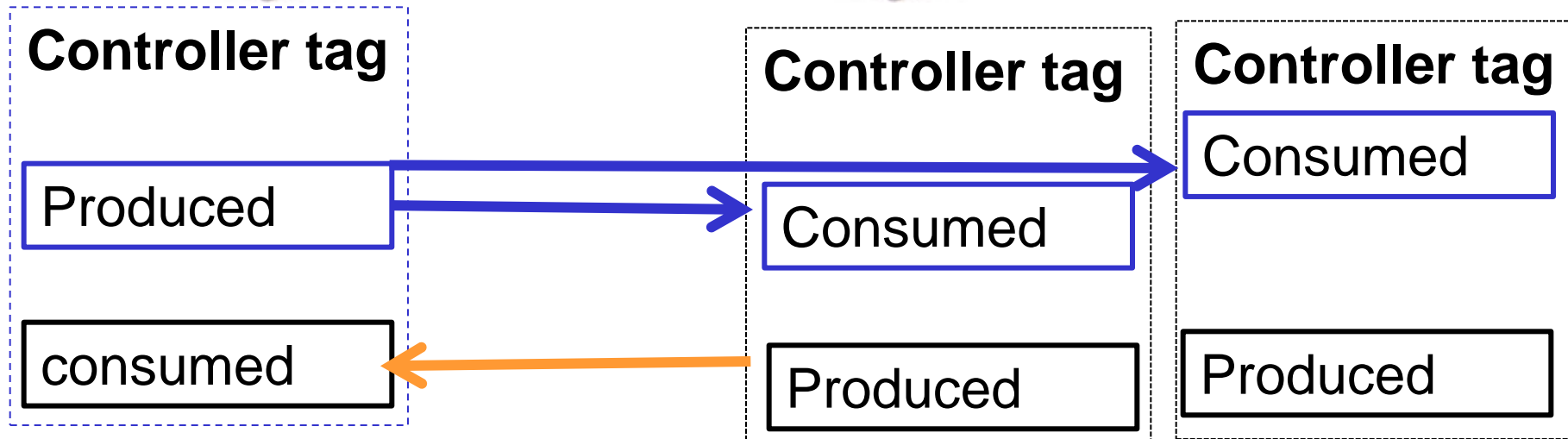
192.168.1.21  
255.255.255.0



192.168.1.2  
255.255.255.20



192.168.1.4  
255.255.255.20



# DATA TRANSFER BETWEEN CONTROLLERS

## Tag guidelines for Produced and Consumed Data

Guideline	Details
Create the tags at the controller scope.	You can share only controller-scoped tags.
Use one of these data types: <ul style="list-style-type: none"><li>· DINT</li><li>· REAL</li><li>· array of DINTs or REALs</li><li>· user-defined</li></ul>	<ul style="list-style-type: none"><li>• To share other data types, create a user-defined data type that contains the required data.</li><li>• Use the same data type for the produced tag and corresponding consumed tag or tags.</li></ul>
Limit the size of the tag to $\leq 500$ bytes.	<p>If transferring more than 500 bytes, create logic to transfer the data in packets.</p> <p>A size of <math>&lt; 125</math> DINT words will keep total bytes within 500. This helps reduce the total number of packets for transactions.</p>
Combine data that goes to the same controller.	<p>If producing several tags for the same controller:</p> <ul style="list-style-type: none"><li>• Group the data into one or more user-defined data types. This method uses fewer connections than does producing each tag separately.</li><li>• Group the data according to similar update intervals. To conserve network bandwidth, use a greater RPI for less critical data.</li></ul> <p>For example, you could create one tag for data that is critical and another tag for data that is not as critical.</p>

# DATA TRANSFER BETWEEN CONTROLLERS

## Produced and Consumed Tag Definitions

Term	Definition
Produced tag	A tag that a controller makes available for use by other controllers. Multiple controllers can simultaneously consume (receive) the data. A produced tag sends its data to one or more consumed tags (consumers) without using logic. The produced tag sends its data at the RPI of the consuming tag.
Consumed tag	A tag that receives the data of a produced tag. The data type of the consumed tag must match the data type (including any array dimensions) of the produced tag. The RPI of the consumed tag determines the period at which the data updates.

## Connections for Produced and Consumed Tags

Tag Type	Required Connections
Produced	<p>The local controller (producing) must have one connection for the produced tag and the first consumer and one more connection for each additional consumer (heartbeat). The produced tag requires two connections.</p> <p>As you increase the number of controllers that can consume a produced tag, you also reduce the number of connections the controller has available for other operations, such as communication and I/O.</p>
Consumed	Each consumed tag requires one connection for the controller that is consuming the tag.

# DATA TRANSFER BETWEEN CONTROLLERS

## Number Connections for Produced and Consumed Tags

Type of Tag	Device	Number of Connections Used
Produced tag	Logix5000 controller	Number_of_consumers + 1
	EtherNet/IP module	1
Consumed tag	Logix5000 controller	1
	EtherNet/IP module	

## Example of Number Connections for Produced and Consumed Tags

Local Controller - 1768 CompactLogix controller

Remote Controller - 1756 ControlLogix controller

Connections Used = 2 (1 + Number of consumers)

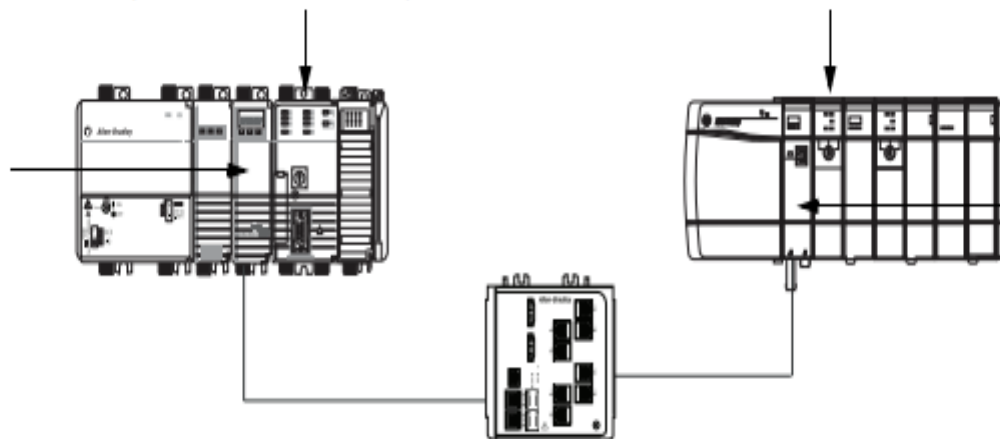
Connections Used = 1

Local Communication  
Module - 1768-ENBT

Remote Communication  
Module - 1756-EN2T

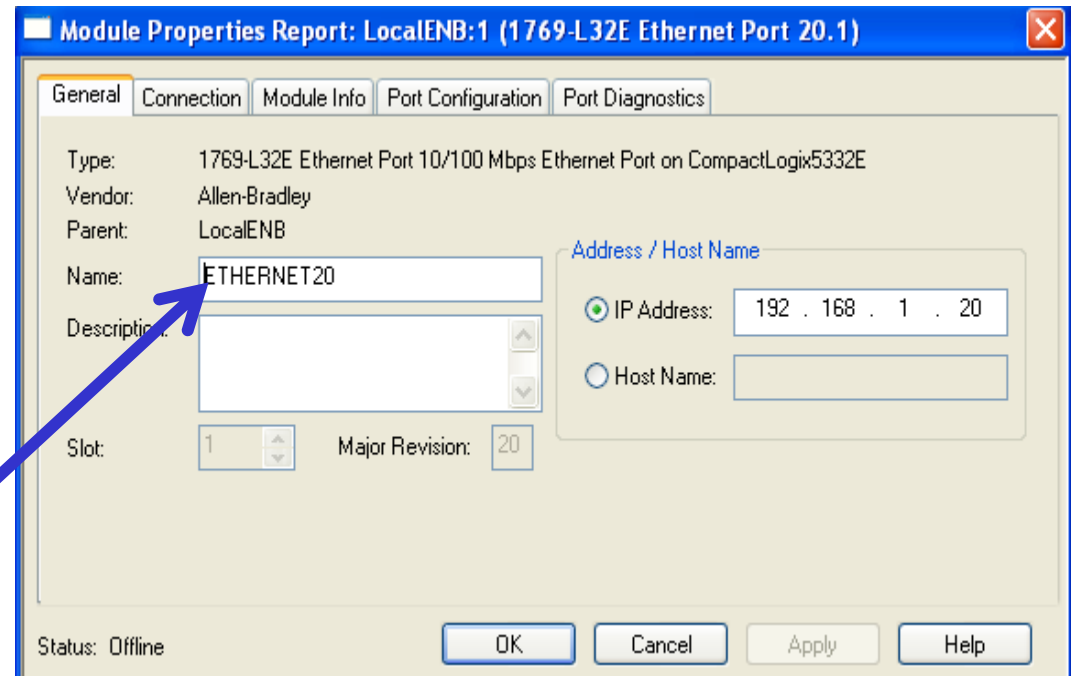
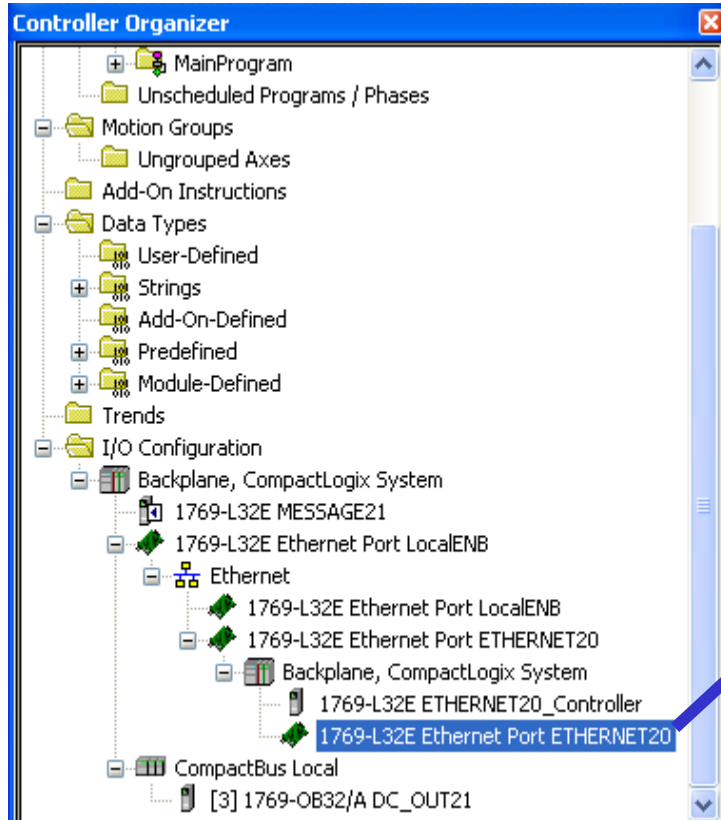
Connections Used = 1

Connections Used = 1



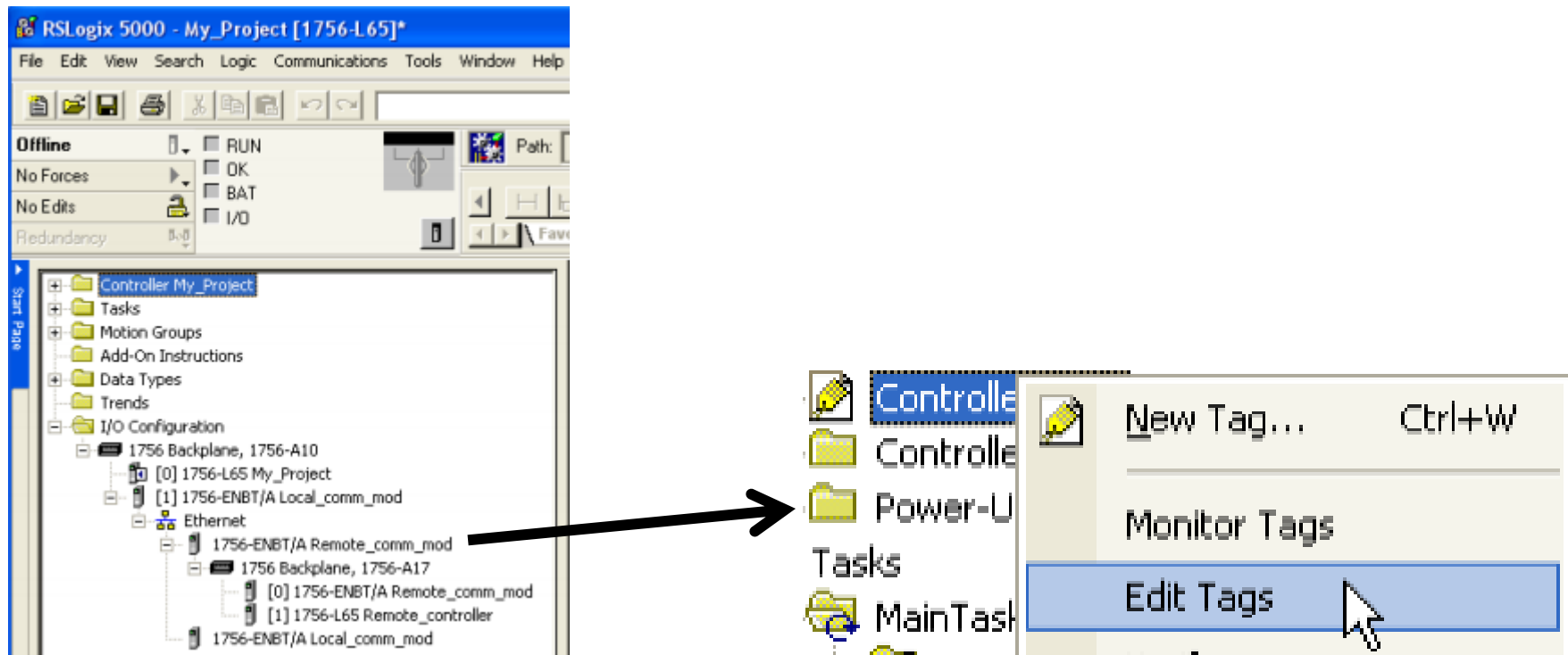
# DATA TRANSFER BETWEEN CONTROLLERS

Create and configure a Produced Tag: Add the **consumer controller** via ethernet network then create controller tags



# DATA TRANSFER BETWEEN CONTROLLERS

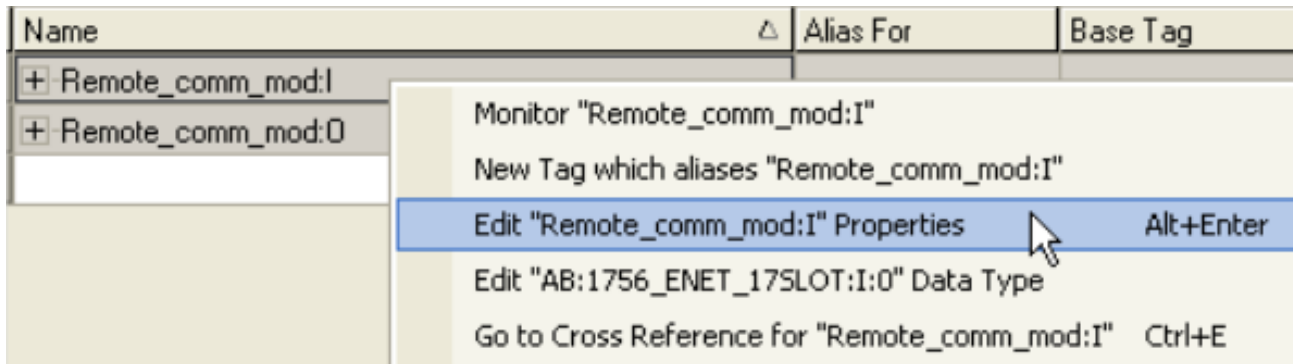
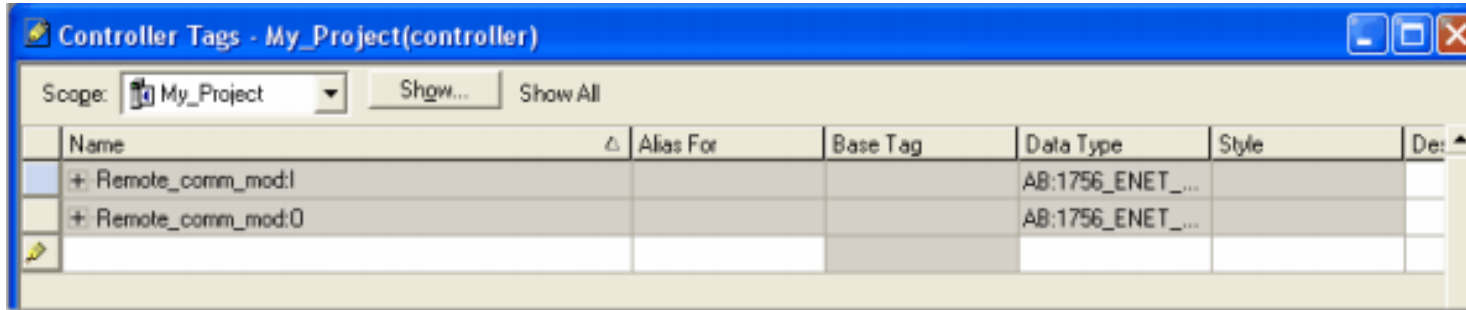
Create and configure a Produced Tag: Add the **consumer controller** via ethernet network then create controller tags





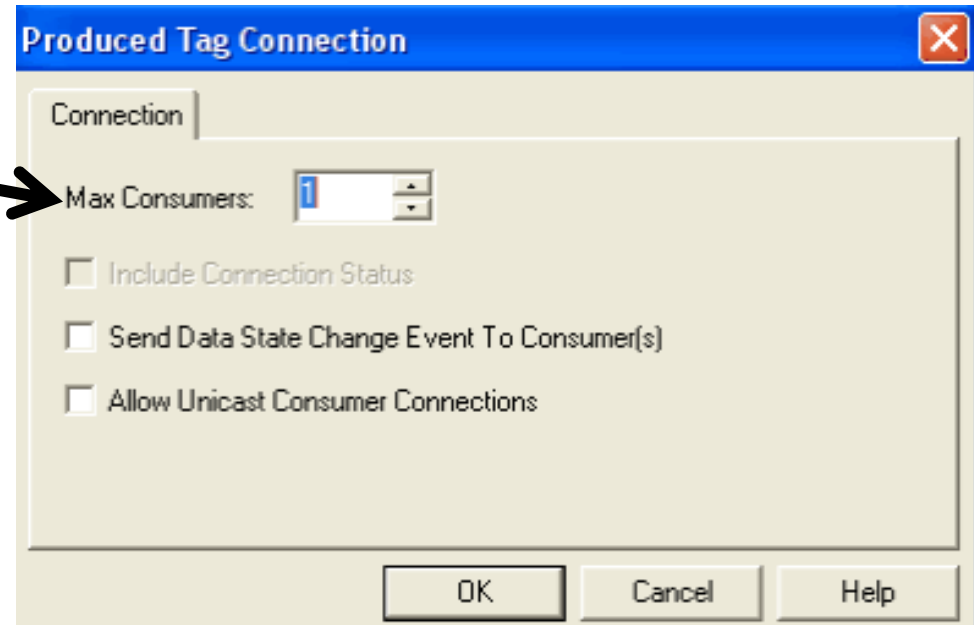
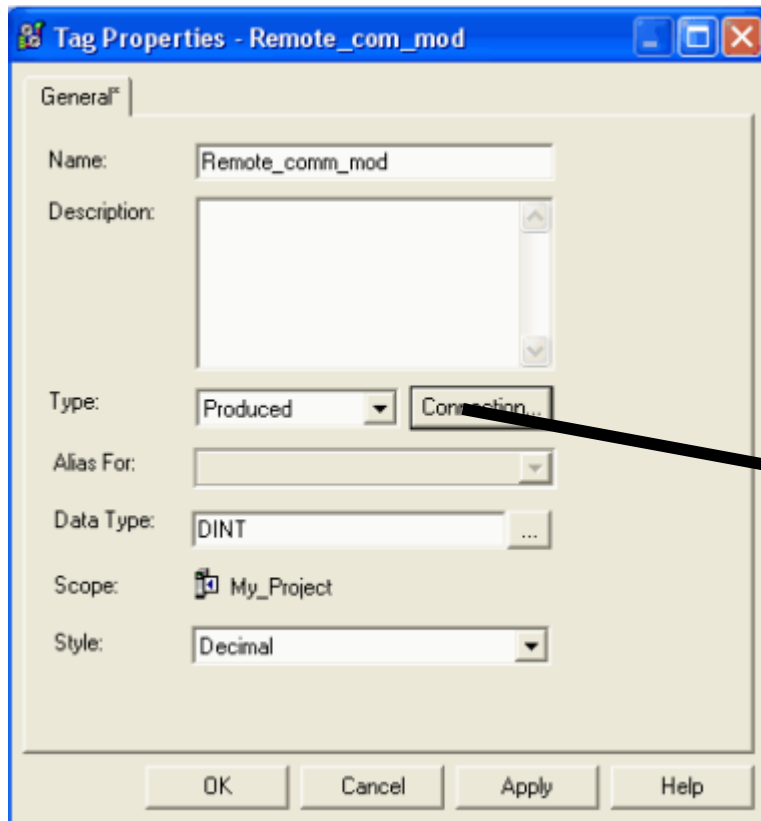
# DATA TRANSFER BETWEEN CONTROLLERS

## Edit properties of Produced Tags



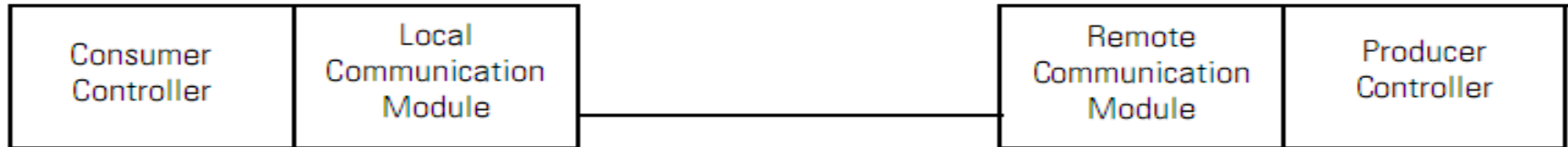
# DATA TRANSFER BETWEEN CONTROLLERS

Choose Tag type is **Produced** then choose connection tab to add Max consumers

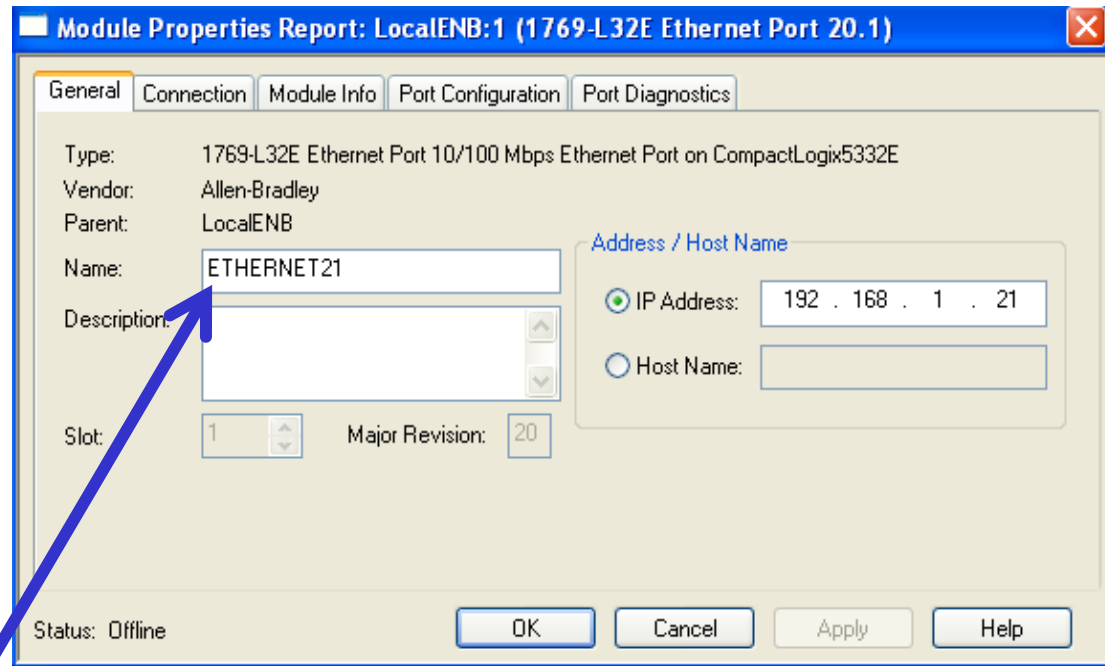
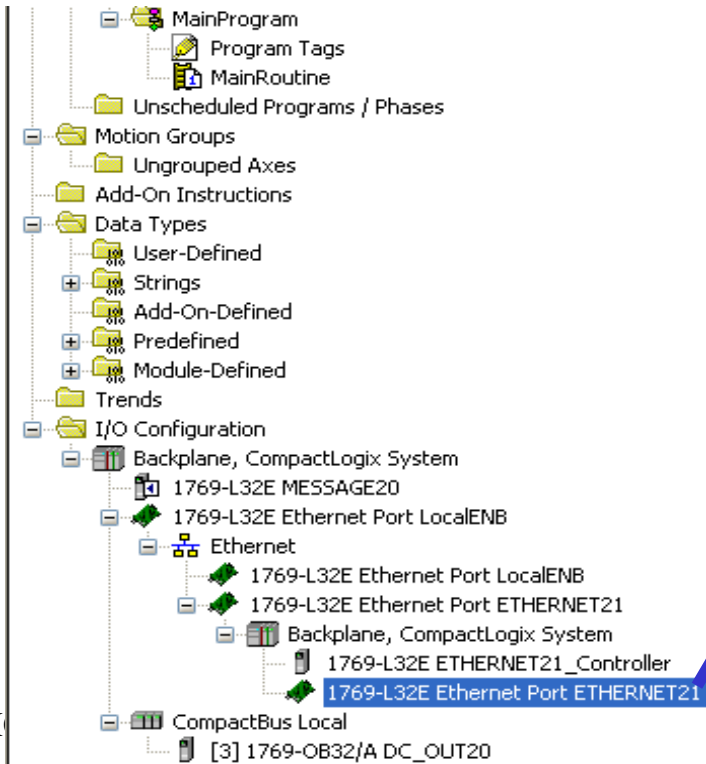


# DATA TRANSFER BETWEEN CONTROLLERS

Consumed Tag is produced and configured by another controller

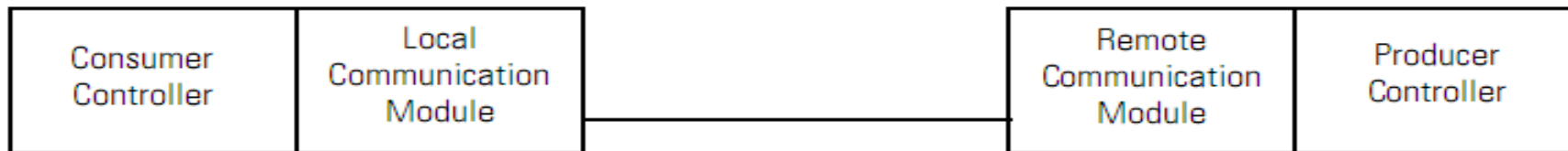


Create and configure a consumed Tag: Add the producer controller via ethernet network then create controller tags

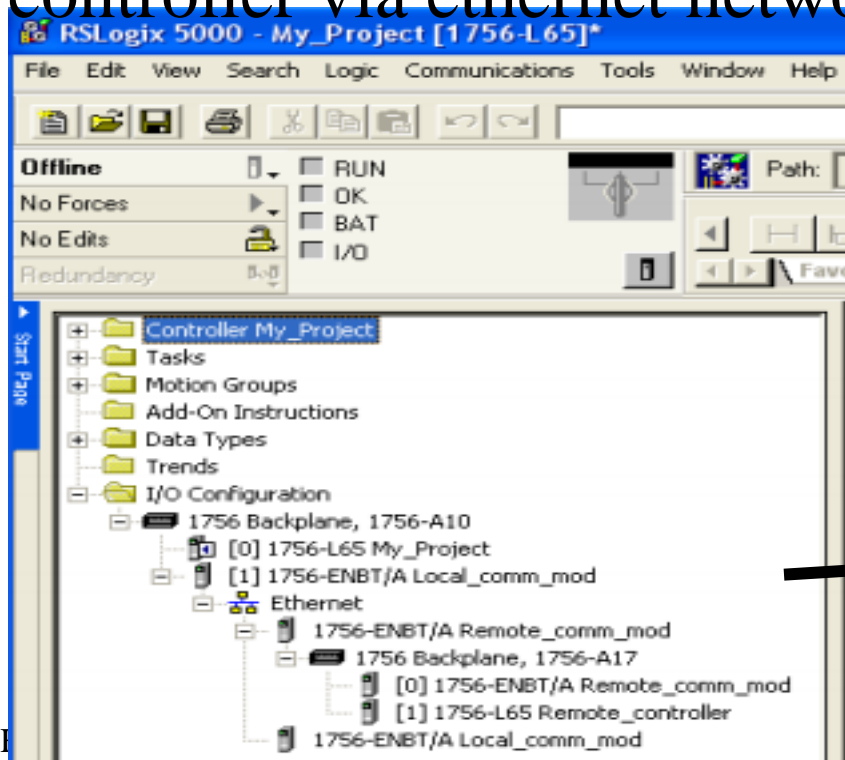


# DATA TRANSFER BETWEEN CONTROLLERS

Consumed Tag is produced and configured by another controller

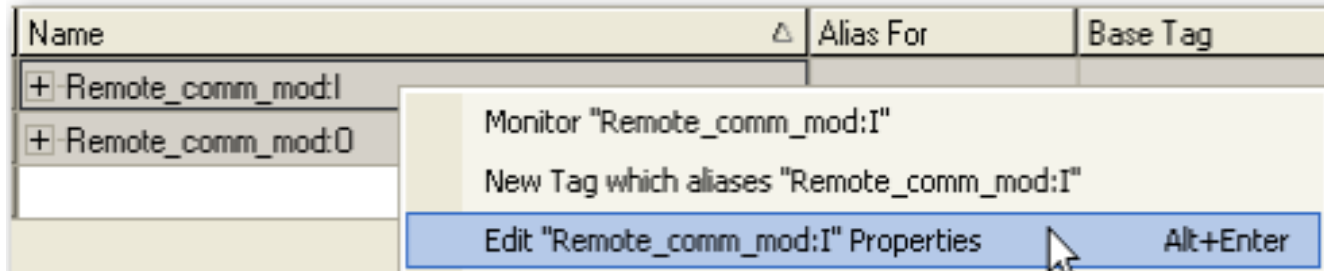


Create and configure a consumed Tag: Add the producer controller via ethernet network then create controller tags

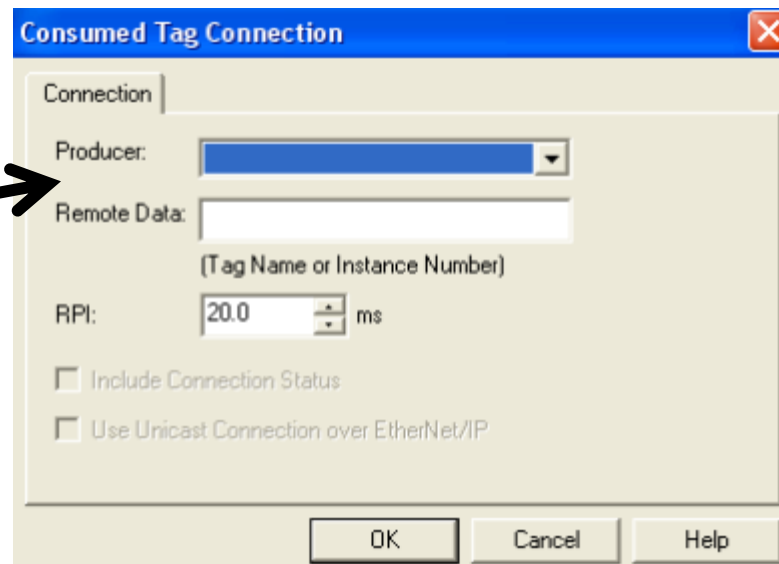
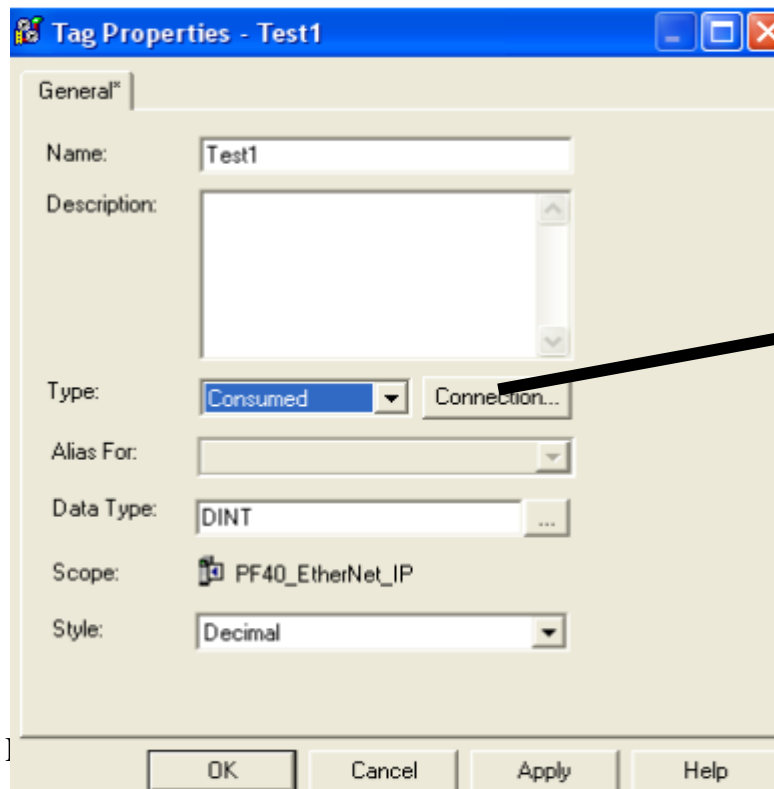


# DATA TRANSFER BETWEEN CONTROLLERS

## Edit properties of controller tag



Choose Tag type is **consumed** then choose connection tab to connect to producer controllers



# DATA TRANSFER BETWEEN CONTROLLERS

## Transfer data via Ethernet network by Message control

**Ins** 192.168.1.21  
255.255.255.0



**Controller tag**  
Tag1, tag2..

192.168.1.4  
255.255.255.20



**Controller tag**  
Tag1, tag2..

**Message control**

Data table write

Data table read

**Message control**

Data table write

Data table read

# DATA TRANSFER BETWEEN CONTROLLERS

Use Message instruction to read, write data

**Message Configuration - Message\_1**

Configuration\* | Communication | Tag

Message Type: CIP Data Table Read

Source Element:

Number Of Elements: 1

Destination Element:

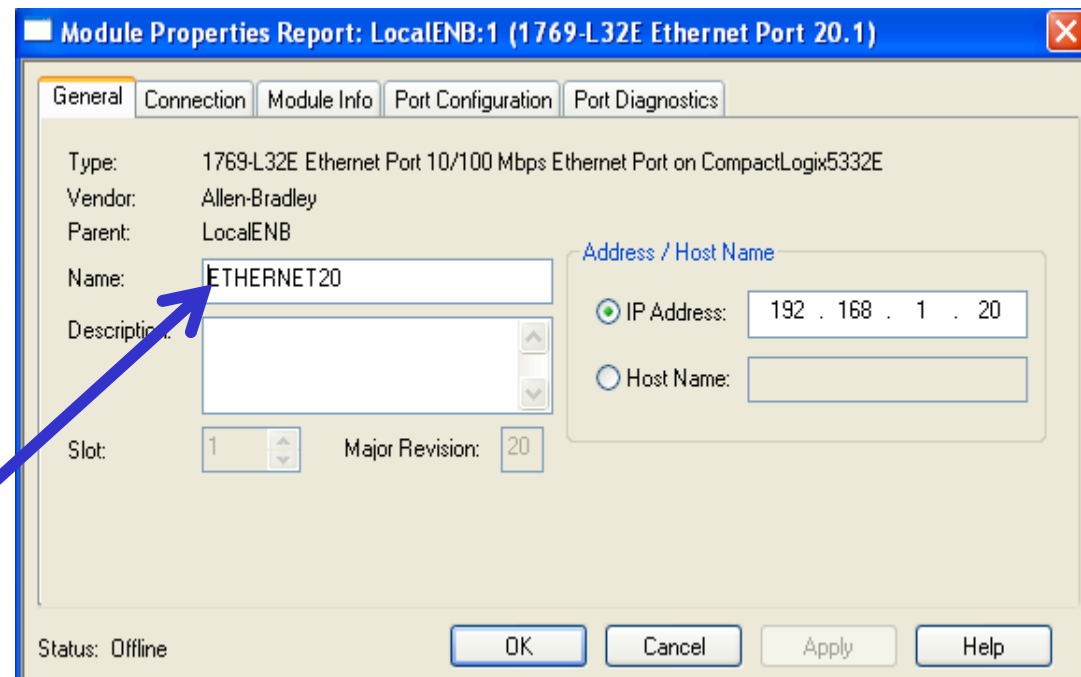
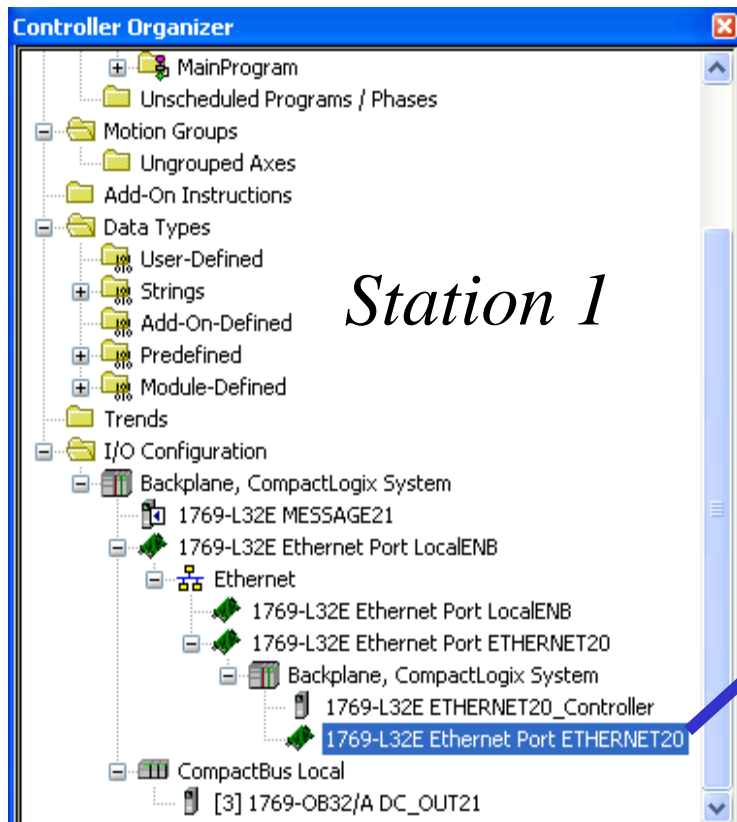
New Tag...

<b>If The Target Device Is a</b>	<b>Select One Of These Message Types</b>
Logix5000 controller	CIP Data Table Read
	CIP Data Table Write
I/O module that you configure using RSLogix 5000 software	Module Reconfigure
	CIP Generic

# DATA TRANSFER BETWEEN CONTROLLERS

Use Message instruction to read, write data

To transfer data, add an Ethernet module and set IP address of remote station in hardware configuration transfer station and vice versa.

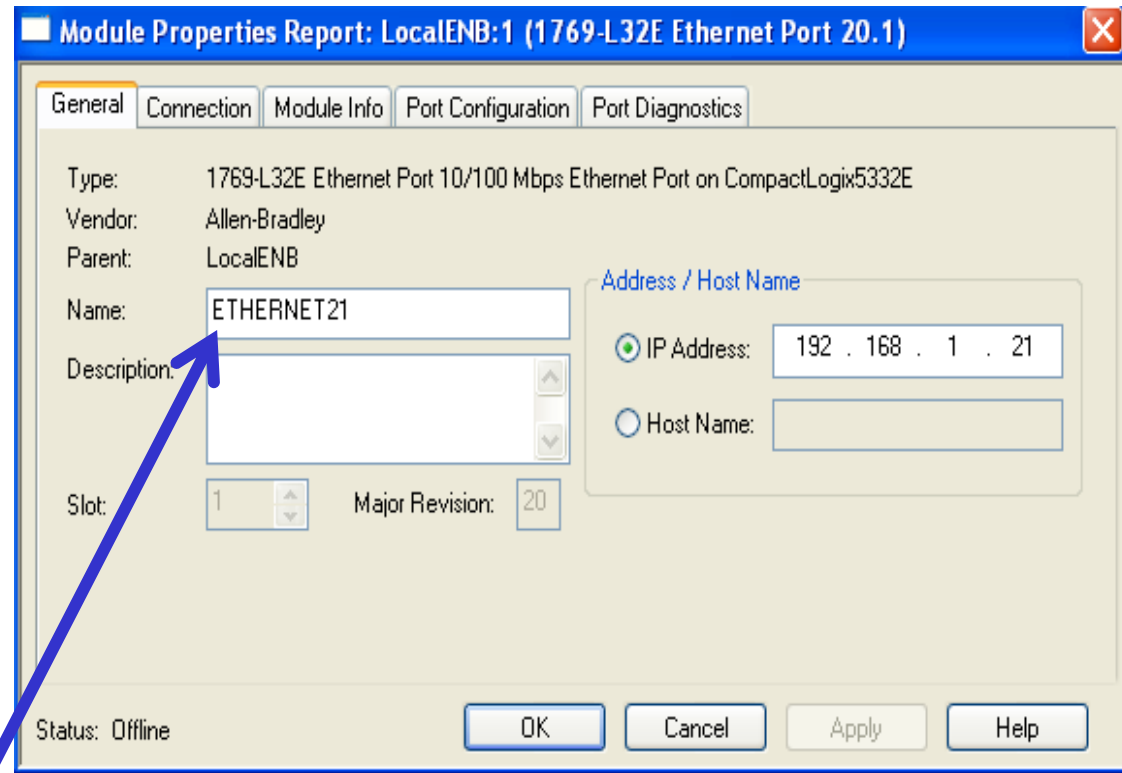
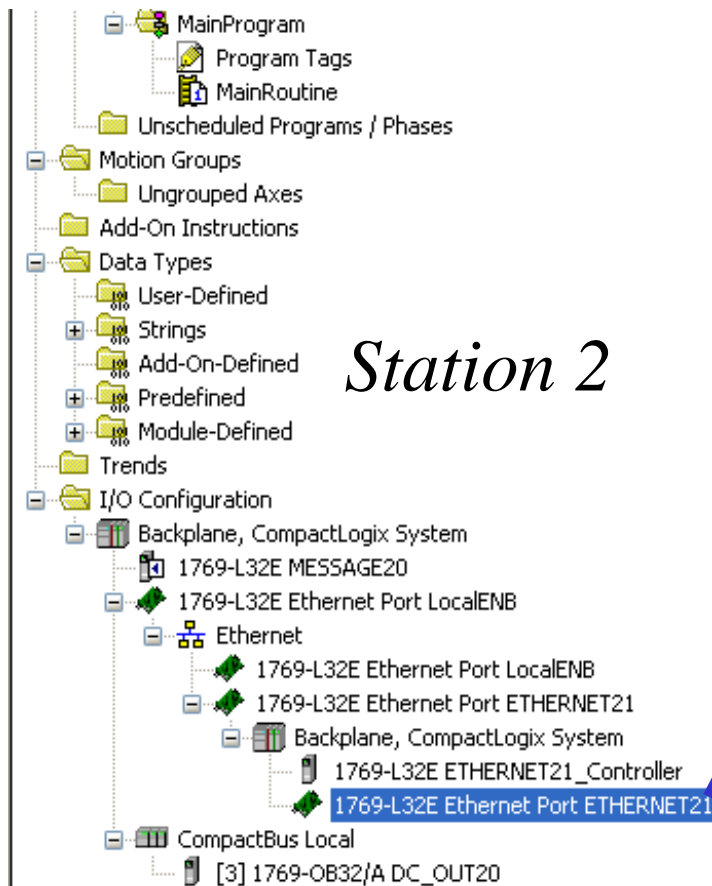




# DATA TRANSFER BETWEEN CONTROLLERS

Use Message instruction to read, write data

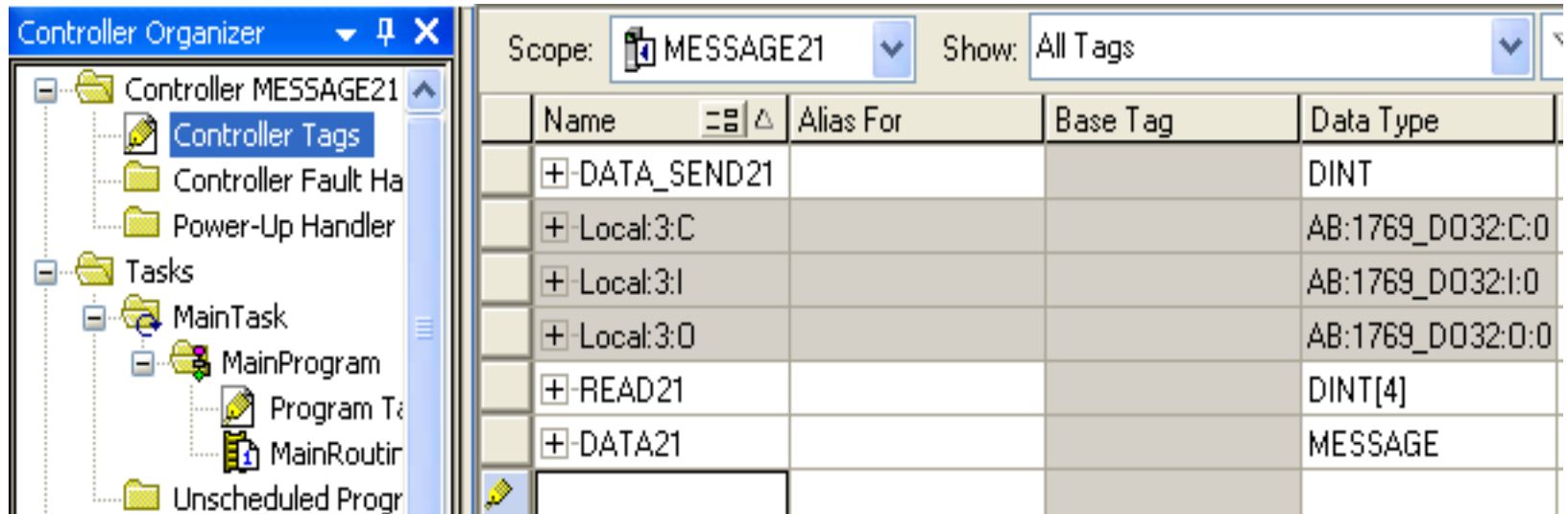
add an Ethernet module and set IP address of remote station in hardware configuration transfer station and vice versa.



# DATA TRANSFER BETWEEN CONTROLLERS

Use Message instruction to read, write data

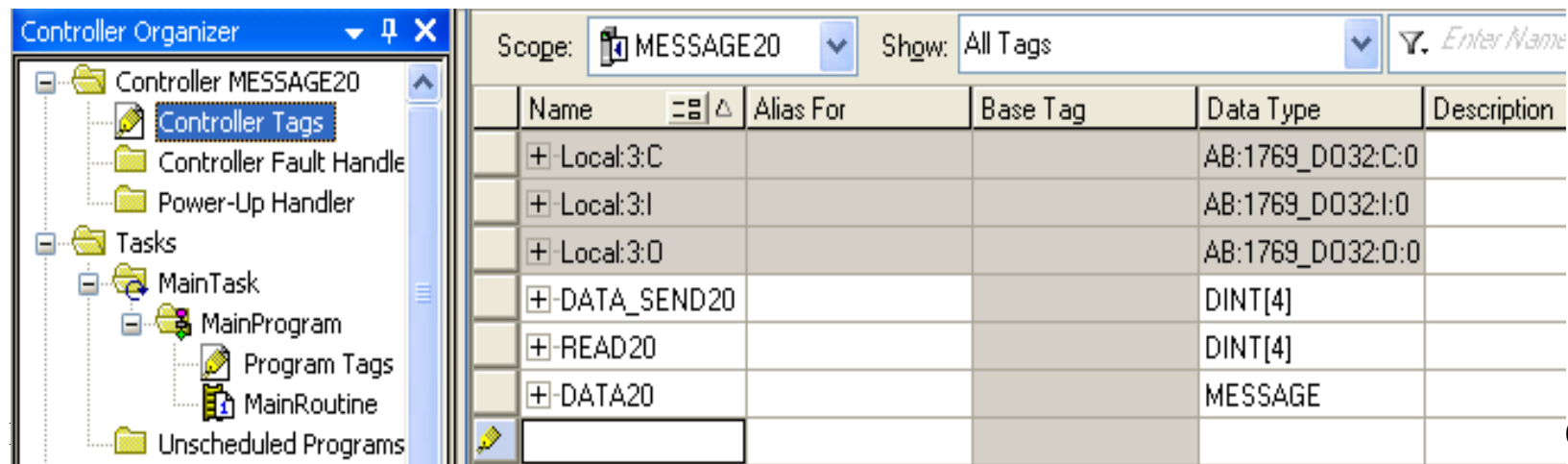
Create tags in controller tag for each station.



Controller Organizer

Scope: MESSAGE21 Show: All Tags

Name	Alias For	Base Tag	Data Type
+ DATA_SEND21			DINT
+ Local:3:C			AB:1769_D032:C:0
+ Local:3:I			AB:1769_D032:I:0
+ Local:3:O			AB:1769_D032:O:0
+ READ21			DINT[4]
+ DATA21			MESSAGE



Controller Organizer

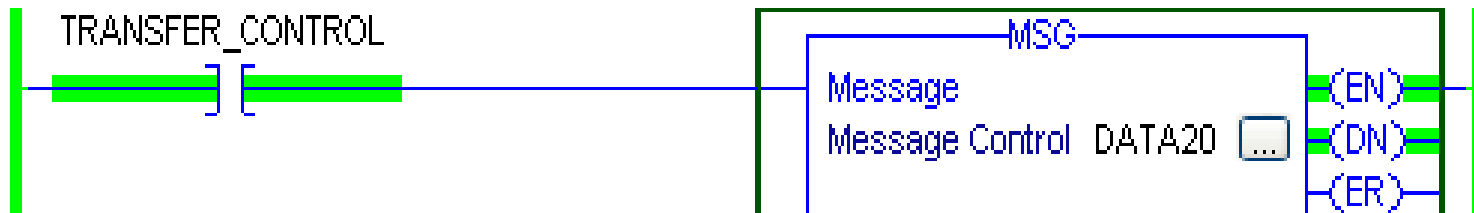
Scope: MESSAGE20 Show: All Tags

Name	Alias For	Base Tag	Data Type	Description
+ Local:3:C			AB:1769_D032:C:0	
+ Local:3:I			AB:1769_D032:I:0	
+ Local:3:O			AB:1769_D032:O:0	
+ DATA_SEND20			DINT[4]	
+ READ20			DINT[4]	
+ DATA20			MESSAGE	

# DATA TRANSFER BETWEEN CONTROLLERS

Use Message instruction to read, write data

Add and configure parameters for Message control Ins ( only for one station)



## Message Configuration - DATA20

Configuration

Communication

Tag

Message Type: CIP Data Table Write

Source Element: DATA\_SEND20

Number Of Elements: 4

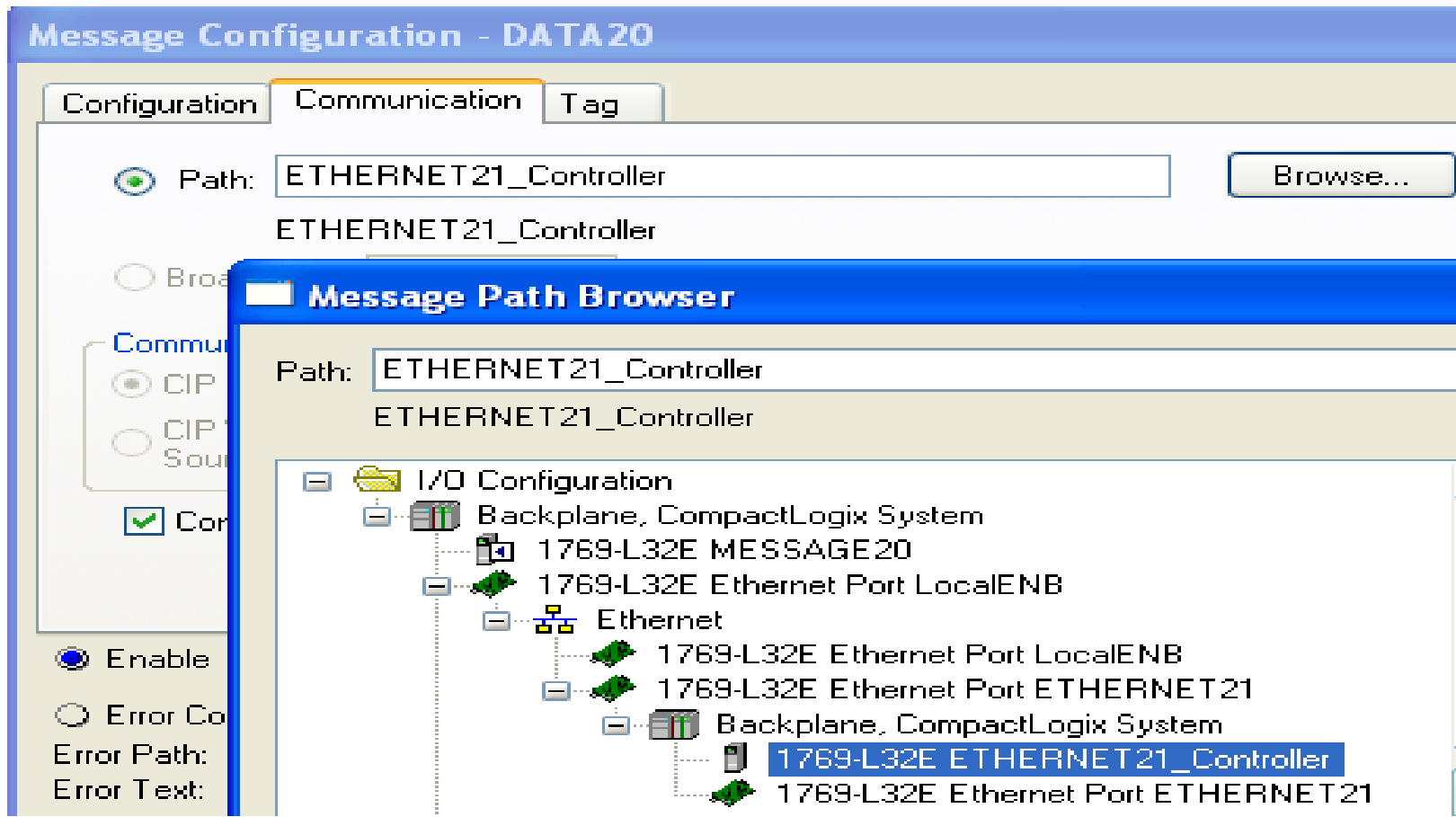
Destination Element: READ21

New Tag...

# DATA TRANSFER BETWEEN CONTROLLERS

Use Message instruction to read, write data

Add and configure parameters for Message control Ins



# DATA TRANSFER BETWEEN CONTROLLERS

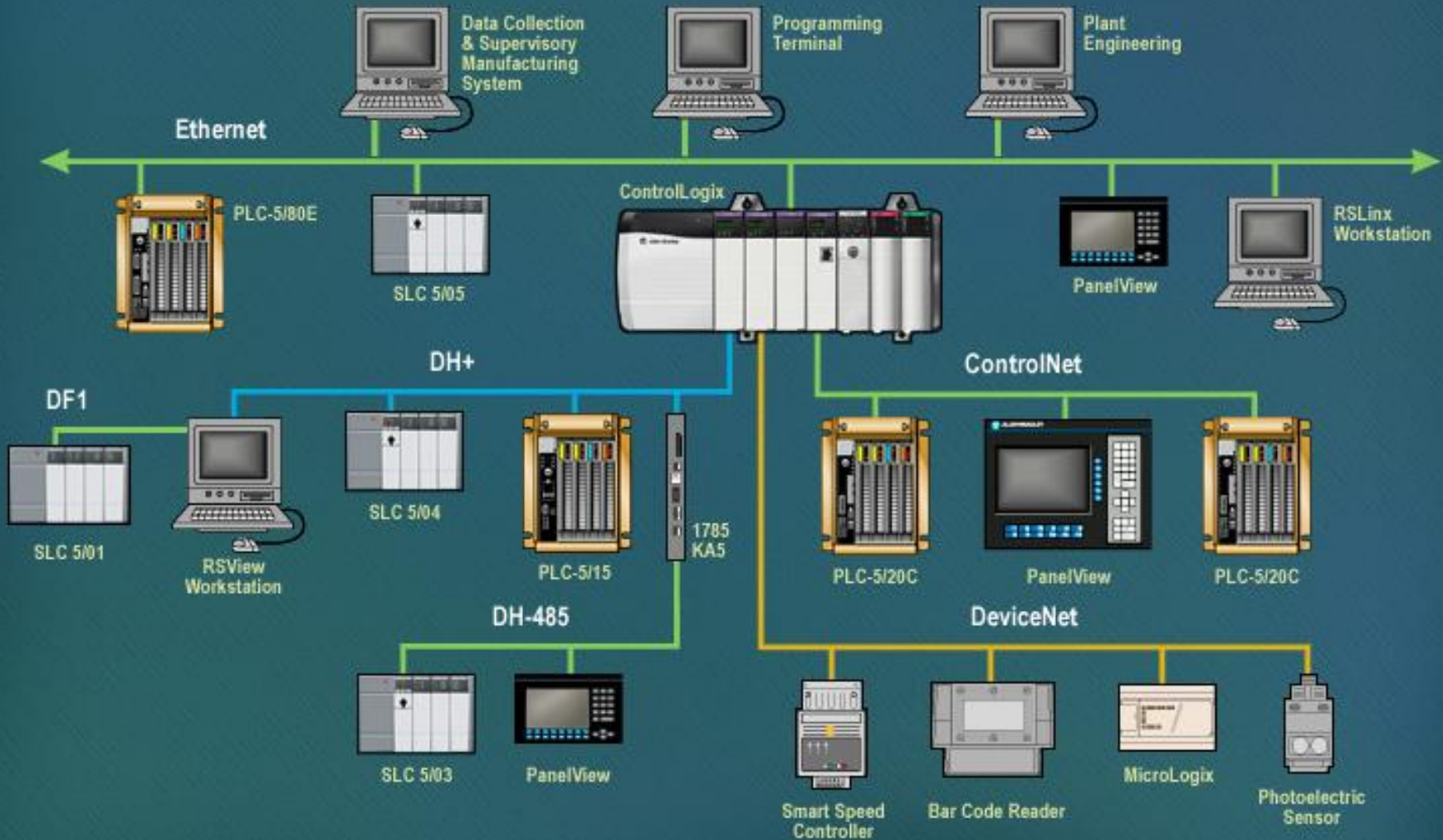
Use Message instruction to read, write data

Download program to plc and check respondent of data

The screenshot displays a PLC software interface. At the top, there are status indicators for 'Rem Run', 'No Forces', and 'No Edits'. To the right, a path is shown as 'AB\_ETHIP-1\192.168.1.20\Backplane\0\*'. Below this, there are navigation buttons for 'Bit', 'Timer/Counter', 'Input/Output', 'Compare', and 'Compute'. The 'Controller Organizer' window on the left shows a tree structure with folders like 'Controller MESSAGE20', 'Tasks', 'MainTask', and 'MainProgram'. The main window shows a table of tags for the selected scope 'MESSAGE20'.

Name	Value	Force Mask	Style
+ Local:3:C	{...}	{...}	
+ Local:3:I	{...}	{...}	
+ Local:3:O	{...}	{...}	
- DATA_SEND20	{...}	{...}	Decimal
+ DATA_SEND20[0]	0		Decimal
+ DATA_SEND20[1]	0		Decimal
+ DATA_SEND20[2]	0		Decimal
+ DATA_SEND20[3]	255		Decimal
+ READ20	{...}	{...}	Decimal
+ DATA20	{...}	{...}	
TRANSFER_CONTROL	1		Decimal

# CONTROLNET OVERVIEW



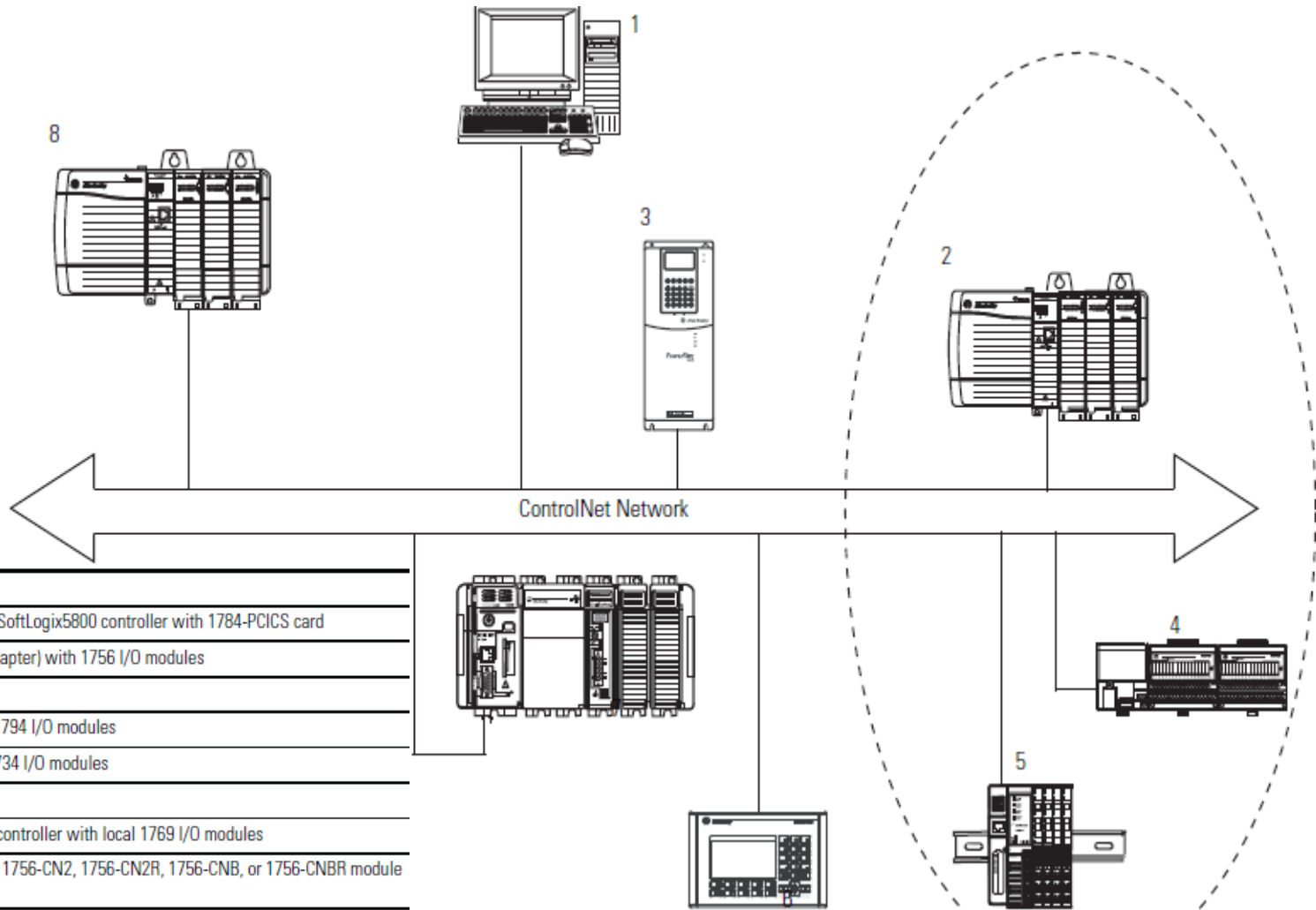
# CONTROLNET OVERVIEW

The ControlNet network provides high-speed transmission of **time-critical I/O** and **interlocking data** and **messaging data**.

The ControlNet network is most often used in these ways:

- As the default network for the ControlLogix platform
- As a backbone to multiple distributed DeviceNet networks
- As a peer interlocking network

# Use ControlNet Communication Modules in a Control System



Item	Description
1	Personal computer running SoftLogix5800 controller with 1784-PCICS card
2	1756-CNB module (as an adapter) with 1756 I/O modules
3	PowerFlex 700S drive
4	1794-ACN15 adapter with 1794 I/O modules
5	1734-ACNR adapter with 1734 I/O modules
6	PanelView terminal
7	CompactLogix 1769-L35CR controller with local 1769 I/O modules
8	ControlLogix controller with 1756-CN2, 1756-CN2R, 1756-CNB, or 1756-CNBR module as the scanner



# Use ControlNet Communication Modules in a Control System

- The controllers produce and consume tags among themselves.
- The controllers initiate MSG instructions that send/receive data or configure devices.
- The computer uploads and downloads projects to the controllers.
- The computer configures devices on the ControlNet network and configures the network itself.

# Exchange Information on a ControlNet Network

- With unscheduled data, the device from which a message originates, such as a Logix5000 controller, contains the path information that makes sure the message reaches its consumers
- Scheduled data in Logix-based systems use the producer/consumer networking model
- The controller can also produce data for other controllers to consume.
- Digital input modules produce (multicast) their data either upon a change of state (COS) or periodically

# Control I/O Via ControlNet Network

- With unscheduled data, the device from which a message originates, such as a Logix5000 controller, contains the path information that makes sure the message reaches its consumers
- Scheduled data in Logix-based systems use the producer/consumer networking model
- The controller can also produce data for other controllers to consume.
- Digital input modules produce (multicast) their data either upon a change of state (COS) or periodically

# CONTROLNET MODULES

1756-CNB



1756-CNBR



Select Module

Module	Description	Vendor
1756-CN2/A	1756 ControlNet Bridge	Allen-Bra
1756-CN2/B	1756 ControlNet Bridge	Allen-Bra
1756-CN2R/A	1756 ControlNet Bridge	Allen-Bra
1756-CN2R/B	1756 ControlNet Bridge	Allen-Bra
1756-CNB/A	1756 ControlNet Bridge	Allen-Bra
1756-CNB/B	1756 ControlNet Bridge	Allen-Bra
1756-CNB/D	1756 ControlNet Bridge	Allen-Bra
1756-CNB/E	1756 ControlNet Bridge	Allen-Bra
1756-CNBR/A	1756 ControlNet Bridge, Redundant Media	Allen-Bra
1756-CNBR/B	1756 ControlNet Bridge, Redundant Media	Allen-Bra
1756-CNBR/D	1756 ControlNet Bridge, Redundant Media	Allen-Bra
1756-CNBR/E	1756 ControlNet Bridge, Redundant Media	Allen-Bra
1756-DHRIO/B	1756 DH+ Bridge/RIO Scanner	Allen-Bra

Find... Add Favorite

By Category By Vendor Favorites

OK Cancel Help

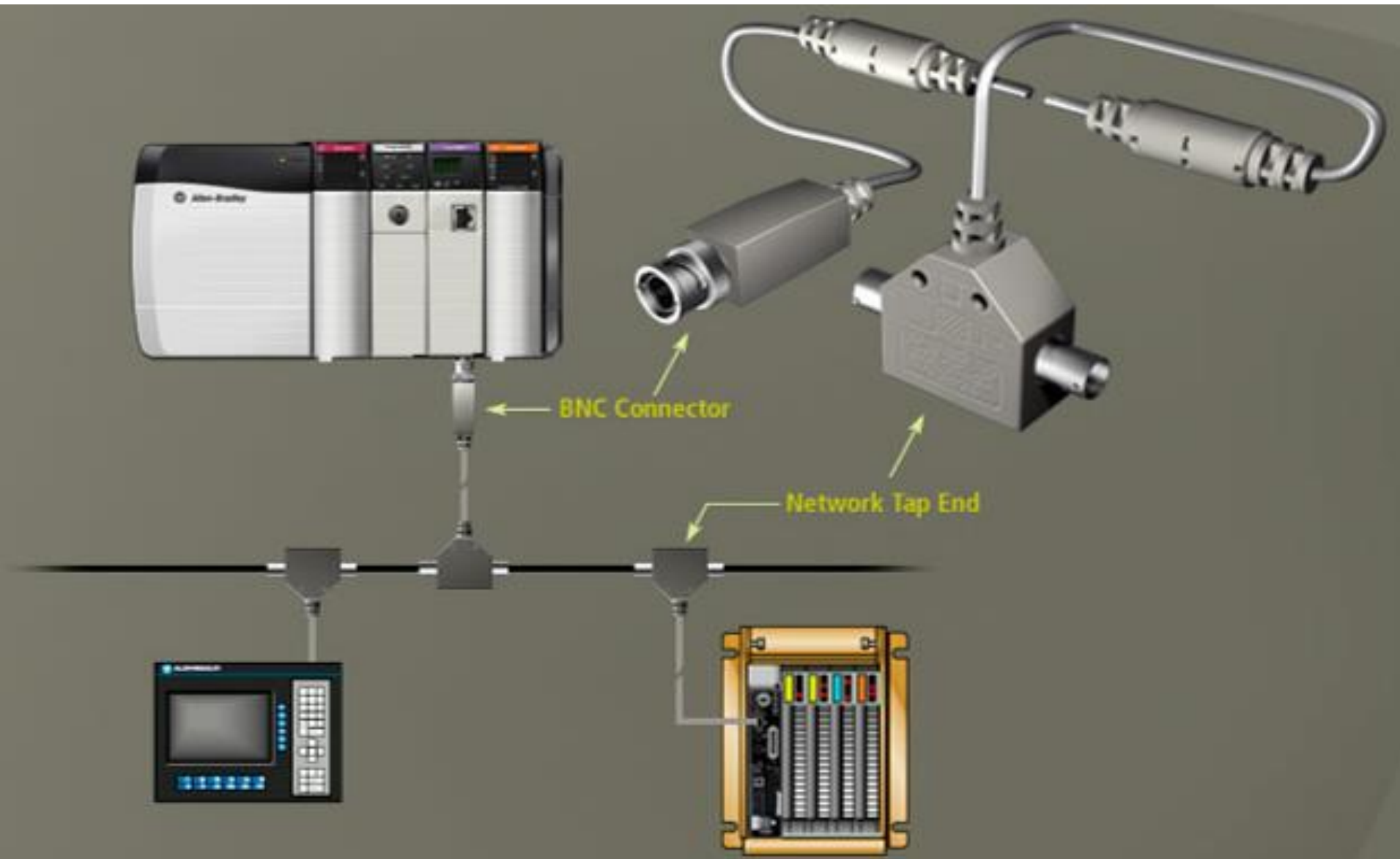
# INSTALLING CONTROLNET MODULES



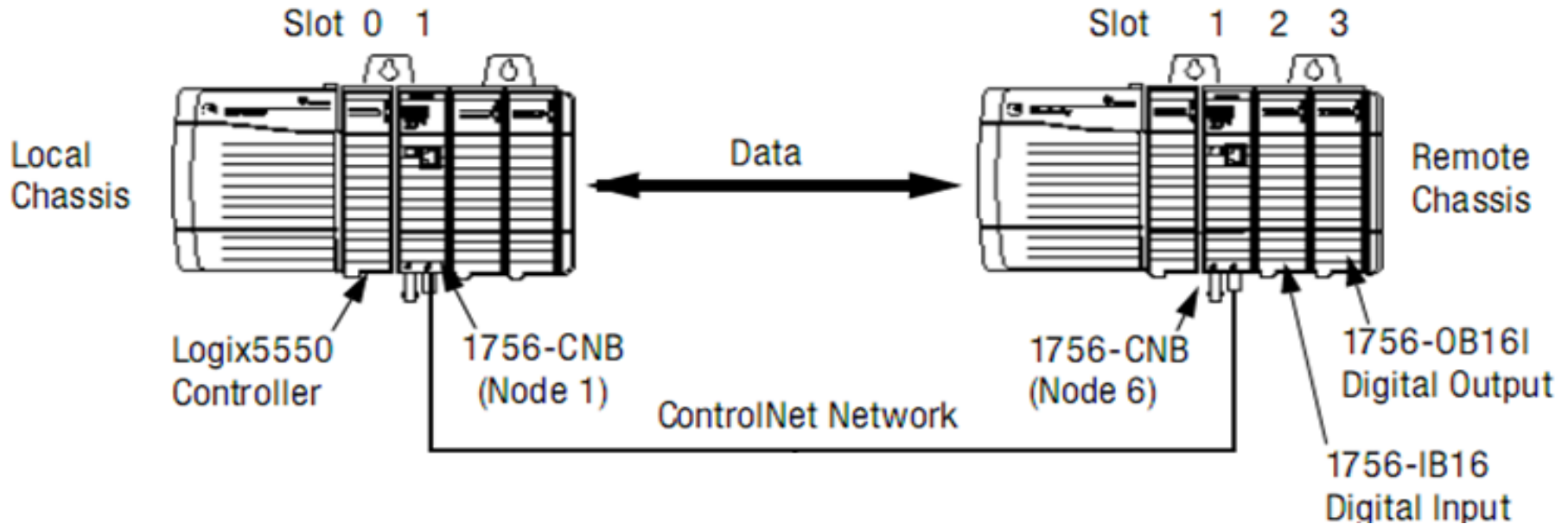
# SETTING MODULE ADDRESS



# CONTROLNET CABLE



# Control I/O Via ControlNet Network



When configure a remote ControlNet communication module or an I/O module, connection format must be choosen. There are two type of communication format **Direct or Rack-optimized connection**



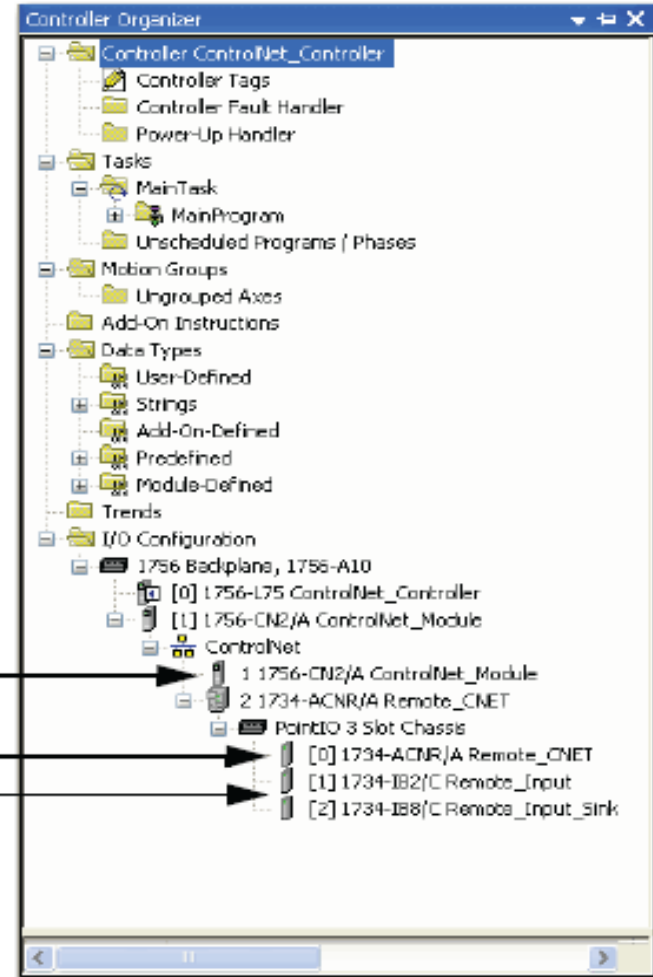
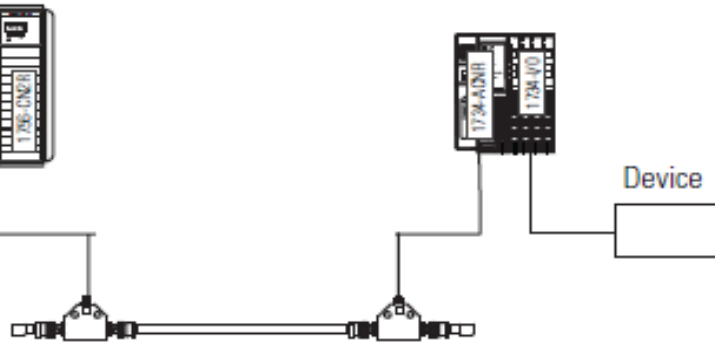
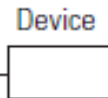
# Control I/O Via ControlNet Network

## Add Distributed I/O

Controller and Local  
Communication Modules



Remote Adapter  
and I/O Modules



Local Communication Bridge Module

Remote Adapter for the Distributed I/O Chassis or DIN Rail  
Distributed I/O Module

# Control I/O Via ControlNet Network

## Validate Connections

The screenshot displays the RSLogix 5000 software interface. The main window title is "RSLogix 5000 - Controllogix20 in POWERFLEX\_F1\_GOOD.ACD [1756-L61 20.3]\*". The interface includes a menu bar (File, Edit, View, Search, Logic, Communications, Tools, Window, Help), a toolbar with various icons, and a status bar. The "Rem Run" section shows the system is in "Run Mode" with "Controller OK", "Battery OK", and "I/O Not Responding" indicators. The "Path" field is set to "AB\_ETHIP-1\192.168.1.150\Backplane\0\*".

The "Controller Organizer" pane on the left shows the project structure under "Module-Defined". The "I/O Configuration" section is expanded to show "1756 Backplane, 1756-A10" containing several modules, including "1756-CNBR/E LOCAL\_CONTROLNET" (module 7). This module is highlighted, and its properties are displayed in the "Module Properties Report" pane on the right.

The "Module Properties Report: LOCAL\_CONTROLNET:0 (1756-CNBR/E 11.1)" pane has tabs for "General", "Connection", "Module Info", and "Backplane". The "Connection" tab is active, showing the "Requested Packet Interval (RPI)" set to 20.0 ms. Other options include "Inhibit Module", "Major Fault On Controller If Connection Fails While in Run Mode", and "Use Scheduled Connection over ControlNet" (checked). The "Module Fault" section shows a fault with the code 16#0317 and the message "Connection Request Error: Connection not scheduled." The status at the bottom is "Status: Faulted".

# CHOOSING COMMUNICATION FORMAT

The screenshot displays a software configuration window titled "New Module" with a search bar containing "1756". On the left, a tree view shows a project structure under "I/O Configuration" with "1756 Backplane, 1756-A10" expanded to show modules [0] through [7], including "1756-CNBR/E LOCAL\_CONTROLNET".

The "New Module" dialog box is open, showing the following configuration details:

- Type: 1756-CNBR/E 1756 ControlNet Bridge, Redundant Media
- Vendor: Allen-Bradley
- Parent: LOCAL\_CONTROLNET
- Name: REMOTE\_CONTROLNET
- Node: 1
- Chassis Size: 10
- Slot: 0
- Comm Format: Rack Optimization (dropdown menu is open showing options: Rack Optimization, Listen Only - Rack Optimization, None)
- Revision: None
- Keying: Compatible Keying

At the bottom of the dialog, there is a checked checkbox for "Open Module Properties" and buttons for "OK", "Cancel", and "Help". Below the dialog, a status bar indicates "4 of 116 Module Types Found" and an "Add to Favorites" button. At the very bottom of the window, there are "Close on Create", "Create", "Close", and "Help" buttons.

# CHOOSING COMMUNICATION FORMAT

## Communication Format with a Remote ControlNet Communication Module

Communication Format with a Remote ControlNet Communication Module	Criteria for Use
None	<ul style="list-style-type: none"><li>• When all of the remote I/O communicating with a controller via the remote ControlNet communication module use a Direct Connection communication format</li><li>• When the connection is used for scheduled peer interlocking</li><li>• When I/O will be mostly direct connections</li><li>• When multiple controllers control the outputs in the chassis</li></ul>
Rack-optimized	<ul style="list-style-type: none"><li>• When some or all of the remote I/O communicating with a controller via the remote ControlNet communication module use a rack-optimized communication format</li><li>• To minimize ControlNet network bandwidth when using large volume of digital I/O</li><li>• If only one controller will control the I/O</li></ul>
Rack-optimized—Listen only	When some or all of the remote I/O communicating with a controller via the remote ControlNet communication module use a rack-optimized communication format

# CHOOSING COMMUNICATION FORMAT

## Communication Format with a Remote ControlNet Communication Module

The image shows a screenshot of the Rockwell Automation software interface. On the left is the 'Controller Organizer' tree view, and on the right is the 'Select Module Type' dialog box.

**Controller Organizer (Left):**

- Tasks
  - MainTask
    - MainProgram
    - Unscheduled Programs / Phases
  - Motion Groups
    - Ungrouped Axes
  - Add-On Instructions
  - Data Types
    - User-Defined
    - Strings
      - Add-On-Defined
      - Predefined
    - Module-Defined
  - Trends
  - I/O Configuration
    - 1756 Backplane, 1756-A10
      - [0] 1756-L61 ControlLogix20
      - [1] 1756-ENBT/A Ethernet
        - Ethernet
      - [2] 1756-DMB DeviceNet
        - DeviceNet
      - [3] 1756-IB16D DI
      - [4] 1756-OB16D D\_O
      - [5] 1756-IA16 A\_I
      - [6] 1756-OA16 A\_O
      - [7] 1756-CNBR/E LOCAL\_CONTROLNET
    - ControlNet
      - 1 1756-CNBR/E REMOTE\_CONTROLNET
        - 1756 Backplane, 1756-A10
          - [0] 1756-CNBR/E REMOTE\_COM
          - [1] 1756-IB16

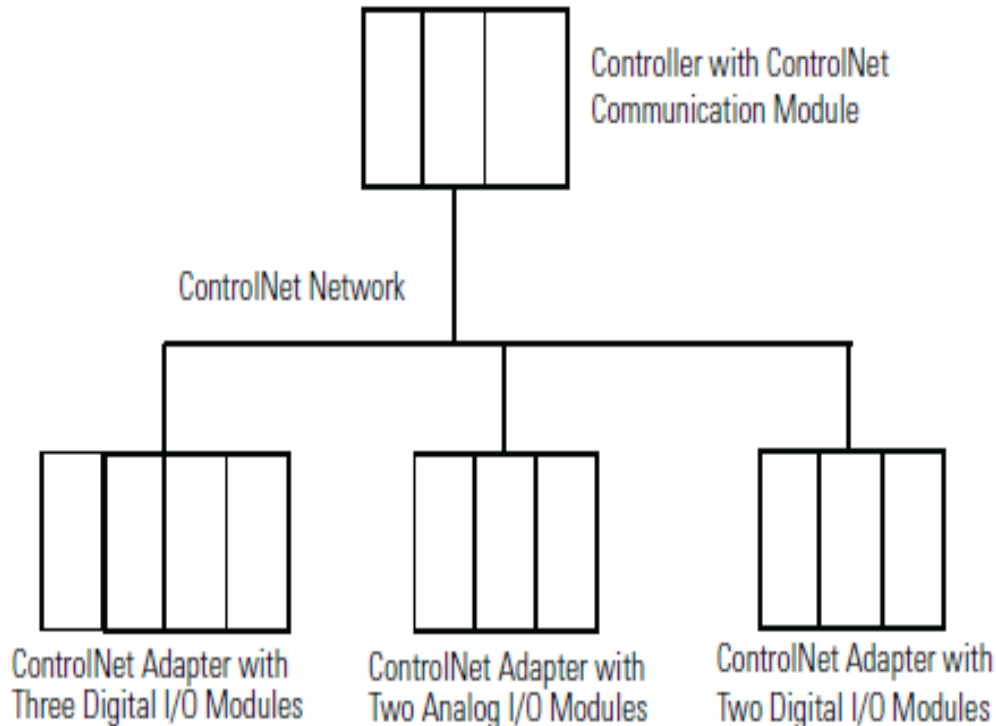
# CHOOSING COMMUNICATION FORMAT

## Communication Format with a Remote ControlNet Communication Module

I/O Module Type	Desired Connection	Required Communication Format
Digital module	Rack-optimized connection	Rack-optimized
	Direct connection or to use specialty features of the module, such as diagnostics, timestamps, or electronic fuses or Connection for listening to data from the module	<ul style="list-style-type: none"><li>• The data your controller needs from the I/O module. For example, if your application uses a 1756-IA16I module in a remote chassis that must provide timestamped input data, choose the CST Timestamped Input Data communication format.</li><li>• A listen-only communication format that matches the data the I/O module is broadcasting to other controllers.</li></ul>
Analog module	Direct connection or to use specialty features of the module, such as diagnostics, timestamps, or electronic fuses or Connection for listening to data from the module	<ul style="list-style-type: none"><li>• The data your controller needs from the I/O module. For example, if your application uses a 1756-OF6CI module in a remote chassis that must provide floating point output data, choose the Float Data communication format.</li><li>• A listen-only communication format that matches the data the I/O module is broadcasting to other controllers.</li></ul>

# CHOOSING COMMUNICATION FORMAT

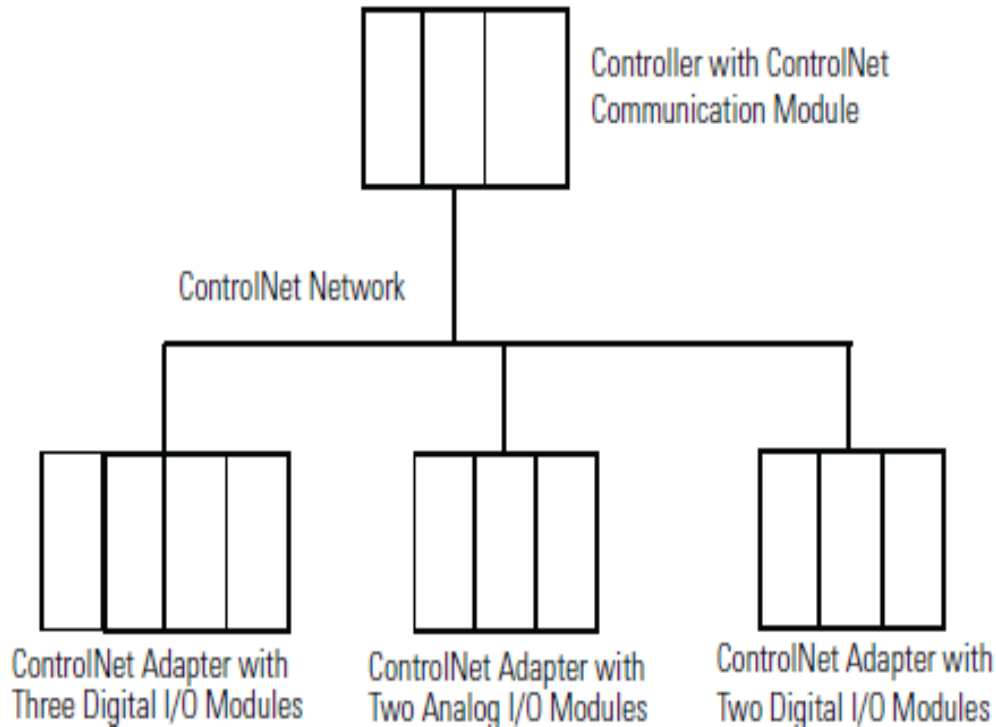
## Direct connections for I/O modules



System Connections	Amount
Controller to local ControlNet communication module	0
Controller to ControlNet adapter <sup>(1)</sup>	0
Direct connection for digital I/O modules	5 digital
Direct connection for analog I/O modules	2 analog
Total connections used	7

# CHOOSING COMMUNICATION FORMAT

## Rack-optimized connections for I/O Modules



System Connections	Amount
Controller to local ControlNet communication module	0
Controller to ControlNet adapters with digital modules (rack-optimized connection to each adapter)	2
Controller to ControlNet adapter with analog modules (direct connection for each analog I/O module)	2
Total connections used	4



# Control distributed I/O over a ControlNet network.

## **RSLogix 5000**

- Configure hardware system: CPU, Local modules, Local ControlNet Module, Remote ControlNet Modules, Remote I/O Modules.
- Write program to control system
- Download program to the CPU

## **RSNetworx for ControlNet**

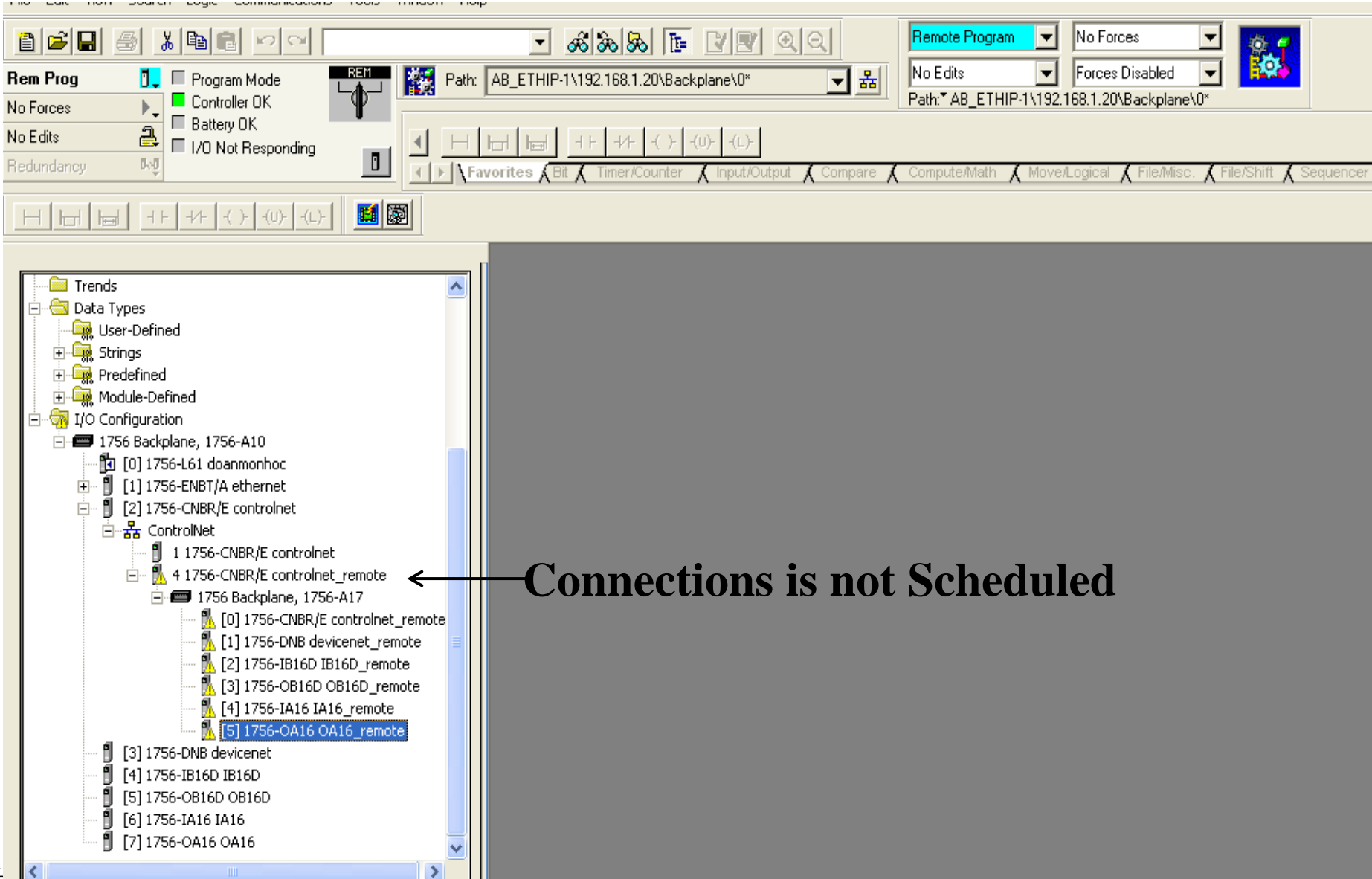
Schedule ControlNet Network

Rechedule controlnet network

Download to network

# Control distributed I/O over a ControlNet network.

Using RSLogix 5000 to control system over ControlNet network

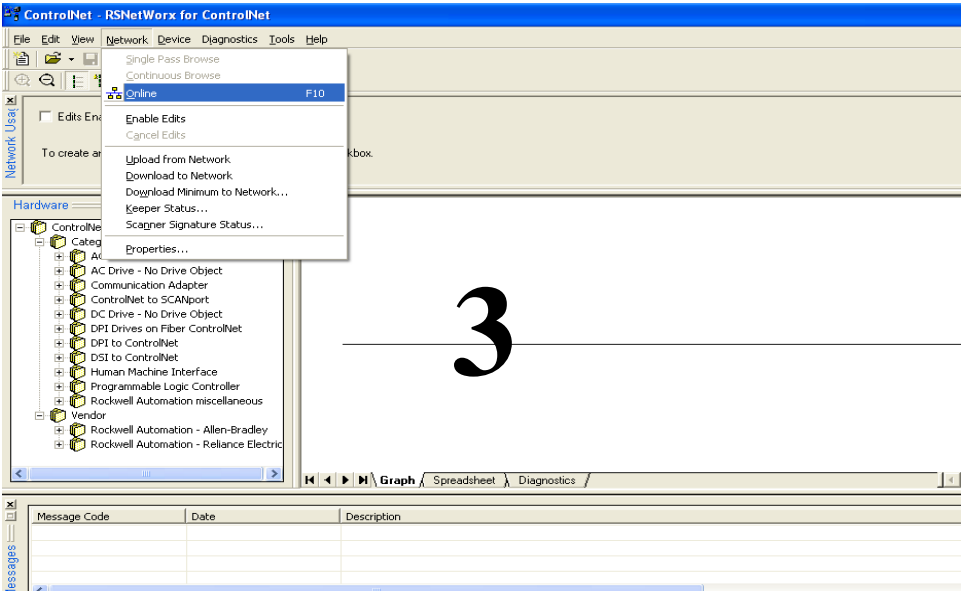
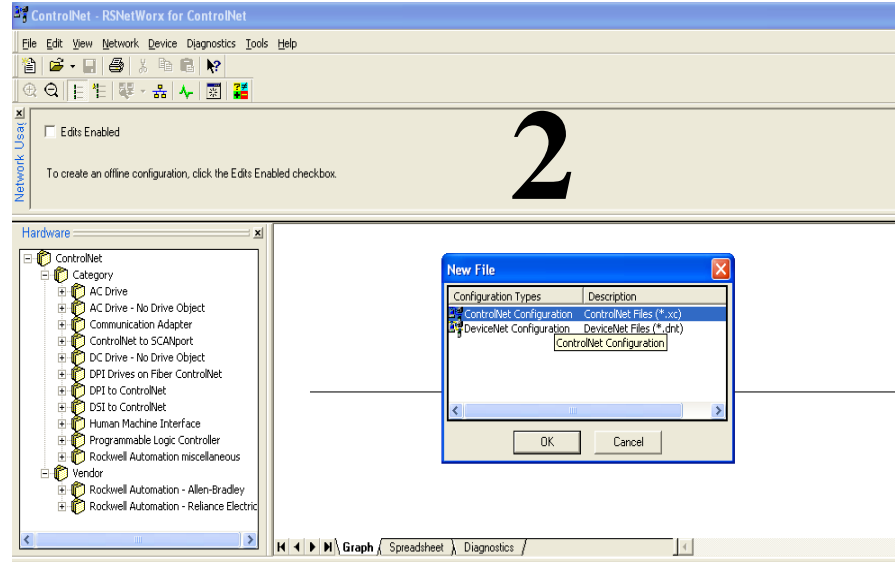
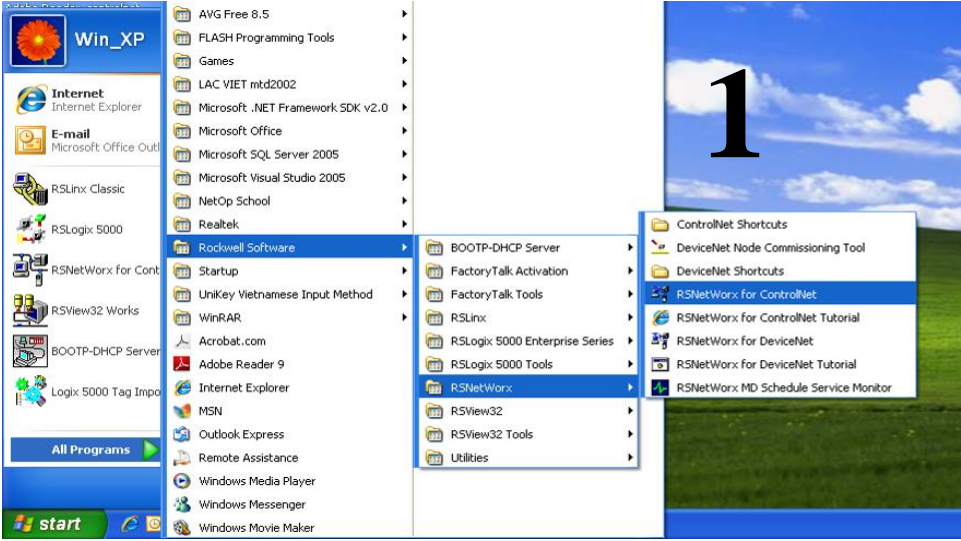


Connections is not Scheduled

F

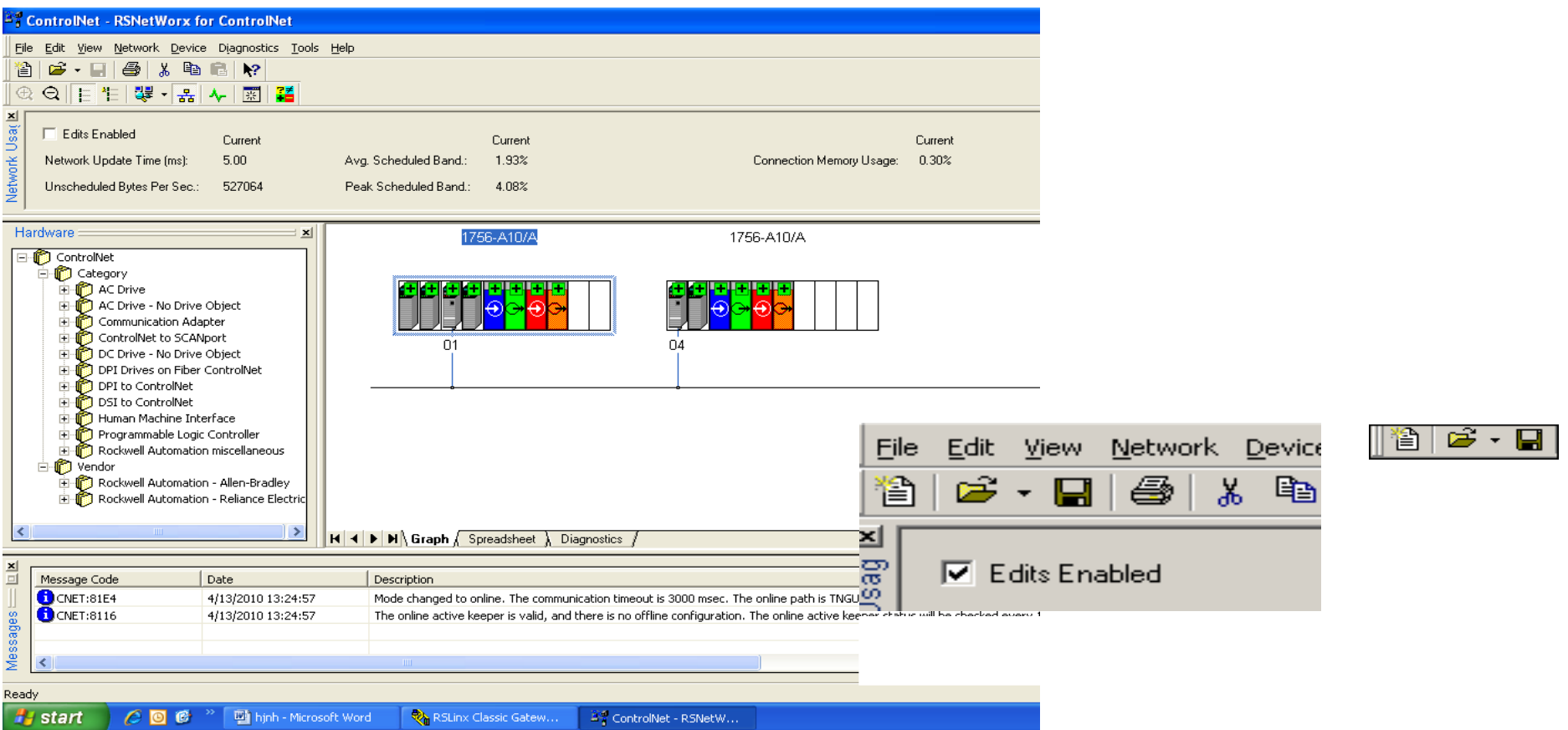
# Control distributed I/O over a ControlNet network.

## Schedule the Network



# Control distributed I/O over a ControlNet network.

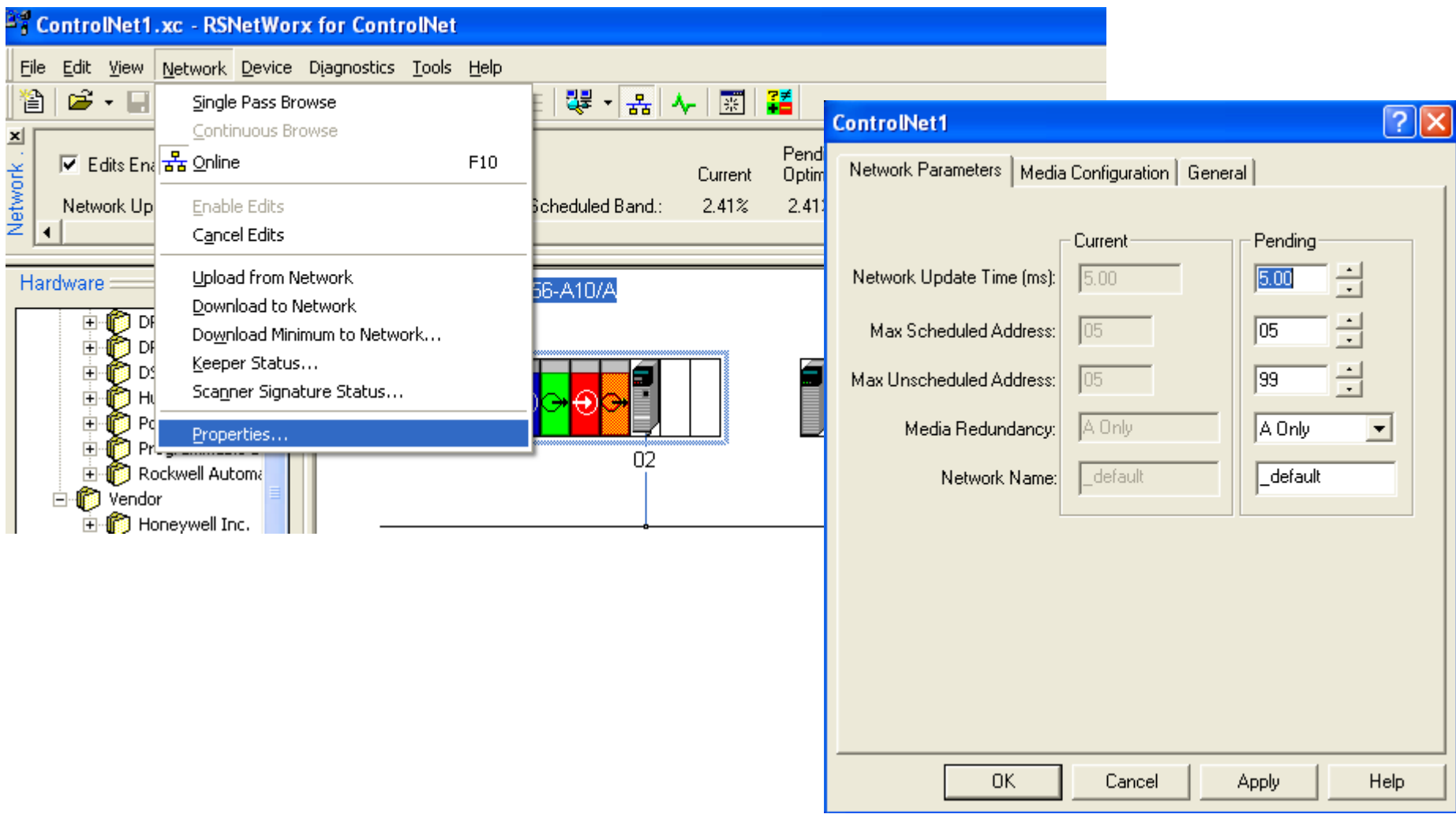
## Schedule the Network



**Check Edits Enabled**, RSNetWorx for ControlNet software reads data in the ControlNet modules and builds a schedule for the network.

# Control distributed I/O over a ControlNet network.

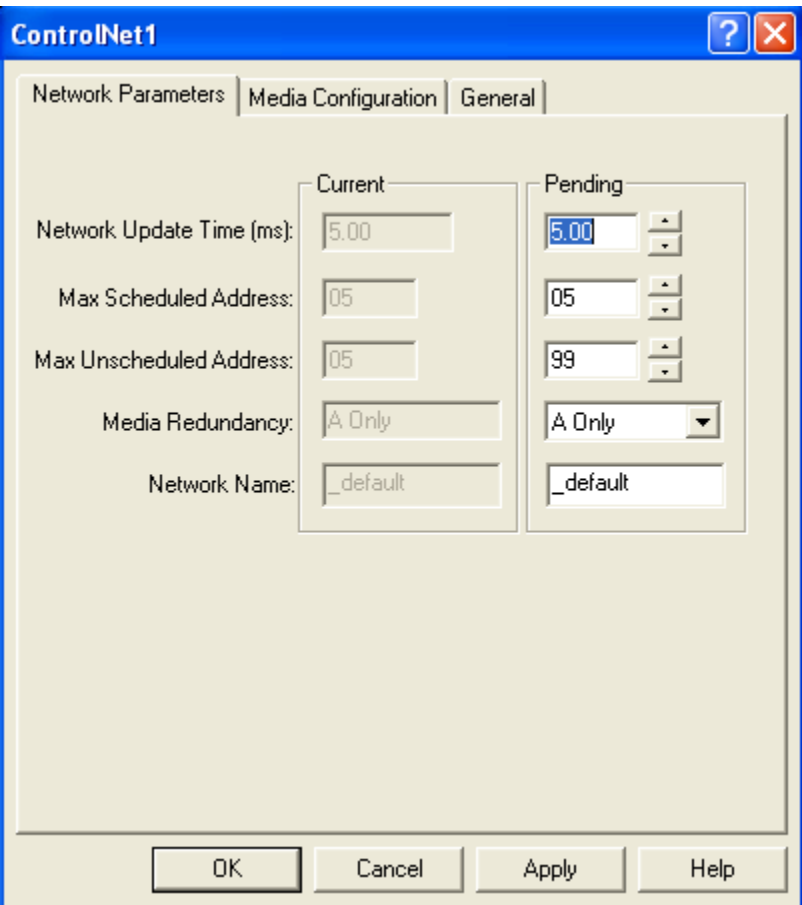
## Schedule the Network



To change the network properties from default settings to those that best fit your network, from the Network menu, choose Properties

# Control distributed I/O over a ControlNet network.

## Schedule the Network



Parameter	Description
Network Update Time	The smallest user-configurable repetitive time cycle in milliseconds at which data can be sent on a ControlNet link.
Max Scheduled Address	The node with the highest network address that can use scheduled time on a ControlNet link. I/O data is transferred during scheduled time. RSNetWorx for ControlNet software sets this value. We recommend that you do not change it.
Max Unscheduled Address	The node with the highest network address that can use unscheduled time on a ControlNet link. Messaging data is transferred during unscheduled time. Nodes set at addresses higher than the maximum unscheduled node do not communicate on the network. For example, they will not display in RSLinx software.
Media Redundancy	Designates if the network uses media redundancy on any of the network communication modules.
Network Name	A user-defined name for the network.

change the network properties from default settings

# Control distributed I/O over a ControlNet network.

## Schedule the Network online

The screenshot shows the RSNetWorx for ControlNet software interface. At the top, there is a menu bar (File, Edit, View, Network, Device, Diagnostics, Tools, Help) and a toolbar. Below the toolbar is a 'Network Usage' section with a table of statistics:

Edits Enabled	Current	Pending	Avg. Scheduled Band:	Current	Pending Optimized Edits	Pending Merged Edits	Connection Memory Usage
Network Update Time (ms):	5.00	5.00	1.81%	1.81%	1.81%	1.81%	
Unscheduled Bytes Per Sec.:	555722	555722	3.83%	3.83%	3.83%	3.83%	

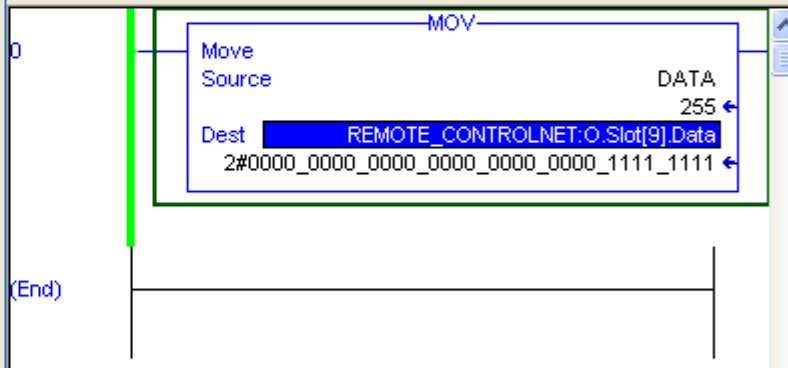
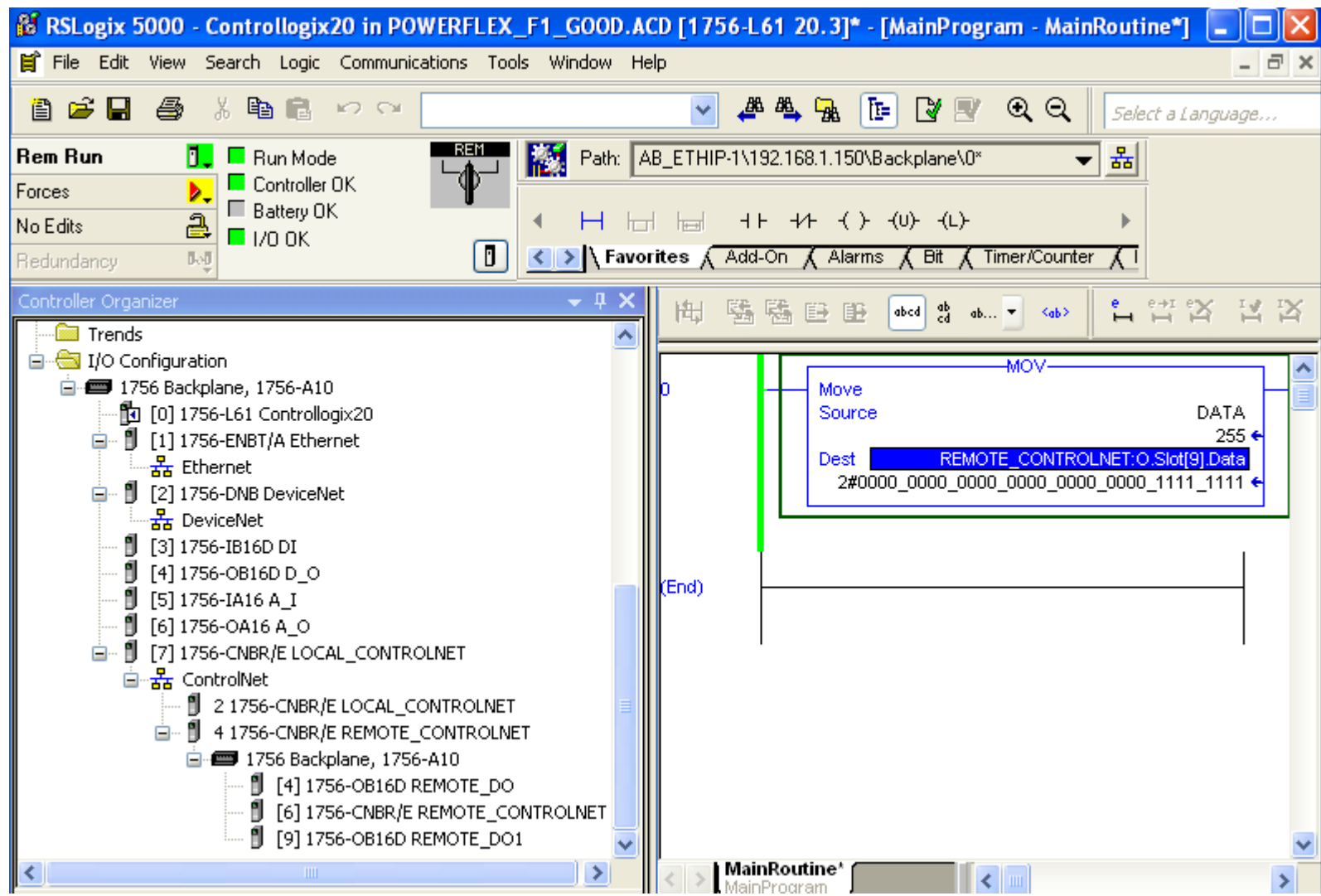
Below the table is a 'Hardware' section showing a tree view of the network topology. It includes categories like AC Drive, Communication Adapter, ControlNet to SCANport, DC Drive, DPI Drives on Fiber ControlNet, DPI to ControlNet, DSI to ControlNet, Human Machine Interface, Programmable Logic Controller, Rockwell Automation miscellaneous, Vendor, Rockwell Automation - Allen-Bradley, and Rockwell Automation - Reliance Electric. The main workspace displays two ControlNet segments, 1756-A10/A and 1756-A10/A-1, with their respective hardware components and connections. A 'Save As' dialog box is open, showing the file name 'TRAFFIC\_LIGHT' and the save type 'ControlNet Files (\*.xc)'. A 'Save Configuration' dialog box is also open, with the 'Optimize and re-write schedule for all connections' option selected.

**Save Scheduled File:** From the Save Configuration dialog box, click Optimize and rewrite the schedule for all connections.

Selecting merge changes into the existing schedule, those controllers whose connections have not changed remain in Run mode rather than changing to Program mode

# Control distributed I/O over a ControlNet network.

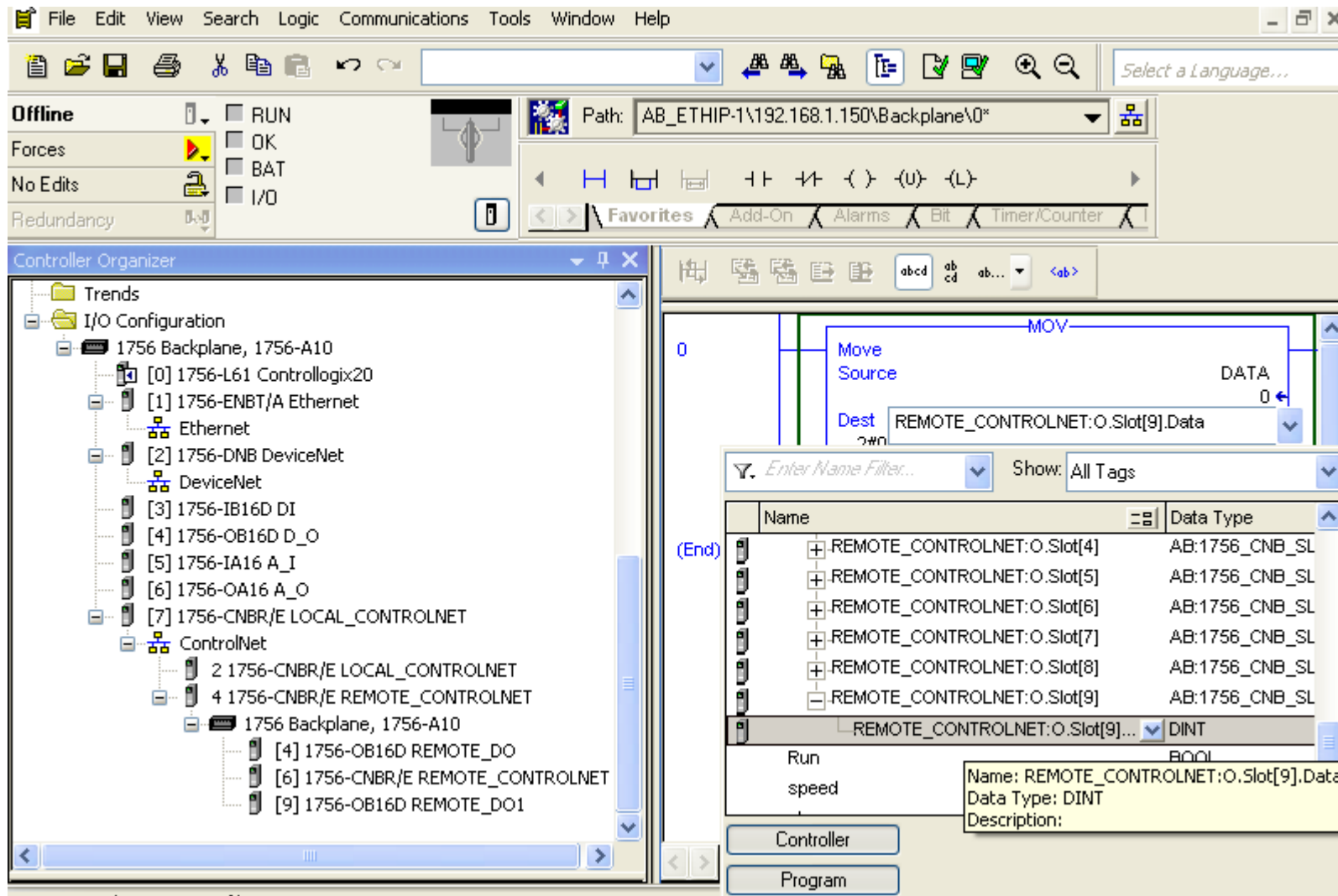
## Schedule the Network online



After saving, I/O modules are scheduled and good connection



# Control distributed I/O over a ControlNet network.



The screenshot displays the RSLogix 5000 software interface. The top menu bar includes File, Edit, View, Search, Logic, Communications, Tools, Window, and Help. The toolbar contains various icons for file operations and logic editing. The path is set to AB\_ETHIP-1\192.168.1.150\Backplane\0\*. The Controller Organizer on the left shows the following tree structure:

- Trends
- I/O Configuration
  - 1756 Backplane, 1756-A10
    - [0] 1756-L61 Controllogix20
    - [1] 1756-ENBT/A Ethernet
      - Ethernet
    - [2] 1756-DNB DeviceNet
      - DeviceNet
    - [3] 1756-IB16D DI
    - [4] 1756-OB16D D\_O
    - [5] 1756-IA16 A\_I
    - [6] 1756-OA16 A\_O
    - [7] 1756-CNBR/E LOCAL\_CONTROLNET
      - ControlNet
        - 2 1756-CNBR/E LOCAL\_CONTROLNET
        - 4 1756-CNBR/E REMOTE\_CONTROLNET
          - 1756 Backplane, 1756-A10
            - [4] 1756-OB16D REMOTE\_DO
            - [6] 1756-CNBR/E REMOTE\_CONTROLNET
            - [9] 1756-OB16D REMOTE\_DO1

The main workspace shows a ladder logic program with a MOV instruction. The destination address is REMOTE\_CONTROLNET:O.Slot[9].Data. A pop-up window displays a list of tags with the following data:

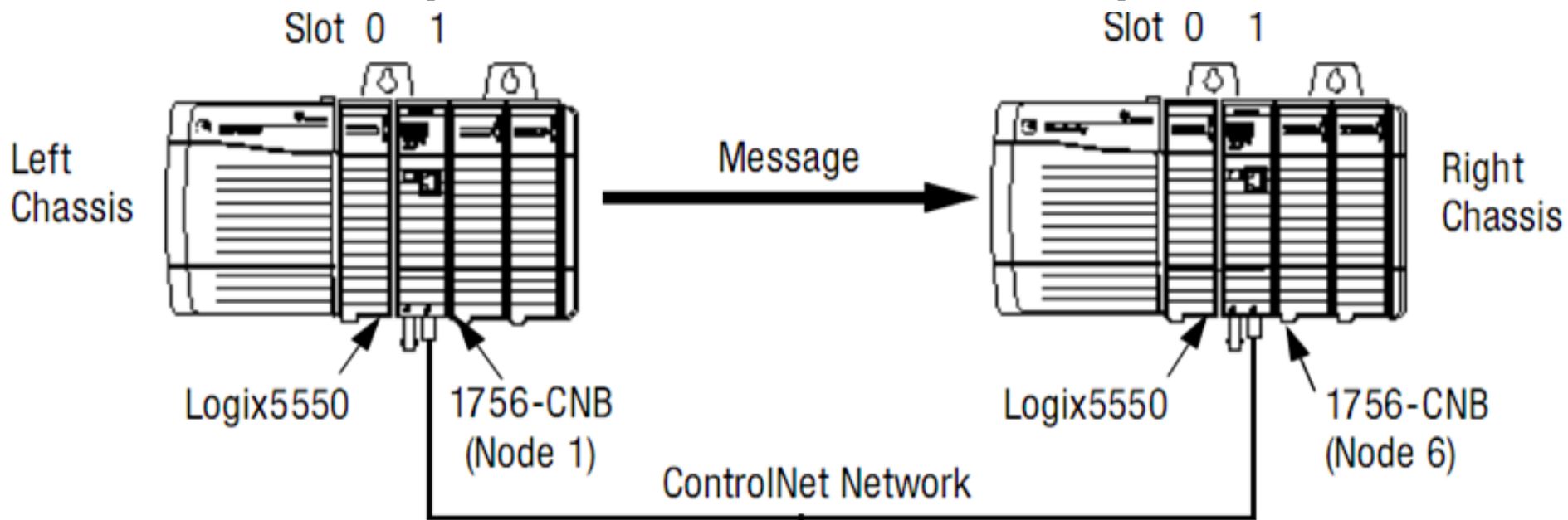
Name	Data Type
+ REMOTE_CONTROLNET:O.Slot[4]	AB:1756_CNB_SL
+ REMOTE_CONTROLNET:O.Slot[5]	AB:1756_CNB_SL
+ REMOTE_CONTROLNET:O.Slot[6]	AB:1756_CNB_SL
+ REMOTE_CONTROLNET:O.Slot[7]	AB:1756_CNB_SL
+ REMOTE_CONTROLNET:O.Slot[8]	AB:1756_CNB_SL
- REMOTE_CONTROLNET:O.Slot[9]	AB:1756_CNB_SL
REMOTE_CONTROLNET:O.Slot[9]...	DINT

The pop-up window also shows the following information:

- Run speed
- Name: REMOTE\_CONTROLNET:O.Slot[9].Data
- Data Type: DINT
- Description:

Writing logic to exchange data with remote I/O modules based on their address in the system.

# Produce and Consume Tags (interlock controllers)



➤ Scheduled Message (Produced and consume)

➤ Unscheduled Message (MSG)

*Configure hardware system and write logic as the same ethernet network except path information*