4

System Design and Configuration

In This Chapter. . . .

— DL405 System Design Strategies
— Module Placement and Configuration
— Calculating the Power Budget
— Local I/O Expansion
— Remote I/O Expansion
— Slice I/O Expansion
— Network Connections to MODBUS® and *Direct*NET
— Network Slave Operation
— Network Master Operation

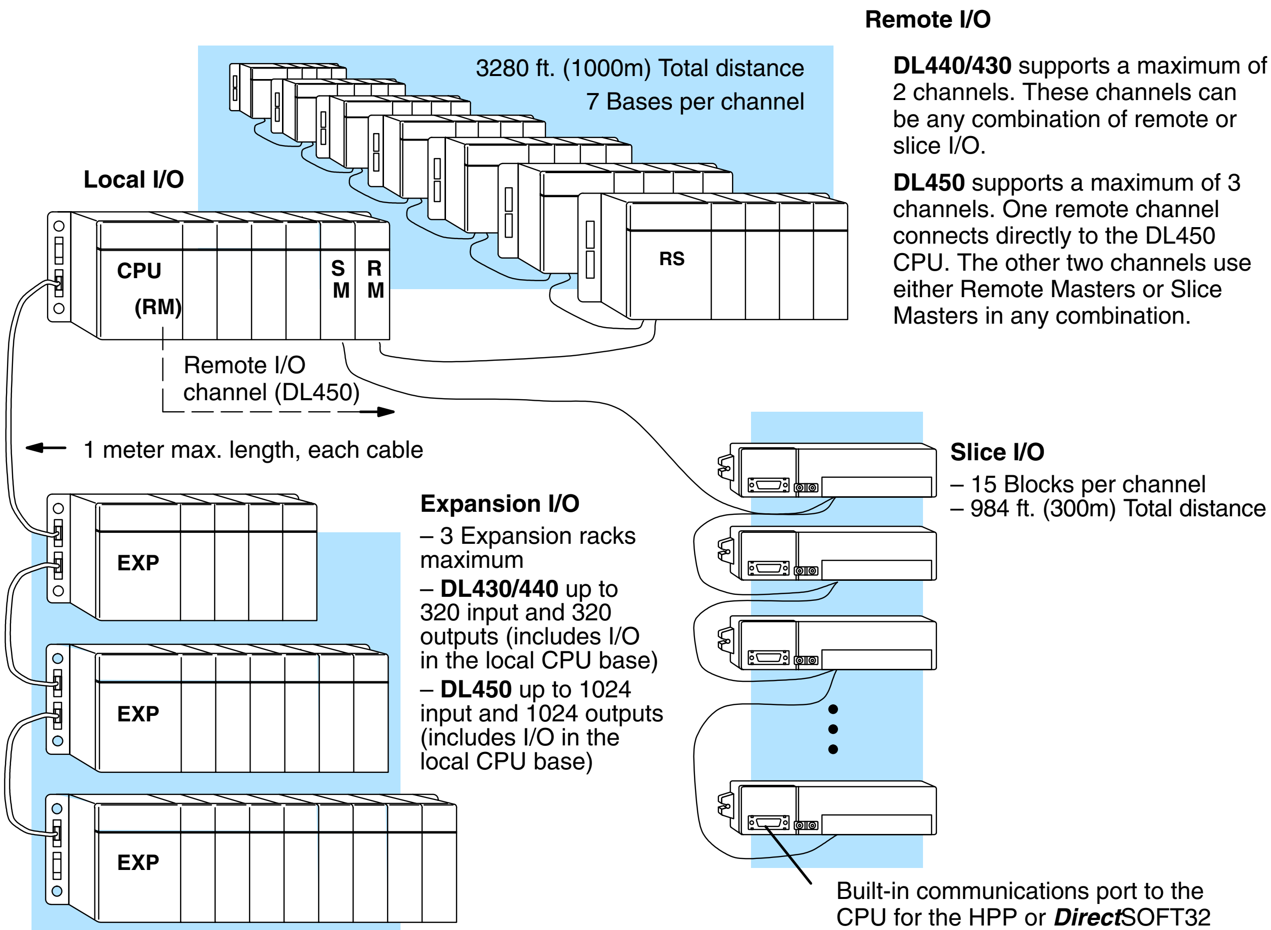
DL405 System Design Strategies

I/O System Configurations

The DL405 PLCs offer the following ways to add netwroking to the system:

- **Local I/O** – consists of I/O modules located in the same base as the CPU.
- **Expansion I/O** – consists of I/O modules in expansion bases located close to the the local base. Expansion cables connect them to the local CPU base's serial bus in daisy-chain fashion.
- **Remote I/O** – consists of I/O modules located in bases which are serially connected to the local CPU base through a Remote Master module, or may connect directly to port 3 on a DL450 CPU.
- **Slice I/O** – consists of small fixed blocks of I/O (not modules) which are serially connected to the local CPU base through the Slice Master module. Note that you cannot connect Slice I/O directly to a DL450 CPU.

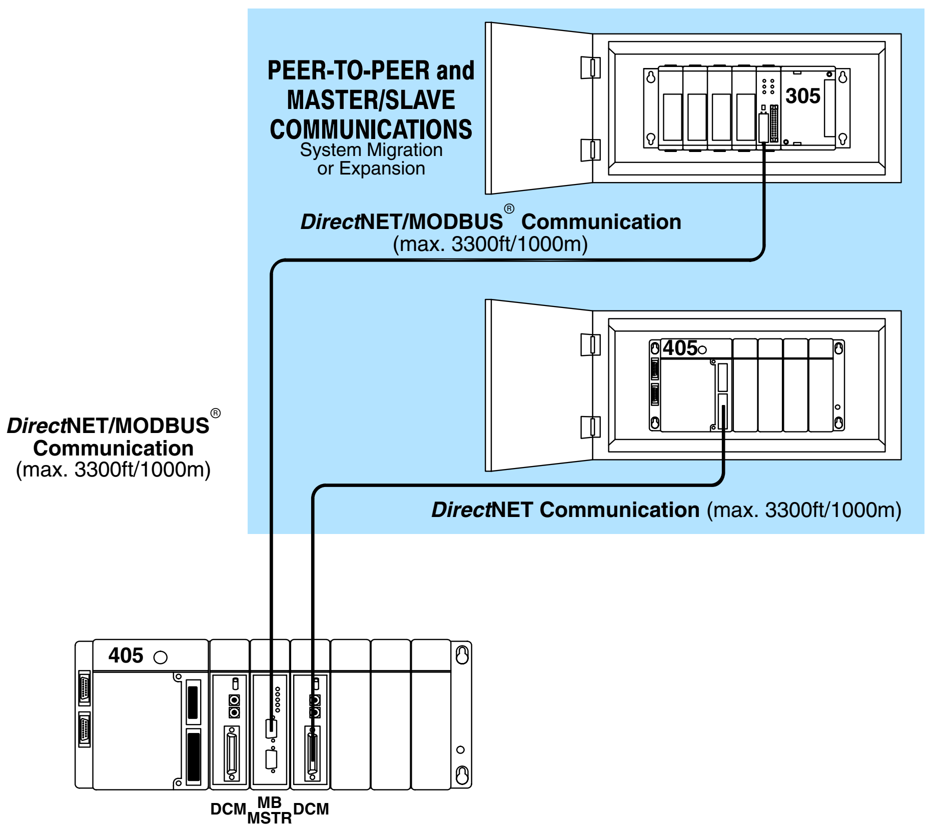
A DL405 system can be developed using many different arrangements of these configurations. All I/O configurations, except Slice slaves use the standard complement of DL405 I/O modules and bases. Slice slaves act as their own base and I/O. Below is a brief description of each of these configurations. Examples of each configuration are discussed in detail later in this chapter.



Networking Configurations

The DL405 PLCs offer the following four ways to add I/O modules to the system:

- **Data Communications Module** – connects a DL405 system to devices using the ***Direct***NET protocol, or connects as a slave to a MODBUS network.
- **DL450 Communications Ports** – the DL450 CPU has two extra (total of four) built-in comm ports. It allow two network connections directly from the CPU. See Chapter 3, CPU Specifications and Operation, for individual port specifications, and the sections at the end of this chapter for network connections.
- **MODBUS Master Module** – You can use MODBUS master modules in any slot of a DL405 system for connecting it as a master to a MODBUS network, using the RTU protocol.
- **MODBUS Slave Module** – You can use MODBUS slave modules in any slot of a DL405 system for connecting it as a slave to a MODBUS network, using the RTU protocol.
- **TIWAY® Network Interface Module** – Interface to Texas Instruments and Siemens TIWAY networks by using this module as a slave.
- **Shared Data Network Module** – The Shared Data Network Module lets you make peer–to–peer connections between DL405 PLC systems.



Module Placement and Configuration

Valid Module/Unit Locations

The most commonly used I/O modules for the DL405 system (AC, DC, AC/DC, Relay and Analog) can be used in any base in your system. The table below lists by category the valid locations for all modules/units in a DL405 system. Remember that the power budget can limit the number of modules in a base (discussed later).

Module/Unit	Local CPU Base	Local Exp. Base	Remote Base
CPUs	CPU Slot Only		
Expansion Units		CPU Slot Only	
8/16/32pt DC Input Modules	✓	✓	✓
64pt DC Input Modules	✓ Note 1	✓ Note 1, 2	
AC Input Modules	✓	✓	✓
AC/DC Input Modules	✓	✓	✓
8/16/32pt DC Output Modules	✓	✓	✓
64pt DC Output Modules	✓ Note 1	✓ Note 1, 2	
AC Output Modules	✓	✓	✓
Relay Output Modules	✓	✓	✓
Analog Modules	✓	✓	✓
Remote I/O			
Remote Master	✓		
Remote Slave Unit			CPU Slot Only
Slice Master	✓		
Communications and Networking Modules	✓	✓ Note 2	
CoProcessor Modules	✓		
Specialty Modules			
Interrupt	DL430 – Slot 0 Only DL440 – Slots 0 & 1 DL450 – Slots 0 & 1		
High Speed Counter	✓	✓	
PID Module	✓		
SDS	✓		
4 Loop Temp. Controller	✓		
Input Simulator	✓	✓	✓
Filler	✓	✓	✓

Note 1: When using 64 pt modules, you cannot use any specialty modules in slots 5, 6, and 7 in the same base.

Note 2: Specialty modules are allowed in expansion bases only if you are using the DL450 CPU and all bases in the system are the D4–xxB–1 type bases.

I/O Configuration Methods

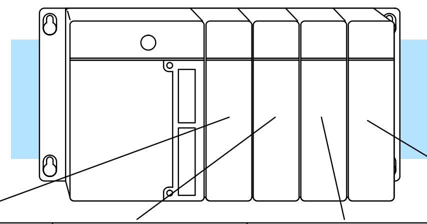
There are two methods of I/O configuration for the DL405 CPUs:

- **Auto configuration** – the CPU automatically configures the I/O. It assigns the lowest I/O numbers to the module in slot 0 (the slot next to the CPU), the next set of I/O numbers to the next module in the base, etc. The numbers are assigned only to modules actually in the base, not to empty slots in the base. This is the default mode of the CPU.
- **Manual configuration** – (DL440/DL450 only) allows you assign I/O numbers. Numbers can be assigned to empty slots or in any order as long as the numbers are assigned in groups of 16 or 32.

Automatic Configuration

The DL405 CPUs automatically detect any installed I/O modules (including specialty modules)at powerup, and establish the correct I/O configuration and addresses. For most applications, you will never have to change the configuration.

I/O addresses use octal numbering, starting at X0 and Y0 in the slot next to the CPU. The addresses are assigned in groups of 8, 16, 32, or 64 depending on the number of points for the I/O module. The discrete input and output modules can be mixed in any order, but there may be restrictions placed on some specialty modules. The following diagram shows the I/O numbering convention for an example system.



Slot 0	Slot 1	Slot 2	Slot 3
8pt. Input	32pt. Output	16pt. Input	8pt. Input
X0–X7	Y0–Y37	X10–X27	X30–X37

Both the Handheld Programmer and ***Direct***SOFT32 provide AUX functions that allow you to automatically configure the I/O. For example, with the Handheld Programmer AUX 46 executes an automatic configuration, which allows the CPU to examine the installed modules and determine the I/O configuration and addressing. With ***Direct***SOFT32, the PLC Configure I/O menu option would be used.

Manual Configuration

✗ ✓ ✓
430 440 450

It may never become necessary, but DL440 and DL450 CPUs allow manual I/O address assignment for any I/O slot(s) in local or expansion bases. You can manually modify an auto configuration to match arbitrary I/O numbering. For example, two adjacent input modules can have starting addresses at X10 and X200.

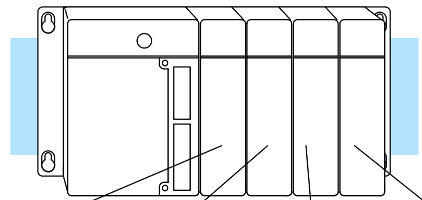
In automatic configuration, the addresses are assigned on 8-point boundaries. Manual configuration, however, assumes that all modules are at least 16 points, so you can only assign addresses that are a multiple of 20 (octal). For example, X30 and Y50 are not valid addresses. You can still use 8 point modules, but 16 addresses will be assigned and the upper eight addresses will be unused.

WARNING: If you manually configure an I/O slot, the I/O addressing for the other modules may change. This is because the DL405 CPUs do not allow you to assign duplicate I/O addresses. You must always correct any I/O configuration errors before you place the CPU in RUN mode. Uncorrected errors can cause unpredictable machine operation that can result in a risk of personal injury or damage to equipment.

Removing a Manual Configuration

After a manual configuration, the system will automatically retain the new I/O addresses through a power cycle. You can remove (overwrite) any manual configuration changes by simply performing an automatic configuration.

The following diagram shows how I/O addresses change after manually configuring a slot.



	Slot 0	Slot 1	Slot 2	Slot 3
Automatic	8pt. Input X0–X7	32pt. Output Y0–Y37	16pt. Input X10–X27	8pt. Input X30–X37
Manual	8pt. Input X0–X7	32pt. Output Y0–Y37	16pt. Input X100–X117	8pt. Input X20–X27

Power–On I/O Configuration Check

The DL405 CPUs can also be set to automatically check the I/O configuration on power-up. By selecting this feature you can detect any changes that may have occurred while the power was disconnected. For example, if someone places an output module in a slot that previously held an input module, the configuration check will detect the change and print a message on the Handheld Programmer or ***Direct***SOFT32 screen (use AUX 44 on the HPP to enable the configuration check).

If the system detects a change in the I/O configuration at power-up, an error code E252 NEW I/O CONFIGURATION will be generated. You can use AUX 42 to determine the exact base and slot location where the change occurred.

WARNING: You should always correct any I/O configuration errors before you place the CPU into RUN mode. Uncorrected errors can cause unpredictable machine operation that can result in a risk of personal injury or damage to equipment.

When a configuration error is generated, you may actually want to use the new I/O configuration. For example, you may have intentionally changed an I/O module to use with a program change. You can use AUX 45 to select the new configuration, or, keep the existing configuration stored in memory.

WARNING: Verify the I/O configuration being selected will work properly with the CPU program. Always correct any I/O configuration errors before placing the CPU in RUN mode. Uncorrected errors can cause unpredictable machine operation that can result in a risk of personal injury or damage to equipment.

Calculating the Power Budget

Managing your Power Resource

As you have seen, the I/O configuration depends on your choice of I/O modules, bases, and I/O location. When determining the types and quantity of I/O modules you will be using in the DL405 system it is important to remember there is a limited amount of power available from the power supply to the system. We have provided a chart to help you easily see the amount of power you will have with your CPU, Expansion Unit or Remote Slave selection. The following chart will help you calculate the amount of power you need with your I/O selections. At the end of this section you will also find an example of power budgeting and a worksheet for your own calculations.

If the I/O you chose exceeds the maximum power available from the power supply you can resolve the problem by shifting some of the modules to an expansion base which contains another power supply.

WARNING: It is *extremely* important to calculate the power budget correctly. If you exceed the power budget, the system may operate in an unpredictable manner which may result in a risk of personal injury or equipment damage.

CPU Power Specifications

The following chart shows the amount of current **available** for the two voltages supplied on the DL405 CPU, Expansion unit or Remote Slave unit. Use these currents when calculating the power budget for you system. The Auxiliary 24V Power Source mentioned in the table is a connection at the base terminal strip allowing you to connect to devices or DL405 modules that require 24VDC.

CPUs	5V Current Supplied in mA.	Auxiliary 24V Power Source Current Supplied in mA.		Remote and Expansion Units	5V Current Supplied in mA.	Auxiliary 24V Power Source Current Supplied in mA.
D4–430	3700	400		D4–EX	4000	400
D4–440	3700	400		D4–EXDC	4000	None
D4–440DC–1	3700	None		D4–EXDC–2	3700	None
D4–440DC–2	3700	None		D4–RS	3700	400
D4–450	3100	400		D4–RSDC	3700	None
D4–450DC–1	3100	None		H4–EBC	3470	400
D4–450DC–2	3100	None		H4–EBC–F	3300	400

Module Power Requirements

The chart on the next page shows the amount of maximum current **required** for each of the DL405 modules. Use these currents when calculating the power budget for your system. If external 24VDC is required, the external 24V from the CPU power supply may be used as long as the power budget is not exceeded.

Device	5V Current Required (mA)	External 24V Current Req. (mA)
I/O Bases		
D4–04B, D4–04BNX, D4–04B–1	80	None
D4–06B, D4–06BNX, D4–06B–1	80	None
D4–08B, D4–08BNX, D4–08B–1	80	None
DC Input Modules		
D4–08ND3S	100	None
D4–16ND2	150	None
D4–16ND2F	150	None
D4–32ND3–1	150	None
D4–32ND3–2	150	None
D4–64ND2	300 (max)	None
AC Input Modules		
D4–08NA	100	None
D4–16NA	150	None
AC/DC Input Modules		
D4–16NE3	150	None
F4–08NES	90	None
DC Output Modules		
D4–08TD1	150	35
F4–08TD1S	295	None
D4–16TD1	200	125
D4–16TD2	400	None
D4–32TD1	250	140
D4–32TD1–1	250	140 (5–15VDC)
D4–32TD2	350	120 / (4A max including loads)
D4–64TD1	800 (max)	None
AC Output Modules		
D4–08TA	250	None
D4–16TA	450	None
Relay Output Modules		
D4–08TR	550	None
F4–08TRS–1	575	None
F4–08TRS–2	575	None
D4–16TR	1000	None
Programming		
D4–HPP	320	None
DV–1000	150	None

Device	5V Current Required (mA)	External 24V Current Req. (mA)
Analog Modules		
F4–04AD	85	100
F4–04ADS	270	120
F4–08AD	75	90
D4–02DA	250	300
F4–04DA	120	180
F4–04DA–1	70	75 + 20 per circuit
F4–04DA–2	90	75 + 20 per circuit
F4–04DAS–1	60	50 per circuit
F4–08DA–1	90	100 + 20 per circuit
F4–16DA–1	90	100 + 20 per circuit
F4–08THM–n	120	50 + 20 per circuit
F4–08RTD	80	None
Remote I/O		
D4–RM	300	None
D4–SM	300	None
D4–SS–88	None	100, (250 w/ HPP)
D4–SS–106	None	100, (250 w/ HPP)
D4–SS–16T	None	100, (250 w/ HPP)
D4–SS–16N	None	100, (250 w/ HPP)
Communications and Networking		
D4–DCM	500	None
H4–ECOM	530	None
H4–ECOM–F	670	None
F4–MAS–MB	235	None
F4–SLV–MB	235	None
F4–SLV–TW	250	None
F4–SDN	235	None
CoProcessors™		
F4–CP128	305	None
F4–CP512	235	None
F4–CP128–T	350	None
Specialty Modules		
D4–INT	100	None
D4–HSC	300	None
F4–16PID	160	None
F4–8MPI	225	170
D4–16SIM	150	None
F4–SDS	110	None
F4–4LTC	280	75

Power Budget Calculation Example

The following example shows how to calculate the power budget for the DL405 system.

Base # 0	**Module Type**	**5 VDC (mA)**	**Auxiliary Power Source 24 VDC Output (mA)**
CPU/ Expansion Unit/ Remote Slave Used	D4–430	**3700**	**400**
Slot 0	D4–16ND2	+ 150	+ 0
Slot 1	D4–16ND2	+ 150	+ 0
Slot 2	D4–02DA	+ 250	+ 300
Slot 3	D4–08ND3S	+ 100	+ 0
Slot 4	D4–08ND3S	+ 100	+ 0
Slot 5	D4–16TD2	+ 400	+ 0
Slot 6	D4–16TD2	+ 400	+ 0
Slot 7	D4–16TR	+ 1000	+ 0
Other			
Base	D4–08B	+ 80	+ 0
Handheld Prog	D4–HPP	+ 320	+ 0
Maximum power required		**2950**	**300**
Remaining Power Available		3700–2950 =**750**	400 – 300 = **100**

1. Using the tables at the beginning of the Power Budgeting section of this chapter fill in the information for the CPU/Expansion Unit/Remote Slave, I/O modules, and any other devices that will use system power including devices that use the 24 VDC output. Pay special attention to the current supplied by either the CPU, Expansion Unit, and Remote Slave since they do differ. Devices which fall into the "**Other**" category are devices such as the Base and the Handheld programmer which also have power requirements but do not directly plug into the base.
2. Add the current columns starting with Slot 0 and put the total in the row labeled "**Maximum power required**"**.**
3. Subtract the row labeled "**Maximum power required**" from the row labeled "**CPU/Expansion Unit/Remote Slave Used**". Place the difference in the row labeled **"Remaining Power Available**".
4. If **"Maximum Power Required"** is greater than "**CPU/Expansion Unit/Remote Slave Used**" in any of the three columns, the power budget will be exceeded. It will be unsafe to used this configuration and you will need to restructure your I/O configuration.

Power Budget Calculation Worksheet

You may copy and use the following blank chart for your power budget calculations.

Base # ______	Module Type	5 VDC (mA)	Auxiliary Power Source 24 VDC Output (mA)
CPU/ Expansion Unit/ Remote Slave Used			
Slot 0			
Slot 1			
Slot 2			
Slot 3			
Slot 4			
Slot 5			
Slot 6			
Slot 7			
Other			
Maximum Power Required			
Remaining Power Available			

1. Using the tables at the beginning of the Power Budgeting section of this chapter fill in the information for the CPU/Expansion Unit/Remote Slave, I/O modules, and any other devices that will use system power including devices that use the 24 VDC output. Pay special attention to the current supplied by either the CPU, Expansion Unit, and Remote Slave since they do differ. Devices which fall into the "**Other**" category are devices such as the Base and the Handheld programmer which also have power requirements but do not directly plug into the base.
2. Add the current columns starting with Slot 0 and put the total in the row labeled "**Maximum power required**"**.**
3. Subtract the row labeled "**Maximum power required**" from the row labeled "**CPU/Expansion Unit/Remote Slave Used**". Place the difference in the row labeled **"Remaining Power Available**".
4. If **"Maximum Power Required"** is greater than "**CPU/Expansion Unit/Remote Slave Used**" in any of the three columns, the power budget will be exceeded. It will be unsafe to used this configuration and you will need to restructure your I/O configuration.

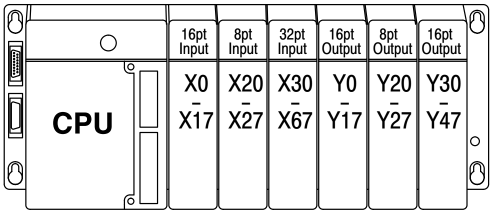
Local I/O Expansion

430 440 450

The following I/O base configurations will assist you in understanding the options available in the DL405 series. Local and expanded bases are the most common and cost effective way of installing I/O. With local and expanded I/O the CPU can automatically configure the I/O for you. Use Remote and Slice I/O when it is necessary to locate I/O at distances away from the CPU. Both Remote and Slice I/O do require additional ladder programming to operate.

Local Base and I/O

The local base is the base in which the CPU resides. Local I/O modules reside in the same base as the CPU. For example, placing 32 point modules in all eight slots in an 8-slot base will use 256 I/O points . The status of each I/O point is updated each I/O scan of the CPU.

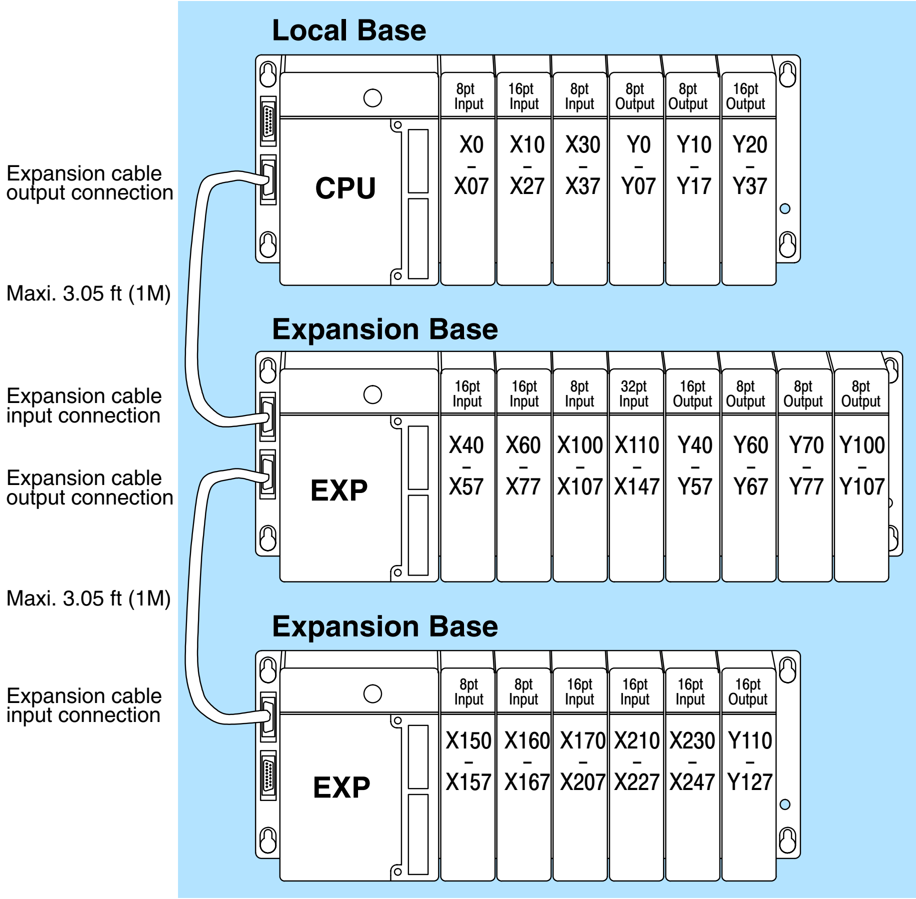


Local Expansion Base and I/O

Use local expansion when you need more I/O points or a greater power budget than the local base provides. The expansion bases require a Local Expansion Unit (in the place of a CPU), and a cable (either D4–EXCBL–1 or D4–EXCBL–2) to connect to the local CPU base. The following figure shows one CPU base, two expansion bases and examples of I/O numbering.

DL430/440: supports a maximum of 3 expansion bases, and maximum of 320 input points and 320 output points (includes local base I/O)

DL450: supports a maximum of 3 expansion bases, and maximum of 1024 input points and 1024 output points (includes local base I/O)



Remote I/O Expansion

How to Add Remote I/O Channels

✓ ✓ ✓
430 440 450

Remote I/O is useful for a system that has a sufficient number of sensors and other field devices located a relative long distance away (up to 1000 meters, or 3050 feet) from the more central location of the CPU. The methods of adding remote I/O are:

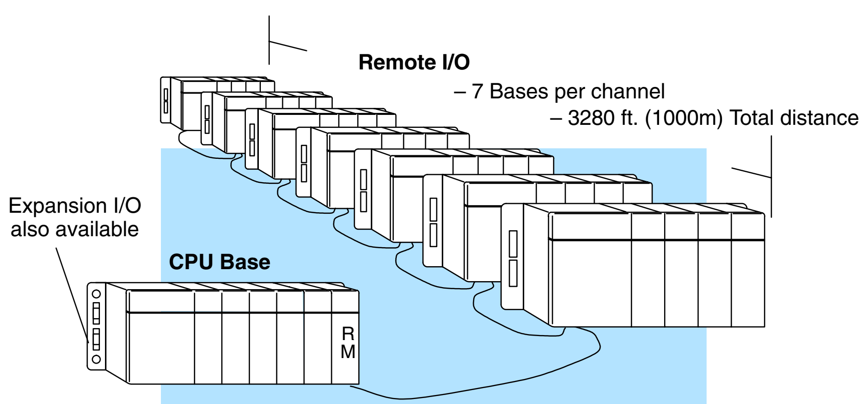
- **DL430 / DL440 CPUs**: Remote I/O requires a remote master module (D4–RM) to be installed in the local CPU base. The CPU updates the remote master, then the remote master handles all communication to and from the remote I/O base by communicating to the remote slave module (D4–RS) installed in each remote base.
- **DL450 CPU**: The CPU's comm port 3 features a built-in Remote I/O channel. You may also use one or two D4–RM remote masters in the local base as described above (can use either or both methods).

	DL430	DL440	DL450
Maximum number of Remote Masters supported in the local CPU base (1 channel per Remote Master)	2	2	2
CPU built-in Remote I/O channels	none	none	1
Maximum I/O points supported by each channel	512	512	512
Maximum Remote I/O points supported	512	1024	1536
Maximum number of remote I/O bases per channel	7	7	7

The use of Remote I/O does not limit the use of local expansion I/O discussed in the previous section. In fact, Remote I/O point numbering is assignable. Depending on the CPU scan time, remote I/O updates may be slower than local and expansion I/O, due to the serial communications involved.

Remote I/O points map into different CPU memory locations than local/local expansion I/O. So, the addition of remote I/O does not reduce the number of local I/O points. Refer to the DL405 Remote I/O manual for details on remote I/O configuration and numbering.

The following figure shows 1 CPU base, and one remote I/O channel with seven remote bases. If the CPU is a DL450, adding the first remote I/O channel does not require installing a remote master module (we use the CPU's built-in remote I/O channel on port 3).



Configuring the CPU's Remote I/O Channel

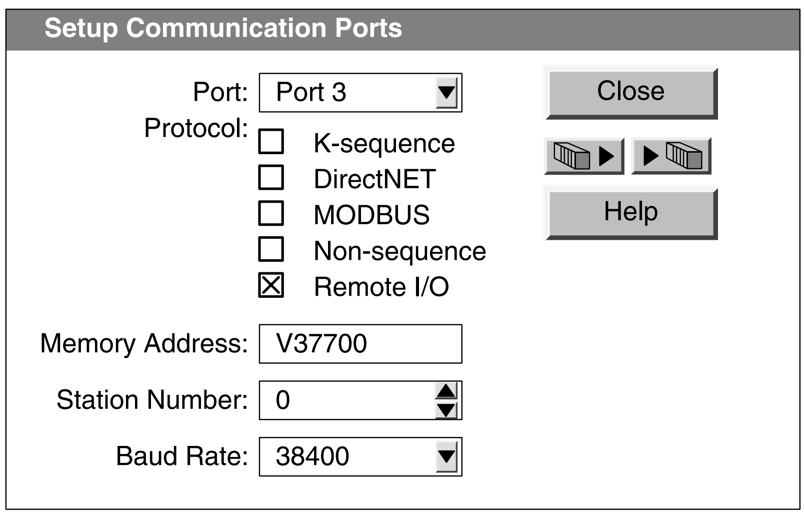
430 440 450

This section describes how to configure the DL450's built-in remote I/O channel. Additional information is in the Remote I/O manual, D4–REMIO–M, which you will need in configuring the Remote slave units on the network. You can use the D4–REMIO–M manual exclusively when using regular Remote Masters and Remote Slaves for remote I/O in any DL405 system.

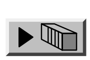
The DL450 CPU's built-in remote I/O channel has the same capability as a Remote Master module, the D4–RM. Specifically, it can communicate with up to seven remote bases containing a maximum of 512 I/O points, at a maximum distance of 1000 meters. If required, you can still use Remote Master modules in the local CPU base (512 I/O points on each channel), for a total of three channels providing 1536 total remote I/O points. First, we'll need to set up the Remote I/O communications.

You may recall from the CPU specifications in Chapter 3 that the DL450's Port 3 is capable of several protocols. To configure the port using the Handheld Programmer, use AUX 56 and follow the prompts, making the same choices as indicated below on this page. To configure the port in ***Direct***SOFT32, choose the PLC menu, then Setup, then Setup Secondary Comm Port...

- **Port:** From the port number list box at the top, choose "Port 3".
- **Protocol:** Click the check box to the left of "Remote I/O" (called "M–NET" on the HPP), and then you'll see the dialog box shown below.



- **Memory Address:** Choose a V-memory address to use as the starting location of a Remote I/O configuration table (V37700 is the default). This table is separate and independent from the table for any Remote Master(s) in the system.
- **Station Number:** Choose "0" as the station number, which makes the DL450 the master. Station numbers 1–7 are reserved for remote slaves.
- **Baud Rate:** The baud rates 19200 and 38400 baud are available. Choose 38400 initially as the remote I/O baud rate, and revert to 19200 baud if you experience data errors or noise problems on the link. Important: You must configure the baud rate on the Remote Slaves (via DIP switches) to match the baud rate selection for the CPU's Port 3.

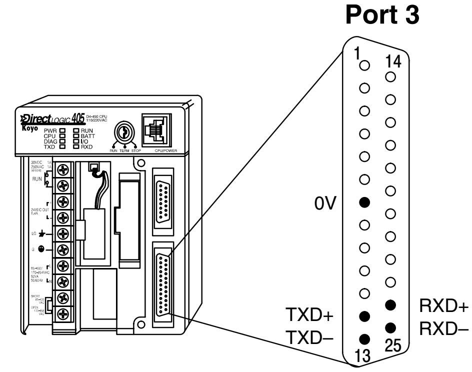


Then click the button indicated to send the Port 3 configuration to the CPU, and click Close.

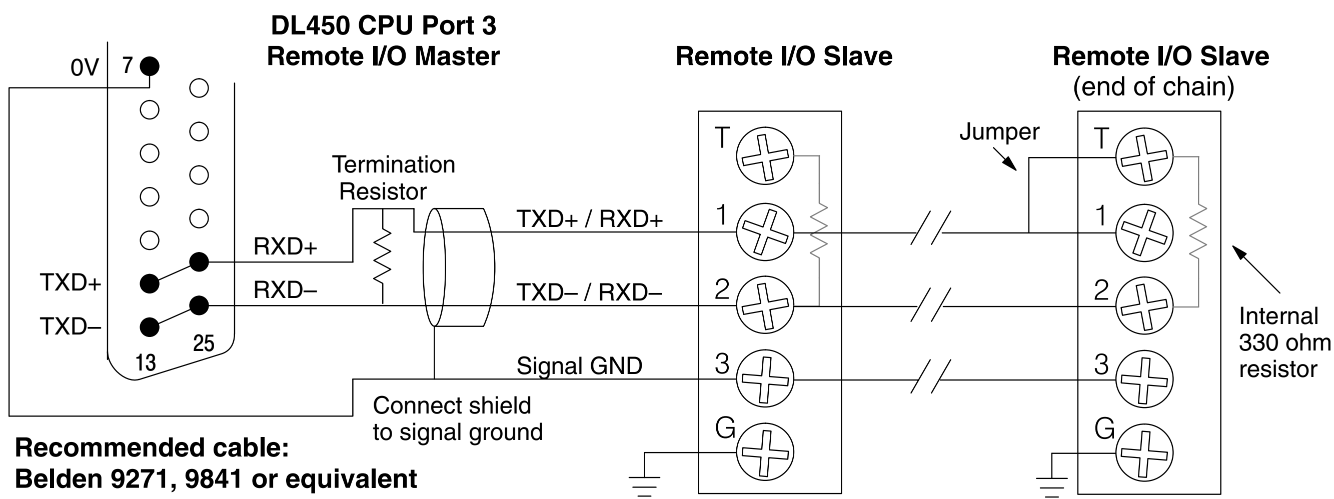
The next step is to make the connections between all devices on the Remote I/O link.

The location of the Port 3 on the DL450 is on the 25-pin connector , as pictured to the right. Remember that ports 1 and 3 are "logical" ports that share the 25-pin connector. Port 3 is an RS–422 non-isolated port. The pin assignments are:

- Pin 7 Signal GND
- Pin 12 TXD+
- Pin 13 TXD–
- Pin 24 RXD+
- Pin 25 RXD–



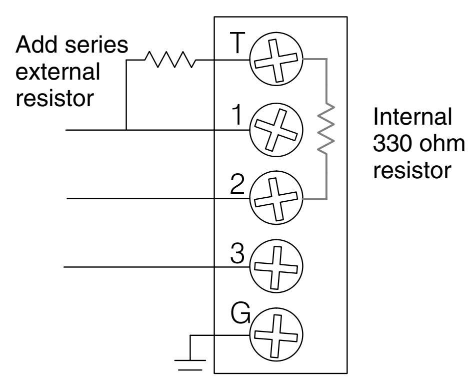
Now we are ready to discuss wiring the DL450 to the remote slaves on the remote base(s). The remote I/O link is a 3-wire, half-duplex type. Since Port 3 of the DL450 CPU is a 5-wire full duplex–capable port, we must jumper its transmit and receive lines together as shown below (converts it to 3-wire, half-duplex).



The twisted/shielded pair connects to the DL450 Port 3 as shown. Be sure to connect the cable shield wire to the signal ground connection. A termination resistor must be added externally to the CPU, as close as possible to the connector pins. Its purpose is to minimize electrical reflections that occur over long cables. Be sure to add the jumper at the last slave to connect the required internal termination resistor.

Ideally, the two termination resistors at the cables opposite ends and the cable's rated impedance should match. For cable impedances greater than 330 ohms, add a series resistor at the last slave as shown to the right. If less than 330 ohms, just parallel a matching resistance across the slave's pins 1 and 2 instead.

Remember to size the termination resistor at Port 3 to match. *The resistance values should be between 100 and 500 ohms.*



Configure Remote I/O Slaves

After configuring the DL450 CPU's Port 3 and wiring it to the remote slave(s), use the following checklist to complete the configuration of the remote slaves. Full instructions for these steps are in the Remote I/O manual.

- Set the baud rate DIP switches to match CPU's Port 3 setting.
- Select a station address for each slave, from 1 to 7. Each device on the remote link *must* have a unique station address. There can be only one master (address 0) on the remote link.

If you're familiar with configuring remote bases, then you'll recall the fixed table location in V-memory (V7404–V7477) to configure up to two remote I/O channels. However, we use a separate table for configuring the DL450 CPU's built-in remote I/O channel. You will still need the table at V7404 to configure any Remote Master modules.

Configuring the Remote I/O Table

The beginning of the configuration table for the built-in remote I/O channel is the memory address we selected in the Port 3 setup.

The table consists of blocks of four words which correspond to each slave in the system, as shown to the right. The first four table locations are reserved.

The CPU reads data from the table just after powerup, interpreting the four data words in each block with these meanings:

1. Starting address of slave's input data
2. Number of slave's input points
3. Starting address of outputs in slave
4. Number of slave's output points

The table is 32 words long. If your system has fewer than seven remote slave bases, then the remainder of the table must be filled with zeros. For example, a 3–slave system will have a remote configuration table containing 4 reserved words,12 words of data and 16 words of "0000".

A portion of the ladder program must configure this table (just once) at powerup. Use the LDA instruction as shown to the right, to load an address to place in the table. Use the regular LD constant to load the number of the slave's input or output points.

The D4–REMIO–M manual contains thorough examples for configuring the table at V7404, which you can adapt for this table as well. The following page give a shorter program example for one slave.

Memory Addr. Pointer	**37700**

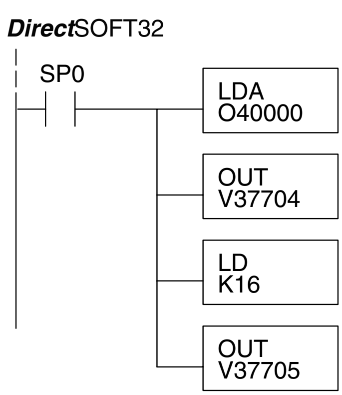
Remote I/O data

Reserved	**V37700**	xxxx
	V37701	xxxx
	V37702	xxxx
	V37703	xxxx

Slave 1 (or last slave)	V37704	xxxx
	V37705	xxxx
	V37706	xxxx
	V37707	xxxx

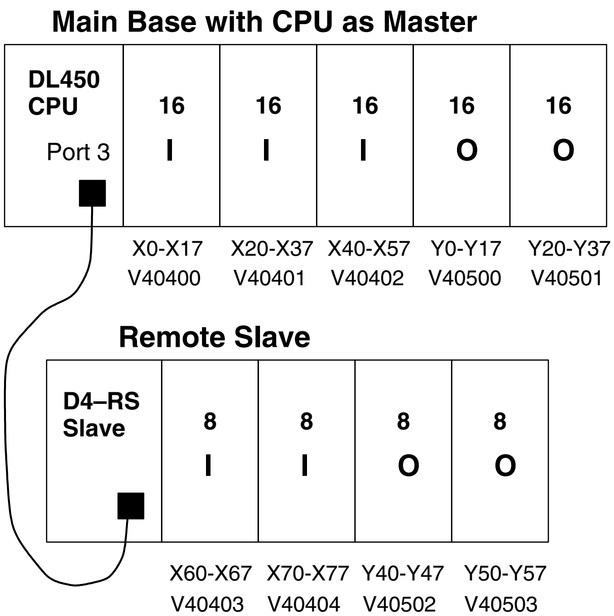
•
•
•

Slave 7 (or last slave)	V37734	0000
	V37735	0000
	V37736	0000
	V37737	0000



Consider the simple system featuring Remote I/O shown below. The DL450's built-in Remote I/O channel connects to one slave base, which we will assign a station address=1. The baud rates on the master and slave will be 38400 kB.

We can map the remote I/O points as any type of I/O point, simply by choosing the appropriate range of V-memory. Remember that on the DL450, you have both GX and GY data types available. Since we have plenty of standard I/O addresses available (X and Y), we will have the remote I/O points start at the next X and Y addresses after the main base points (X60 and Y40, respectively).



Remote Slave Worksheet

Remote Base Address 1 **(Choose 1–7)**

Slot Number	Module Name	INPUT		OUTPUT	
		Input Addr.	No. Inputs	Output Addr.	No.Outputs
0	08ND3S	X060	8		
1	08ND3S	X070	8		
2	08TD1			Y040	8
3	08TD1			Y050	8
4					
5					
6					
7					

Input Bit Start Address: X060 **V-Memory Address:V** 40403

Total Input Points 16

Output Bit Start Address: Y040 **V-Memory Address:V** 40502

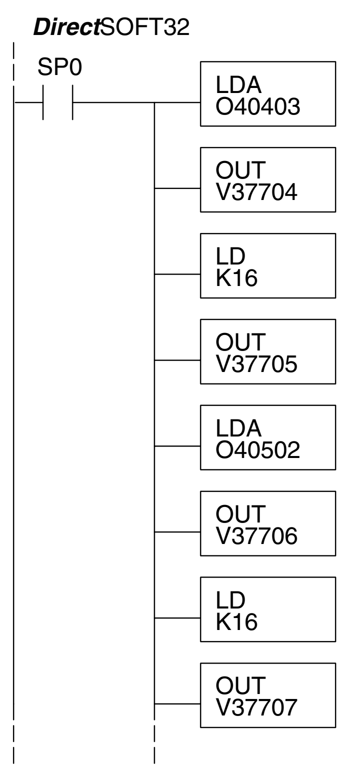
Total Output Points 16

Remote I/O Setup Program

Using the Remote Slave Worksheet shown above can help organize our system data in preparation for writing our ladder program (a blank full-page copy of this worksheet is in Appendix A of the D4–REMIO–M manual for your use and duplication). The four key parameters we need to place in our Remote I/O configuration table is in the lower right corner of the worksheet. You can determine the address values by using the memory map given at the end of Chapter 3, CPU Specifications and Operation.

The program segment required to transfer our worksheet results to the Remote I/O configuration table is shown to the right. Remember to use the LDA or LD instructions appropriately.

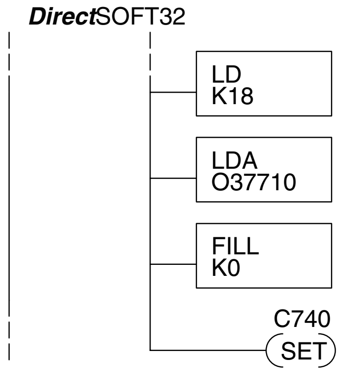
The next page covers the remainder of the required program to get this remote I/O link up and running.



System Design and Configuration

When configuring a Remote I/O channel for fewer than 7 slaves, we must fill the remainder of the table with zeros. This is necessary because the CPU will try to interpret any non-zero number as slave information.

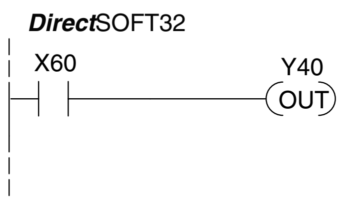
We continue our setup program from the previous page by adding a segment which fills the remainder of the table with zeros. The easiest way is the use the fill command as shown. The example to the right fills zeros for slave numbers 2–7, which do not exist in our example system (6 bases x 4 = 24 locations, = 18 hex).



On the last rung in the example program above, we set a special relay contact C740. This particular contact indicates to the CPU that the ladder program has just finished specifying a remote I/O system. At that moment the CPU begins remote I/O communications. Be sure to include this contact after any Remote I/O setup program.

Remote I/O Test Program

Now we can verify the remote I/O link and setup program operation. A simple quick check can be done with just one rung of ladder, shown to the right. It connects the first input of the remote base with the first output. After placing the PLC in RUN mode, we can go to the remote base and activate its first input. Then its first output should turn on.



SLICE I/O Expansion

✓ ✓ ✓
430 440 450

Slice I/O is similar to remote I/O since it also requires a special module in the local CPU base called a Slice Master (D4–SM) to control the Slice Slave(s). The Slice Slave (D4–SS–xxx) is a block of fixed I/O and a communication port in one package.

	DL430	DL440	DL450
Maximum number of Slice Masters supported in the local CPU base (number of channels)	2	2	2
Maximum number of Slice Slaves supported by 1 Slice Master	15	15	15
Maximum number of Slice Slaves	30	30	30

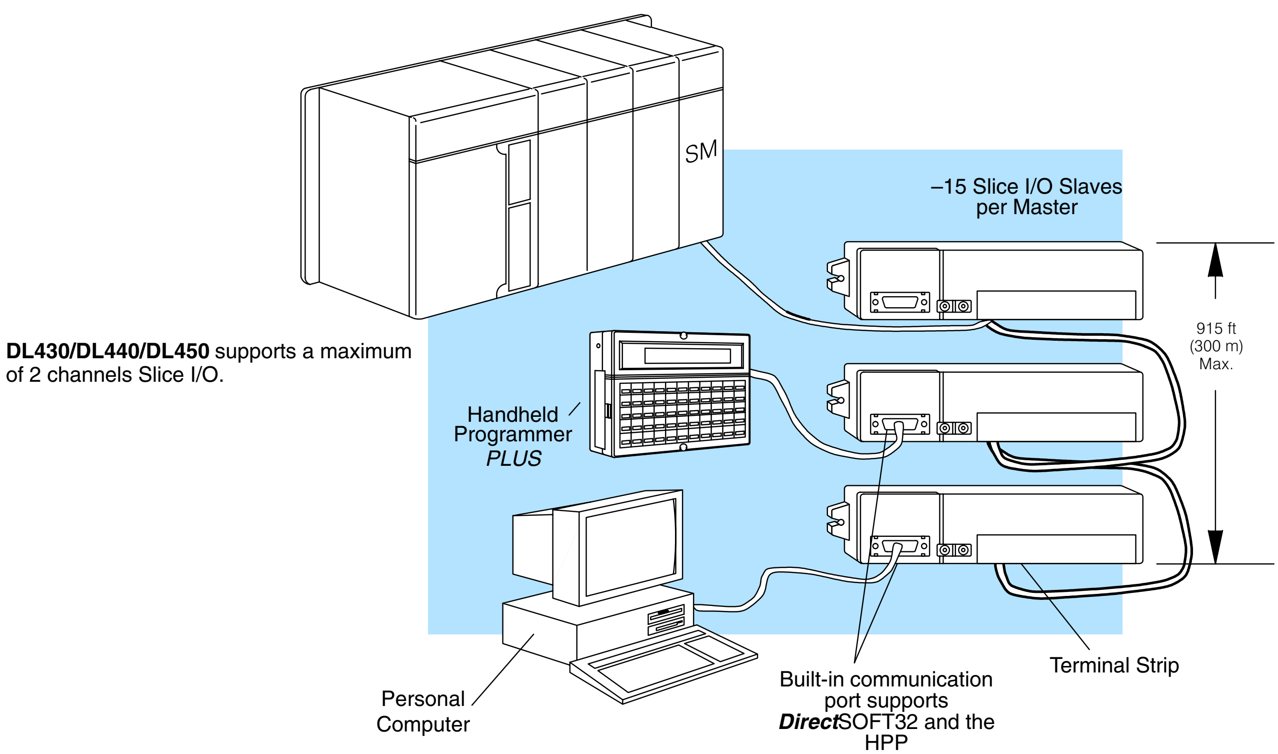
The maximum distance between a slice master and Slice I/O is 915 feet (300 m). Due to the added communication time it is possible that Slice I/O may not be updated every CPU scan. This along with other specifications regarding slice I/O is covered in the DL405 Slice I/O manual.

Slice I/O Memory Allocation

Memory allocation for Slice I/O is similar to remote I/O since Slice I/O points do not use the same memory locations as local/local expansion I/O. Refer to the DL405 Slice I/O manual for details on Slice I/O configuration and numbering.

Slice I/O Example

The following figure shows one CPU base, three Slice slaves and programming with a handheld programmer or ***Direct***SOFT32.



System Design and Configuration

Network Connections to MODBUS® and *Direct*Net

Configuring the CPU's Comm Ports

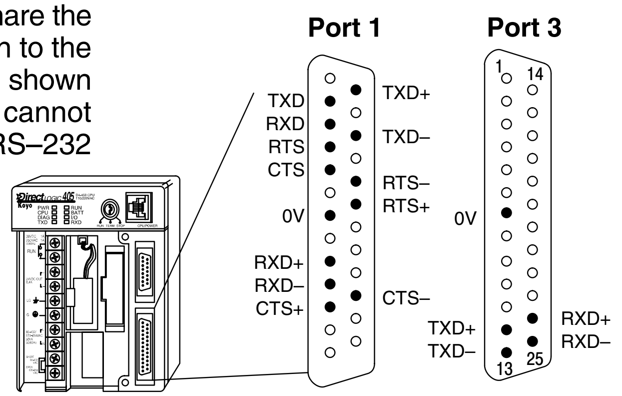
✓ ✓ ✓
430 440 450

This section describes how to configure the CPU's built-in networking ports. for either MODBUS or ***Direct***NET. This will allow you to connect the DL405 PLC system directly to MODBUS networks using the RTU protocol, or to other devices on a ***Direct***NET network. MODBUS hosts system on the network must be capable of issuing the MODBUS commands to read or write the appropriate data. For details on the MODBUS protocol, please refer to the Gould MODBUS Protocol reference Guide (P1–MBUS–300 Rev. B). In the event a more recent version is available, check with your MODBUS supplier before ordering the documentation. For more details on ***Direct***NET, order our ***Direct***NET manual, part number DA–DNET–M.

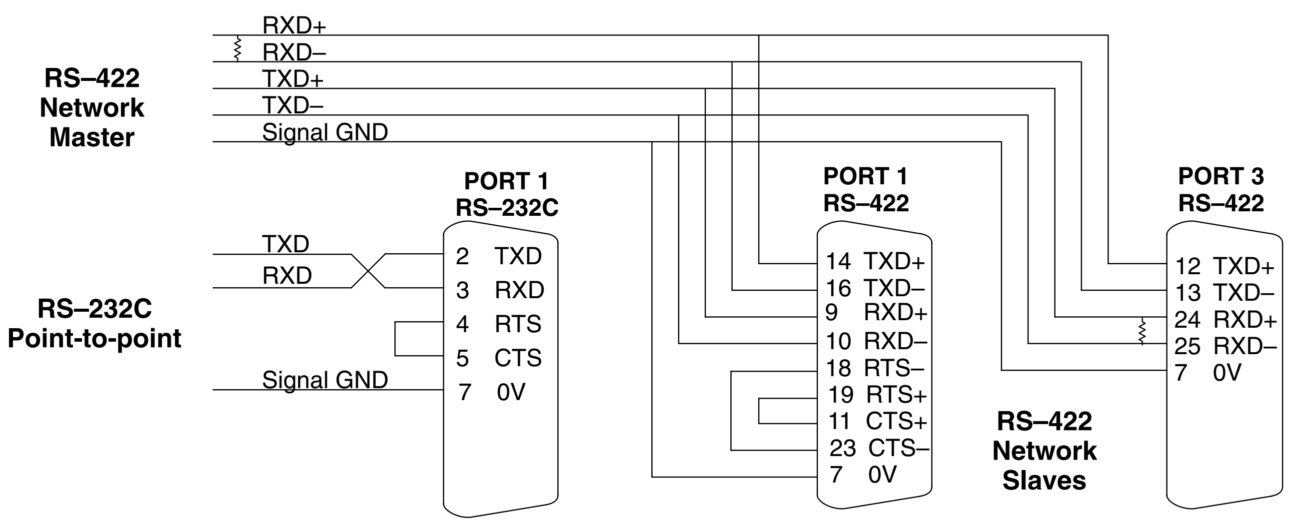
The DL430 and DL440 can be DirectNET slaves on port 1. Both the DL450's Port 1 and Port 3 can operate as master or slave for both MODBUS and ***Direct***NET. Port 1 has RS–232 and RS–422 signal levels available on separate pins, and Port 3 (DL450) uses RS–422 signal levels.

Ports 1 and Port 3 on the DL450 share the 25-pin D-shell connector, as shown to the right. Connect one or both ports as shown below. Note that you cannot simultaneously use Port 1's RS–232 signals and its RS–422 signals.

	Port 1	Port 3
DL430 and DL440	***Direct***NET, slave only	N/A
DL450	***Direct***NET or MODBUS, master/slave	***Direct***NET or MODBUS, master/slave



You will need to determine whether the network connection is a 3-wire RS–232 type, or a 5-wire RS–422 type. Normally, we use RS–232 signals for shorter distances (15 meters max), for communications between just two devices. Use RS–422 signals for longer distances (1000 meters max.), and for multi-drop networks (from 2 to 248 devices). Be sure to use termination resistors at the both ends of RS–422 network wiring, matching the impedance rating of the cable (between 100 and 500 ohms).

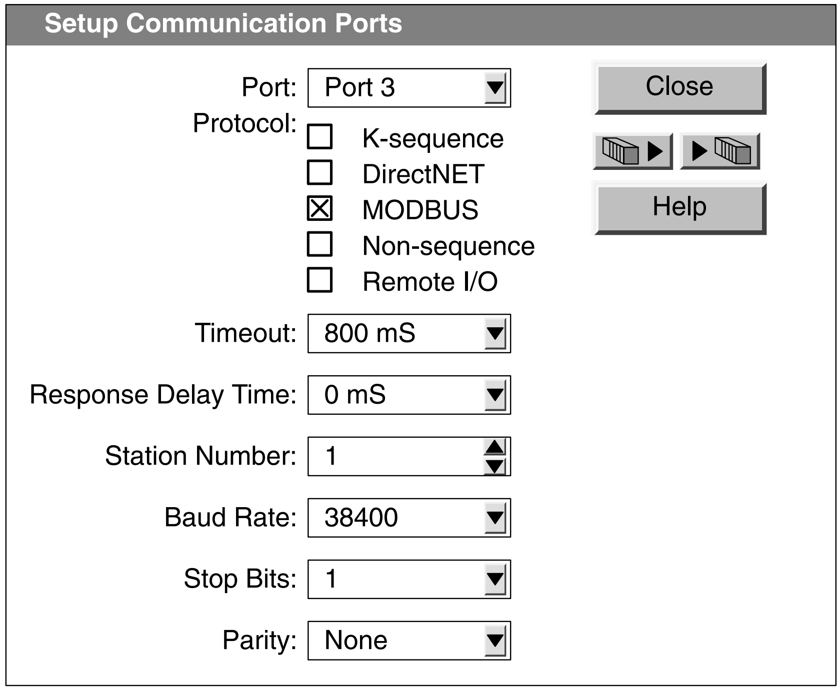


MODBUS Port Configuration

✗ 430 ✗ 440 ✓ 450

In ***Direct***SOFT32, choose the PLC menu, then Setup, then Secondary Comm Port.

- **Port:** From the port number list box at the top, choose Port 1 or 3.
- **Protocol:** Click the check box to the left of MODBUS (use AUX 56 on the HPP, andselect MBUS), and then you'll see the dialog box below.



- **Timeout:** amount of time the port will wait after it sends a message to get a response before logging an error.
- **Response Delay Time:** the amount of time the port waits to send a message after it's ready to send. For port 1, it activates the RTS line before it begins transmitting (assuming CTS is already active). The port will not transmit if the CTS input is false.
- **Station Number:** For making the CPU port a MODBUS® master, choose "1". The possible range for MODBUS slave numbers is from 1 to 247, but the DL450 network instructions will access only slaves 1 to 90. Each slave must have a unique number. At powerup, the port is automatically a slave, unless and until the DL450 executes ladder logic network instructions which use the port as a master. Thereafter, the port reverts back to slave mode until ladder logic uses the port again.
- **Baud Rate:** The available baud rates include 300, 600, 900, 2400, 4800, 9600, 19200, and 38400 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. Refer to the appropriate product manual for details.
- **Stop Bits:** Choose 1 or 2 stop bits for use in the protocol.
- **Parity:** Choose none, even, or odd parity for error checking.

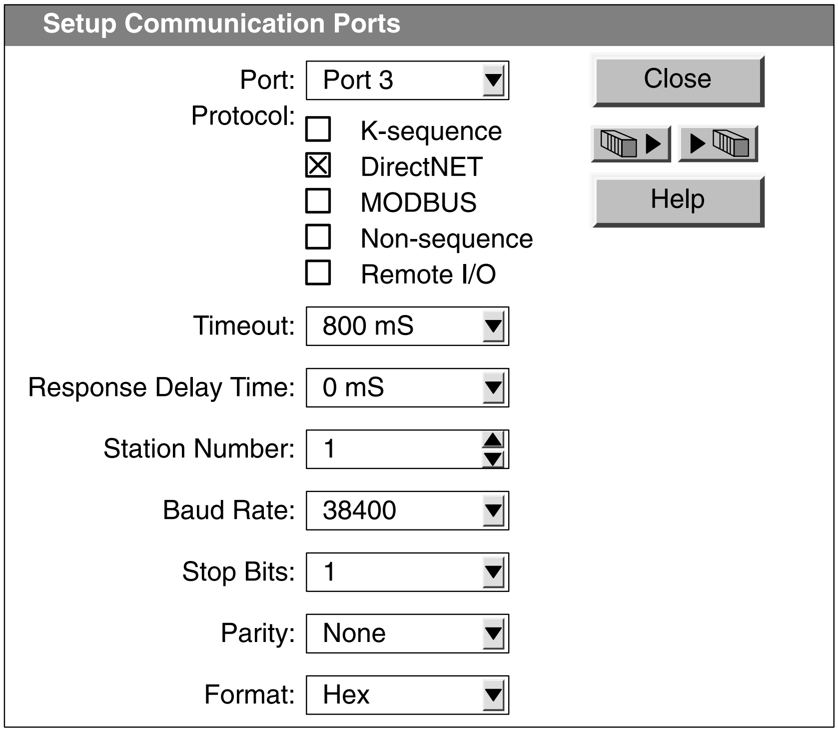
Then click the button indicated to send the Port configuration to the CPU, and click Close.

*Direct*NET Port Configuration

✓ ✓ ✓
430 440 450

In ***Direct***SOFT32, choose the PLC menu, then Setup, then Secondary Comm Port.

- **Port:** From the port number list box, choose Port 1 or 3 (DL450 only).
- **Protocol:** Click the check box to the left of DirectNET (use AUX 56 on the HPP, then select DNET), and then you'll see the dialog box below.



- **Timeout:** amount of time the port will wait after it sends a message to get a response before logging an error.
- **Response Delay Time:** the amount of time the port waits to send a message after it's ready to send. For port 1, it activates the RTS line before it begins transmitting (assuming CTS is already active). The port will not transmit if the CTS input is false.
- **Station Number:** For making the CPU port a ***Direct***NET master, choose "1". The allowable range for ***DIrect***NET slaves is from 1 to 90 (each slave must have a unique number). At powerup, the port is automatically a slave, unless and until the DL450 executes ladder logic instructions which attempt to use the port as a master. Thereafter, the port reverts back to slave mode until ladder logic uses the port again.
- **Baud Rate:** The available baud rates include 300, 600, 900, 2400, 4800, 9600, 19200, and 38400 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 between hex or ASCII formats.

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

Network Slave Operation

✓ ✓ ✓
430 440 450

This section describes how other devices on a network can communicate with a CPU port that you have configured as a ***Direct***NETslave or MODBUS slave (DL450). A MODBUS host must use the MODBUS RTU protocol to communicate with the DL450 as a slave. The host software must send a MODBUS function code and MODBUS address to specify a PLC memory location the DL450 comprehends. The ***Direct***NET host just uses normal I/O addresses to access any DL405 CPU and system. No CPU ladder logic is required to support either MODBUS slave or ***Direct***NET slave operation.

MODBUS Function Codes Supported

✗ ✗ ✓
430 440 450

The MODBUS function code determines whether the access is a read or a write, and whether to access a single data point or a group of them. The DL450 supports the MODBUS function codes described below.

MODBUS Function Code	Function	DL405 Data Types Available
01	Read a group of coils	Y, CR, T, CT, GY
02	Read a group of inputs	X, SP, GX
05 (slave only)	Set / Reset a single coil	Y, CR, T, CT
15	Set / Reset a group of coils	Y, CR, T, CT
03, 04	Read a value from one or more registers	V
06 (slave only)	Write a value into a single register	V
16	Write a value into a group of registers	V

MODBUS Data Types Supported

The memory types in a DL405 system include X input, Y output, C control relay, V memory data registers, etc. MODBUS uses differently named data types. So, you will need to determine which MODBUS data type corresponds to any desired PLC memory location by using the cross-reference table below.

DL450 Memory Type	Quantity (Decimal)	PLC Range (Octal)	Corresponding MODBUS Data Type	Rx Function Code
Inputs (X)	1024	X0 – X1777	Input	02
Global Inputs (GX)	1536	GX0 – GX2777	Input	02
Special Relays (SP)	512	SP0 – SP137 SP320 – SP717	Input	02
Outputs (Y)	1024	Y0 – Y1777	Coil	01
Global Outputs (GY)	1536	GY0 – GY2777	Coil	01
Control Relays (CR)	2048	C0 – C3777	Coil	01
Timer Contacts (T)	256	T0 – T377	Coil	01
Counter Contacts (CT)	256	CT0 – CT377	Coil	01
Stage Status Bits (S)	1024	S0 – S1777	Coil	01
Timer Current Values (V)	256	V0 – V377	Input Register	03
Counter Current Value (V)	256	V1000 – V1377	Input Register	03
V Memory, user data (V)	3072 12288	V1400 – V7377 V10000 – V37777	Holding Register	03
V Memory, system (V)	320	V700 – V777 V7400 – V7777	Holding Register	03

Determining the MODBUS Address

There are typically two ways that most host software 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 Requires the Data Type and Address...

Many host software packages 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 various MODBUS data types were presented earlier, but they have been included again in the following table.

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, CR, S, T, C (contacts)
- Word – V, Timer current value, Counter current value

In either case, you basically just convert the PLC octal address to decimal and add the appropriate MODBUS address (if required). The table below shows the exact equation used for each group of data.

DL450 Memory Type	QTY (Dec.)	PLC Range (Octal)	MODBUS Address Range (Decimal)	MODBUS Data Type
For Discrete Data Types	**Convert PLC Addr. to Dec.**	**+**	**Start of Range +**	**Data Type**
Inputs (X)	1024	X0 – X1777	2048 – 3071	Input
Special Relays (SP)	512	SP0 – SP137 SP320 – SP717	3072 – 3167 3280 – 3535	Input
Outputs (Y)	1024	Y0 – Y1777	2048 – 3071	Coil
Control Relays (CR)	2048	C0 – C3777	3072 – 5119	Coil
Timer Contacts (T)	256	T0 – T377	6144 – 6399	Coil
Counter Contacts (CT)	256	CT0 – CT377	6400 – 6655	Coil
Stage Status Bits (S)	1024	S0 – S1777	5120 – 6143	Coil
Global Inputs (GX) *	1536	GX0 – GX2777	0 – 1535	Input
Global Outputs (GY) *	1536	GY0 – GY2777	0 – 1535	Coil
For Word Data Types	**Convert PLC Addr. to Dec.**		**+**	**Data Type**
Timer Current Values (V)	256	V0 – V377	0 – 255	Input Register
Counter Current Values (V)	256	V1000 – V1377	512 – 767	Input Register
V Memory, user data (V)	3072 12288	V1400 – V7377 V10000 – V37777	768 – 3839 4096 – 16383	Holding Register
V Memory, system (V)	320	V700 – V777 V7400 – V7777	448 – 768 3480 – 3735	Holding Register

* **Note:** The total of GX and GY global I/O points cannot exceed 1536 points.

System Design and Configuration

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 Address (Dec.) + Data Type

V2100 = 1088 decimal

1088 + Hold. Reg. = **Holding Reg. 1088**

V Memory, user data (V)	3072 12288	V1400 – V7377 V10000–V37777	768 – 3839 4096 – 16383	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 (2048).
4. Use the MODBUS data type from the table.

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

Y20 = 16 decimal

16 + 2048 + Coil = **Coil 2064**

Outputs (Y)	1024	Y0 – Y1777	2048 – 3071	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 Address (Dec.) + Data Type

T10 = 8 decimal

8 + Input Reg. = **Input Reg. 8**

Timer Current Values (V)	256	V0 – V377	0 – 255	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 Type

C54 = 44 decimal

44 + 3072 + Coil = **Coil 3116**

Control Relays (CR)	2048	C0 – C3777	3072 – 5119	Coil

If Your MODBUS Host Software Requires an Address ONLY

Some host software does 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's still fairly simple. Basically, MODBUS also separates the data types by address ranges as well. So this means an address alone can actually describe the type of data and location. This is often referred to as "adding the offset". One important thing to remember here is that two different addressing modes may be available in your host software package. These are:

- 484 Mode
- 584/984 Mode

We recommend that you use the 584/984 addressing mode if your host software allows you to choose. This is because the 584/984 mode allows access to a higher number of memory locations within each data type. If your software only supports 484 mode, then there may be some PLC memory locations that will be unavailable. 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 (contacts), C (contacts)
- Word – V, Timer current value, Counter current value

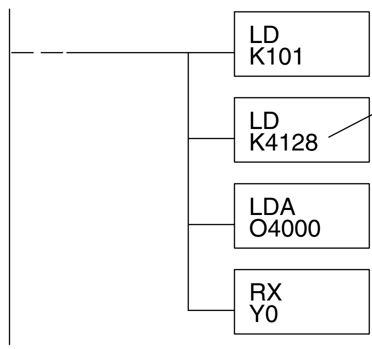
In either case, you basically just convert the PLC octal address to decimal and add the appropriate MODBUS addresses (as required). The table below shows the exact equation used for each group of data.

Discrete Data Types				
Memory Type	**PLC Range (Octal)**	**Address (484 Mode)**	**Address (584/984 Mode)**	**Data Type**
Global Inputs (GX)	GX0 – GX1746	1001 – 1999	10001 – 10999	Input
	GX1747 – GX3777	——	11000 – 12048	Input
Inputs (X)	X0 – X1777	——	12049 – 13072	Input
Special Relays (SP)	SP0 – SP777	——	13073 – 13584	Input
Global Outputs (GY)	GY0 – GY3777	1 – 2048	1 – 2048	Output
Outputs (Y)	Y0 – Y1777	2049 – 3072	2049 – 3072	Output
Control Relays (CR)	C0 – C3777	3073 – 5120	3073 – 5120	Output
Timer Contacts (T)	T0 – T377	6145 – 6400	6145 – 6400	Output
Counter Contacts (CT)	CT0 – CT377	6401 – 6656	6401 – 6656	Output
Stage Status Bits (S)	S0 – S1777	5121 – 6144	5121 – 6144	Output

Word Data Types			
Registers	PLC Range (Octal)	Input/Holding (484 Mode)*	Input/Holding (585/984 Mode)*
V Memory (Timers)	V0 – V377	3001/4001	30001/40001
V Memory (Counters)	V1000 – V1177	3513/4513	30513/40513
V Memory (Data Words)	V1200 – V1377	3641/4641	30641/40641
V Memory (Data Words)	V1400 – V1746	3769/4769	30769/40769
V Memory (Data Words)	V1747 – V1777	——	31000/41000
V Memory (Data Words)	V2000 – V7377	——	41025
V Memory (Data Words)	V10000 – V17777	——	44097

*MODBUS: Function 04 (New feature)

The DL450 will support **function 04** read input register **(Address 30001)**. To use function 04, put the number '4' into the most significant position (4xxx). Four digits must be entered for the instruction to work properly with this mode.



The Maximum constant possible is 4128. This is due to the 128 maximum number of Bytes that the RX/WX instruction can allow. The value of 4 in the most significant position of the word will cause the RX to use function 04 (30001 range).

1. Refer to the Memory Mapping section of this manual for the correct memory mapping size. Some of the addresses shown above might not pertain to your CPU.

2. For an automated MODBUS/Koyo address conversion utility, download the file **modbus_conversion.xls** from our website, **www.automationdirect.com**.

Example 1: V2100 584/984 Mode

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 Address (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	Hold Reg.

Example 2: Y20 584/984 Mode

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: T10 Current Value 484 Mode

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. Add the MODBUS starting address for the mode (3001).

PLC Address (Dec.) + Mode Address

TA10 = 8 decimal

8 + 3001 = **3009**

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	Hold Reg.

Example 4: C54 584/984 Mode

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

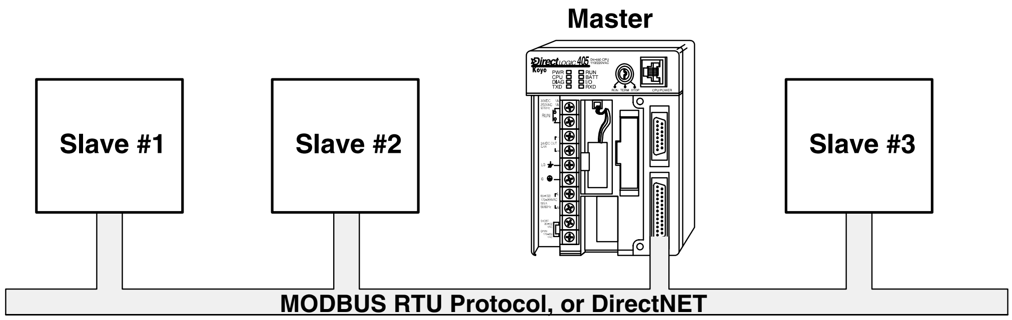
44 + 3072 + 1 = **3117**

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

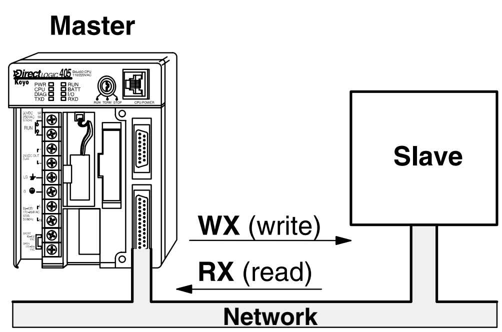
Network Master Operation

430 440 450

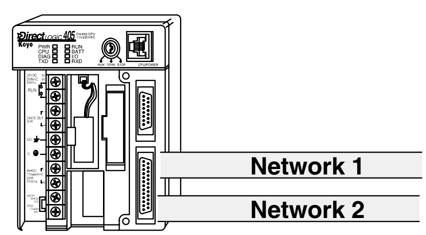
This section describes how the DL450 can communicate on a MODBUS or ***Direct***NET network as a master. For MODBUS networks, it uses the MODBUS RTU protocol, which must be interpreted by all the slaves on the network. Since MODBUS and ***Direct***NET are master / slave networks, the master station must initiate requests for network data transfers. This section teaches you how to design the required ladder logic for network master operation.



When using the DL450 CPU as the master station, 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 will 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, which goes to the port.



It's possible to use both Port 1 and Port 3 for either MODBUS or ***Direct***NET, and to use either or both as masters. You must tell the WX and RX instructions the intended port for each communications transaction.

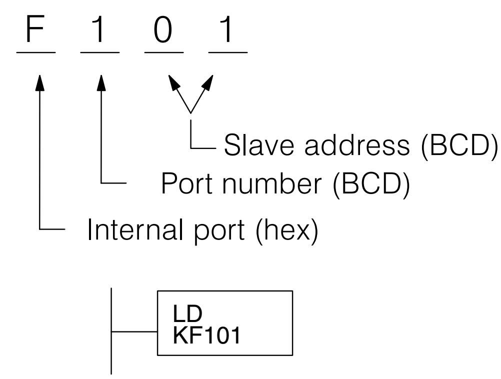


To summarize, the RLL instructions identify the following items.

1. Port number on the master (Port 1 or 3), and the slave station address. (LD instruction)
2. Amount of data (in bytes) you want to transfer. (LD instruction)
3. Area of memory to be used by the master. (LDA instruction)
4. Area of CPU V-memory to be used in communication with the slave, and whether it is a write or read operation. (WX or RX instruction)
5. Interlocks for communication timing for multiple WX and RX routines.

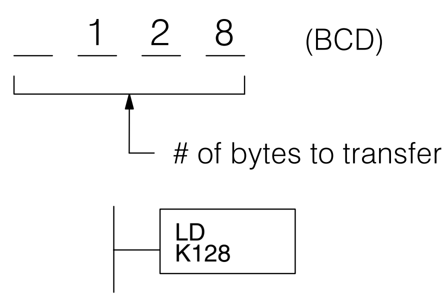
Step 1: Identify Master Port # and Slave #

The first Load (LD) instruction identifies the communications port number on the network master (DL450) and the address of the slave station. This instruction can address up to 90 MODBUS slaves, or 90 ***Direct***NET slaves. The format of the word is shown to the right. The “F” in the upper nibble tells the CPU the port is internal to the CPU (and not in a slot in the base). The second nibble indicates the port number, 1 or 3. The lower byte contains the slave address number in BCD (01 to 90).



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.



The number of bytes specified also depends on the type of data you want to obtain. For example, the DL405 Input points can be accessed by V-memory locations or as X input locations. However, if you only want X0 – X27, you'll have to use the X input data type because the V-memory locations can only be accessed in 2-byte increments. The following table shows the byte ranges for the various types of ***Direct***LOGIC™ products.

DL 205 / 405 Memory	Bits per unit	Bytes
V memory	16	2
T / C current value	16	2
Inputs (X, GX, SP)	8	1
Outputs (Y, C, Stage, T/C bits)	8	1
Scratch Pad Memory	8	1
Diagnostic Status	8	1

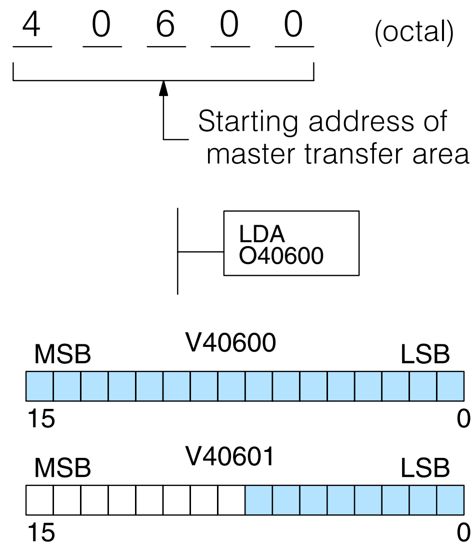
DL305 Memory	Bits per unit	Bytes
Data registers	8	1
T / C accumulator	16	2
I/O, internal relays, shift register bits, T/C bits, stage bits	1	1
Scratch Pad Memory	8	2
Diagnostic Status(5 word R/W)	16	10

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 DL450 CPU sends the number of bytes previously specified from its memory area beginning at the LDA address specified.

For an RX instruction, the DL450 CPU reads the number of bytes previously specified from the slave, placing the received data into its memory area beginning at the LDA address specified.

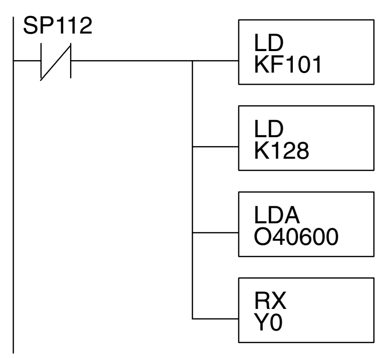


NOTE: Since V memory words are always 16 bits, you may not always use the whole word. For example, if you only specify 3 bytes and you are reading Y outputs from the slave, you will only get 24 bits of data. In this case, only the 8 least significant bits of the last word location will be modified. The 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.

The RX instruction reads data from the slave starting at the address specified. The WX instruction writes data to the slave starting at the address specified.



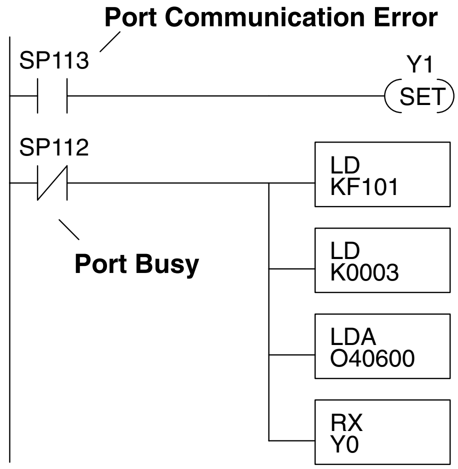
- ***Direct*NET slaves** – specify the same address in the WX and RX instruction as the slave's native I/O address
- MODBUS DL405 or DL205 slaves – specify the same address in the WX and RX instruction as the slave's native I/O address
- MODBUS 305 slaves – use the following table to convert DL305 addresses to MODBUS addresses

DL305 Series CPU Memory Type–to–DL405 Series CPU Memory					
PLC Memory type	305 base address	405 base addr.	PLC Memory Type	305 base address	405 base addr.
TMR/CNT Current Values	R600	V0	TMR/CNT Status Bits	CT600	GY600
I/O Points	IO 000	GY0	Control Relays	CR160	GY160
Data Registers	R401, R400	V100	Shift Registers	SR400	GY400
Stage Status Bits (D3–330P only)	S0	GY200			

Communications from a Ladder Program

In some applications, the DL450 CPU as a network master will communicate only periodically to slaves(s) on the network. However, most applications will probably want to make a "continuous" update of memory areas from a slave to the master.

This normally means starting the task on each PLC SCAN. However, a single WX or RX network communication will probably last longer than one PLC scan time. *And we must wait before executing another RX or WX until the port has finished transmitting the previous WX or RX data.*



Each port which can be a master has two Special Relay contacts associated with it (see Appendix D for comm port special relays).One indicates "Port busy", and the other indicates "Port Communication Error". The example above shows the use of these contacts for a network master that only reads a device (RX). The Port Busy contact ensures one network transaction finishes before we begin another.

Use of the communication error SP relay is optional. If used, be sure to place it at the beginning of the communication routines, because a comm error relay is always reset (turned off) whenever an RX or WX instruction using the same port executes.

Multiple Read and Write Interlocks

If you're 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, C0 is set. When the port has finished the communication task, the second routine is executed and C0 is reset.

If you're using RLL*PLUS* Stage Programing, you can just put each routine in a separate program stage to ensure proper execution. In most cases, RLL*PLUS* is a much more efficient way to create an automation program.

The ***Direct***NET manual provides a master / slave example with both RLL and Stage program descriptions (they are easily adapted for use with MODBUS).