AUTOMATIONDIRECT.com

D0-DCM User Manual

Manual Number: D0-DCM-M

~ WARNING ~

Thank you for purchasing automation equipment from **AutomationDirect.com®**, doing business as, **AutomationDirect**. We want your new automation equipment to operate safely. Anyone who installs or uses this equipment should read this publication (and any other relevant publications) before installing or operating the equipment.

To minimize the risk of potential safety problems, you should follow all applicable local and national codes that regulate the installation and operation of your equipment. These codes vary from area to area and usually change with time. It is your responsibility to determine which codes should be followed, and to verify that the equipment, installation, and operation is in compliance with the latest revision of these codes.

At a minimum, you should follow all applicable sections of the National Fire Code, National Electrical Code, and the codes of the National Electrical Manufacturer's Association (NEMA). There may be local regulatory or government offices that can also help determine which codes and standards are necessary for safe installation and operation.

Equipment damage or serious injury to personnel can result from the failure to follow all applicable codes and standards. We do not guarantee the products described in this publication are suitable for your particular application, nor do we assume any responsibility for your product design, installation, or operation.

Our products are not fault-tolerant and are not designed, manufactured or intended for use or resale as on-line control equipment in hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems, in which the failure of the product could lead directly to death, personal injury, or severe physical or environmental damage ("High Risk Activities"). **AutomationDirect** specifically disclaims any expressed or implied warranty of fitness for High Risk Activities.

For additional warranty and safety information, see the Terms and Conditions section of our catalog. If you have any questions concerning the installation or operation of this equipment, or if you need additional information, please call us at 770-844-4200.

This publication is based on information that was available at the time it was printed. At **AutomationDirect** we constantly strive to improve our products and services, so we reserve the right to make changes to the products and/or publications at any time without notice and without any obligation. This publication may also discuss features that may not be available in certain revisions of the product.

Trademarks

This publication may contain references to products produced and/or offered by other companies. The product and company names may be trademarked and are the sole property of their respective owners. **AutomationDirect** disclaims any proprietary interest in the marks and names of others.

ADVERTENCIA

Gracias por comprar equipo de automatización de **Automationdirect.com**®. Deseamos que su nuevo equipo de automatización opere de manera segura. Cualquier persona que instale o use este equipo debe leer esta publicación (y cualquier otra publicación pertinente) antes de instalar u operar el equipo.

Para reducir al mínimo el riesgo debido a problemas de seguridad, debe seguir todos los códigos de seguridad locales o nacionales aplicables que regulan la instalación y operación de su equipo. Estos códigos varian de área en área y usualmente cambian con el tiempo. Es su responsabilidad determinar cuales códigos deben ser seguidos y verificar que el equipo, instalación y operación estén en cumplimiento con la revisión mas reciente de estos códigos.

Como mínimo, debe seguir las secciones aplicables del Código Nacional de Incendio, Código Nacional Eléctrico, y los códigos de (NEMA) la Asociación Nacional de Fabricantes Eléctricos de USA. Puede haber oficinas de normas locales o del gobierno que pueden ayudar a determinar cuales códigos y normas son necesarios para una instalación y operación segura.

Si no se siguen todos los códigos y normas aplicables, puede resultar en daños al equipo o lesiones serias a personas. No garantizamos los productos descritos en esta publicación para ser adecuados para su aplicación en particular, ni asumimos ninguna responsabilidad por el diseño de su producto, la instalación u operación.

Nuestros productos no son tolerantes a fallas y no han sido diseñados, fabricados o intencionados para uso o reventa como equipo de control en línea en ambientes peligrosos que requieren una ejecución sin fallas, tales como operación en instalaciones nucleares, sistemas de navegación aérea, o de comunicación, control de tráfico aéreo, máquinas de soporte de vida o sistemas de armamentos en las cuales la falla del producto puede resultar directamente en muerte, heridas personales, o daños físicos o ambientales severos ("Actividades de Alto Riesgo"). **Automationdirect.com** específicamente rechaza cualquier garantía ya sea expresada o implicada para actividades de alto riesgo.

Para información adicional acerca de garantía e información de seguridad, vea la sección de Términos y Condiciones de nuestro catálogo. Si tiene alguna pregunta sobre instalación u operación de este equipo, o si necesita información adicional, por favor llámenos al número 770-844-4200 en Estados Unidos.

Esta publicación está basada en la información disponible al momento de impresión. En **Automationdirect.com** nos esforzamos constantemente para mejorar nuestros productos y servicios, así que nos reservamos el derecho de hacer cambios al producto y/o a las publicaciones en cualquier momento sin notificación y sin ninguna obligación. Esta publicación también puede discutir características que no estén disponibles en ciertas revisiones del producto.

Marcas Registradas

Esta publicación puede contener referencias a productos producidos y/u ofrecidos por otras compañías. Los nombres de las compañías y productos pueden tener marcas registradas y son propiedad única de sus respectivos dueños. Automationdirect.com, renuncia cualquier interés propietario en las marcas y nombres de otros.

~ AVERTISSEMENT ~

Nous vous remercions d'avoir acheté l'équipement d'automatisation de **Automationdirect.comMC**, en faisant des affaires comme, **AutomationDirect**. Nous tenons à ce que votre nouvel équipement d'automatisation fonctionne en toute sécurité. Toute personne qui installe ou utilise cet équipement doit lire la présente publication (et toutes les autres publications pertinentes) avant de l'installer ou de l'utiliser.

Afin de réduire au minimum le risque d'éventuels problèmes de sécurité, vous devez respecter tous les codes locaux et nationaux applicables régissant l'installation et le fonctionnement de votre équipement. Ces codes diffèrent d'une région à l'autre et, habituellement, évoluent au fil du temps. Il vous incombe de déterminer les codes à respecter et de vous assurer que l'équipement, l'installation et le fonctionnement sont conformes aux exigences de la version la plus récente de ces codes.

Vous devez, à tout le moins, respecter toutes les sections applicables du Code national de prévention des incendies, du Code national de l'électricité et des codes de la National Electrical Manufacturer's Association (NEMA). Des organismes de réglementation ou des services gouvernementaux locaux peuvent également vous aider à déterminer les codes ainsi que les normes à respecter pour assurer une installation et un fonctionnement sûrs.

L'omission de respecter la totalité des codes et des normes applicables peut entraîner des dommages à l'équipement ou causer de graves blessures au personnel. Nous ne garantissons pas que les produits décrits dans cette publication conviennent à votre application particulière et nous n'assumons aucune responsabilité à l'égard de la conception, de l'installation ou du fonctionnement de votre produit.

Nos produits ne sont pas insensibles aux défaillances et ne sont ni conçus ni fabriqués pour l'utilisation ou la revente en tant qu'équipement de commande en ligne dans des environnements dangereux nécessitant une sécurité absolue, par exemple, l'exploitation d'installations nucléaires, les systèmes de navigation aérienne ou de communication, le contrôle de la circulation aérienne, les équipements de survie ou les systèmes d'armes, pour lesquels la défaillance du produit peut provoquer la mort, des blessures corporelles ou de graves dommages matériels ou environnementaux («activités à risque élevé»). La société **AutomationDirect** nie toute garantie expresse ou implicite d'aptitude à l'emploi en ce qui a trait aux activités à risque élevé.

Pour des renseignements additionnels touchant la garantie et la sécurité, veuillez consulter la section Modalités et conditions de notre documentation. Si vous avez des questions au sujet de l'installation ou du fonctionnement de cet équipement, ou encore si vous avez besoin de renseignements supplémentaires, n'hésitez pas à nous téléphoner au 770-844-4200.

Cette publication s'appuie sur l'information qui était disponible au moment de l'impression. À la société **AutomationDirect**, nous nous efforçons constamment d'améliorer nos produits et services. C'est pourquoi nous nous réservons le droit d'apporter des modifications aux produits ou aux publications en tout temps, sans préavis ni quelque obligation que ce soit. La présente publication peut aussi porter sur des caractéristiques susceptibles de ne pas être offertes dans certaines versions révisées du produit.

Marques de commerce

La présente publication peut contenir des références à des produits fabriqués ou offerts par d'autres entreprises. Les désignations des produits et des entreprises peuvent être des marques de commerce et appartiennent exclusivement à leurs propriétaires respectifs. **AutomationDirect** nie tout intérêt dans les autres marques et désignations.

DL05/06 Data Communications Module

AUTOMATIONDIRECT.com

Please include the Manual Number and the Manual Issue, both shown below, when communicating with Technical Support regarding this publication.

Manual Number: D0-DCM-M

Issue: 2nd Edition, Rev. B

Issue Date: 02/20

Publication History		
Issue	**Date**	**Description of Changes**
Original	08/05	Original issue
2nd Edition	09/07	***Direct***SOFT5 updates
2nd Edition, Rev. A	09/18	Converted to InDesign, general updates to reference D2-262 and D4-454 CPUs
2nd Edition, Rev. B	02/20	Updates

Table of Contents

D0-DCM User Manual

Chapter 1 - Introduction

Chapter 2 - Installation, Network Cabling and Module Specifications

Chapter 3 - D0-DCM Module Setup

Chapter 4 - *Direct*Net Communications Using RX/WX

Chapter 5 - Modbus Communications RX/WX and MRX/MWX

Chapter 6: DCM Using IBox Instructions

CHAPTER
1

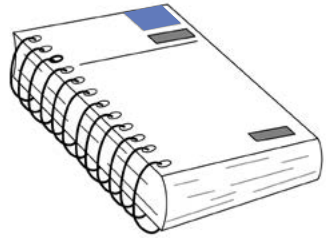
INTRODUCTION

In This Chapter...

Manual Overview

The Purpose of this Manual

This manual is designed to help you install, connect to and setup your DL05/06 Data Communications Module (D0–DCM). This manual explains how to configure the module's communications parameters and defines the important memory locations reserved for the module. Application examples, wiring diagrams, module configuration and programming examples are provided.



Supplemental Manuals

Depending on which products you have purchased, there may be other manuals that are necessary or helpful for your application. These are some suggested manuals:

User Manuals

- PLC User Manuals (D0-USER-M, D0-06USER-M)
- ***Direct***SOFT Programming Software

If you plan to use your D0-DCM module as an interface to HMI or PC Control software or to an Operator Interface panel, you may need to refer to the documentation for that product's specifications.

Technical Support

We strive to make our manuals the best in the industry and rely on your feedback in reaching our goal. If you cannot find the solution to your particular application, or, if for any reason you need additional technical assistance, please call us at

770–844–4200

Our technical support group will work with you to answer your questions. They are available Monday through Friday from 9:00 A.M. to 6:00 P.M. Eastern Time. We also encourage you to visit our web site where you can find technical and non-technical information about our products and our company.

http://www.automationdirect.com

Conventions Used

When the "notepad" icon is in the left–hand margin, the paragraph to its immediate right will be a special note.

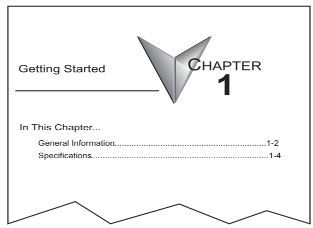
The word **NOTE** in boldface will mark the beginning of the text.

When the "exclamation mark" icon is in the left–hand margin, the paragraph to its immediate right will be a warning. This information could prevent injury, loss of property, or even death (in extreme cases).

The word **WARNING** in boldface will mark the beginning of the text which will also be in boldface..

Key Topics for Each Chapter

The beginning of each chapter will list the key topics tha be found in that chapter.



D0-DCM Overview

Important Configuration Information & PLC Firmware Requirements

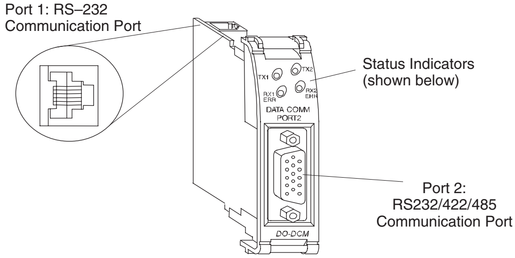
The D0-DCM's communications port parameters are configured using either the ***Direct***SOFT PLC>Setup>D0-DCM setup dialog box or ladder logic programming for DirectSOFT32 users (no DIP switch settings). If port 1 and/or port 2 default parameters are acceptable for your application, no setup is required.

TIP: If you intend to use port 2 as a network master, you must configure the port. Chapter 3 discusses port 1 and port 2 default parameters, V-memory configuration registers and provides port configuration examples. See Chapter 2 for PLC firmware and ***Direct***SOFT requirements.

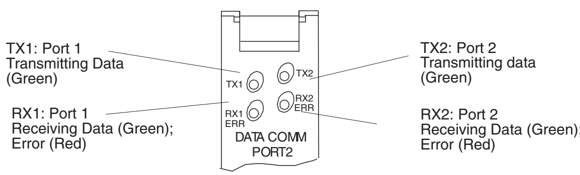
Hardware Features

The following diagrams show the D0-DCM hardware features.

D0–DCM



Status Indicators



Module Uses

The D0–DCM Data Communications Module is a general purpose communications interface that can be used in a DL05/06 PLC system. The module can occupy any option slot in the PLC. This module is primarily used as:

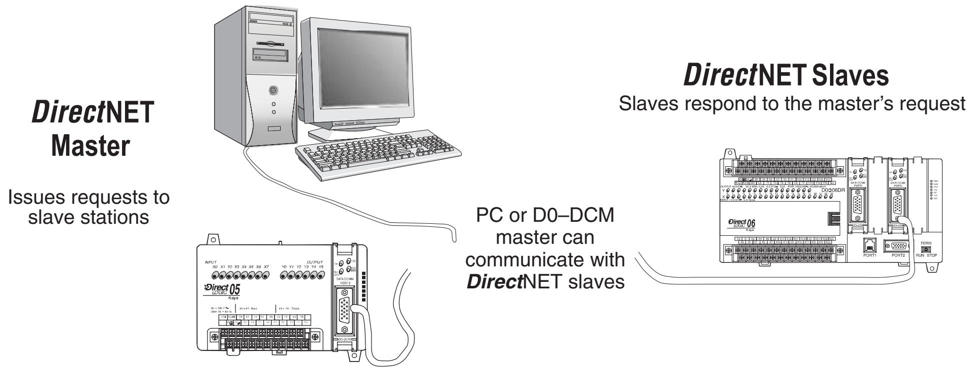
- an extra general purpose communications port to connect to a personal computer or operator interface
- a network master or slave station in a DirectNET network (port 2); port 1 functions as a DirectNET slave only
- a network master or slave station in a Modbus® RTU network (port 2); port 1 functions as a Modbus® RTU slave only
- a K–sequence slave (both ports)

DCM Application Examples

DirectNET Master or Slave

The D0-DCM can be used as a network interface for applications that require data to be shared between PLCs, or between PLCs and other devices such as PCs or operator interfaces. The D0-DCM can be configured as either a DirectNET master or slave station to share any type of system data including timer/counter values, V-memory data, control relay and I/O status, etc.

Using a D0-DCM as a network Master



The D0–DCM can be used with a DL05 or DL06 PLC to serve as a network master. A master is the network station that initiates requests for data from other stations (slaves) on the network. This is accomplished by using RLL network instructions in the master's PLC program. These instructions utilize the DCM's port 2 to read or write blocks of data to slave stations on the same network. RX/WX and NETCFG/NETRX/NETWX are examples of network instructions used in the master PLC's program.

Possible Slaves

- DL05/DL06 CPU (either port)
- DL05/DL06 CPU w/D0-DCM
- D2-240, D2-250-1, D2-260, or D2-262 CPU (either port)
- D2-240, D2-250-1, D2-260, or D2-262 CPU w/ D2-DCM
- D3-330/330P w/ DCU
- D3-340/350 (either port)
- D4-430/440 (bottom port)
- D4-450 or D4-454 (phone jack or bottom port)
- Any DL405 CPU w/ D4-DCM

Using a D0-DCM as a network Slave

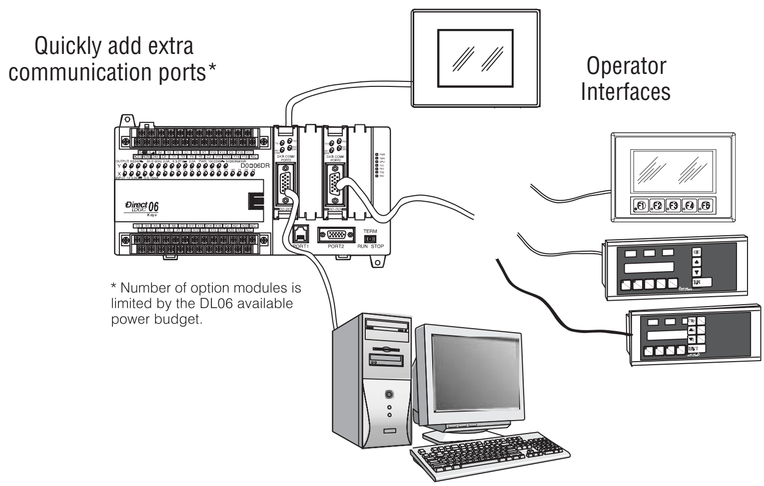
The D0-DCM can also be used with a DL05 or DL06 CPU to serve as a network slave station. In this case, the D0-DCM "listens" to the network for any messages that contain the D0-DCM's address. The D0-DCM carries out the master's request to read or write data, and sends confirmation and/or information to the master station. RLL network instructions are not necessary in the slave's program for the DCM to operate as a slave on the network. Port 1 and port 2 support slave mode operation.

Possible Masters

- DL05/06 CPU (port 2)
- DL05/06 CPU w/ D0-DCM
- D2-250-1, D2-260 or D2-262 CPU (bottom port)
- D2-240, D2-250–1, D2-260 or D2-262 CPU w/ D2-DCM
- D3-340 or D3-350 CPU (bottom port)
- Any DL405 CPU w/ D4-DCM
- D4-450 and D4-454 CPU (bottom port)
- Host computer w/ KEPDirect for PLCs

Additional Communications Port

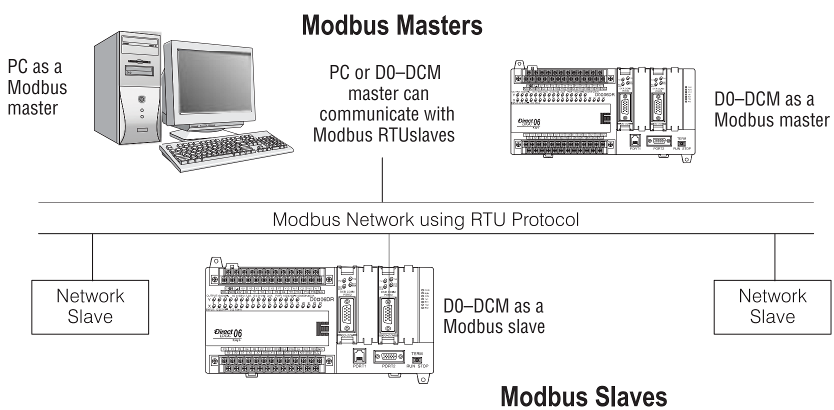
The D0-DCM ports are similar to the ports on the DL05 and DL06 CPUs. In general, if you can connect a device to the CPU ports, then you can also connect the same device to the D0-DCM. These devices can be a variety of things, such as operator interfaces or personal computers. In this application, the DCM is acting as a slave port to the master O/I(s) or PC(s). Port 1 and port 2 support slave mode operation.



Modbus RTU Master or Slave

The D0–DCM can be used as network interface to connect your DL05/06 system to a Modbus RTU network. Port 1 can serve as a Modbus RTU slave only. Port 2 can serve as a Modbus RTU master or slave. This manual does not describe the Modbus protocol. We recommend that you reference the Gould MODBUS Protocol Reference Guide (P1-MBUS-300 Rev. B) for details on the protocol if necessary. There may be more recent editions of this manual, so check with your Modbus supplier before ordering the documentation.

NOTE: *For information about the Modbus protocol see the Group Schneider Web site at:* ***www.schneiderautomation.com****. At the main menu, select Support/Services, Modbus, Modbus Technical Manuals, PI-MBUS-300 Modbus Protocol Reference Guide or search for PIMBUS300.*



Installation, Network Cabling and Module Specifications

Chapter 2

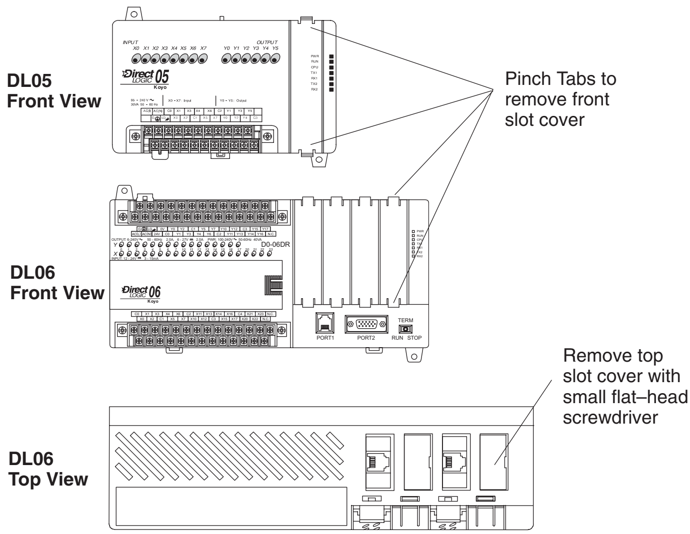
In This Chapter...

Inserting the D0-DCM into the PLC

D0-DCM Module Installation

Remove the front protective option slot cover by squeezing the pinch tabs and lifting the cover off. Remove the top option slot cover using small flat-head screwdriver or similar device. Be sure PLC power is off when installing the D0-DCM module.

Insert the module into the open slot in the DL05 or into any one of the four slots in the DL06. Locate the module so the printed information is oriented in the same direction as the markings on the PLC. Be careful to align the female connector on the printed circuit board of the module with the male connector on the PLC mother board. Press the module into the slot until the front of the module is flush with the front of the PLC. Check the DL06 power budget to be sure that it remains within the power supply limits before installing more modules.



Insert the module into the open slot in the DL05 or into any one of the four slots in the DL06. Locate the module so the printed information is oriented in the same direction as the markings on the PLC. Be careful to align the female connector on the printed circuit board of the module with the male connector on the PLC mother board. Press the module into the slot until the front of the module is flush with the front of the PLC. Check the DL06 power budget to be sure that it remains within the power supply limits before installing more modules.

PLC Firmware and DirectSOFT Requirements

NOTE: *The DL05 CPU's communication feature for the D0-DCM requires DirectSOFT32 Version 3.0c (or later) and firmware version 5.00 (or later). The DL06 requires DirectSOFT32 version V4.0, build 16 (or later) and firmware version 1.90 (or later). See our web site for firmware information and downloads:* ***www.automationdirect.com***

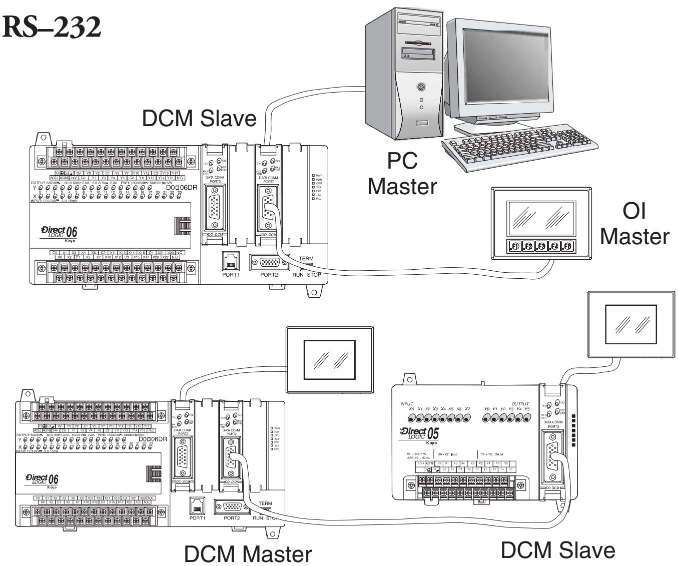
Building the Communication Cable

There are several considerations that help determine the type of cable needed for your D0–DCM application. The next few pages discuss these considerations in detail.

Consideration 1: Physical Configuration

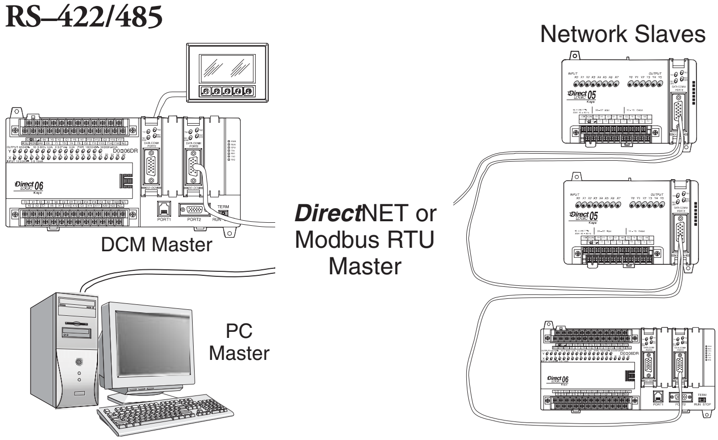
The D0–DCM can be used in either a point-to-point or multi-drop configuration. A point-to-point connection only has two stations, a master and a slave. Use the point-to-point configuration to connect a PC, an operator interface, or an intelligent device to a single D0–DCM. You also use this configuration when you connect a ***Direct***NET or (Modbus RTU) master station to a single ***Direct***NET or (Modbus RTU) slave station, respectfully.

Point to Point – RS–232



Use the multi-drop configuration to connect one master to two or more slaves.

Multi-drop – RS–422/485



Consideration 2: Electrical Specification RS232C or RS422/485

The D0-DCM can support RS-232 (ports 1 and 2) or RS-422/485 (port 2) communication. Your application and configuration choice will help determine which electrical specification is best for you. If you are using multi-drop, you must use RS-422 or RS-485. If you are using point-to-point, you may choose between RS-232, RS-422 or RS-485.

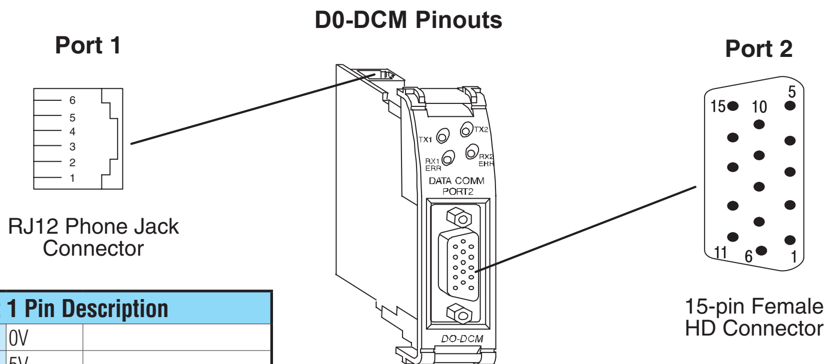
You can use RS-232 if the cable length is less than 50 feet and if the cable will not be subjected to induced electrical noise that is commonly found near welders, large motors, or other devices that create large magnetic fields.

You must use RS-422/485 for all other applications. RS–422/485 allows longer cable distances (up to 3300 feet) and provides higher noise immunity.

Although the network configuration and electrical specification are important, the type of devices being connected to the D0-DCM are just as important. The exact cable schematic needed really depends on a combination of all three things.

The following diagram shows the port pinouts for the D0-DCM.

Consideration 3: Cable Schematics



Port 1 Pin Description		
1	0V	
2	5V	
3	RXD	Receive Data (RS-232)
4	TXD	Transmit Data (RS-232)
5	RTS	Request to Send
6	0V	

Port 1 Pin Description		
1	5V	5VDC
2	TXD2	Transmit Data (RS-232)
3	RXD2	Receive Data (RS-232)
4	RTS2	Ready to Send Data (RS-232)
5	CTS2	Clear to Send (RS-232)
6	RXD2 –	Receive Data – (RS-422/485)
7	0V	Logic Ground
8	0V	Logic Ground
9	TXD2 +	Transmit Data + (RS-422/485)
10	TXD2 –	Transmit Data – (RS-422/485)
11	RTS2 +	Request to Send + (RS-422/485)
12	RTS2 –	Request to Send – (RS-422/485)
13	RXD2 +	Receive Data + (RS-422/485)
14	CTS2 +	Clear to Send + (RS-422/485)
15	CTS2 –	Clear to Send – (RS-422/485

Consideration 4: Cable Specifications

Although many types of cables may work for your application, we recommend you use a cable that is constructed to offer a high degree of noise immunity. The following specifications are to be used as a guideline.

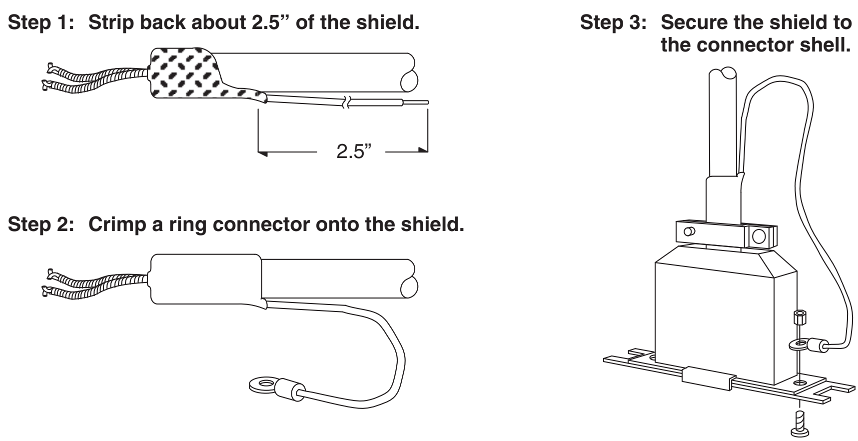
Structure	Shielded, twisted-pair (RS232 only uses two wires and a ground)
Conductor size	24AWG or larger
Insulation	Polyethylene
Shield	Copper braid or aluminum foil
Impedance	100Q @ 1MHz
Capacitance	60pf / meter or less

Consideration 5: Installation Guidelines

Your company may have guidelines for cable installation. If so, you must check those before you begin the installation. Here are some general things to consider.

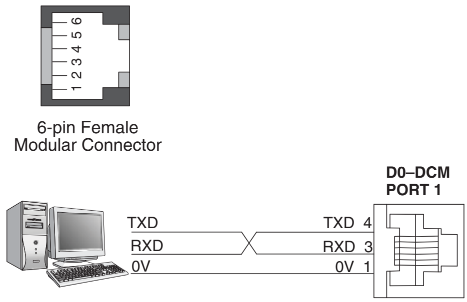
- Don't run cable next to larger motors, high current switches, or transformers. This may cause noise problems.
- Route the cable through an approved cable housing to minimize the risk of accidental cable damage. Check local and national codes to choose the correct method for your application.
- Consider redundant cabling if the application data is critical. This allows you to quickly reconnect all stations while the primary cable is being repaired.

Cable Shield Grounding — It is important to ground the cable shield to minimize the possibility of noise. The preferred method is to connect one end of the cable shield to the connector housing. If noise problems are still present and you have a good earth ground for the cabinet, you must connect one end of the shield to the cabinet earth ground. Don't ground both ends of the shield because this will create induced noise on the cable.

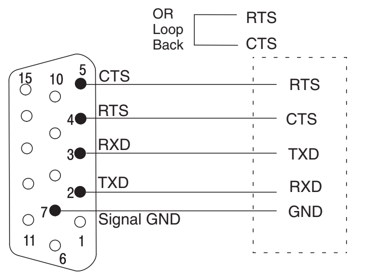


Wiring Diagrams

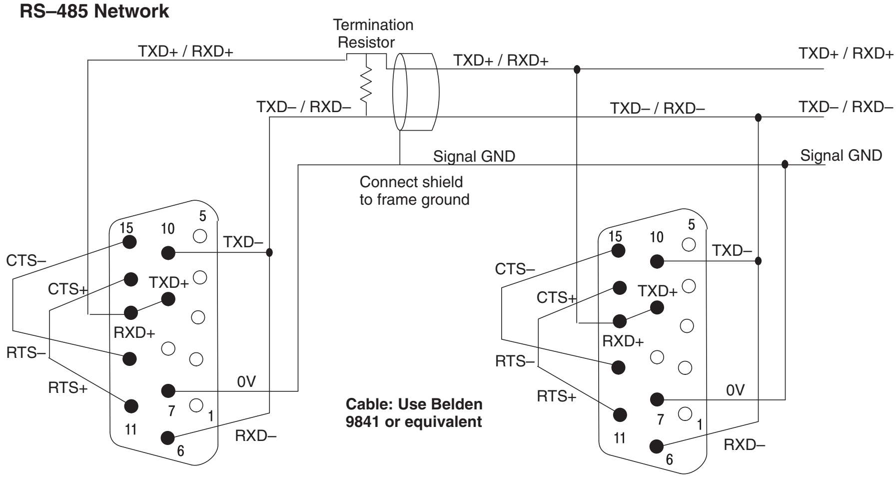
D0–DCM Port 1
RS–232 Network



D0–DCM Port 2
RS–232 Network

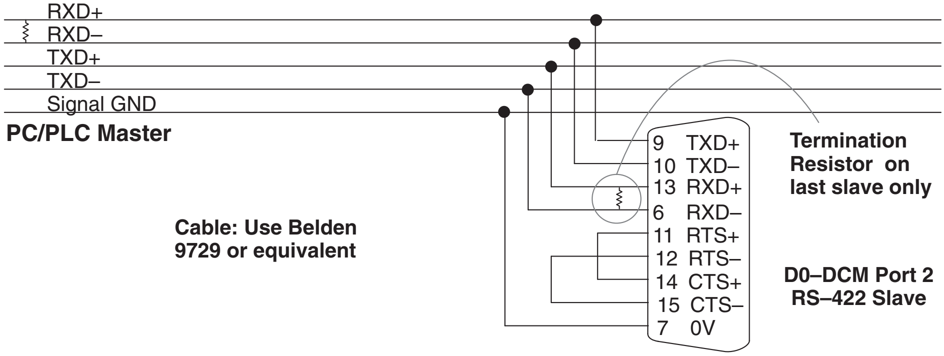


D0–DCM Port 2
RS–485 Network



D0–DCM Port 2

RS–422 Network



RS–422/485 Multi-drop Termination Resistors — It is important you add termination resistors at each end of the RS422/485 line. This helps reduce data errors during data transmission. You must select resistors that match the cable impedance. For example, a typical 22AWG solid conductor cable with 4.5 twists per foot has a typical impedance of about 120 ohm.

There are two ways to connect the resistors:

- Line-to-Line — this method balances the receive data lines (IN+ and IN–) and requires one resistor at each end of the line. (The cable diagrams we've provided show this method, but you can use either).
- Line-to-Ground — this method also balances the receive data lines, but common mode noise rejection is improved significantly. This method requires two resistors at each end of the line. Also, since there are two resistors, the sum total of both resistors must match the cable impedance.

Module Specifications

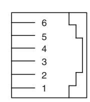
General Specifications

General Specifications	
Power Budget Requirement	250mA @ 5VDC (Not including external 5VDC consumption)
Maximum Number of Modules	DL05: 1; DL06: 4
Operating Temperature	32°F to 131°F (0°C to 55°C)
Storage Temperature	–4°F to 158°F (–20°C to 80°C)
Operating Humidity	5 to 95% (non-condensing)
Air Composition	No corrosive gases permitted
Vibration	MIL STD 810C, Method 514.2
Shock	MIL STD 810C, Method 516.2
Voltage Isolation	1000VAC, 1 minute duration
Insulation Resistance	10M ohms at 500VDC
Noise Immunity	NEMA ICS3–304, UL, CE, (FCC Class A) Class 1, Division 2 (C1D2)
Weight	1.75 oz. (50g)

Port 1 Specifications

Connector	6-pin Female Modular (RJ12)
Communications	RS-232
Protocol (auto-detection)	***Direct***Net slave K-sequence slave Modbus RTU slave
Station Number	0–247
Communication Data	8 data bits, 1 start bit, 1 stop bit (fixed)
Parity Bit	None, Odd
Baud Rates	9600, 19200, 38400, 57600, 115200 bps
Transmit Mode	ASCII, Hex
Maximum Distance	RS–232: 50ft (15 meters)

Port 1



RJ12 Phone Jack Connector

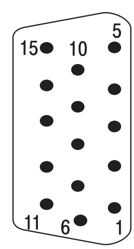
Port 1 Pin Description		
1	0V	
2	5V	
3	RXD	Receive Data (RS-232)
4	TXD	Transmit Data (RS-232)
5	RTS	Request to Send
6	0V	

Port 2 Specifications

Connector	15-pin Female Modular (RJ12)
Communications	RS-232 RS-422/485
Protocol (auto-detection)	***Direct***Net slave K-sequence slave Modbus RTU slave Non–sequence (ASCII IN/OUT)
Station Number	0–247
Communication Data	8/7 data bits, 1 start bit, 1/2 stop bit (fixed)
Parity Bit	None, Odd
Baud Rates	300, 600, 1200, 4800, 9600, 19200, 38400, 57600, 115200 bps
Transmit Mode	ASCII, Hex
Communications Time–out	Base time x (1–50)
Response Delay Time	0/2/5/10/20/50/100/500ms
Character Time–out	0–9999ms
Maximum Distance	RS–232: 50ft (15m) RS422/485 – 1000m

Port 1 Pin Description		
1	5V	5VDC
2	TXD2	Transmit Data (RS-232)
3	RXD2	Receive Data (RS-232)
4	RTS2	Ready to Send Data (RS-232)
5	CTS2	Clear to Send (RS-232)
6	RXD2 –	Receive Data – (RS-422/485)
7	0V	Logic Ground
8	0V	Logic Ground
9	TXD2 +	Transmit Data + (RS-422/485)
10	TXD2 –	Transmit Data – (RS-422/485)
11	RTS2 +	Request to Send + (RS-422/485)
12	RTS2 –	Request to Send – (RS-422/485)
13	RXD2 +	Receive Data + (RS-422/485)
14	CTS2 +	Clear to Send + (RS-422/485)
15	CTS2 –	Clear to Send – (RS-422/485

Port 2



15-pin Female
HD Connector

Indicator	State		Definition
TXT	ON	Green	Port 1 transmitting data
RX1	ON	Green	Port 1 receiving data
(ERR)	ON	Red	Port 1 Timeout, NAK or Exception Response
TXT2	ON	Green	Port 2 transmitting data
RX2	ON	Green	Port 2 receiving data
(ERR)	ON	Red	Port 2 Timeout, NAK or Exception Response

D0-DCM MODULE SETUP

CHAPTER 3

In This Chapter...

Important Module Configuration Information

The D0-DCM's communications port parameters are configured using either the ***Direct***SOFT PLC>Setup>D0-DCM setup dialog box or ladder logic programming for ***Direct***SOFT users. If port 1 and/or port 2 default parameters are acceptable for your application, no setup is required. (Tip: If you intend to use port 2 as a network master, you must configure the port).

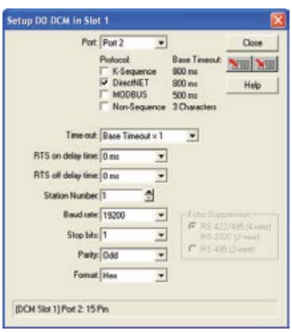
The "DCM Port 1 and Port 2 Configuration Registers" section lists port 1 and port 2 default parameters and V-memory configuration registers used by the DCM module(s).

NOTE: *The DL05 CPU's communication feature for the D0-DCM requires* ***Direct****SOFT Version 3.0c (or later) and firmware version 5.00 (or later). The DL06 requires* ***Direct****SOFT version V4.0, build 16 (or later) and firmware version 1.90 (or later). See our web site for firmware information and downloads:* ***www.automationdirect.com.***

Tip for *Direct*SOFT Users (optional)

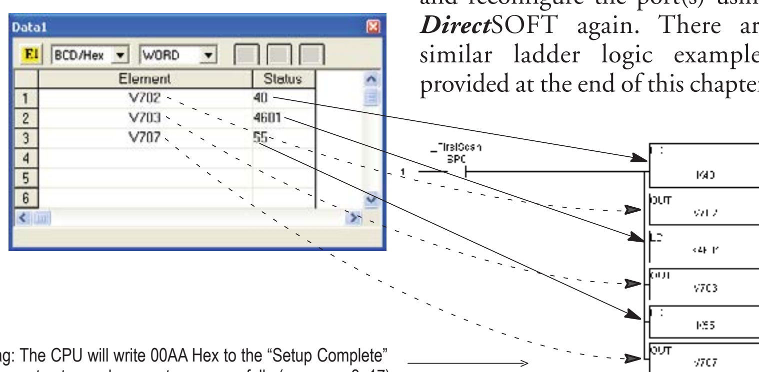
If you intend to use ladder logic in your program to configure the DCM ports, you can use the DCM setup dialog box and a Data View window to quickly determine the BCD/HEX values to use in your port setup ladder logic code. This method greatly simplifies the process. The procedure is summarized below and covered in this chapter.

Step 1: Use the ***Direct***SOFT PLC>Setup>D0–DCM setup dialog box to configure the DCM port(s) as needed for your application. Save the port configuration to the CPU.



Step 2: In ***Direct***SOFT, open a Data View window and type in the special V-memory locations used for the DCM module based on the slot the module is occupying (see page 3–9). The BCD/HEX register values needed to support your specific port communications selections will be displayed in the Data View window.

Step 3: Create a ladder rung in your PLC program to write the values determined in the previous step to the special V-memory locations used for the module. Creating this rung ensures that your specified communications port parameters will be maintained after a power outage, power cycle, etc. without having to reconnect and reconfigure the port(s) using ***Direct***SOFT again. There are similar ladder logic examples provided at the end of this chapter.

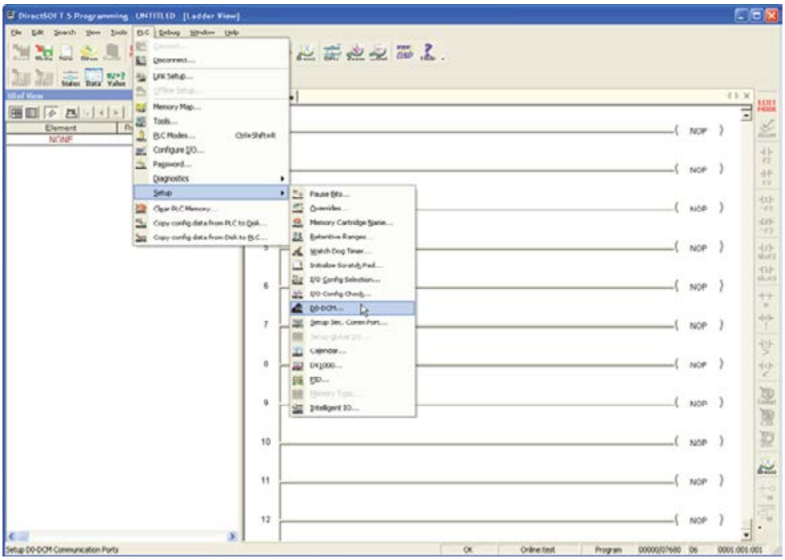


Setup Completion Flag: The CPU will write 00AA Hex to the "Setup Complete" register after and if the port setup code executes successfully (see page 3–17).

Using *Direct*SOFT to Configure the DCM

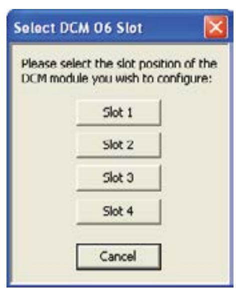
*Direct*SOFT PLC>Menu>Setup

Using the D0-DCM setup dialog box in ***Direct***SOFT is the easiest way to configure the D0–DCM communications port parameters. The DCM must be installed in an option slot and the PLC must be powered up and connected to a PC running ***Direct***SOFT or later. It is recommended to connect your PC to port 1 on the DL05/06 CPU to setup the DCM module, however, CPU port 2 or an ECOM Ethernet link will work. Once you're on–line with the PLC, click on PLC>Setup>D0–DCM.



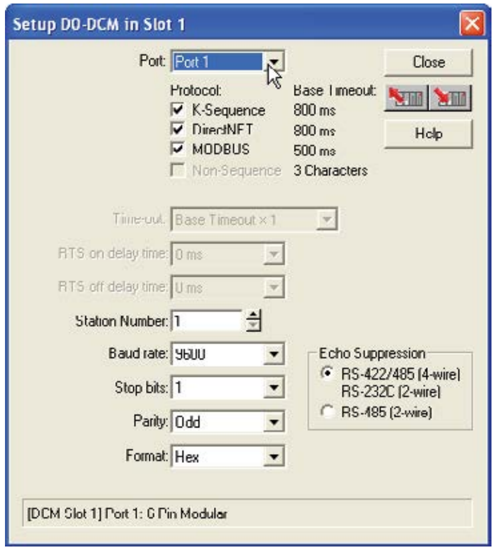
Select DCM Slot

Select the option slot number that the target DCM is occupying. The DL06 PLC can support multiple DCM's, so be sure to click on the proper slot number 1-4. Once you click on a slot number, the D0-DCM port setup window will display as shown on the following pages.

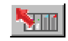


Port 1 Configuration (slave only)

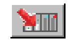
The D0-DCM Setup window allows you to verify or make any necessary changes to the communications port parameters for your specific application.



Read from module



Write to module



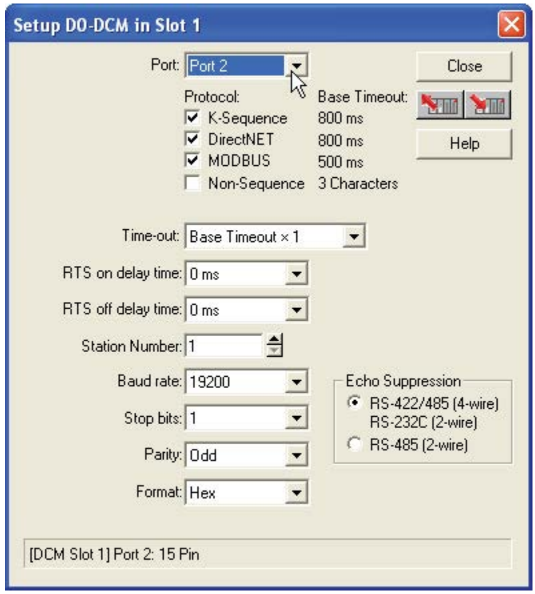
- **Port:** From the port number list box, choose "Port 1 ".
- **Protocol:** By default, all protocols are selected which results in "auto–detect" mode. The port will automatically determine which protocol is being used to communicate with it and operate accordingly. There is no advantage in deselecting unused protocols (selecting single protocols is for master mode use on port 2).
- **Station Number:** The allowable range for ***Direct***NET slaves is from 1 to 90. The allowable range for Modbus RTU slaves is from 1 to 247. Each slave must have a unique, but do not need to be consecutive.
- **Baud Rate:** The available baud rates include 9.6K to 115.2K baud. Choose a higher baud rate initially, reverting to lower baud rates if you experience data errors or noise problems on the network. Important: You must configure the baud rates of all devices on the network to the same value.
- **Stop Bits:** Select 1 or 2 stop bits for use in the protocol.
- **Parity:** Select none, even, or odd parity for error checking.
- **Format:** Select hex or ASCII formats.
- **Echo Suppression:** Select port 1 wiring (applies to Modbus protocol only).

Write to module

Then click the button indicated to send the Port configuration to the Module, and click Close

Port 2 Configuration (slave mode)

Click on “Port 2” to display its parameter settings. Make any changes as necessary for your application.



Read from module

Write to module

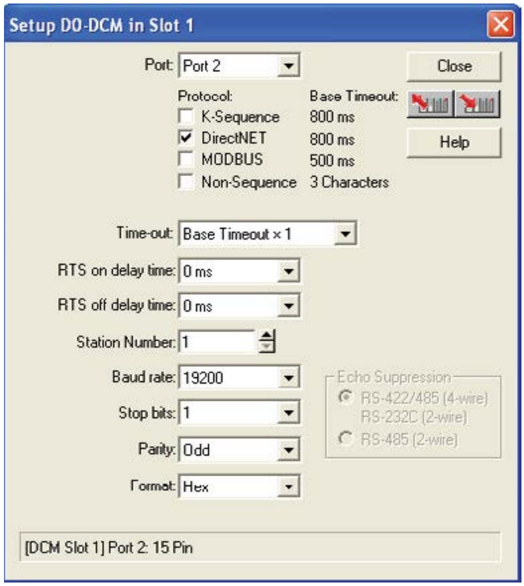
- **Port:** From the port number list box, choose “Port 2”.
- **Protocol:** By default, all protocols (except non–sequence) are selected which results in “auto–detect” mode. The port will automatically determine which protocol is being used to communicate with it and operate accordingly. There is no advantage in deselecting unused protocols (selecting a single protocol is for master mode use on port 2).
- **Timeout:** amount of time the port will wait after it sends a message to get a response before logging an error.
- **RTS On Delay Time:** The amount of time between raising the RTS line and sending the data.
- **RTS Off Delay Time:** The amount of time between resetting the RTS line after sending the data.
- **Station Number:** The allowable range for ***Direct***NET slaves is from 1 to 90. The allowable range for Modbus RTU slaves is from 1 to 247. Each slave must have a unique, but do not need to be consecutive.
- **Baud Rate:** The available baud rates include 300 to 115.2K baud. Choose a higher baud rate initially, reverting to lower baud rates if you experience data errors or noise problems on the network. Important: You must configure the baud rates of all devices on the network to the same value.
- **Stop Bits:** Choose 1 or 2 stop bits for use in the protocol. Parity: Choose none, even, or odd parity for error checking.
- **Format:** Choose hex or ASCII formats.
- **Echo Suppression:** Select port 2 wiring (applies to Modbus protocol only)

Write to module

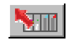
Then click the button indicated to send the Port configuration to the Module, and click Close.

Port 2 Configuration (*Direct*NET

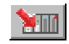
To configure Port 2 for ***Direct*****NET** master



Read from module

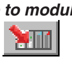


Write to module



- **Port:** From the port number list box, choose "Port 2 ".
- **Protocol:** Click the check box to the left of "***Direct***NET".
- **Timeout:** amount of time the port will wait after it sends a message to get a response before logging an error.
- **RTS On Delay Time:** The amount of time between raising the RTS line and sending the data.
- **RTS Off Delay Time:** The amount of time between resetting the RTS line after sending the data.
- **Station Number:** For making the DCM port a ***Direct***NET master, choose "1". The allowable range for ***Direc***tNET slaves is from 1 to 90 (each slave must have a unique number). At power up, the port is a slave, unless and until the CPU executes network read/write instructions which uses the DCM port as a master. Thereafter, the port reverts back to slave mode until network read/write instructions use the port again.
- **Baud Rate:** The available baud rates include 300 to 115.2K baud. Choose a higher baud rate initially, reverting to lower baud rates if you experience data errors or noise problems on the network. Important: You must configure the baud rates of all devices on the network to the same value.
- **Stop Bits:** Choose 1 or 2 stop bits for use in the protocol. Parity: Choose none, even, or odd parity for error checking. Format: Choose hex or ASCII formats.
- **Parity:** Choose none, even, or odd parity for error checking
- **Format:** Choose hex or ASCII formats.

Write to module

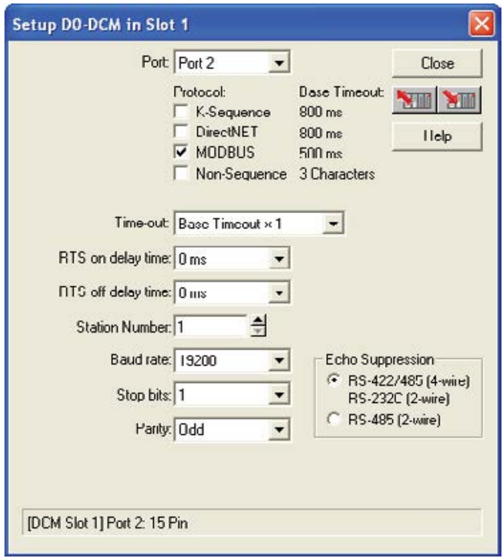


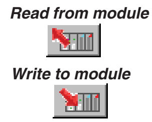
Then click the button indicated to send the Port configuration to the Module, and click Close.

Port 2 Configuration (Modbus Master)

To configure Port 2 for Modbus® RTU master operation:

- **Port:** From the port number list box at the top, choose “Port 2”.
- **Protocol:** Click the check box to the left of “MODBUS”.
- **Timeout:** amount of time the port will wait after it sends a message to get a response before logging an error.
- **RTS On Delay Time:** The amount of time between raising the RTS line and sending the data.
- **RTS Off Delay Time:** The amount of time between resetting the RTS line after sending the data.



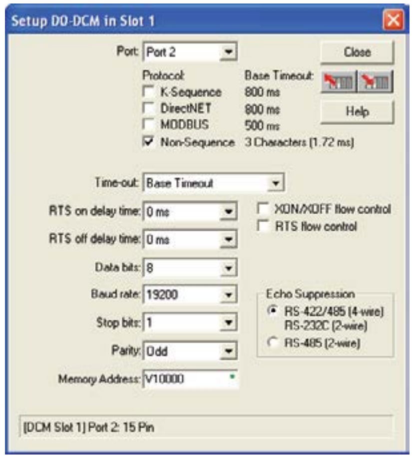


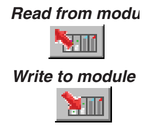
- **Station Number:** For making the DCM port a Modbus master, choose “1”. The possible range for Modbus slave numbers is from 1 to 247 when using the MRX/MWX network instructions (WX/RX network instructions limits slaves 1 to 90). Each slave must have a unique number. At power up, the port is a slave, unless and until the CPU executes network read/write instructions which uses the DCM port as a master. Thereafter, the port reverts back to slave mode until network read/write instructions use the port again.
- **Baud Rate:** The available baud rates include 300 to 115.2K baud. Choose a higher baud rate initially, reverting to lower baud rates if you experience data errors or noise problems on the network. Important: You must configure the baud rates of all devices on the network to the same value.
- **Stop Bits:** Choose 1 or 2 stop bits for use in the protocol. Parity: Choose none, even, or odd parity for error checking. Echo Suppression: Select port 2 wiring method.
- **Parity:** Choose none, even, or odd parity for error checking.
- **Echo Suppression:** Select port 2 wiring method.

Then click the button indicated to send the Port configuration to the Module, and click Close.

Port 2 Configuration (Non–Sequence)

Configuring port 2 on the DCM for Non–Sequence allows the CPU to use the DCM port to read/write raw ASCII strings using the DL05/06 ASCII instructions. Refer to Chapter 5 in the DL05/06 PLC User Manual for all available ASCII/Print instructions. In ***Direct***SOFT, select the PLC menu, then Setup, then “D0-DCM”.





- **Port:** From the port number list box at the top, choose “Port 2”.
- **Protocol:** Click the check box to the left of “Non–Sequence”.
- **Timeout:** amount of time the port will wait after it sends a message to get a response before logging an error.
- **RTS On Delay Time:** The amount of time between raising the RTS line and sending the data.
- **RTS Off Delay Time:** The amount of time between resetting the RTS line after sending the data.
- **Data Bits:** Select either 7-bits or 8-bits to match the number of data bits specified for the connected devices.
- **Baud Rate:** The available baud rates include 300 to 115.2K baud. Choose a higher baud rate initially, reverting to lower baud rates if you experience data errors or noise problems on the network. Important: You must configure the baud rates of all devices on the network to the same value.
- **Stop Bits:** Choose 1 or 2 stop bits to match the number of stop bits specified for the connected devices.
- **Parity:** Choose none, even, or odd parity for error checking. Be sure to match the parity specified for the connected devices.
- **Memory Address:** Please choose a memory address with 64 words of contiguous free memory for use by Non-Sequence Protocol.
- **Xon/Xoff Flow Control:** Choose this selection if you have port 2 wired for Hardware Flow Control (Xon/Xoff) with RTS and CTS signal connected between all devices.
- **RTS Flow Control:** Choose this selection if you have Port 2 RTS signal wired between all devices.
- **Echo Suppression:** Select the appropriate radio button based on the wiring configuration used on port 2.

Then click the button indicated to send the Port configuration to the Module, and click Close.

D0–DCM Port Configuration Registers

Module Configuration Registers

The table below lists the special V-memory locations used by the DL05/DL06 PLCs for the D0–DCM module based on the slot the module is occupying. The registers, by slot, are used regardless of the method you use to configure the module (***Direct***SOFT or ladder logic). The following pages define each register's function as referenced by the letter (A,B,C,...) in the table. DL05 and DL06 ladder logic examples are provided beginning on page 3–18.

Module Configuration Parameters		DL05 and DL06 Option Slot					
		Word Offset	DL05 Slot 1	DL06 Slot 1	DL06 Slot 2	DL06 Slot 3	DL06 Slot 4
A	**Port 1**–Transmit Mode (ASCII/Hex), Protocol	+0000	V7700	V700	V710	V720	V730
B	**Port 1**–Station Address, Baud Rate , Parity	+0001	V7701	V701	V711	V721	V731
C	**Port 2**–RTS On/Off Delay, Transmit Mode (ASCII/ Hex), Protocol, Comm Time–out, RS–485 Mode Select	+0002	V7702	V702	V712	V722	V732
D	**Port 2**–Station Address, Baud Rate, Data Bit , Stop Bit , Parity	+0003	V7703	V703	V713	V723	V733
E	**Port 2** – Memory Address (Non-Sequence protocol)	+0005	V7705	V705	V715	V725	V735
F	**Port 2** – Character Time–out	+0006	V7706	V706	V716	V726	V736
G	**Port1/Port 2** – Setup Completion Code	+0007	V7707	V707	V717	V727	V737
H	**Port 1/Port 2** – Reset Time-out		V7730	V7730	V7731	V7732	V7733

Default Communications Parameters

On power up, the DCM will write the necessary data to the V-memory configuration registers to result with the following default port communications parameters. If you need to change any of the default settings or configure the module for network master operation, you must use either the ***Direct***SOFT >PLC>Setup>D0–DCM setup dialog box or ladder logic programming for DirectSOFT users.

Parameter	Port 1	Port 2
Mode	Slave	
Baud Rate	9600bps	19200bps
Parity	Odd	
Protocol	K-Sequence/***Direct***Net/Modbus (auto–detect)	
Station Address	1	
Data Bits	8 (fixed)	8
Stop Bits	1 (fixed)	1

Parameter Descriptions

Protocol Selection: Slave mode (ports 1 and 2) – The default protocol setting for ports 1 and 2 is referred to as "auto–detect" mode (all protocols are selected except non–sequence for port 2). With this selection, the port will automatically determine which protocol is being used to communicate to it and operate accordingly. This selection is fine if you intend to program/monitor the CPU through the D0-DCM using ***Direct***SOFT, or connect it to an operator interface, etc. You can select a single protocol if desired.

Master mode (port 2 only) – The DCM can serve as a ***Direct***Net or Modbus master. When using port 2 as a master, you must select the single appropriate protocol for the master port to use when communicating to the slave device(s) and set the station address to "1". At power up, the port is a slave, unless and until the CPU executes network read/write instructions which uses the DCM port as a master. Thereafter, the port reverts back to slave mode until network read/write instructions use the port again.

Communication Timeout: Communication Timeout Disable is normally used only if you're developing your own ***Direct***NET programs. By disabling the timeout, you can send one ***Direct***NET component without any communication timeout problems. If you have this timeout disabled and a communication error does occur, you must restart communications by sending a retry or an End of Transmission (EOT) command. If you want to know more, see the ***Direct***NET manual for details.

Transmit Mode: Select between ASCII and HEX modes of data representation. If you want the fastest communication possible, use HEX mode, which is the default. The difference is in the way the data is represented. The same data is twice as long in ASCII format, so if there's more data, it takes longer to transfer. If you have a device on the network that requires ASCII mode, then configure the DCM for ASCII mode, otherwise, use HEX mode.

Baud Rate: There are several baud rate selections available ranging from 300bps to 115.2Kbps. All stations must have the same baud rate setting before the communications will operate correctly. Usually, you should use the highest baud rate possible unless noise problems appear. If noise problems appear, then try reducing the baud rates.

Parity: Choose between none, even and odd parity for error checking.

RTS Delay Times: On Delay – The delay time specifies the amount of time the D0–DCM waits to send the data after it has raised the RTS signal line. This is normally set to 0, and is typically only adjusted if you are using the D0–DCM with a radio modem. If you are using the D0–DCM with a radio modem, check your modem documentation to help you choose the proper setting.

RTS Off Delay – the delay time specifies the amount of time the D0–DCM will wait to reset the RTS line after sending the data.

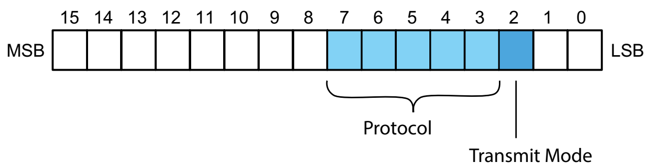
Station Address: The decimal addresses do not have to be consecutive, but each station must have a unique number. See protocol description above for port 2 master operation.

A: Port 1 – Transmit Mode, Protocol

Use word +0000 to set Port 1:

- K–Sequence slave, ***Direct***NET slave or Modbus RTU slave protocol (or auto–detect for all three protocols)
- HEX or ASCII transmit mode

Word +0000



Set all unused bits to zero.

Port 1: Transmit Mode	
Mode	**Bit 3**
Hex Mode	0
ASCII Mode	1

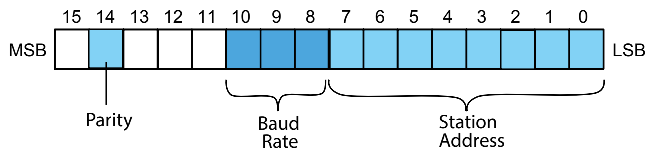
Port 1 Protocol					
Protocol	**Bit 7-4 (Hex)**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**
K-Sequence	8	1	0	0	0
DirectNet	4	0	1	0	0
MODBUS RTU	2	0	0	1	0
K-Seq / D-Net /MODBUS RTU"	E	1	1	1	0

B: Port 1 – Station Address, Baud Rate, Parity

Use word +0001 to set Port 1:

- Station address ranges from 0–247 (00–F7 Hex)
- Baud rates ranging from 9.6K to 115.2K bps
- Odd or No parity

Word +0001



Set all unused bits to zero.

Port 1: Station Address								
Address	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
00	0	0	0	0	0	0	0	0
01	0	0	0	0	0	0	0	1
02	0	0	0	0	0	0	1	0
– – – – –								
F6	1	1	1	1	0	1	1	0
F7	1	1	1	1	0	1	1	1

Port 1: Baud Rate			
Baud Rate	**Bit 10**	**Bit 9**	**Bit 8**
9.5 Kbps	0	0	0
19.2 Kbps	0	0	1
38.4 Kbps	0	1	0
57.6 Kbps	0	1	1
115.2 Kbps	1	0	0

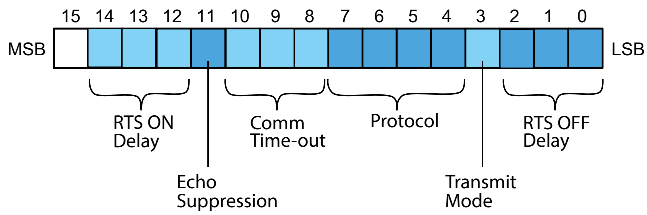
Port 1: Parity	
Parity	**Bit 14**
No Parity	0
Odd Parity	1

C: Port 2 – RTS On/Off delay, Transmit Mode, Protocol, Comm Time–out, RS–485 Mode

Use word +0002 to set Port 2:

- K–Sequence slave, ***Direct***NET slave or Modbus RTU slave protocol (or auto–detect for all three protocols)
- HEX or ASCII transmit mode
- RTS ON and OFF delay times
- Communication Time–out
- Echo Suppression

Word +0002



Set all unused bits to zero.

Port 2: RTS OFF Delay			
Time (ms)	**Bit 2**	**Bit 1**	**Bit 1**
0	0	0	0
2	0	0	1
5	0	1	0
10	0	1	1
20	1	0	0
50	1	0	1
100	1	1	0
500	1	1	1

Base Time-out for K-Seq/D-Net = 800ms

Base Time–out for MODBUS = 500ms

Port 2: Transmit Mode	
Mode	**Bit 3**
Hex Mode	0
ASCII Mode	1

Port 2: Protocol					
Protocol	Bit 7-4 (Hex)	Bit 7	Bit 6	Bit 5	Bit 4
K-Sequence	8	1	0	0	0
DirectNet	4	0	1	0	0
Modbus RTU	2	0	0	1	0
Non-Sequence	1	0	0	0	1
K-Seq / D-Net /Modbus RTU"	E	1	1	1	0

	Port 2: Communication Time-out (ms)			
K-Seq, D-Net, Modbus RTU	Non-Sequence Protocol	Bit 10	Bit 9	Bit 8
Base Time x 1	0	0	0	0
Base Time x 1.2	2	0	0	1
Base Time x 1.5	5	0	1	0
Base Time x 2	10	0	1	1
Base Time x 5	20	1	0	0
Base Time x 10	50	1	0	1
Base Time x 20	100	1	1	0
Base Time x 50	500	1	1	1

Port 2: Echo Suppression	
Mode	Bit 11
RS-422/485: 4-wire RS-232	0
RS-485: 2-wire	1

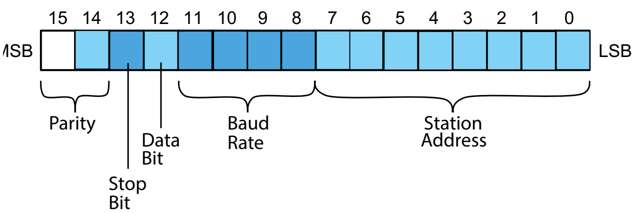
Port 2: RTS ON Delay			
Time (ms)	Bit 14	Bit 13	Bit 12
0	0	0	0
2	0	0	1
5	0	1	0
10	0	1	1
20	1	0	0
50	1	0	1
100	1	1	0
500	1	1	1

D: Port 2 – Station Address, Baud Rate, Data Bit, Stop Bit, Parity

Use word +0003 to set Port 2:

- Station address ranges from 0–247 (00–F7 Hex)
- Baud rates ranging from 9.6K to 115.2K bps
- Data Bit Length (7 or 8 bits)
- Stop Bit Length (1 or 2 bits)
- Odd, Even or No Parity

Word +0003



Set all unused bits to zero.

Port 2: Station Address								
Address	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
00	0	0	0	0	0	0	0	0
01	0	0	0	0	0	0	0	1
02	0	0	0	0	0	0	1	0
– – – – –								
F6	1	1	1	1	0	1	1	0
F7	1	1	1	1	0	1	1	1

Port 2: Baud Rate				
Baud Rate	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
300 bps	0	0	0	0
600 bps	0	0	0	1
1200 bps	0	0	1	0
2400 bps	0	0	1	1
4800 bps	0	1	0	0
9600 bps	0	1	0	1
19.2 Kbps	0	1	1	0
38.4 Kbps	0	1	1	1
57.6 Kbps	1	0	0	0
115.2 Kbps	1	0	0	1

Port 2: Data Bit	
Length	**Bit 12**
8 Bits	0
7 Bits	1

Port 2: Stop Bit	
Length	**Bit 13**
1 Bit	0
2 Bits	1

Port 2: Parity		
Parity	**Bit 15**	**Bit 14**
No Parity	0	0
Odd Parity	0	1
Even Parity	1	1

E: Port 2 – Memory Address - (Non-Sequence protocol)

Use word +0005 to set Port 2 Memory Address:

- This parameter is used when Port 2 is configured for Non-Sequence protocol.
- Hex value of the V-memory location to temporarily store ASCII data coming into the PLC
- This is the starting address of a contiguous block of 64 unused V-memory locations.

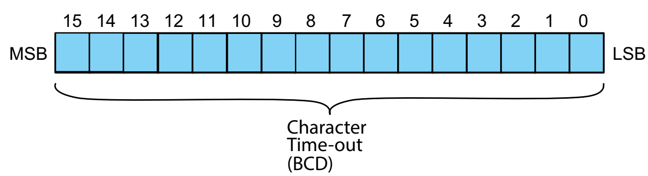
F: Port 2 – Character Time–out

Use word +0006 to set Port 2:

- Character Time–out (BCD)

This parameter is used when the DCM is used as a Modbus RTU master.

Word +0006

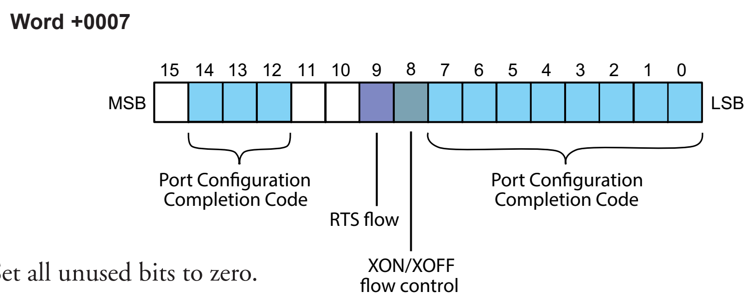


Port 2: Character Time-out	
Bit 15–0	**Time**
0	3.5 Characters
1–9999 (BCD)	Time-out (ms)

G: Port 1 and 2 Setup and Completion Code, XON/XOFF flow control (Port2) RTS flow control (Port 2)

Word +0007 is used for Port 1 and Port 2 to:

- Request that the DCM recognize changes in the port(s) communication parameters from default or previous settings
- Confirm that Port 1 and Port 2 configuration is complete
- Indicate if there are any errors in the Port 1 or Port 2 configuration parameters
- Allows enabling of XON/XOFF flow control in Port 2 when using Non-Sequence protocol
- Allows enabling RTS flow control in Port 2 when using Non-Sequence protocol



Set all unused bits to zero.

Port 2 NOT used for Non-Sequence protocol:

Loading a K0055 (BCD) into word +0007 will request that the DCM recognizes (looks for) new or desired port communication parameters that are written to the DCM port configuration registers. If the requested changes are valid, a 00AA (hex) will then be written to word +0007. If the desired or new communications are invalid or out of range, an error code 00E* (Hex) will be written to word +0007. The last digit (represented here with and asterisk) indicates the address that has an error. For example, error code 00E2 means that word +0002 has an error.

Port 2 used for Non-Sequence protocol:

Loading a K7055 (BCD) into word +0007 will request that the DCM recognize (looks for) new or desired port communication parameters that are written to the DCM port configuration registers, with XON/XOFF flow control turned OFF and RTS flow control turned OFF. If the requested changes are valid, a 70AA (hex) will be written to word +0007.

If the requested changes are invalid or out of range, an error code 70E* (hex) will be written to word +0007. The last digit (represented here with an asterisk) indicates the address that has an error. For example, a value of 70E2 means that word +0002 has an error.

Loading a value of 7155 (BCD) into word +0007 will request that the DCM recognize the requested port communication parameters and to enable XON/XOFF flow control. If the requested changes are valid, the CPU writes a value of 71AA (hex) to word +0007.

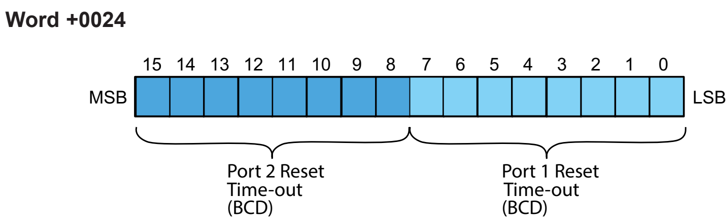
Loading a value of 7255 (BCD) into word +0007 will request that the DCM recognize the requested port communication parameters and to enable RTS flow control. If the requested changes are valid, the CPU writes a value of 72AA (hex) to word +0007.

Loading a value of 7355 (BCD) into word +0007 will request that the DCM recognize the requested port configuration parameters, enable XON/XOFF flow control and enable RTS flow control. If the requested changes are valid, the CPU writes a value of 73AA (hex) to word +0007.

H: Port 1 and 2 Reset Time–out

Use word +0024 to set Port 1 and Port 2:

- Reset Time–out in seconds
- Default value: 3030

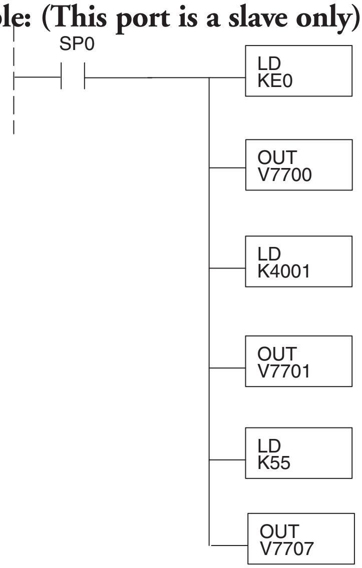


Reset Time-out		
Port	**Time(s)**	**Function**
Port 1: Bit 0–7	0	Disable
Port 2: Bit 15–8	1–99	Enable

Using Ladder Logic to Setup the D0-DCM (DL05)

The following tested ladder logic examples are provided by our technical support department for assistance only . We do not guarantee that the data is suitable for your particular application, nor do we assume any responsibility for them in your application.

Port 1 Example: (This port is a slave only)



Selects auto–detect protocol mode (K–seq, D–Net and MODBUS) and HEX transmit mode.

Write configuration data into V7700.

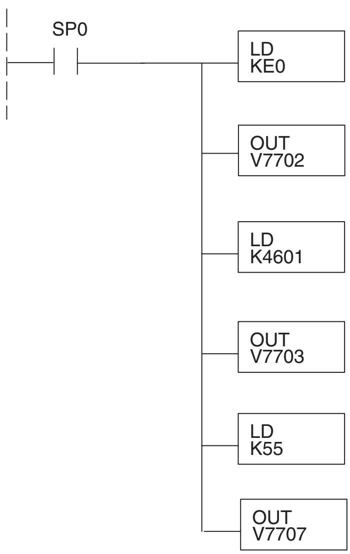
Selects station address 1, baud rate of 9600bps and odd parity.

Write configuration data into V7701.

K55 is the port setup completion code that must be used to request that the CPU recognizes the new/desired port parameters

Write configuration data into V7707.

Port 2 Example: Slave Mode



Selects auto–detect protocol mode (K–seq, D–Net and MODBUS), HEX transmit mode, RTS On/Off delay times are set to zero and RS–422/485 (4–wire) / RS–232

Write configuration data into V7702.

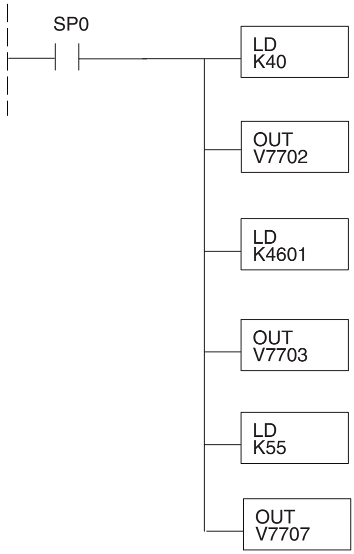
Selects station address 1, baud rate of 19200bps, odd parity, 8–bit data length and 1 stop bit.

Write configuration data into V7703.

K55 is the port setup completion code that must be used to request that the CPU recognizes the new/desired port parameters

Write configuration data into V7707.

Port 2 Example: *Direct*Net Master



Selects DirectNet protocol, HEX transmit mode, RTS On/Off delay times are set to zero and RS–422/485 (4–wire) / RS–232

Write configuration data into V7702.

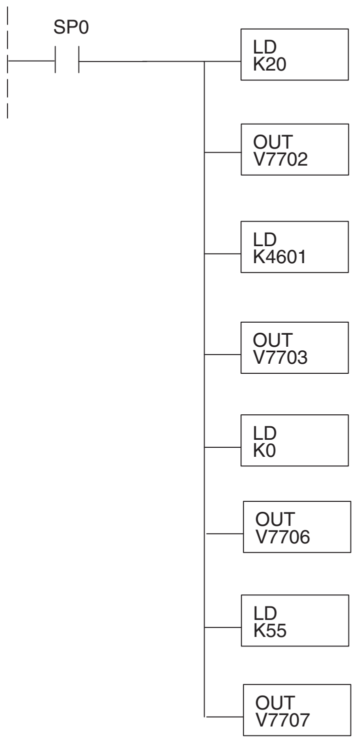
Selects station address 1, baud rate of 19200bps, odd parity, 8–bit data length and 1 stop bit.

Write configuration data into V7703.

K55 is the port setup completion code that must be used to request that the CPU recognizes the new/desired port parameters

Write configuration data into V7707.

Port 2 Example: Modbus RTU Master



Selects MODBUS protocol, HEX transmit mode, RTS On/Off delay times are set to zero and RS–422/485 (4–wire) / RS–232

Write configuration data into V7702.

Selects station address 1, baud rate of 19200bps, odd parity, 8–bit data length and 1 stop bit.

Write configuration data into V7703.

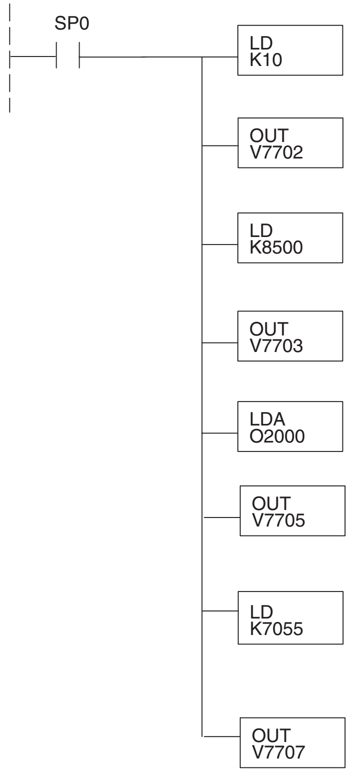
Selects 3.5 characters length time–out.

Write configuration data into V7706.

K55 is the port setup completion code that must be used to request that the CPU recognizes the new/desired port parameters

Write configuration data into V7707.

Port 2 Example: Non-Sequence Protocol



Selects non-sequence protocol, RTS On/Off delay times are set to zero and RS–422/485 (4–wire) / RS–232 is selected

Write configuration data into V7702.

Selects baud rate of 9600bps, even parity, 8–bit data length and 1 stop bit.

Write configuration data into V7703.

Selects V2000 as Memory Address, where ASCII data coming into the PLC is stored temporarily.

Write configuration data into V7705.

Loading a value of 7055 requests that the DCM verify the requested port parameters, does not enable Xon/Xoff flow control and does not enable RTS flow control.

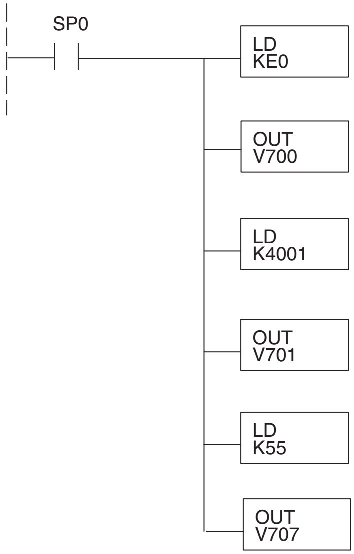
Write configuration data into V7707.

Using ladder Logic to Setup the D0-DCM (DL06)

The following tested ladder logic examples are provided by our technical support department for assistance only . We do not guarantee that the data is suitable for your particular application, nor do we assume any responsibility for them in your application.

The following examples assume the D0-DCM is installed in slot 1.

Port 1 Example: Slave Mode Only



Selects auto–detect protocol mode (K–seq, D–Net and MODBUS) and HEX transmit mode.

Write configuration data into V700 (based on module slot location).

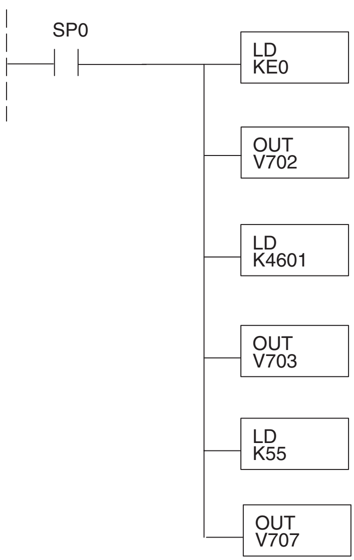
Selects station address 1, baud rate of 9600bps and odd parity.

Write configuration data into V701 (based on module slot location).

K55 is the port setup completion code that must be used to request that the CPU recognizes the new/desired port parameters

Write configuration data into V707 (based on module slot location).

Port 2 Example: Slave Mode



Selects auto–detect protocol mode (K–seq, D–Net and MODBUS), HEX transmit mode, RTS On/Off delay times are set to zero and RS–422/485 (4–wire) / RS–232

Write configuration data into V702 (based on module slot location).

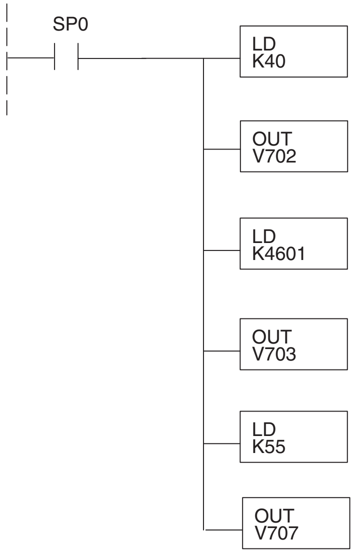
Selects station address 1, baud rate of 19200bps, odd parity, 8–bit data length and 1 stop bit.

Write configuration data into V703 (based on module slot location).

K55 is the port setup completion code that must be used to request that the CPU recognizes the new/desired port parameters

Write configuration data into V707 (based on module slot location).

Port 2 Example: *Direct*Net Master



Selects DirectNet protocol, HEX transmit mode, RTS On/Off delay times are set to zero and RS–422/485 (4–wire) / RS–232

Write configuration data into V702 (based on module slot location).

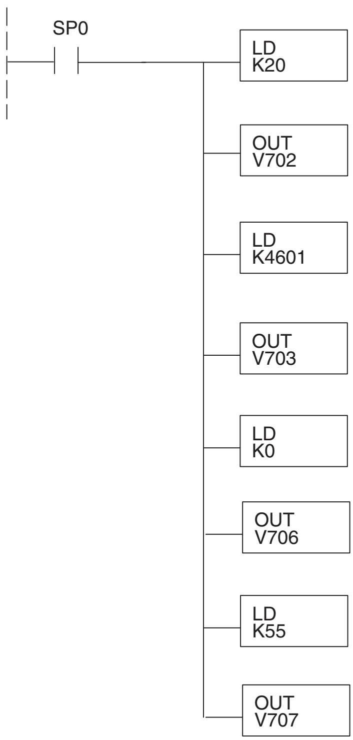
Selects station address 1, baud rate of 19200bps, odd parity, 8–bit data length and 1 stop bit.

Write configuration data into V703 (based on module slot location).

K55 is the port setup completion code that must be used to request that the CPU recognizes the new/desired port parameters

Write configuration data into V707 (based on module slot location).

Port 2 Example: Modbus RTU Master



Selects MODBUS protocol, HEX transmit mode, RTS On/Off delay times are set to zero and RS–422/485 (4–wire) / RS–232C

Write configuration data into V702 (based on module slot location).

Selects station address 1, baud rate of 19200bps, odd parity, 8–bit data length and 1 stop bit.

Write configuration data into V703 (based on module slot location).

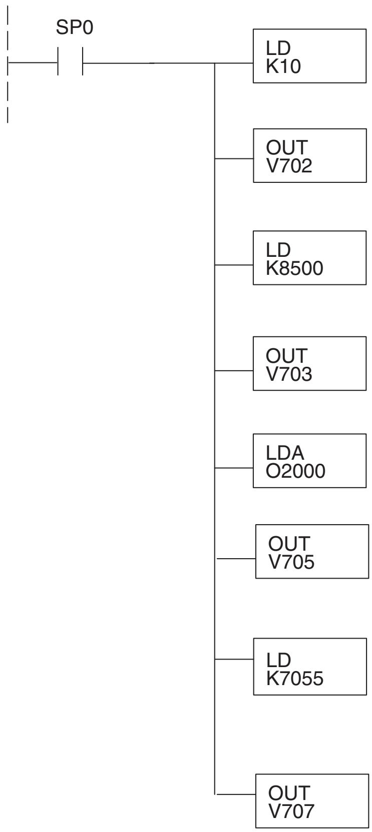
Selects 3.5 characters length time–out.

Write configuration data into V706 (based on module slot location).

K55 is the port setup completion code that must be used to request that the CPU recognizes the new/desired port parameters

Write configuration data into V707 (based on module slot location).

Port 2 Example: Non-Sequence Protocol



Selects non-sequence protocol, RTS On/Off delay times are set to zero and RS–422/485 (4–wire) / RS–232 is selected

Write configuration data into V702 (based on module slot location).

Selects baud rate of 9600bps, even parity, 8–bit data length and 1 stop bit.

Write configuration data into V703 (based on module slot location).

Selects V2000 as Memory Address, where ASCII data coming into the PLC is stored temporarily.

Write configuration data into V705 (based on module slot location).

Loading a value of 7055 requests that the DCM verify the requested port parameters, does not enable Xon/Xoff flow control and does not enable RTS flow control.

Write configuration data into V707 (based on module slot location).

DirectNet Communications Using RX/WX

Chapter 4

In This Chapter...

RX/WX Network Instructions

Read (RX) and Write (WX) Instructions

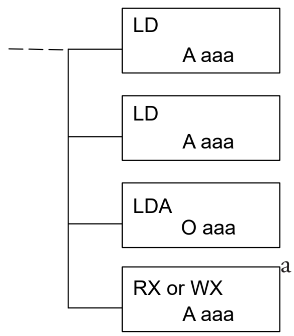
The Read (RX) and Write (WX) instructions are used by the initiating PLC to Read a block of data from another PLC or Write a block of data to another PLC. To perform their functions, the RX and WX boxes must be preceded in the ladder logic program by two Load instructions and one Load Address instruction.

The Load and Load Address instructions load communication parameters into the accumulator and the first and second level of the accumulator stack. The RX or WX instruction takes these parameters from the stack and the accumulator and prepares the data to be sent over the network. If you need to know more about the function of the accumulator and the accumulator stack, refer to the User Manual for your PLC.

***NOTE:** Please review intelligent instructions (IBOX) in Chapter 5 of the user manual for the PLC you are using, which simplify this and other functions.*
Consider the following IBOX instructions: ECOM100, ECRX and ECWX. Building the Read (RX) or Write (WX) Routine

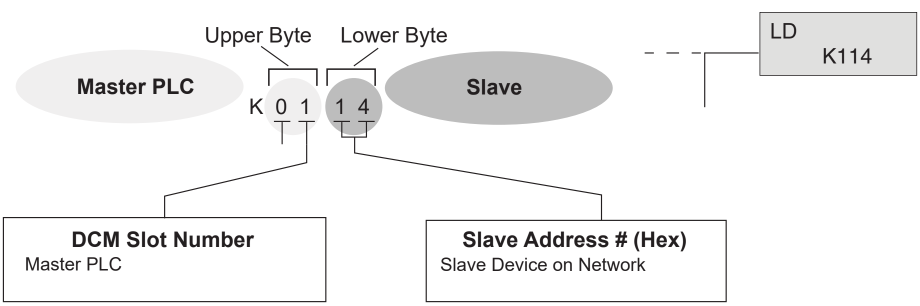
Building the Read (RX) and Write (WX) Routine

For network communications, you build the Read (RX) or Write (WX) instructions into a routine which requires the four instructions you see to the right. The function of each of these instructions is explained below or on the next page. They must be used in the sequence shown.



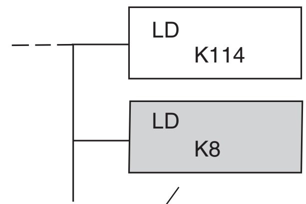
The First LD Instruction

The first Load (LD) instruction accepts either a constant or a variable. Use a "K" to designate the number as a constant. Use "V" if you are entering the address of a register. The contents of that register perform the same function as the constant shown below. For example, you could use V2000 in place of K0114. If the contents of V2000 is the number "114," the function would be the same. Using a variable allows changing parameters while the program is running. It is recommended, however, to use a constant when possible.

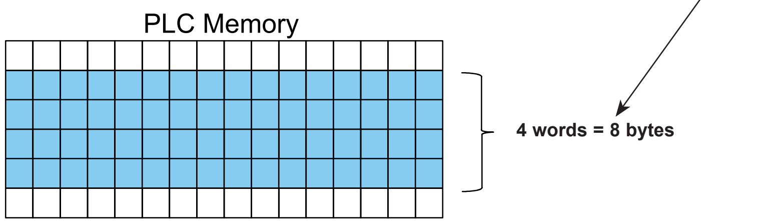


The Second LD Instruction

The second Load (LD) instruction determines the length of the data block to be transmitted during the Read or Write communication. This instruction will also accept two data types. Use a "K" to designate the number as a constant. Use a "V" if you are entering the address of a register.

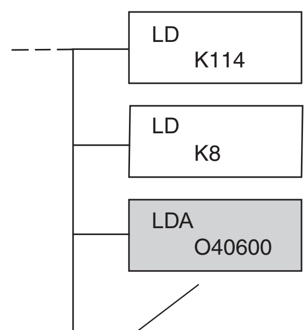


For Word Memory data, you must use a multiple of two bytes between 2 and 128. For Bit Memory data, you can use any multiple of one byte between 1 and 128. More information about addressing Word and Bit Memory is provided under "Addressing Different Memory Types.



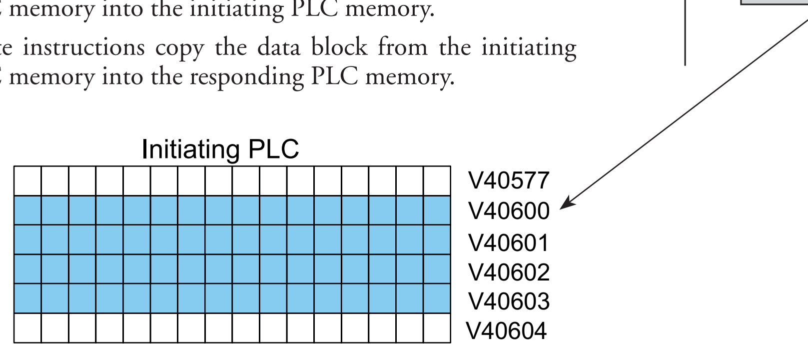
The LDA Instruction

The Load Address (LDA) instruction specifies the V-memory address of the beginning memory register in the initiating, or master, PLC. The data block to be transmitted will begin at this address and extend the number of bytes specified in the preceding LD instruction. The leading "O" indicates this is an octal number. Simply substitute the letter "O" for the "V" in the V-memory designation. For example, V40600 becomes O40600.



Read instructions copy the data block from the responding PLC memory into the initiating PLC memory.

Write instructions copy the data block from the initiating PLC memory into the responding PLC memory.

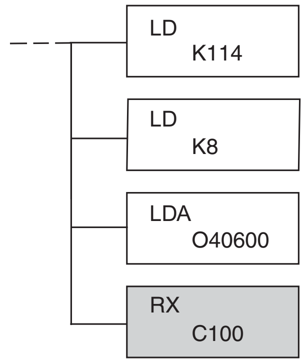


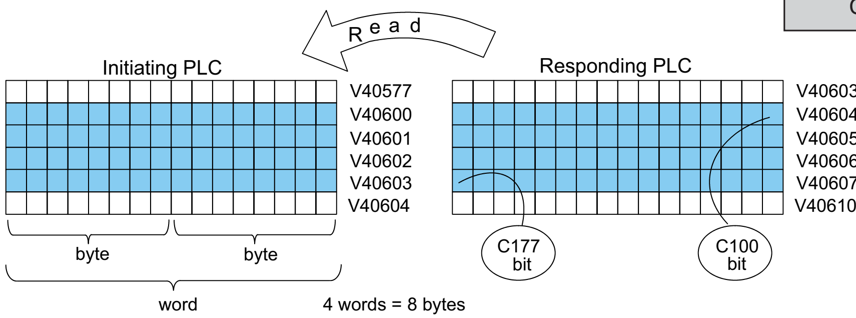
Read (WX) Instruction

The Read (RX) instruction specifies the memory location to be read from the slave PLC.

A block of data is read that begins at the specified memory location and extends the number of bytes specified in the second LD instruction.

In this example, the 8-byte block of data beginning at C100 and ending at C177 in the responding, or slave, PLC is read (copied) into the initiating PLC's memory beginning at V40600.



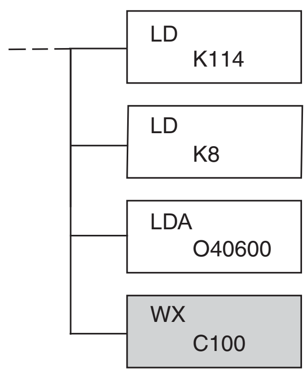


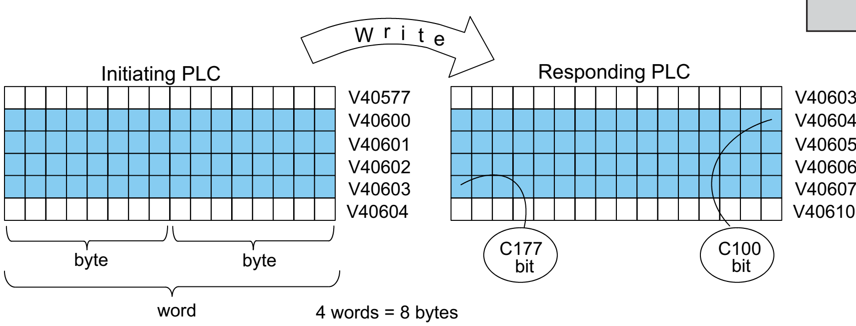
Write (WX) Instruction

The Write (WX) instruction specifies the memory location to be written to in the slave PLC.

A block of data is written that begins at the specified memory location and extends the number of bytes specified in the second LD instruction.

In the example, the 8-byte block of data beginning at V40600 and ending at V40603 in the initiating, or master, PLC is written (copied) into the responding PLC's memory beginning at C100 and ending at C177.





Addressing the Different Memory Types

Some data types are inherently 16 bits long, for example timer and counter current values. Other data types are 1 bit long, for example: discrete inputs and outputs. Word-length and bit-length data are mapped into Word Memory, also known as V-memory, which allows you to address any of the different memory types as 16-bit words.

Bit Memory

Bit memory can be addressed in Read and Write instructions by the name of the first bit of any byte. If your second LD instruction contains the constant K8, eight bytes will be transmitted. If you use C0 in your RX or WX instruction, you will transmit the eight bytes from C0 through C77.

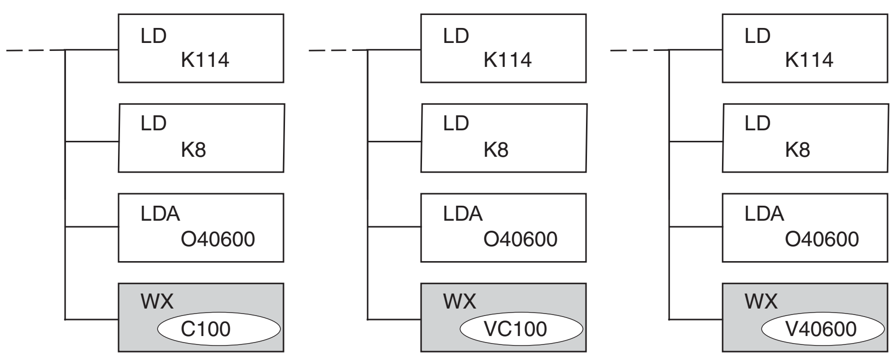
Word Memory and Aliases

In the example below, V40600 is the V-memory designation for the sixteen bits from C0 through C17. Aliases are a convenient substitute for V-memory designations, and can be used interchangeably in Read and Write instructions. VC0 is the alias for V40600. Either nomenclature addresses the same 16 bits.

The alias is simply the name of the first bit in a group of sixteen bits, with V added as a prefix. For example, VC0 represents the 16 bits beginning with C0. Word Memory, Bit Memory and Aliases all use the octal numbering system.

Word Memory Address V40600 = VC0 (Alias)	C17	C16	C15	C14	C13	C12	C11	C10	C7	C6	C5	C4	C3	C2	C1	C0
V40601 = VC20	C37	C36	C35	C34	C33	C32	C31	C30	C27	C26	C25	C24	C23	C22	C21	C20
V40602 = VC40	C57	C56	C55	C54	C53	C52	C51	C50	C47	C46	C45	C44	C43	C42	C41	C40
V40603 = VC60	C77	C76	C75	C74	C73	C72	C71	C70	C67	C66	C65	C64	C63	C62	C61	C60

The following Write routines are all equivalent. *Direct*SOFT gives you the flexibility to identify the responding PLC's memory area in three different ways, as shown below.



Available Data Types

You can address the different data types by any available convention shown in the tables that follow. The largest block of data that can be sent in a single Read or Write operation is 128 bytes. The smallest block of data is one byte for Bit Memory types and two bytes, or one word for Word Memory types. The octal numbering system is used for all addresses in these tables.

DL05 CPU

DL05 CPU			
Data Type	**Bit Memory**	**Word Memory**	**Alias**
Timer Current Values	None	V0 – V177	TA0 – TA177
Counter Current Values		V1000 – V1177	CTA0 – CTA177
User Data Words		V1200 – V7377	None
Input Points (See Note 1)	X0 – X377	V40400 – V40417	VX0 – VX360
Output Points (See Note 1)	Y0 – Y377	V40500 – V40517	VY0 – VY360
Control Relays	C0 – C777	V40600 – V40677	VC0 – VC760
Special Relays	SP0 – SP777	V41200 – V41237	VSP0 – VSP760
Timer Status Bits	T0 – T177	V41100 – V41107	VT0 – VT160
Counter Status Bits	CT0 – CT177	V41140 – V41147	VCT0 – VCT160
Stages	S0 – S377	V41000 – V41017	VS0 – VS360

1 - The DL05 systems are limited to 8 discrete inputs and 6 discrete outputs with the present available hardware, but 256 point addresses exist.

DL06 CPU

DL06 CPU			
Data Type	**Bit Memory**	**Word Memory**	**Alias**
Timer Current Values	None	V0 – V377	TA0 – TA377
Counter Current Values		V1000 – V1177	CTA0 – CTA177
User Data Words		V400 – V677 V1200 – V7377 V10000–V17777	None
Input Points (See Note 1)	X0 – X777	V40400 – V40437	VX0 – VX760
Output Points (See Note 1)	Y0 – Y777	V40500 – V40537	VY0 – VY760
Control Relays	C0 – C1777	V40600 – V40677	VC0 – VC1760
Special Relays	SP0 – SP777	V41200 – V41237	VSP0 – VSP760
Timer Status Bits	T0 – T377	V41100 – V41117	VT0 – VT160
Counter Status Bits	CT0 – CT177	V41140 – V41147	VCT0 – VCT160
Stages	S0 – S1777	V41000 – V41077	VS0 – VS1760
Remote I/O	GX0 – GX3777 GY0 – GY3777	V40000 – V40177 V40200 – V40377	VGX0 – VGX3760 VGY0 – VGY3760

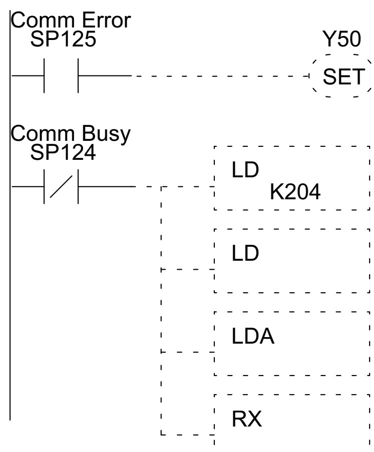
1 - The DL06 systems are limited to 20 discrete inputs and 16 discrete outputs with the present available hardware, but 512 point addresses exist.

Special Relays for Communications

The ***Direct***LOGIC PLCs provide internal contacts (bits) for monitoring the status of communications. The internal contacts are called Special Relays (there are other Special Relays used for other purposes). There are two Special Relays for each slot in the base that will accept the ECOM module. The two relays perform the following functions:

- Communication Busy – This bit is on when the communication module is busy transmitting or receiving. You must use this bit, or relay contact, to prevent overwriting your Read or Write (RX/WX) instructions.
- Communication Error – This bit is on when an error occurred in the last RX or WX communication. This error automatically clears (the bit resets to zero) when another RX or WX instruction executes.

For example, Special Relays SP124 and SP125 correspond to an ECOM module in slot 3 of the PLC base.



The Special Relay SP125 is used in the example to energize the output Y50, indicating a communication error has occurred. This Special Relay must appear earlier in the program than your RX or WX instruction because it is turned off (reset to zero) when a subsequent Read or Write instruction is executed.

The Special Relay SP124 indicates the ECOM is busy. When SP124 is on, the normally closed contact opens to prevent executing another RX or WX instruction until the last one is completed. The appropriate busy bit must be used as a NC contact on every RX/WX instruction rung in the program.

The Special Relays for the DL05 and DL06 are listed in the tables below.

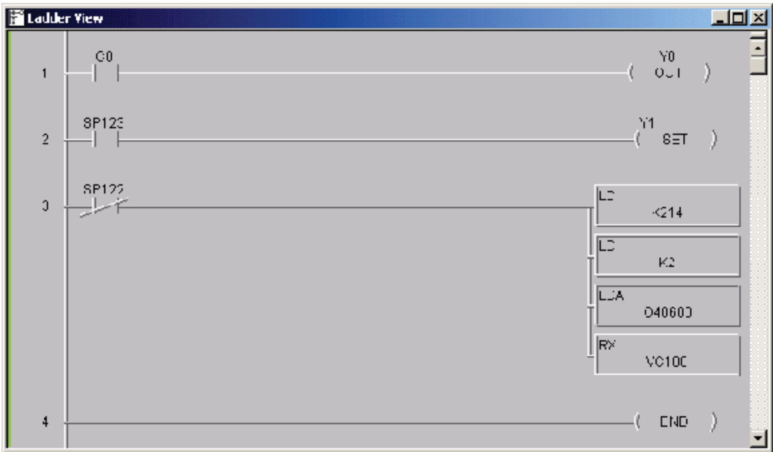
DL05 Special Purpose Communication Relays	
CPU Base	**Option Slot**
Communication Busy	SP120
Communication Error	SP121

DL06 Special Purpose Communication Relays				
CPU Base	**Slot 1**	**Slot 2**	**Slot 3**	**Slot 4**
Communication Busy	SP120	SP122	SP124	SP126
Communication Error	SP121	SP123	SP125	SP127

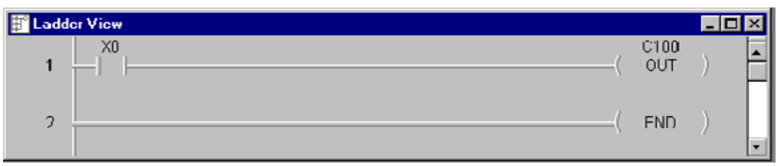
Program with One RX Instruction

The Ladder View screen below is the program development screen in ***Direct***SOFT Programming Software. This four-rung program is explained in detail on the following page. This is a complete program although its function is very limited. There is also a two-rung program that runs in the slave PLC, and it is also explained on the page following. This example assumes the DCM is in slot 2 of a DL06 PLC.

Program for the Master PLC



Program for the Slave PLC



When the input (X0) to the slave PLC is turned on (transitions from 0 to 1), the C0 bit in the master PLC also transitions from 0 to 1. The program in the master PLC causes Y0 to turn on in response to the C0 bit.

NOTE: *The slave PLC logic is a basic example only. If the master/slave communication fails, the bits written to the slaves from the master will remain in the same state last written from the master.*

For example DL05/06 communications programs, go to **www.automationdirect.com** technical support website > Example programs> Communications> **example # EP–COM–005**.

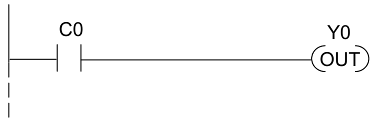
Master example: This project contains simple logic for reading the inputs from a DL05/06 slave and placing their status in C0–C17 in the master. It also writes C17–C37 to the outputs on the slave.

Slave example: This project can be used in conjunction with the master project to setup the slave to turn off its outputs if the master PLC stops communicating with it.

Program for the Master PLC

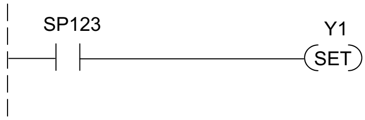
Rung 1

In our example, the normally open contact labeled C0 is an internal control relay. When C0 is on, discrete output Y0 is energized.



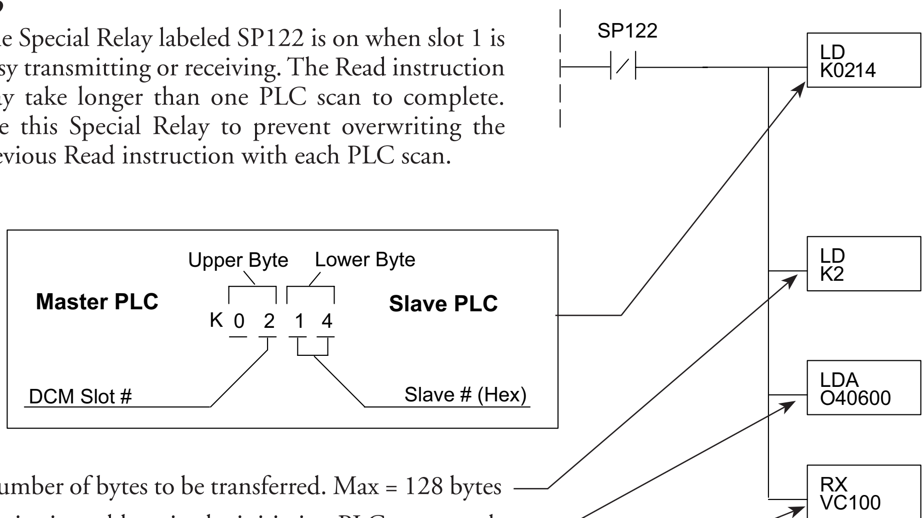
Rung 2

The second rung uses a Special Relay to identify a communication error. In the example, SP123 is on if a communication error is present for slot one. Use different Special Relays if your DCM module is in a different slot (see page 4-7). We use SP123 to turn on an indicator light connected to a discrete output.



Rung 3

The Special Relay labeled SP122 is on when slot 1 is busy transmitting or receiving. The Read instruction may take longer than one PLC scan to complete. Use this Special Relay to prevent overwriting the previous Read instruction with each PLC scan.



Number of bytes to be transferred. Max = 128 bytes

Beginning address in the initiating PLC, expressed as an octal number.

Beginning address in the responding PLC.

Rung 4

All ***Direct***LOGIC PLCs use an END statement to identify the final rung of the main body of the program.

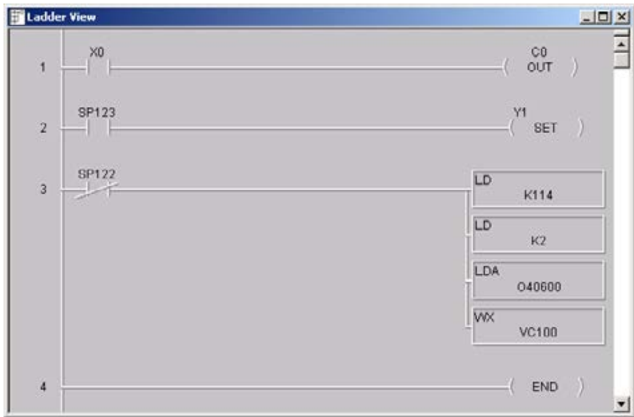
Program for the Slave PLC

This two-rung program resides in the responding PLC's CPU. Its function is simply to use the X0 contact to turn on the internal control relay, C100.

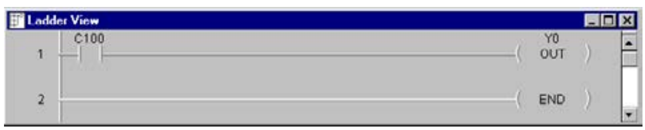
Example Program with One WX Instruction

The Ladder View screen below is the program development screen in ***Direct***SOFT Programming Software. This four-rung program is explained in detail on the page following. This is a complete program although its function is very limited. There is also a two-rung program that runs in the responding PLC. It is also explained on the page following. This example assumes the DCM is in slot 2 of a DL06 PLC.

Program for the Master PLC



Program for the Slave PLC



When the input to the master PLC is turned on (transitions from 0 to 1), the C100 bit in the slave PLC also transitions from 0 to 1. The program in the slave PLC causes Y0 to turn on in response to the C100 bit.

NOTE: *The slave PLC logic is a basic example only. If the master/slave communication fails, the bits written to the slaves from the master will remain in the same state last written from the master.*

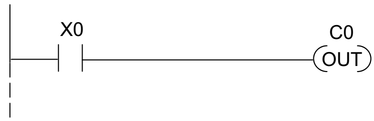
For example DL05/06 communications programs, go to **www.automationdirect.com** technical support website > Example programs> Communications> example # **EP–COM–005**.

Master example: This project contains simple logic for reading the inputs from a DL05/06 slave and placing their status in C0–C17 in the master. It also writes C17–C37 to the outputs on the slave.

Slave example: This project can be used in conjunction with the master project to setup the slave to turn off its outputs if the master PLC stops communicating with it.

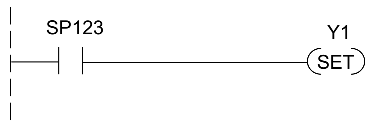
Rung 1

In our example, the normally open contact labeled X0 is a toggle switch input to a discrete input module. When X0 is on, Control Relay C0 is energized.



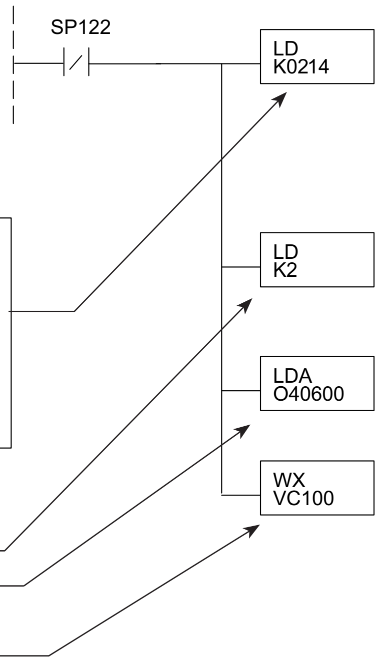
Rung 2

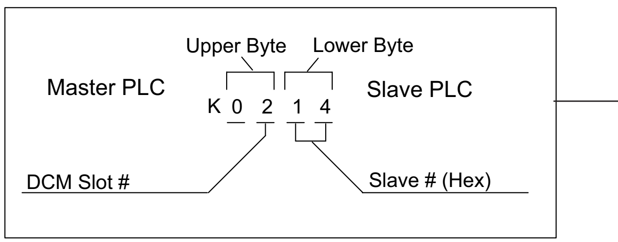
The second rung uses a Special Relay to identify a communication error. In the example, SP123 is on if a communication error is present for slot one. Use different Special Relays if your ECOM module is in a different slot (see page 4-11 and 4-12). We use SP123 to turn on an indicator light connected to a discrete output.



Rung 3

The Special Relay labeled SP122 is on when slot 1 is busy transmitting or receiving. The Write instruction may take longer than one PLC scan to complete. Use this Special Relay to prevent overwriting the previous Write instruction with each PLC scan.





Number of bytes to be transferred. Max = 128 bytes

Beginning address in the initiating PLC, expressed as an octal number.

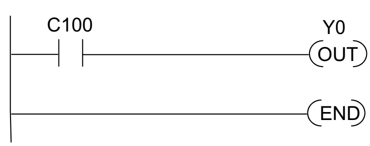
Beginning address in the responding PLC.

Rung 4

All ***Direct***LOGIC PLCs use an END statement to identify the final rung of the main body of the program.

Program for the Responding PLC

This two-rung program resides in the responding PLC's CPU. Its function is simply to take the C100 contact and convert it to a real output, Y0.



Integrating Multiple RX and WX Instructions

Multiple Read and Write instructions require interlocks for sequencing because only one RX/WX instruction can be processed per CPU scan. Using interlocks, one RX/WX instruction is processed in each scan until all RX/WX instructions have been executed. After the last instruction, the sequence then begins again at the first RX/WX instruction.

Without interlocks, the RX/WX instructions would be executed in an unpredictable order, and some might be executed many times before others are executed once. The interlocks serve to open (disconnect) the ladder circuits for all Read and Write instructions except the one that should be processed on the current CPU scan.

We show two methods of creating the interlocks necessary for sequencing multiple Read and Write instructions:

- Sequenced Internal Control Relays
- Shift Register

We will step you through the development of the interlocks using both methods. The two examples shown perform the same function. Only the interlocks are different.

***Note:** To fully understand the material in this section, you will first need to understand the Example Programs on pages 4–8 and 4–10, as well as the material in the Network Instructions section, beginning on page 4–2.*

The following program segment sequences through three RX/WX instructions (two Write instructions and one Read instruction). You can develop your own program incorporating either of the two interlocking control strategies and expanding the number of interlocks to accommodate the number of RX/WX instructions in your program.

Interlocking Relays

It is easy to see the function of the interlocking relays if we construct a truth table first.

Across the top of the truth table we show internal control relays that we are considering using for our sequencing strategy. We have used C50 through C52 for our chart, but any contacts that are not used for other purposes in your program will work just as well.

Down the left side of the chart, we list the number of RX/WX instructions we may want to use in our RLL program.

Truth Table	C52	C51	C50
First RX/WX	0	0	0
Second RX/WX	0	0	1
Third RX/WX	0	1	0
Fourth RX/WX	0	1	1
Fifth RX/WX	1	0	0
Sixth RX/WX	1	0	1
Seventh RX/WX	1	1	0
Eighth RX/WX	1	1	1

The three contacts in this truth table will accommodate as many as eight Read or Write instructions. Our program only has three RX/WX instructions so we only need to use two contacts (see why on page 4-13). We will use C50 and C51. One additional contact (C53) would give us 32 combinations since the number of combinations expands as the power of 2.

Our three RX/WX instructions can be sequenced by the two contacts C50 and C51. Two contacts provide four different binary states:

- Both off
- C50 on and C51 off
- C50 off and C51 on
- Both on

We only need to use three of the four binary states (circled) since we only have three RX/WX instructions to sequence.

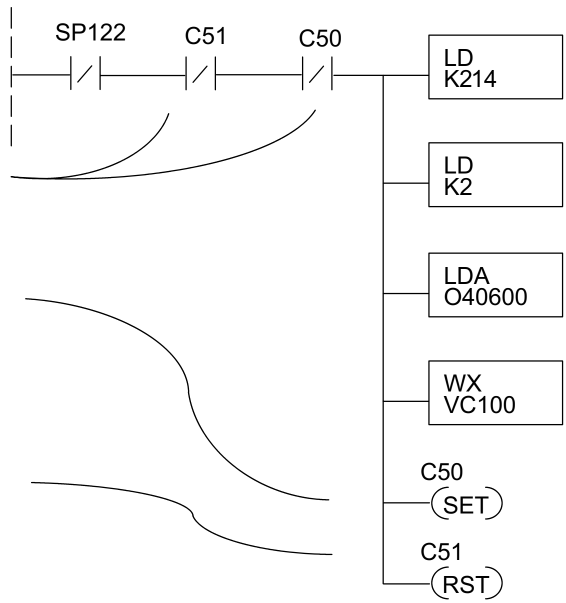
Truth Table	C52	C51	C50
First RX/WX	0	0	0
Second RX/WX	0	0	1
Third RX/WX	0	1	0
Fourth RX/WX	0	1	1
Fifth RX/WX	1	0	0
Sixth RX/WX	1	0	1
Seventh RX/WX	1	1	0
Eighth RX/WX	1	1	1

First RX/WX Instruction

C50 and C51 are interlocking contacts. They are normally closed in this rung to permit power flow to the first WX instruction. Both bits are off, corresponding to the first row of the truth table.

After the WX instruction is executed C50 is SET (turned on) which opens the contact in this rung and closes the C50 contact in the next rung.

C51 is RESET (turned off) which leaves the C51 contact closed for the next rung.

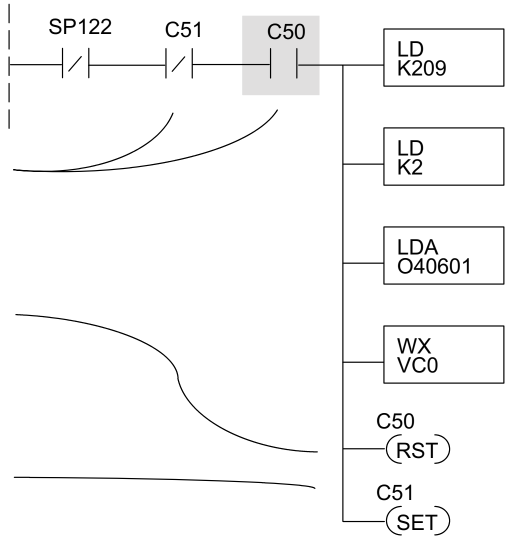


Second RX/WX Instruction

C50 is normally open and C51 is normally closed. For this rung to be executed, the C50 bit must be on and the C51 bit must be off, corresponding to the second row of the truth table. C50 was turned on in the previous rung. C51 was turned off in the previous rung.

After the WX instruction is executed C50 is RESET (turned off) which opens the C50 contact in this rung and closes it in the next rung.

C51 is SET (turned on), which closes the normally open C51 contact in the next rung.

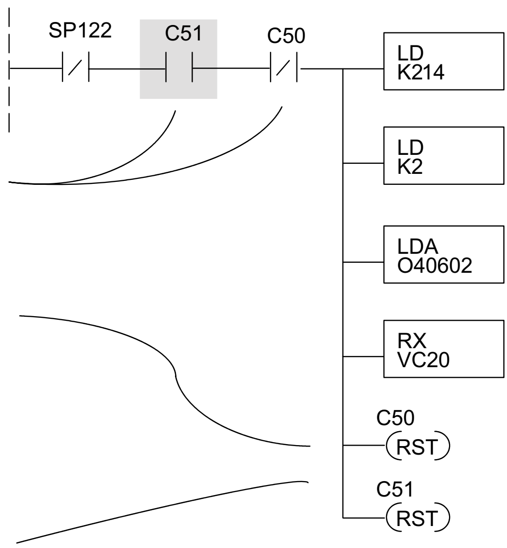


Third RX/WX Instruction

In this last rung, C50 is normally closed and C51 is normally open. For this rung to be executed, the C50 bit must be off and the C51 bit must be on, corresponding to the third row of the truth table. C51 was turned on in the previous rung.

After the RX instruction is executed, C50 is RESET which opens the C50 contact in this rung and allows it to close in preparation for repeating the first communication rung on the next CPU scan.

C51 is also RESET, which allows the C51 contact to close in preparation for repeating the first communication rung on the next CPU scan.



Returning to the First RX/WX Instruction

At the end of the third RX/WX instruction, we cycle back to the top row of the truth table on page 4-13. Both C50 and C51 are off, and the next CPU scan executes the first RX/WX instruction.

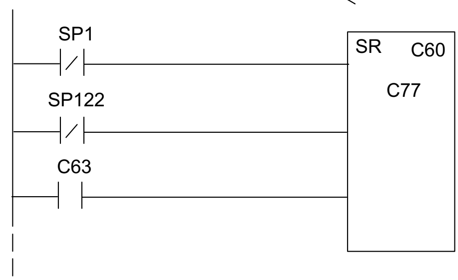
Shift Register

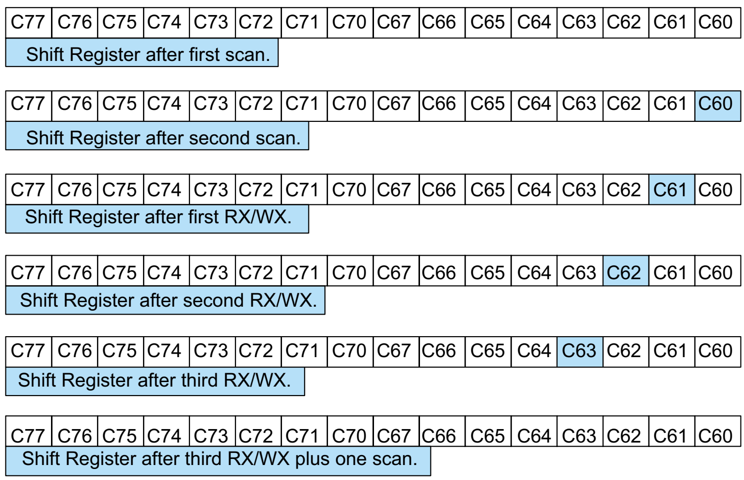
The Shift Register can be used for creating interlocks, as an alternative to using control relays. For a complete explanation of the function of the Shift Register, see the User Manual for your PLC. If you have more than a few RX/WX instructions, using control relays can become cumbersome. The Shift Register allows a single contact to be used in each communication rung as an interlock.

The data input to the Shift Register (SR) is Special Relay SP1. SP1 is the always-on bit. Combined with a normally closed contact it sends zeros to the Shift Register data input.

The clock input to the Shift Register is SP122, the communication busy bit. Each time one of the RX/WX instructions executes, the Shift Register moves the set bit over one place.

C63 is used in this example to reset the Shift Register to all zeros.



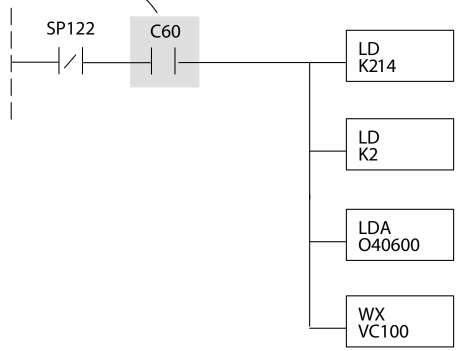


Store If Equal

The Store If Equal instruction detects when the Shift Register is reset to zeros. When that condition is true the C60 bit is SET by this rung. The C60 bit becomes the high bit shifted by the Shift Register until each RX/WX instruction is executed in turn.

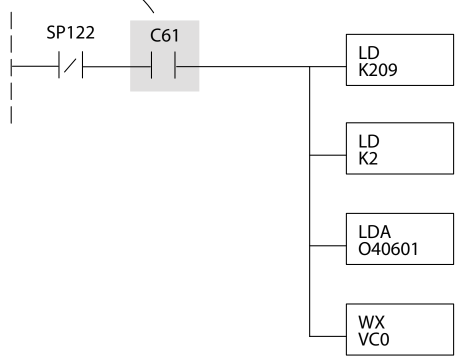
First RX/WX Instruction

C60 is the interlocking contact. It is turned on by the Store If Equal rung preceding this one.



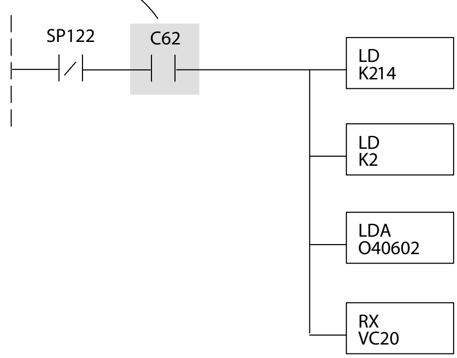
Second RX/WX Instruction

C61 is the interlocking contact. It is turned on by the sequencing steps of the Shift Register in a preceding rung.



Third RX/WX Instruction

C62 is the interlocking contact. It is turned on by the sequencing steps of the Shift Register in a preceding rung.



After this rung is executed, the Shift Register shifts the high bit from C62 to C63 on the next CPU scan. C63 resets the Shift Register to zeros, the Store If Equal sets the C60 bit, and the CPU executes the first RX/WX instruction.

Chapter 5

Modbus *RTU* Communications RX/WX and MRX/MWX

In This Chapter...

Network Slave Operation

This section describes how a Modbus RTU master on a network can communicate with a D0–DCM using the Modbus RTU protocol. A network master must send a Modbus function code and Modbus address to specify a PLC memory location in the DL05/06 CPU. No CPU ladder logic is required to support Modbus slave operation.

Modbus Function Codes Supported

The D0-DCM supports the following Modbus function codes when operating as a Modbus slave.

Modbus Function Code	Function	DL05/06/205/405 Data Types Available
01	Read Discrete Output Table	Y, C, T, CT
02	Read Discrete Input Table	X, SP
03	Read Holding registers (when addressing mode is 584/984, this function is used to access analog output registers)	V
04	Read Input Registers (when addressing mode is 584/984, this function is used to access analog input registers)	V
05	Force Single Discrete Output	Y, C, T, CT
06	Preset Single Holding Registers	V
08	Loop Back / Maintenance	
15	Force Multiple Discrete Outputs	Y, C, T, CT
16	Preset Multiple Holding Registers	V

Determining the Modbus Address

There are typically two ways that most Modbus addressing conventions allow you to specify a PLC memory location. These are:

- By specifying the Modbus data type and address
- By specifying a Modbus address only.

If Your Host Software or Master Requires the Data Type and Address

Many Modbus masters allow you to specify the Modbus data type and the Modbus address that corresponds to the PLC memory location. This is the easiest method, but not all packages allow you to do it this way.

The actual equation used to calculate the address depends on the type of PLC data you are using. The PLC memory types are split into two categories for this purpose.

- Discrete – X, SP, Y, C, S, T(contacts), CT (contacts)
- Word – V-memory, Timer current value, Counter current value

In either case, you basically convert the PLC octal address to decimal and add the appropriate Modbus starting address (as required). The following tables show the exact range used for each group of data.

NOTE: *For an automated Modbus/Koyo address conversion utility, download the file Modbus conversion.xls from the* **www.automationdirect.com** *technical support > Technical and Application notes > PLC hardware> Communications> app note #* **AN–MISC–010**.

DL05 Memory Type	Qty (Dec.)	PLC Range (Octal)	Modbus Address Range	Modbus Data Type
For Discrete Data Types…	**Convert PLC Addr. to Dec.**		**+ Start of Range**	**+ Data Type**
Inputs (X)	256	X0 – X377	2048 – 2303	Input
Special Relays (SP)	512	SP0 – SP777	3072 – 3583	Input
Outputs (Y)	256	Y0 – Y377	2048 – 2303	Coil
Control Relays (C)	512	C0 –C777	3072 – 3583	Coil
Timer Contacts (T)	128	T0 – T177	6144 – 6271	Coil
Counter Contacts (CT)	128	CT0 – CT177	6400 – 6527	Coil
Stage Status Bits (S)	256	S0 – S377	5120 – 5375	Coil
For Word Data Types …	**Convert PLC Addr. to Dec.**		**+**	**Data Type**
Timer Current Values (V)	128	V0 – V177	0 – 127	Input Register
Counter Current Values (V)	128	V1000 – V1177	512 – 639	Input Register
V-Memory, user data (V)	3072	V1400 – V7377	768 – 3839	Holding Register

DL06 Memory Type	Qty (Dec.)	PLC Range (Octal)	Modbus Address Range	Modbus Data Type
For Discrete Data Types...	Convert PLC Addr. to Dec.		+ Start of Range	+ Data Type
Inputs (X)	512	X0 – X777	2048 – 2559	Input
Special Relays (SP)	512	SP0 – SP777	3072 – 3583	Input
Outputs (Y)	512	Y0 – Y777	2048 – 2559	Coil
Control Relays (C)	1024	C0 –C1777	3072 – 4095	Coil
Timer Contacts (T)	256	T0 – T377	6144 – 6399	Coil
Counter Contacts (CT)	128	CT0 – CT177	6400 – 6527	Coil
Stage Status Bits (S)	1024	S0 – S1777	5120 – 6143	Coil
Global Inputs (GX)	2048	GX0 – GX3777	0 – 2047	Input
Global Outputs (GY)	2048	GY0 – GY3777	0 – 2047	Coil
For Word Data Types ...	Convert PLC Addr. to Dec.		+	Data Type
Timer Current Values (V)	128	V0 – V177	0 – 127	Input Register
Counter Current Values (V)	128	V1000 – V1177	512 – 639	Input Register
V-Memory, user data (V)	256 3072 4096	V400 – V677 V1400 – V7377 V10000 – V17777	256 – 511 768 – 3839 4096 – 8191	Holding Register

The following examples show how to generate the Modbus address and data type for hosts which require this format.

Example 1: V2100

Find the Modbus address for User V location V2100.

1. Find V- Memory in the table.
2. Convert V2100 into decimal (1088).
3. Use the Modbus data type from the table.

PLC Addr. (Dec.) + Data Type

V2100 = 1088 decimal

1088 + Hold. Reg. = Holding Reg.

Timer Current Values (V)	128	V0 – V177	0 – 127	Input Register
Counter Current Values (V)	128	V1000 – V1177	512 – 639	Input Register
V-Memory, user data (V)	1024	V2000 – V3777	1024 – 2047	Holding Register

Example 2: Y20

Find the Modbus address for output Y20.

1. Find Y outputs in the table.
2. Convert Y20 into decimal (16).
3. Add the starting address for the range (2049).
4. Use the Modbus data type from the table.

PLC Addr.(Dec.) + Start Addr. + Data Type

Y20 = 16 decimal

16 + 2049 + Coil = Coil 2065

Outputs (Y)	320	Y0 – Y477	2049 – 2367	Coil
Control Relays (C)	256	C0 –C377	3072 – 3551	Coil

Example 3: T10 Current Value

Find the Modbus address to obtain the current value from Timer T10.

1. Find Timer Current Values in the table.
2. Convert T10 into decimal (8).
3. Use the Modbus data type from the table.

PLC Addr.(Dec.) + Data Type

T10 = 8 decimal

8 + Input Reg. = Input Reg. 8

Timer Current Values (V)	128	V0 – V177	0 – 127	Input Register
Counter Current Values (V)	128	V1000 – V1177	512 – 639	Input Register

Example 4: C54

Find the Modbus address for Control Relay C54

1. Find Control Relays in the table.
2. Convert C54 into decimal (44).
3. Add the starting address for the range (3072).
4. Use the Modbus data type from the table.

PLC Addr.(Dec.) + Start Addr. + Data

C54= 44 decimal

44 + 3072 + Coil = Coil 3117

Outputs (Y)	320	Y0 – Y477	2049 – 2367	Coil
Control Relays (CR)	256	C0 –C377	3072 – 3551	Coil

If the Host Software or Client Requires an Address ONLY

Some Modbus TCP clients do not allow you to specify the Modbus data type and address. Instead, you specify an address only. This method requires another step to determine the address, but it is not difficult. Basically, Modbus also separates the data types by address ranges as well. This means an address alone can actually describe the type of data and location. This is often referred to as "adding the offset".

The actual equation used to calculate the address depends on the type of PLC data you are using. The PLC memory types are split into two categories for this purpose.

- Discrete – X, GX, SP, Y, CR, S, T, C (contacts)
- Word – V-memory , Timer current value, Counter current value

In either case, you basically convert the PLC octal address to decimal and add the appropriate Modbus starting address (as required). The following tables show the exact range used for each group of data.

NOTE: *For an automated Modbus/Koyo address conversion utility, download the file Modbus_conversion.xls from www.automationdirect.com technical support > Technical and Application Notes > PLC hardware > Communications > app note # AN-MISC-010.*

DL05 Discrete Data Types

PLC Memory Type	Qty (Dec.)	PLC Range (Octal)	Modbus Address Range	Access
Inputs (X)	256	X0 – X377	12049 – 123047	Read Only
Special Relays (SP)	512	SP0 – SP777	13073 – 13584	
Reserved	–	–	13585 – 20000	
Outputs (Y)	256	Y0 – Y377	2049 – 2304	Read/Write
Control Relays (CR)	512	C0 –C777	3073 – 3584	
Timer Contacts (T)	128	T0 – T377	6145 – 6272	
Counter Contacts (CT)	128	CT0 – CT177	6401 – 6528	
Stage Status Bits (S)	256	S0 – S377	5121 – 5376	
Reserved	—	—	6529 – 10000	

DL06 Discrete Data Types				
PLC Memory Type	Qty (Dec.)	PLC Range (Octal)	Modbus Address Range	Access
Global Inputs (GX)	2048	GX0 – GX1746 GX1747 – GX3777	10001 – 10999 11000 – 12048	Read Only
Inputs (X)	512	X0 – X777	12049 – 12560	
Special Relays (SP)	512	SP0 – SP777	13073 – 13584	
Reserved	–	–	13585 – 20000	
Global Outputs (GY)	2048	GY0 – GY3777	0 – 2048	Read/Write
Outputs (Y)	512	Y0 – Y777	2049 – 2560	
Control Relays (CR)	1028	C0 –C1777	3073 – 4096	
Timer Contacts (T)	256	T0 – T377	6145 – 6400	
Counter Contacts (CT)	128	CT0 – CT177	6401 – 6528	
Stage Status Bits (S)	1024	S0 – S1777	5121 – 6144	
Reserved	—	—	6529 – 10000	

DL06 Word Data					
Registers (Word)	Qty (Dec.)	PLC Range (Octal)	Modbus 40001 Address Range	Modbus 30001 Address Range	Access
V-Memory Timers	256	V0 – V377	40001 – 40256	30001 – 30256	Read/Write
V-Memory Counters	128	V1000 – V1177	40513 – 40460	30513 – 30640	
V-Memory Data Words	256	V400 – V777	40257 – 40512	30257 – 30512	
	3072	V1400 – V7377	40769 – 43840	30769 – 33840	
	4096	V10000 – V17777	44097 – 48192	34097 – 38192	

The following examples show how to generate the Modbus address and data type for hosts which require this format.

Example 1: V2100

Find the Modbus address for User V location V2100.

1. Find V-memory in the table.
2. Convert V2100 into decimal (1088).
3. Add the Modbus starting address for the mode (40001).

PLC Addr. (Dec.) + Mode Address

V2100 = 1088 decimal

1088 + 40001 = 41089

For Word Data Types ...	PLC Address Dec.		+	Appropriate Mode Address		
Timer Current Values (V)	128	V0 – V177	0 – 127	3001	30001	Input Reg.
Counter Current Values (V)	128	V1000 – V1177	512 – 639	3001	30001	Input Reg.
V-Memory, user data (V)	1024	V2000 – V3777	1024 – 2047	4001	40001	Holding Reg.

Example 2: Y20

Find the Modbus address for output Y20.

1. Find Y outputs in the table.
2. Convert Y20 into decimal (16).
3. Add the starting address for the range (2048).
4. Add the Modbus address for the mode (1).

PLC Addr.(Dec.) + Start Address + Mode

Y20 = 16 decimal

16 + 2048 + 1 = 2065

Outputs (Y)	320	Y0 – Y477	2048 – 2367	1	1	Coil
Control Relays (CR)	256	C0 –C377	3072 – 3551	1	1	Coil
Timer Contacts (T)	128	T0 –T177	6144 – 6271	1	1	Coil

Example 3: C54

Find the Modbus address for Control Relay C54.

1. Find Control Relays in the table.
2. Convert C54 into decimal (44).
3. Add the starting address for the range (3072).
4. Add the Modbus address for the mode (1).

PLC Addr.(Dec.) + Start Address + Mode

C54 = 44 decimal

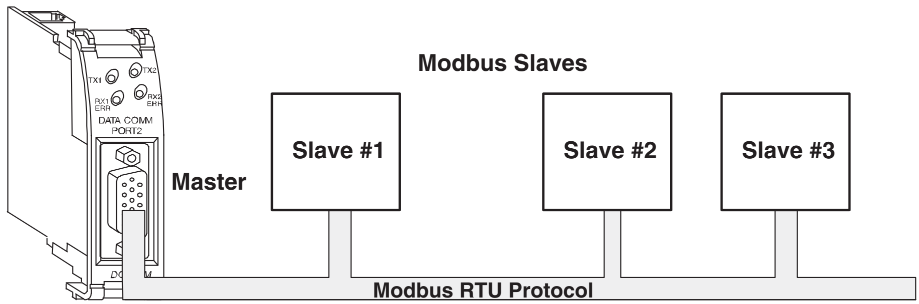
44 + 3072 + 1 = 3117

Outputs (Y)	320	Y0 – Y477	2048 – 2367	1	1	Coil
Control Relays (CR)	256	C0 –C377	3072 – 3551	1	1	Coil
Timer Contacts (T)	128	T0 –T177	6144 – 6271	1	1	Coil

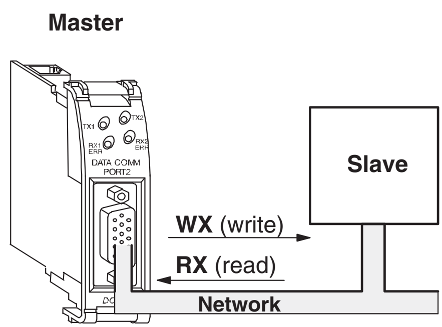
Network Master Operation: RX/WX Instructions

Overview

This section describes how the DL05/06CPU can operate as a master on a Modbus RTU network using the DCM. This section discusses how to design the required ladder logic for network master operation.



When using the DCM as a master on the network, you use simple RLL instructions to initiate the requests. The WX instruction initiates network write operations, and the RX instruction initiates network read operations. Before executing either the WX or RX commands, we need to load data related to the read or write operation onto the CPU's accumulator stack. When the WX or RX instruction executes, it uses the information on the stack combined with data in the instruction box to completely define the task.



Modbus Function Codes Supported

The DCM supports the following Modbus function codes when acting as a Modbus RTU master.

Modbus Function Code	Function	DL05/06 Data Types Available
01	Read Output Table	Y, C, T, CT
02	Read Input Table	X, SP
03	Read holding registers (when addressing mode is 584/984, this function is used to access analog output registers)	V
04	Read Input Registers (when addressing mode is 584/984, this function is used to access analog input registers)	V
15	Force Multiple Outputs	Y, C, T, CT
16	Preset Multiple Registers	V

NOTE: *The D0–DCM, as a master, does not support function code 4. Therefore, 30001 address ranges cannot be read from a slave device.*

PLC Memory Supported for Client Operation

The actual equation used to calculate the address depends on the type of PLC data you are using. The PLC memory types are split into three categories for this purpose.

- Discrete Inputs - GX, X, SP
- Discrete Outputs - Y, C, S, T, CT
- Word - Timer current value, Counter current value, Data Words

In either case, you basically take the Modbus address you are trying to target, subtract the starting Modbus of that range, convert the result to octal and add the octal number to the beginning PLC address in the appropriate PLC range. See the conversion examples on the following page. The following tables show the exact range used for each group of data.

NOTE: *For an automated* Modbus*/Koyo address conversion utility, download the file* ***modbus_conversion.xls*** *from* ***www.automationdirect.com; technical support****>Technical and Application Notes > PLC hardware > Communications > app note # AN-MISC-010.*

DL06 Discrete Data Types

PLC Memory Type	Qty (Dec.)	PLC Range (Octal)	Modbus Address Range	Access
Global Inputs (GX)	2048	GX0 – GX1746 GX1747 – GX3777	10001 – 10999 11000 – 12048	Read Only
Inputs (X)	512	X0 – X777	12049 – 12560	
Special Relays (SP)	512	SP0 – SP777	13073 – 13584	
Reserved	–	–	13585 – 20000	
Global Outputs (GY)	2048	GY0 – GY3777	0 – 2048	Read/Write
Outputs (Y)	512	Y0 – Y777	2049 – 2560	
Control Relays (CR)	1024	C0 –C1777	3073 – 4096	
Timer Contacts (T)	256	T0 – T377	6145 – 6400	
Counter Contacts (CT)	128	CT0 – CT177	6401 – 6528	
Stage Status Bits (S)	1024	S0 – S1777	5121 – 6144	
Reserved	—	—	6529 – 10000	

DL06 Word Data

Registers (Word)	Qty (Dec.)	PLC Range (Octal)	Modbus 40001 Address Range	Modbus 30001 Address Range	Access
V-Memory (Timers)	256	V0 – V377	40001 – 40256	30001 – 30256	Read/Write
V-Memory (Counters)	128	V1000 – V1177	40513 – 40460	30513 – 30640	
V-Memory (Data Words)	256	V400 – V777	40257 – 40512	30257 – 30512	
	3072	V1400 – V7377	40769 – 43840	30769 – 33840	
	4096	V10000 – V17777	44097 – 48192	34097 – 38192	

Your PC's Windows calculator can be used for number conversions (i.e. decimal to octal). The Windows calculator must be in Calculator>View>Scientific mode to enable number conversions capability.

Example 1: Calculating Word PLC Address

Find the PLC address to use to target Modbus address 41025 in a server device.

1. Subtract the beginning of the Modbus word address range (40001) from the desired Modbus address to target.
2. Convert decimal result into octal.
3. Add octal result to beginning PLC range (Input, Output or Word).

1. 41025 - 40001 = 1024 decimal
2. 1024 decimal = 2000 octal
3. V0 (octal) + 2000 (octal) = **V2000** octal

Example 2: Calculating Discrete Input PLC Address

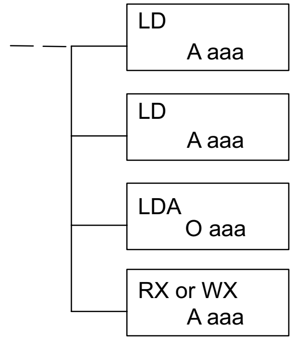
Find the PLC address to use to target Modbus address 12060 in a server device.

1. Subtract the beginning of the Modbus Input address range (12049) from the desired Modbus address to target.
2. Convert decimal result into octal.
3. Add octal result to beginning PLC range (Input, Output or Word).

1. 12060 - 12049 = 11 decimal
2. 11 decimal = 13 octal
3. X0 (octal) + 13 octal = **X13** octal

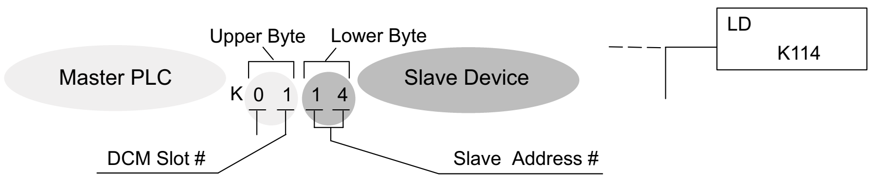
Building the Read (RX) or Write (WX) Routine

For network communications, you build the Read (RX) or Write (WX) instructions into a routine which requires the four instructions you see to the right. They must be used in the sequence shown. The following step-by-step procedure will provide you the information necessary to set up your ladder program to receive data from a network server.



Step 1: Identify ECOM Slot Location and Server Node

The first Load (LD) instruction accepts either a constant or a variable. Use a "K" to designate the number as a constant. Use a "V" if you are entering the address of a register. The contents of that register perform the same function as the constant shown below. For example, you could use V2000 in place of K0114. If the contents of V2000 is the number "114," the function would be the same. Using a variable allows changing parameters while the program is running.



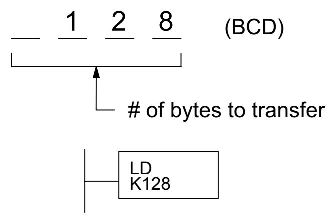
Depending on which slot the DCM is in, it has two Special Relay contacts associated with it. One indicates "Port busy", and the other indicates "Port Communication Error". The "Port Busy" bit is on while the PLC communicates with the slave. When the bit is off the program can initiate the next network request.

DL05 Special Relays	
Relay	**Option Slot**
Communication Busy	SP120
Communication Error	SP121

DL06 Special Relays				
Relay	**Slot 1**	**Slot 2**	**Slot 3**	**Slot 4**
Communication Busy	SP120	SP122	SP124	SP126
Communication Error	SP121	SP123	SP125	SP127

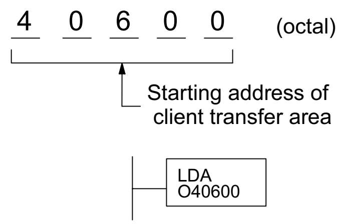
Step 2: Load Number of Bytes to Transfer

The second Load (LD) instruction determines the number of bytes which will be transferred between the master and slave in the subsequent WX or RX instruction. The value to be loaded is in BCD format (decimal), from 1 to 128 bytes. Requesting an even number of bytes, generates a Modbus message using Function 03, Read Holding Registers. If you need to Read Input Registers, Function Code 04, enter an odd number of bytes. For example, to read 10 Input Holding Registers, enter 2 (bytes/word) X 10 registers + 1, 21 bytes. This will request ten 30001 range addresses from the Modbus server (slave) device.



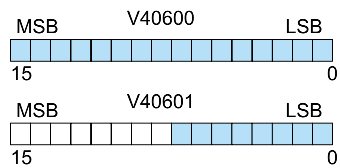
Step 3: Specify Master Memory Area

The third instruction in the RX or WX sequence is a Load Address (LDA) instruction. Its purpose is to load the starting address of the memory area to be transferred. Entered as an octal number, the LDA instruction converts it to hex and places the result in the accumulator.



For a WX instruction, the DL05/06 CPU sends the number of bytes previously specified from its memory area beginning at the LDA address specified.

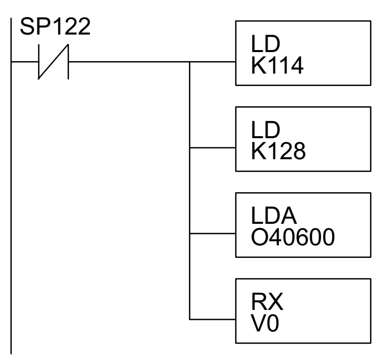
For an RX instruction, the DL05/06 CPU reads the number of bytes previously specified from the server, placing the received data into its memory area beginning at the LDA address specified.



NOTE: *Since V-memory words are always 16 bits, you may not always use the whole word. For example, if you only specify to read 3 bytes, you will only get 24 bits of data. In this case, only the 8 least significant bits of the last word location will be modified. The remaining 8 bits are not affected.*

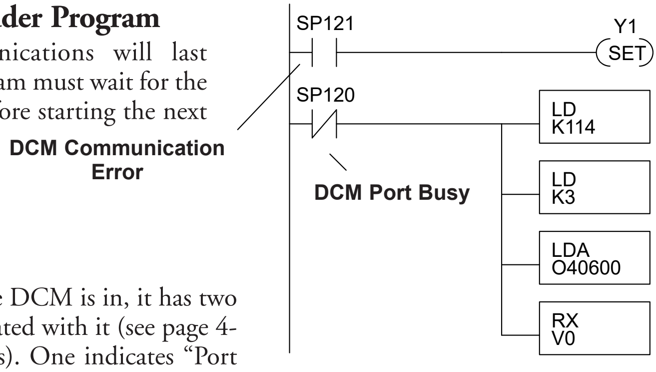
Step 4: Specify Slave Memory Area

The last instruction in our sequence is the WX or RX instruction itself. Use WX to write to the slave, and RX to read from the slave. All four of our instructions are shown to the right. In the last instruction, you must specify the starting address and a valid data type for the slave



Communications from a Ladder Program

Typically network communications will last longer than 1 scan. The program must wait for the communications to finish before starting the next transaction.



Depending on which slot the DCM is in, it has two Special Relay contacts associated with it (see page 4-11 to 4- 12 for special relays). One indicates "Port busy", and the other indicates "Port Communication Error". The example above shows the use of these contacts for an DCM that is in slot 1. The "Port Busy" bit is on while the PLC communicates with the slave. When the bit is off the program can initiate the next network request.

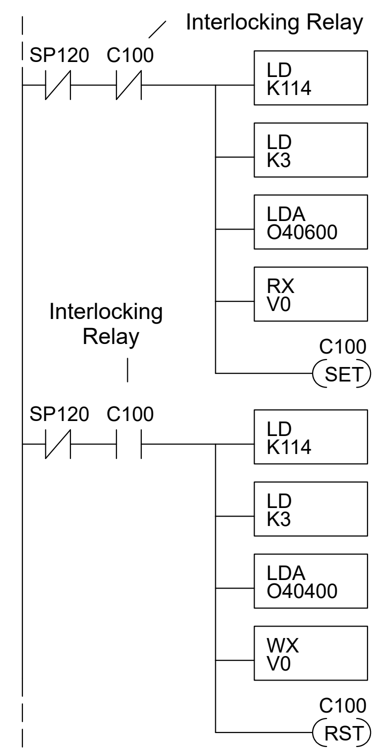
The "Port Communication Error" bit turns on when the PLC has detected an error. Use of this bit is optional. When used, it should be ahead of any network instruction boxes since the error bit is reset when an RX or WX instruction is executed.

Multiple Read and Write Interlocks

If you are using multiple reads and writes in the RLL program, you have to interlock the routines to make sure all the routines are executed. If you don't use the interlocks, then the CPU will only execute the first routine. This is because each port can only handle one transaction at a time.

In the example to the right, after the RX instruction is executed, C100 is set. When the port has finished the communication task, the second routine is executed and C100 is reset.

If your are using RLL*PLUS* Stage Programming, you can put each routine in a separate program stage to ensure proper execution and switch from stage to stage allowing only one of them to be active at a time.

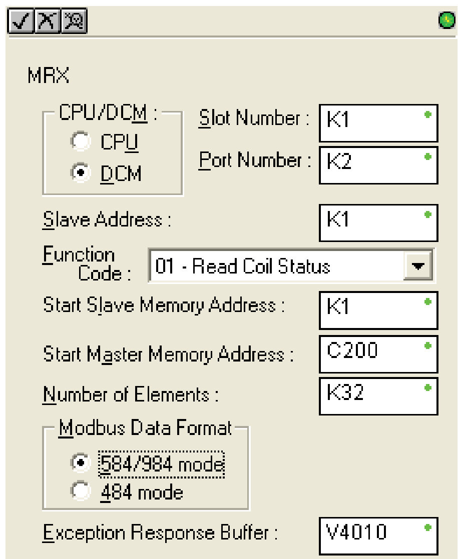


Network Master Operation: DL06 MRX / MWX Instructions

NOTE: *DirectSOFT (version 5.2 or later) and a DL06 PLC is required to use the MRX/MWX instructions.*

Modbus Read from Network (MRX)

The Modbus Read from Network (MRX) instruction is used by the DL06 DCM network master to read a block of data from a connected slave device and to write the data into V–memory addresses within the DL06 master CPU. The instruction allows the user to specify the DCM slot and port number, Modbus function code, slave station address, starting master and slave memory addresses, number of elements to transfer, Modbus data format and the exception response buffer.



- **CPU/DCM:** select DCM Port
- **Slot Number:** select the option slot the target DCM occupies (1–4)
- **Port Number:** must be D0–DCM Port 2 (K2)
- **Slave Address:** specify a slave station address (0–247)
- **Function Code:** The following Modbus function codes are supported by the MRX instruction:

  01 – Read Coil Status

  02 – Read Input Status

  03 – Read Holding Registers

  04 – Read Input Registers

  07 – Read Exception Status
- **Start Slave Memory Address:** specifies the starting slave memory address of the data to be read. See the table on the following page.
- **Start Master Memory Address:** specifies the starting memory address in the master where the data will be placed. See the table on the following page.
- **Number of Elements:** specifies how many coils, inputs, holding registers or input register will be read. See the table on the following page.
- **Modbus Data Format:** specifies Modbus 584/984 or 484 data format to be used
- **Exception Response Buffer:** specifies the master memory address where the Exception Response will be placed (requires 3 registers).

MRX Slave Memory Address

MRX Slave Address Ranges		
Function Code	**Modbus Data Format**	**Modbus Address Range(s)**
01 – Read Coil Status	484 Mode	1 – 999
	594/984 Mode	1 – 65535
02 – Read Input Status	484 Mode	1001 – 1999
	594/984 Mode	10001 – 19999 (5 digit) or 1000001 – 165535 (6 digit)
03 – Read Holding Registers	484 Mode	4001 – 4999
	594/984 Mode	40001 – 49999 (5 digit) or 4000001 – 465535 (6 digit)
04 – Read Input Registers	484 Mode	3001 – 3999
	594/984 Mode	30001 – 39999 (5 digit) or 3000001 – 365535 (6 digit)
07 – Read Exception Status	484 and 594/984 Mode	N/A

MRX Master Memory Addresses

MRX Master Memory Address Ranges		
Operand Data Type		**DL06 Range**
Inputs	**X**	0–1777
Outputs	**Y**	0–1777
Control Relays	**C**	0–3777
Stage Status Bits	**S**	0–1777
Timer Bits	**T**	0–377
Counter Bits	**CT**	0–377
Special Relays	**SP**	0–777
V-memory	**V**	All
Global Inputs	**GX**	0–3777
Global Outputs	**GY**	0–3777

MRX Number of Elements

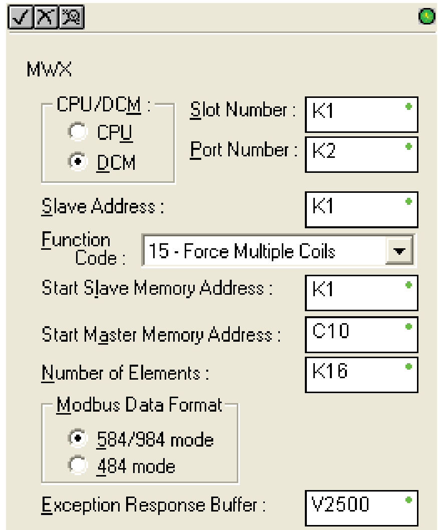
Number of Elements		
Operand Data Type		**DL06 Range**
V-memory	**V**	All
Constant	**K**	Bits: 1–2000 Registers: 1–125

MRX Exception Response Buffer

Exception Response Buffer		
Operand Data Type		**DL06 Range**
V-memory	**V**	All

Modbus Write to Network (MWX)

The Modbus Write to Network (MWX) instruction is used to write a block of data from the DL06 DCM network master memory to Modbus memory addresses within a slave device on the network. The instruction allows the user to specify the Modbus function code, slave station address, starting master and slave memory addresses, number of elements to transfer, Modbus data format and the exception response buffer.



- **CPU/DCM:** select DCM
- **Slot Number:** select the option slot the target DCM occupies (1–4)
- **Port Number:** must be D0–DCM Port 2 (K2)
- **Slave Address:** specify a slave station address (0–247)
- **Function Code:** The following Modbus function codes are supported by the MWX instruction:

  05 – Force Single coil

  06 – Preset Single Register

  15 – Force Multiple Coils

  16 – Preset Multiple Registers
- **Start Slave Memory Address:** specifies the starting slave memory address where the data will be written. See the table on the following page.
- **Start Master Memory Address:** specifies the starting address of the data in the master that is to written to the slave. See the table on the following page.
- **Number of Elements:** specifies how many consecutive coils or registers will be written to. This field is only active when either function code 15 or 16 is selected.
- **Modbus Data Format:** specifies Modbus 584/984 or 484 data format to be used
- **Exception Response Buffer:** specifies the master memory address where the exception response will be placed (requires 3 registers)

MWX Slave Memory Address

MWX Slave Address Ranges		
Function Code	**Modbus Data Format**	**Modbus Address Range(s)**
05 – Force Single Coil	484 Mode	1 – 999
	594/984 Mode	1 – 65535
06 – Preset Single Registers	484 Mode	4001 – 4999
	594/984 Mode	40001 – 49999 (5 digit) or 4000001 – 465535 (6 digit)
15 – Force Multiple Coils	484 Mode	1 – 999
	594/984 Mode	1 – 65535
16 – Preset Multiple Registers	484 Mode	4001 – 4999
	594/984 Mode	40001 – 49999 (5 digit) or 4000001 – 465535 (6 digit)

MWX Master Memory Addresses

MWX Master Memory Address Ranges		
Operand Data Type		**DL06 Range**
Inputs	**X**	0–1777
Outputs	**Y**	0–1777
Control Relays	**C**	0–3777
Stage Status Bits	**S**	0–1777
Timer Bits	**T**	0–377
Counter Bits	**CT**	0–377
Special Relays	**SP**	0–777
V-memory	**V**	All
Global Inputs	**GX**	0–3777
Global Outputs	**GY**	0–3777

MRX Number of Elements

Number of Elements		
Operand Data Type		**DL06 Range**
V-memory	**V**	All
Constant	**K**	Bits: 1–2000 Registers: 1–125

MRX Exception Response Buffer

Exception Response Buffer		
Operand Data Type		**DL06 Range**
V-memory	**V**	All

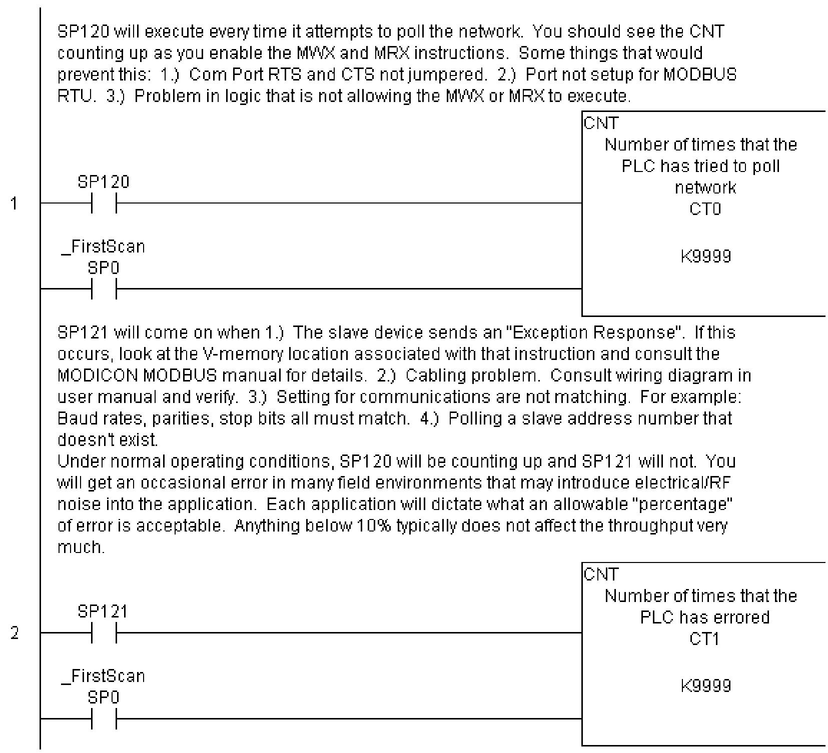
MRX/MWX Example in *Direct*SOFT

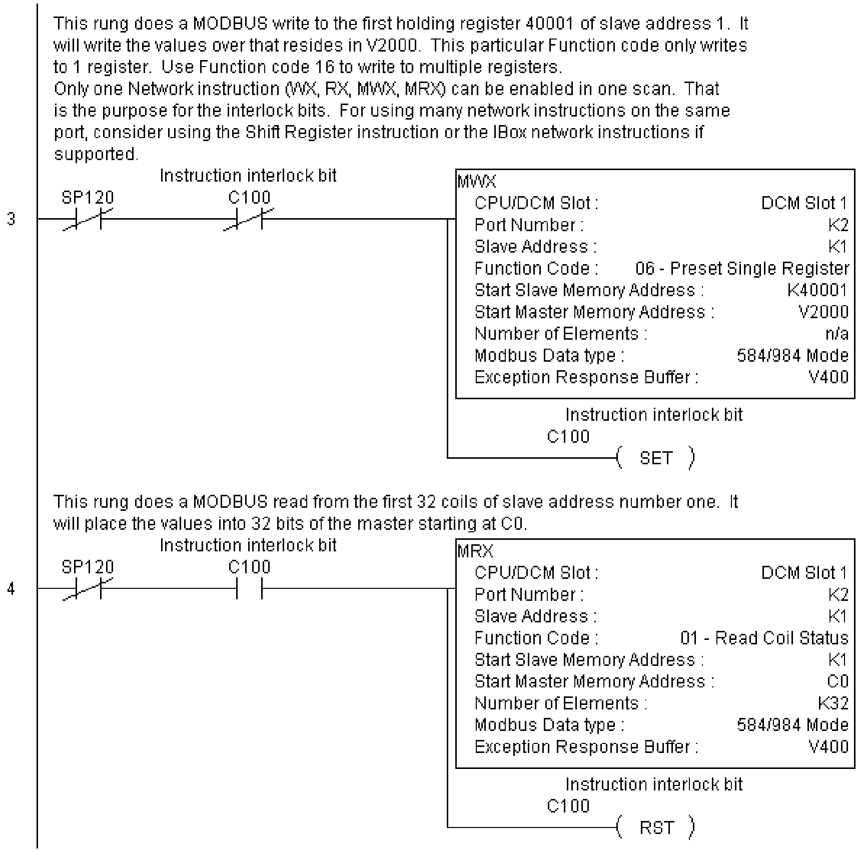
The DCM (port 2) has two Special Relay contacts associated with it (see 5–12 for comm port special relays). One indicates "Port busy" and the other indicates "Port Communication Error". The "Port Busy" bit is on while the PLC communicates with the slave. When the bit is off the program can initiate the next network request. The "Port Communication Error" bit turns on when the PLC has detected an error and use of this bit is optional. When used, it should be ahead of any network instruction boxes since the error bit is reset when an MRX or MWX instruction is executed. Typically network communications will last longer than 1 CPU scan. The program must wait for the communications to finish before starting the next transaction.

Multiple Read and Write Interlocks

If you are using multiple reads and writes in the RLL program, you have to interlock the routines to make sure all the routines are executed. If you don't use the interlocks, then the CPU will only execute the first routine. This is because each port can only handle one transaction at a time. In the example below, after the MRX instruction is executed, C100 is set. When the port has finished the communication task, the second routine is executed and C100 is reset. If you're using RLL^{PLUS} Stage Programing, you can put each routine in a separate program stage to ensure proper execution and switch from stage to stage allowing only one of them to be active at a time.

The following MRX/MWX example is for a DL06 CPU with a D0–DCM in option slot 1.




This rung does a MODBUS write to the first holding register 40001 of slave address 1. It will write the values over that resides in V2000. This particular Function code only writes to 1 register. Use Function code 16 to write to multiple registers.
Only one Network instruction (WX, RX, MWX, MRX) can be enabled in one scan. That is the purpose for the interlock bits. For using many network instructions on the same port, consider using the Shift Register instruction or the IBox network instructions if supported.
3
SP120
Instruction interlock bit
C100
MWX
CPU/DCM Slot : DCM Slot 1
Port Number : K2
Slave Address : K1
Function Code : 06 - Preset Single Register
Start Slave Memory Address : K40001
Start Master Memory Address : V2000
Number of Elements : n/a
Modbus Data type : 584/984 Mode
Exception Response Buffer : V400
Instruction interlock bit
C100
SET
This rung does a MODBUS read from the first 32 coils of slave address number one. It will place the values into 32 bits of the master starting at C0.
4
SP120
Instruction interlock bit
C100
MRX
CPU/DCM Slot : DCM Slot 1
Port Number : K2
Slave Address : K1
Function Code : 01 - Read Coil Status
Start Slave Memory Address : K1
Start Master Memory Address : C0
Number of Elements : K32
Modbus Data type : 584/984 Mode
Exception Response Buffer : V400
Instruction interlock bit
C100
RST

Communications Using Network IBox Instructions

Chapter 6

In This Chapter...

NOTE: Direct**SOFT (version 5.2 or later) is required to program using the IBox instructions. The DL05 CPU requires firmware version v5.10 or later, and the DL06 CPU requires firmware version v2.10 or later to support use of the IBox instructions. See our web site for firmware information and downloads:* ***www.automationdirect.com

Network Configuration Instruction (NETCFG)

NETCFG IB–700

Network Config defines all the common information necessary for performing network reading and writing using the NETRX and NETWX IBox instructions via the D0–DCM module.

You must have the Network Config instruction at the top of your ladder/stage program with any other configuration IBoxes.

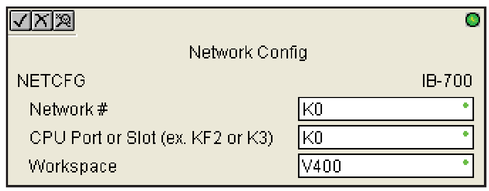
If you use more than one local serial port, DCM or ECOM in your PLC for Networking, you must have a different Network Config instruction for each network in your system that utilizes any NETRX/NETWX IBox instructions.

The Workspace parameter is an internal, private register used by the Network Config IBox and must be unique in this one instruction and must not be used anywhere else in your program.

The “CPU Port or Slot” parameter is the PLC option slot the DCM is occupying.

NETCFG Parameters:

- **Network#:** specifies a unique # for each DCM network to use.
- **CPU Port or Slot:** specifies the option slot number of used by the DCM.
- **Workspace:** specifies a V-memory location that will be used by the instruction.



Parameter		DL05/DL06 Range
Network #	K	K0–255
CPU Port or Slot	K	K0–FF
Workspace	V	See PLC V-memory map – Data Words

Network Read Instruction (NETRX)

NETRX IB–701

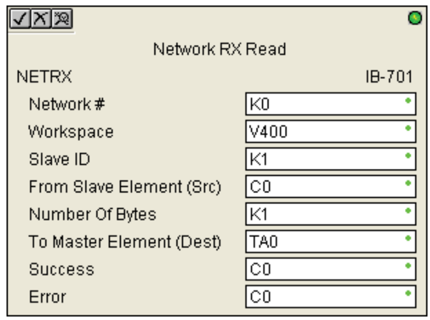
Network RX Read performs the RX instruction with built–in interlocking with all other Network RX (NETRX) and Network WX (NETWX) IBoxes in your program to simplify communications networking. It will perform the RX on the specified Network #, which corresponds to a specific unique Network Configuration (NETCFG) at the top of your program.

The Workspace parameter is an internal, private register used by this IBox and must be unique in this one instruction and must not be used anywhere else in your program.

Whenever this IBox has power, it will read element data from the specified slave into the given destination V-memory buffer, giving other Network RX and Network WX IBoxes on that Network# a chance to execute.

For example, if you wish to read and write data continuously from 5 different slaves, you can have all of these NETRX and NETWX instructions in one rung driven by SP1 (Always On). They will execute round–robin style, automatically.

NETRX Parameters:



- **Network#:** specifies the (CPU port's, DCM's, ECOM's) Network# defined by the NETCFG instruction.
- **Workspace:** specifies a V-memory location that will be used by the instruction Slave ID: specifies the slave PLC that will be targeted by the NETRX instruction From Slave Element (Src): specifies the slave address of the data to be read Number of Bytes: specifies the number of bytes to read from the slave device.
- **To Master Element (Dest):** specifies the location where the slave data will be placed in the master PLC.
- **Success:** specifies a bit that will turn on once the request is completed successfully.
- **Error:** specifies a bit that will turn on if the instruction is not successfully completed.

Parameter		DL05/DL06 Range
Network #	K	K0–255
Workspace	V	See PLC V-memory map – Data Words
Slave ID	K	K0–90
SRC	X,Y,C,S,T,CT,GX,GY,V	See PLC V-memory map
# of Bytes	K	K1–128
Dest	V	See PLC V-memory map – Data Words
Success	X,Y,C,GX,GY,B	See PLC V-memory map
Error	X,Y,C,GX,GY,B	See PLC V-memory map

Network Write Instruction (NETWX)

NETWX IB–702

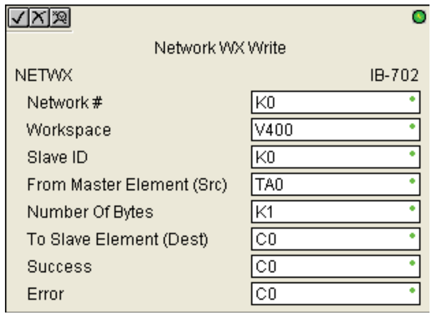
Network WX Write performs the WX instruction with built–in interlocking with all other Network RX (NETRX) and Network WX (NETWX) IBoxes in your program to simplify communications networking. It will perform the WX on the specified Network #, which corresponds to a specific unique Network Configuration (NETCFG) at the top of your program.

The Workspace parameter is an internal, private register used by this IBox and must be unique in this one instruction and must not be used anywhere else in your program.

Whenever this IBox has power, it will write data from the master's V-memory buffer to the specified slave starting with the given slave element, giving other Network RX and Network WX IBoxes on that Network # a chance to execute.

For example, if you wish to read and write data continuously from 5 different slaves, you can have all of these NETRX and NETWX instructions in one rung driven by SP1 (Always On). They will execute round–robin style, automatically.

NETWX Parameters:



- **Network#:** specifies the (CPU port's, DCM's, ECOM's) Network# defined by the NETCFG instruction.
- **Workspace:** specifies a V-memory location that will be used by the instruction.
- **Slave ID:** specifies the slave PLC that will be targeted by the NETWX instruction.
- **From Slave Element (Src):** specifies the location in the master PLC where the data will be sourced from.
- **Number of Bytes:** specifies the number of bytes to write to the slave device.
- **To Master Element (Dest):** specifies the slave address the data will be written to.
- **Success:** specifies a bit that will turn on once the request is completed successfully.
- **Error:** specifies a bit that will turn on if the instruction is not successfully completed.

Parameter		DL05/DL06 Range
Network #	K	K0–255
Workspace	V	See PLC V-memory map – Data Words
Slave ID	K	K0–90
SRC	X,Y,C,S,T,CT,GX,GY,V	See PLC V-memory map – Data Words
# of Bytes	K	K1–128
Dest	X,Y,C,S,T,CT,GX,GY,V	See PLC V-memory map
Success	X,Y,C,GX,GY,B	See PLC V-memory map
Error	X,Y,C,GX,GY,B	See PLC V-memory map

Example Using NETCFG, NETRX and NETWX

Rung 1: The Network Configuration IBox coordinates all of the interaction with other Network IBoxes (NETRX/NETWX). You must have a Network Configuration IBox for each CPU serial port, DCM or ECOM network in your system. Configuration IBoxes must be at the top of your program and must execute every scan.

This IBox defines Network# K0 to be for a DCM in slot 1.

The Workspace register is used to maintain state information about the port or module, along with proper sharing and interlocking with the other NETRX and NETWX IBoxes in the program. This V memory register must not be used anywhere else in the entire program.

Rung 2: Using Network# K0, read X0–X7 from Slave K7 and write them to slave K5 as fast as possible. Store them in this local PLC in C200–C207, and write them to C300–C307 in slave K5.

Both the NETRX and NETWX work with the Network Config IBox to simplify all networking by handling all of the interlocks and proper resource sharing. They also provide very simplified error reporting. You no longer need to worry about any SP "busy bits" or "error bits", or what port number or slot number a module is in, or have any counters or shift registers or any other interlocks for resource management.

In this example, SP1 (always ON) is driving both the NETRX and NETWX IBoxes in the same rung. On the scan that the Network Read completes, the Network Write will start that same scan. As soon as the Network Write completes, any pending operations below it in the program would get a turn. If there are no pending NETRX or NETWX IBoxes below this IBox, then the very next scan the NETRX would start its request again.