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
DEVICENET NETWORK

DeviceNet Overview

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

There are one Master and Slaves

HCMUTE 11_2014

phuongtv@hcmute.edu.vn_0908248231
DEVICENET NETWORK
Communication between DeviceNet Devices and CompactLogix via 1769 SDN Module

PLC with RSNetWorx for DeviceNet software

CompactLogix controller with 1769-SDN scanner
MicroLogix 1500 controller with 1769-SDN scanner
1784-PCD Interface card

1734 POINT I/O
E3 Overload Relay
PowerFlex 40 Drive Enhanced DeviceNet Communications Module
MicroLogix 1000 Controller with 1761-NET-DNI
MicroLogix 1200 Controller with 1761-NET-DNI
DEVICENET NETWORK
Communication between DeviceNet Devices and CompactLogix via 1756 DNB Module
## DEVICE NET NETWORK

**DeviceNet Features and Functionality**

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

## Assign an address to each Devices

<table>
<thead>
<tr>
<th>Give this device</th>
<th>This address</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scanner</td>
<td>0</td>
<td>If you have multiple scanners, give them the lowest addresses in sequence (0, 1…).</td>
</tr>
<tr>
<td>Any device on the network except</td>
<td>1…61</td>
<td>• Give the lower addresses to devices with 15 bytes or more of input or output data.</td>
</tr>
<tr>
<td>the scanner</td>
<td></td>
<td>• 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.</td>
</tr>
<tr>
<td>Computer interface to the network</td>
<td>62</td>
<td>If you connect a computer directly to the DeviceNet network, use address 62 for the computer.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Many computer interface devices use this address as their default.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The 1784-U2DN device can connect a computer directly to a DeviceNet network.</td>
</tr>
<tr>
<td>No device</td>
<td>63</td>
<td>Always leave address 63 open. Out of the box, most DeviceNet devices are preset for address 63.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Leaving address 63 open makes it possible to configure a new device.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The auto-address recovery feature also requires address 63 to be open.</td>
</tr>
</tbody>
</table>
DEVICENET NETWORK

Bus Topology

- Terminator
- Trunk
- Tap
- Power Supply
- Node
- Total Trunk Distance
- Drop Length
- Branching Drop
- Zero Drop
- Daisy Chain Drop

max 20 feet

HCMUTE 11_2014

phuongtv@hcmute.edu.vn_0908248231
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

D=101

R=101

V_CANH

V_CANL

0 Dominant 1 Recessive

1 Recessive 0 Dominant

phuongtv@hcmute.edu.vn_0908248231
DEVICENET NETWORK

The Inverted Logic of a DeviceNet Bus

![Diagram of DeviceNet bus logic]

- **CANH** (High Voltage): 3.5, 2.5, 1.5
- **CANL** (Low Voltage): 3.5, 2.5, 1.5

- **Dominant Differential Output Range**: 3.0, 1.5, 0.05
- **Recessive Differential Output Range**: -0.5, 0.05

- **Dominant Differential Input Range**: 5.0, 0.9, 0.5
- **Recessive Differential Input Range**: 0.0, -0.5
### DeviceNet Data Frame

<table>
<thead>
<tr>
<th>Field</th>
<th>Bit(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start of Frame</td>
<td>1 bit</td>
</tr>
<tr>
<td>Identifier</td>
<td>11 bits</td>
</tr>
<tr>
<td>Data Field</td>
<td>6 bytes</td>
</tr>
<tr>
<td>Control: RTR, CRC, ACK, DLC</td>
<td>16 bits</td>
</tr>
<tr>
<td>RTR Bit</td>
<td>1 bit</td>
</tr>
<tr>
<td>ACK Slot</td>
<td>1 bit</td>
</tr>
<tr>
<td>CLC Delimiter</td>
<td>1 bit</td>
</tr>
<tr>
<td>CRC Sequence</td>
<td>15 bits</td>
</tr>
<tr>
<td>End of Space</td>
<td>7 bits</td>
</tr>
<tr>
<td>Interframe Space</td>
<td>&gt;= 3 bits</td>
</tr>
</tbody>
</table>

**Notes:**
- The data frame starts with an identifier, followed by the data field, control information, and finally the end markers such as ACK slot and RTR bit.
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**

<table>
<thead>
<tr>
<th></th>
<th>S</th>
<th>O</th>
<th>F</th>
<th>10</th>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
<th>R</th>
<th>T</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Node 1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Node 2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Node 3</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Bus</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 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**

<table>
<thead>
<tr>
<th>Destination node address</th>
<th>Service code</th>
<th>Class ID</th>
<th>Instance ID</th>
<th>Attribute ID</th>
<th>Data</th>
</tr>
</thead>
</table>

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

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

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
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.
I/O MESSAGE

Selecting I/O message to transfer between Master and Slave
MASTER SCANER DATA FLOW

Ladder Processor

Input Image

B
D
A
C
E

Output Image

A
B
C
D
E

......

Internal Input Data Storage

Y
X
Z

......

Internal Output Data Storage

Y
X
Z

......

Master Scanner

Input Data From Devicenet Device

Output Data to Devicenet Device
MASTER SCANNER

CompactLogix™ Scanner 1769-SDN

ControlLogix® Scanner 1756-DNB

DriveLogix™ and FlexLogix™
Communication Card 1788-DNBO

EtherNet/IP to DeviceNet Linking
Device 1788-EN2DN

Allen-Bradley
EtherNet/IP
DeviceNet
Linking Device

IP ADDRESS
DeviceNet Node Address
DeviceNet Baud Rate
SCANNER MEMORY

The memory of a Logix5000 scanner is organized as an array of DINTs (4-byte elements). A Logix5000 system is easier to program if you give each device its own DINT or DINTs within the scanner.

If a device sends input data, it uses input memory in the scanner.

If a device gets output data, it uses output memory in the scanner.

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.

<table>
<thead>
<tr>
<th>Scanner</th>
<th>Maximum input data (DINTs)</th>
<th>Maximum output data (DINTs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1756-DNB</td>
<td>124</td>
<td>123</td>
</tr>
<tr>
<td>1769-SDN</td>
<td>90</td>
<td>90</td>
</tr>
<tr>
<td>1788-CN2DN</td>
<td>124</td>
<td>123</td>
</tr>
<tr>
<td>1788-EN2DN</td>
<td>124</td>
<td>123</td>
</tr>
<tr>
<td>1788-DNBO</td>
<td>124</td>
<td>123</td>
</tr>
</tbody>
</table>
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
     - controller
     - scanner
     - device
     - device
     - device
     - device
     - device

2. Choose a scanner.

<table>
<thead>
<tr>
<th>If you are using:</th>
<th>And the main network is:</th>
<th>Use this scanner:</th>
</tr>
</thead>
<tbody>
<tr>
<td>single network</td>
<td>Corresponding scanner for your controller</td>
<td></td>
</tr>
<tr>
<td>subnets</td>
<td>EtherNet/IP to DeviceNet Linking Device 1788-EN2DN</td>
<td></td>
</tr>
<tr>
<td>ControlNet™</td>
<td>ControlNet to DeviceNet Linking Device 1788-CN2DN</td>
<td></td>
</tr>
</tbody>
</table>
DEVICENET NETWORK CONFIGURATION

- ControlLogix Controller With 1756-ENBT Module
- Ethernet/IP Network
- Linking Devices
- CompactLogix Controller
- DeviceNet Network
- Flex Adapter & I/O
- Input/output Devices
- Motor Starter
- Sensor
- Pushbutton Cluster
- PowerFlex Ac Drive
- Indicator Lights
- Bar Code Scanner
DEVICENET NETWORK CONFIGURATION
Configure I/O modules for Slaves by RSnetwork for Devicenet
Configure I/O address for modules in Master Scanner
DEVICENET NETWORK CONFIGURATION

Insert Master Scanner into Project using RSLogix5000

![RSLogix 5000 - DEVICENET [1769-L32E]*](image)

**New Module**

- **Type:** 1769-SDN/B 1769 Scanner DeviceNet
- **Vendor:** Allen-Bradley
- **Name:** SDN
- **Description:**
- **Input Size:** 90 (32-bit)
- **Output Size:** 90 (32-bit)
- **Revision:** 3.1
- **Electronic Keying:** Disable Keying
- **Open Module Properties:** OK, Cancel, Help
DEVICENET NETWORK CONFIGURATION
Creating logic to exchange I/O data over Devicenet Network
COMPACTLOGIX DEVICENET SCANNER

1769 SDN(Devicenet Master)

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

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

A = Sink input
B = Common
C = +V DC
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
+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
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

A

Outputs

B

Outputs

C

NC +V1 COM1 +V1 COM1 +V1 COM1 +V1 COM1 +V2 COM2 +V2 COM2 +V2 COM2 +V2 COM2 NC

+V1 = Terminals 35, 37, 39 and 41
+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
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.
CONTROL DEVICES VIA D_NET

- Creating a project in RSLogix5000 and add a Scanner
CONTROL DEVICES VIA D_NET

Configuring the Scanner Module
CONTROL DEVICES VIA D_NET

Writing a simple program in Rslogix 5000.

Notice: `CommandRegister.Run` instruction must be executed to run devicenet network
CONTROL DEVICES VIA D_NET

Writing logic to access I/O modules.
## DeviceNet Network Troubleshooting

<table>
<thead>
<tr>
<th>Status code (decimal)</th>
<th>Description</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>65</td>
<td>The AutoScan option is on and the device is in idle mode.</td>
<td>None.</td>
</tr>
<tr>
<td>70</td>
<td>The address of the device is already in use by another device on the network.</td>
<td>Change the address of the device to an unused address.</td>
</tr>
<tr>
<td>71</td>
<td>Illegal data in scan list.</td>
<td>Reconfigure the scan list and remove any illegal data.</td>
</tr>
<tr>
<td>72</td>
<td>No communication with the device.</td>
<td>Inspect the device and verify connections.</td>
</tr>
<tr>
<td>73</td>
<td>Device’s identity information does not match electronic key in scanner</td>
<td>• Make sure that the correct device is at this address.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Make sure that the device matches the specified electronic key (vendor, product code, product type).</td>
</tr>
<tr>
<td>74</td>
<td>Data overrun on port detected.</td>
<td>• Modify your configuration and check for invalid data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Check network communication traffic.</td>
</tr>
<tr>
<td>75</td>
<td>Either or both of the following:</td>
<td>Check that the device has:</td>
</tr>
<tr>
<td></td>
<td>• The device does not have a scan list.</td>
<td>• scan list</td>
</tr>
<tr>
<td></td>
<td>• The device has not received communication from any other device.</td>
<td>• properly wired connection to the network</td>
</tr>
<tr>
<td>76</td>
<td>No direct network traffic for scanner.</td>
<td>None. The scanner hears other network communication but does not hear any directed to it.</td>
</tr>
<tr>
<td>77</td>
<td>During initialization, the data size expected by the device does not match the scan list entry.</td>
<td>Check the device and the scan list for the correct input and output size for the device.</td>
</tr>
<tr>
<td>Line</td>
<td>Description</td>
<td>Troubleshooting Steps</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>-----------------------</td>
</tr>
</tbody>
</table>
| 78   | Device is not communicating or communication is intermittent. | • Check that the device has a properly wired connection to the network.  
• Check that the device has power.  
• If the device is polled, make sure the interscan delay is long enough for the device to return its data. |
| 79   | Scanner has failed to transmit a message. | • Make sure that your scanner is connected to a valid network.  
• Check for disconnected cables. |
| 80   | Scanner is in idle mode. | To run the network:  
1. Put controller in run/remote run mode.  
2. Turn on the following member of command register for the scanner:  
   ...O.CommandRegister.Fault |
| 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. | • Check scan list device to make sure that its input and output data sizes are correct.  
• Check the configuration of the device. |
| 83   | Device returns error responses when the scanner attempts to communicate with it. | • Check the accuracy of the scan list.  
• Check the configuration of the device. The device may be in another scanner's scan list.  
• Cycle power to the device. |
<p>| 84   | Scanner is initializing the DeviceNet network. | None. This code clears itself once the scanner attempts to initialize all the devices on the network. |</p>
<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
<th>Recommendations</th>
</tr>
</thead>
<tbody>
<tr>
<td>85</td>
<td>During runtime, the device is sending the wrong size of data.</td>
<td>Contact Rockwell Automation support. See the back of this publication.</td>
</tr>
</tbody>
</table>
| 86     | Device is in idle state/mode (not producing data) while the scanner is in run mode. | - Check the configuration and status of the device.  
- If you set up an interlock between 2 scanners (controllers), make sure both scanners are in run mode. |
| 88     | In shared inputs, the I/O parameters (polled, strobed, etc.) do not 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 | - Make sure that you installed a compatible device.  
- The offline configuration for the device does not match the actual (online) configuration of the device. |
| 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)                                   | - Cycle power to the device.  
- Make sure all devices are at the same baud rate.  
- Make sure there is no short circuit between a CAN line (blue or white) and a power or shield line (black, red, shield).  
- Check for any of the following sources of noise:  
  - Close proximity to a high voltage power cable  
  - Improper or no termination resistor  
  - Improper grounding  
- Check for a device that is producing noise or inappropriate data on the network. |
<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>92</td>
<td>The DeviceNet cable is <em>not</em> supplying power to the communication port.</td>
<td>• Make sure the network has 24V dc power.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Check the connection to the device.</td>
</tr>
<tr>
<td>95</td>
<td>The firmware of the device is currently being updated.</td>
<td>None. Do not disconnect the device while the update is in progress. You will lose any existing data in the device's memory.</td>
</tr>
<tr>
<td>96</td>
<td>Communication port is in test mode.</td>
<td>None.</td>
</tr>
<tr>
<td>97</td>
<td>Controller has set the scanner to the halted mode.</td>
<td>1. See if the following bit of the command register for the scanner is on:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>...0.CommandRegister.HaltScanner</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Cycle power to the scanner.</td>
</tr>
<tr>
<td>98</td>
<td>General firmware error.</td>
<td>Replace device.</td>
</tr>
<tr>
<td>99</td>
<td>System failure.</td>
<td>Replace device.</td>
</tr>
</tbody>
</table>
CONTROL DEVICES VIA D_NET

EX_1
Set up a devicenet network includes two nodes.

- Scaner 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.

- Scaner 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

Primary controller

Secondary controller

DeviceNet

All backed up devices must operate on DeviceNet
BACK UP ON DEVICENET NETWORK

Node 1 (Active) → Ownership → Shared Storage → Ownership → Node 2 (Passive)

Services Transfer

Heartbeat

Error node
BACK UP ON DEVICENET NETWORK

Start

Set Heartbeat to SDN

Read Heartbeat from SDN

Current heartbeat = Set heartbeat

Run Network

Set Heartbeat to SDN

Read Heartbeat verify from SDN

Current heartbeat = Set heartbeat

Change SDN after two heartbeat constant
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)*
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).
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**

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias For</th>
<th>Base Tag</th>
<th>Data Type</th>
<th>Style</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure_Heartbeat</td>
<td></td>
<td></td>
<td>INT</td>
<td>Decimal</td>
<td></td>
</tr>
<tr>
<td>Current_Heartbeat</td>
<td></td>
<td></td>
<td>INT</td>
<td>Decimal</td>
<td></td>
</tr>
<tr>
<td>heartbeat</td>
<td></td>
<td></td>
<td>INT</td>
<td>Decimal</td>
<td></td>
</tr>
<tr>
<td>Local:1:1</td>
<td></td>
<td>A8:1768_SDN_1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local:1:0</td>
<td></td>
<td>A8:1768_SDN_3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read_Heartbeat</td>
<td></td>
<td></td>
<td>MESSAGE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read_Heartbeat_Verify</td>
<td></td>
<td></td>
<td>MESSAGE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Set_Heartbeat</td>
<td></td>
<td></td>
<td>MESSAGE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
After CommandRegister.Run Command is executed, users can I/Os data via devicenet network.
CONFIGURING MSG TO GET HEARBEAT

Message Configuration - Read_Heartbeat

- **Message Type:** CIP Generic
- **Service Type:** Get Attribute Single
- **Service Code:** e (Hex)
- **Class:** 90 (Hex)
- **Instance:** 1
- **Attribute:** 12 (Hex)
- **Source Element:**
- **Source Length:** 0 (Bytes)
- **Destination:** Current_Heartbeat
- **Enable:**
- **Enable Waiting:**
- **Start:**
- **Done:**
- **Done Length:** 0
- **Error Code:**
- **Extended Error Code:**
- **Timed Out:**

Local:1:0.CommandRegister.Run Read_Heartbeat.EN MSG
Message Message Control Read_Heartbeat...
SELECTING SDN TO GET HEARTBEAT
CONFIGURING MSG TO SET HEARTBEAT
SELECTING SDM TO SET HEARTBEAT

![Image of a software interface for selecting SDMs to set heartbeat]
DATA TRANSFER BETWEEN CONTROLLERS

Transfer data via Ethernet network

*Each device must have 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

Controller tag
Produced
consumed

Controller tag
Consumed
Produced

Controller tag
Consumed
Produced

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

<table>
<thead>
<tr>
<th>Guideline</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create the tags at the controller scope.</td>
<td>You can share only controller-scoped tags.</td>
</tr>
<tr>
<td>Use one of these data types:</td>
<td>• To share other data types, create a user-defined data type that contains the required data.</td>
</tr>
<tr>
<td>• DINT</td>
<td>• Use the same data type for the produced tag and corresponding consumed tag or tags.</td>
</tr>
<tr>
<td>• REAL</td>
<td></td>
</tr>
<tr>
<td>• array of DINTs or REALs</td>
<td></td>
</tr>
<tr>
<td>• user-defined</td>
<td></td>
</tr>
<tr>
<td>Limit the size of the tag to (\leq 500) bytes.</td>
<td>If transferring more than 500 bytes, create logic to transfer the data in packets.</td>
</tr>
<tr>
<td></td>
<td>A size of (&lt; 125) DINT words will keep total bytes within 500. This helps reduce the total number of packets for transactions.</td>
</tr>
<tr>
<td>Combine data that goes to the same controller.</td>
<td>If producing several tags for the same controller:</td>
</tr>
<tr>
<td></td>
<td>• Group the data into one or more user-defined data types. This method uses fewer connections than does producing each tag separately.</td>
</tr>
<tr>
<td></td>
<td>• Group the data according to similar update intervals. To conserve network bandwidth, use a greater RPI for less critical data.</td>
</tr>
<tr>
<td></td>
<td>For example, you could create one tag for data that is critical and another tag for data that is not as critical.</td>
</tr>
</tbody>
</table>
DATA TRANSFER BETWEEN CONTROLLERS

Produced and Consumed Tag Definitions

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Produced tag</td>
<td>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.</td>
</tr>
<tr>
<td>Consumed tag</td>
<td>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.</td>
</tr>
</tbody>
</table>

Connections for Produced and Consumed Tags

<table>
<thead>
<tr>
<th>Tag Type</th>
<th>Required Connections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Produced</td>
<td>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. 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.</td>
</tr>
<tr>
<td>Consumed</td>
<td>Each consumed tag requires one connection for the controller that is consuming the tag.</td>
</tr>
</tbody>
</table>
DATA TRANSFER BETWEEN CONTROLLERS

Number Connections for Produced and Consumed Tags

<table>
<thead>
<tr>
<th>Type of Tag</th>
<th>Device</th>
<th>Number of Connections Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Produced tag</td>
<td>Logix5000 controller</td>
<td>Number_of_consumers + 1</td>
</tr>
<tr>
<td></td>
<td>EtherNet/IP module</td>
<td>1</td>
</tr>
<tr>
<td>Consumed tag</td>
<td>Logix5000 controller</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>EtherNet/IP module</td>
<td></td>
</tr>
</tbody>
</table>

Example of Number Connections for Produced and Consumed 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

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

Controller tag
Tag1, tag2...

Controller tag
Tag1, tag2...

Message control
Data table write
Data table read

Message control
Data table write
Data table read

Ins 192.168.1.21
255.255.255.0

192.168.1.4
255.255.255.20
DATA TRANSFER BETWEEN CONTROLLERS

Use Message instruction to read, write data

![Message Configuration - Message_1](image)

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

Station 1
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.

Station 2
DATA TRANSFER BETWEEN CONTROLLERS
Use Message instruction to read, write data
Create tags in controller tag for each station.
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

- **Message Type:** CIP Data Table Write
- **Source Element:** DATA_SEND20
- **Number Of Elements:** 4
- **Destination Element:** READ21
DATA TRANSFER BETWEEN CONTROLLERS

Use Message instruction to read, write data

Add and configure parameters for Message control Ins

![Message Configuration - DATA20](image)
DATA TRANSFER BETWEEN CONTROLLERS

Use Message instruction to read, write data

Download program to plc and check respondent of data
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

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Personal computer running SoftLogix5800 controller with 1784-PCICS card</td>
</tr>
<tr>
<td>2</td>
<td>1756-CN8 module (as an adapter) with 1756 I/O modules</td>
</tr>
<tr>
<td>3</td>
<td>PowerFlex 700S drive</td>
</tr>
<tr>
<td>4</td>
<td>1794-ACN15 adapter with 1794 I/O modules</td>
</tr>
<tr>
<td>5</td>
<td>1734-ACNF adapter with 1734 I/O modules</td>
</tr>
<tr>
<td>6</td>
<td>PanelView terminal</td>
</tr>
<tr>
<td>7</td>
<td>CompactLogix 1769-L35CR controller with local 1769 I/O modules</td>
</tr>
<tr>
<td>8</td>
<td>ControlLogix controller with 1756-CN2, 1756-CN2R, 1756-CN8, or 1756-CNBR module as the scanner</td>
</tr>
</tbody>
</table>
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

HCMUTE 11_2014

phuongtv@hcmute.edu.vn_0908248231
INSTALLING CONTROLNET MODULES
SETTING MODULE ADDRESS

Network Node Address Switches (Set for Network Address 21)
CONTROLNET CABLE

BNC Connector

Network Tap End

Diagram of a network cable connection with labels for BNC Connector and Network Tap End.
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
Control I/O Via ControlNet Network

Validate Connections
CHOOSING COMMUNICATION FORMAT

New Module

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
Revision: Listen Only - Rack Optimization
Keiyng: Compatible Keiyng

Open Module Properties

4 of 116 Module Types Found

Add to Favorites

Close on Create
## CHOOSING COMMUNICATION FORMAT

Communication Format with a Remote ControlNet Communication Module

<table>
<thead>
<tr>
<th>Communication Format with a Remote ControlNet Communication Module</th>
<th>Criteria for Use</th>
</tr>
</thead>
</table>
| None                                                              | • When all of the remote I/O communicating with a controller via the remote ControlNet communication module use a Direct Connection communication format  
• When the connection is used for scheduled peer interlocking  
• When I/O will be mostly direct connections  
• When multiple controllers control the outputs in the chassis |
| Rack-optimized                                                    | • 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  
• To minimize ControlNet network bandwidth when using large volume of digital I/O  
• If only one controller will control the I/O |
| 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

Controller Organizer

Select Module Type

New Module

Type: 1756-IB16 16 Point 10V-31.2V DC Input
Vendor: Allen-Bradley
Parent: REMOTE_CONTROLNET
Name: REMOTE_DI
Slot: 1

Comm Format: Rack Optimization
Revision: CST Timestamped Input Data

OK Cancel Help

HCMUTE 11_2014

phuongtv@hcmute.edu.vn_0908248231
## CHOOSING COMMUNICATION FORMAT

Communication Format with a Remote ControlNet Communication Module

<table>
<thead>
<tr>
<th>I/O Module Type</th>
<th>Desired Connection</th>
<th>Required Communication Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital module</td>
<td>Rack-optimized connection</td>
<td>Rack-optimized</td>
</tr>
<tr>
<td></td>
<td>Direct connection or to use specialty features of the module, such as diagnostics,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>timestamps, or electronic fuses</td>
<td></td>
</tr>
<tr>
<td></td>
<td>or Connection for listening to data from the module</td>
<td></td>
</tr>
<tr>
<td>Analog module</td>
<td>Direct connection or to use specialty features of the module, such as diagnostics,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>timestamps, or electronic fuses</td>
<td></td>
</tr>
<tr>
<td></td>
<td>or Connection for listening to data from the module</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• A listen-only communication format that matches the data the I/O module is broadcasting to other controllers.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• A listen-only communication format that matches the data the I/O module is broadcasting to other controllers.</td>
<td></td>
</tr>
</tbody>
</table>
CHOOSING COMMUNICATION FORMAT

Direct connections for I/O modules

System Connections

<table>
<thead>
<tr>
<th>Connection</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller to local ControlNet communication module</td>
<td>0</td>
</tr>
<tr>
<td>Controller to ControlNet adapter(^{(1)})</td>
<td>0</td>
</tr>
<tr>
<td>Direct connection for digital I/O modules</td>
<td>5 digital</td>
</tr>
<tr>
<td>Direct connection for analog I/O modules</td>
<td>2 analog</td>
</tr>
<tr>
<td>Total connections used</td>
<td>7</td>
</tr>
</tbody>
</table>
CHOOSING COMMUNICATION FORMAT

Rack-optimized connections for I/O Modules

<table>
<thead>
<tr>
<th>System Connections</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller to local ControlNet communication module</td>
<td>0</td>
</tr>
<tr>
<td>Controller to ControlNet adapters with digital modules (rack-optimized connection to each adapter)</td>
<td>2</td>
</tr>
<tr>
<td>Controller to ControlNet adapter with analog modules (direct connection for each analog I/O module)</td>
<td>2</td>
</tr>
<tr>
<td>Total connections used</td>
<td>4</td>
</tr>
</tbody>
</table>
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
Reschedule 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.
Control distributed I/O over a ControlNet network.

Schedule the Network

1. Win XP desktop
2. ControlNet - I-SNetWorks for ControlNet
3. ControlNet - I-SNetWorks for ControlNet
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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Update Time</td>
<td>The smallest user-configurable repetitive time cycle in milliseconds at which data can be sent on a ControlNet link.</td>
</tr>
<tr>
<td>Max Scheduled Address</td>
<td>The node with the highest network address that can use scheduled time on a ControlNet link. I/O data is transferred during scheduled time. RSNNetWorx for ControlNet software sets this value. We recommend that you do not change it.</td>
</tr>
<tr>
<td>Max Unscheduled Address</td>
<td>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.</td>
</tr>
<tr>
<td>Media Redundancy</td>
<td>Designates if the network uses media redundancy on any of the network communication modules.</td>
</tr>
<tr>
<td>Network Name</td>
<td>A user-defined name for the network.</td>
</tr>
</tbody>
</table>

change the network properties from default settings
Control distributed I/O over a ControlNet network.

Schedule the Network online

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.

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