



# Errata Sheet

This Errata Sheet contains corrections or changes made after the publication of this manual.

**Product Family:** DL305 **Date:** August 2018  
**Manual Number:** D3-350-M  
**Revision and Date:** 2nd Edition, Rev. D; March 2010

## Changes to Chapter 3. CPU Specifications and Operations

Page 3-6. Using Battery Backup; Enabling the Battery Backup

The ladder example shown to enable the backup battery is incorrect. Use the following example instead. Note: This example assumes that the original content of V7633 was a 0 (zero).



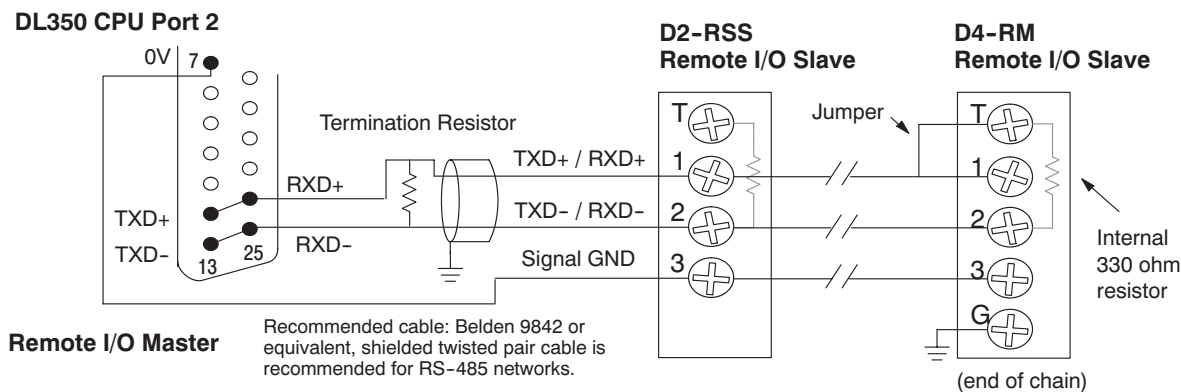
## Changes to Chapter 4. System Design and Configuration

Page 4-15. I/O Configurations with a 10 Slot Local CPU Base

The four drawings on this page to the left of the bases showing jumper switch SW2 are mislabeled. They should say "700" and "100 EXP", not "700 EXP" and "100" as shown.

Page 4-18. CPU Specifications; Remote I/O Expansion; Configuring the CPU's Remote I/O Channel

Replace the remote I/O wiring diagram shown with this one.



## Changes to Chapter 5. Standard RLL Instructions; Timer, Counter and Shift Register; Accumulating Timer (TMRA) and Accumulating Fast Timer (TMRAF)

Page 5-38. In the first paragraph, second sentence on this page, the maximum value for the TMRAF instruction reads 99999.99 (total of seven 9's). This is incorrect - it should be 999999.99 (total of eight 9's).



# Errata Sheet

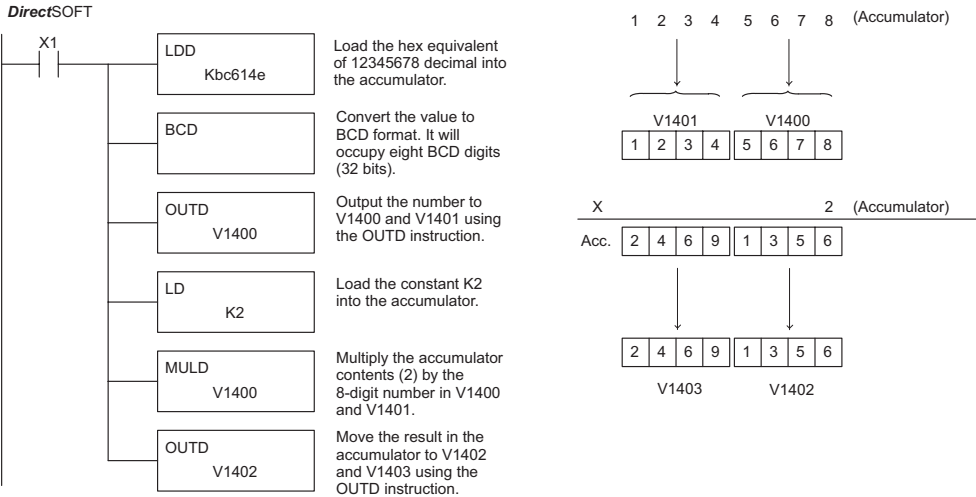
This Errata Sheet contains corrections or changes made after the publication of this manual.

## Changes to Chapter 5. Standard RLL Instructions; Accumulator Logical Instructions; Compare Real Number (CMPR)

Page 5-76. In the ladder example, the contact reference is incorrect. It should be SP62 turning on output C1, not SP60.

## Changes to Chapter 5. Standard RLL Instructions; Math Instructions; Multiply Double (MULD)

Page 5-84. The ladder example shown is incorrect. Replace it with the following example.



### Handheld Programmer Keystrokes

\$	STR	→	B	1	ENT										
SHFT	L	D	D	→	PREV	SHFT	B	C	SHFT	G	B	E	SHFT	E	ENT
SHFT	B	C	D	ENT											
GX	SHFT	D	→	B	E	A	A	ENT							
SHFT	L	D	→	PREV	C	ENT									
SHFT	M	U	L	→	B	E	A	A	ENT						
GX	SHFT	D	→	B	E	A	C	ENT							

## Changes to Chapter 5. Standard RLL Instructions; Network Instructions

Please change Step 1 on pages 5-137 and 5-139 (RX & WX Commands) to read as follows:

Step 1: - Load the slave address (0-90 BCD) into the first byte and the slot number of the master DCM (0-7) into the second byte of the second level of the accumulator stack. When using Port 2 of the CPU, the formatting should be Kf1xx where xx is the slave address (0-90 BCD).



# Errata Sheet

**This Errata Sheet contains corrections or changes made after the publication of this manual.**

## Changes to Chapter 8. PID Loop Operation

For more recent and complete information on PID loop operation, refer to Chapter 8 in the DL06 user manual (manual number D0-06USER-M).

## Changes to Chapter 9. Maintenance and Troubleshooting

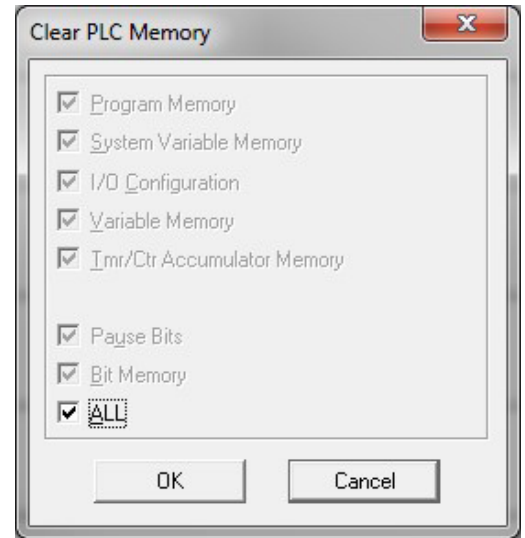
Page 9-25. Add the following to the end of this chapter (right after Regular Forcing with Direct Access):

### Reset the PLC to Factory Defaults

**NOTE:** Resetting to factory defaults will not clear any password stored in the PLC.

Resetting a DirectLogic PLC to Factory Defaults is a two-step process. Be sure to have a verified backup of your program using “Save Project to Disk” from the File menu before performing this procedure. Please be aware that the program as well as any settings will be erased and not all settings are stored in the project. In particular you will need to write down any settings for Secondary Communications Ports and manually set the ports up after resetting the PLC to factory defaults.

Step 1 – While connected to the PLC with DirectSoft, go to the PLC menu and select; “Clear PLC Memory”. Check the “ALL” box at the bottom of the list and press “OK”.



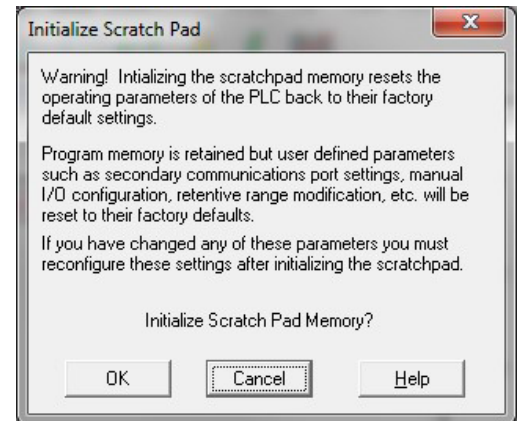
Step 2 – While connected with DirectSoft, go the PLC menu and then to the “Setup” submenu and select “Initialize Scratch Pad”. Press “OK”.

**NOTE:** All configurable communications ports will be reset to factory default state. If you are connected via Port 2 or another configurable port, you may be disconnected when this operation is complete.

**NOTE:** Retentive ranges will be reset to the factory settings.

**NOTE:** Manually addressed IO will be reset to factory default settings.

The PLC has now been reset to factory defaults and you can proceed to program the PLC.



# System Design and Configuration

---

## In This Chapter. . . .

- DL305 System Design Strategies
  - Module Placement
  - Calculating the Power Budget
  - Expansion I/O
  - Remote I/O
  - Network Connections to MODBUS and *DirectNET*
  - Network Slave Operation
  - Network Master Operation
-

## DL305 System Design Strategies

### I/O System Configurations

The DL350 CPU offers the following ways to add I/O to the system:

- **Local I/O** - consists of I/O modules located in the same base as the CPU.
- **Remote I/O** - consists of I/O modules located in bases which are serially connected to the bottom port on a DL350 CPU.
- **Expansion I/O** - consists of I/O modules located in expansion bases located close to the local base. Expansion cables connect them to the local CPU base's serial bus in a daisy-chain fashion.

A DL305 system can be developed using many different arrangements of these configurations. All I/O configurations use the standard complement of DL305 I/O modules and bases.

### Networking Configurations

The DL350 CPU offers the following way to add networking to the system:

- **DL350 Communications Port** - The DL350 CPU has a 25-Pin connector on Port 2 that provides a built-in RTU MODBUS connection.
- **MODBUS Master Module**- MODBUS master modules can be used in any slot for connecting as a master to a MODBUS network.
- **MODBUS Slave Module**- MODBUS slave modules can be used in any slot for connecting as a slave to a MODBUS network.

Module/Unit	Master	Slave
DL350 CPU	<i>DirectNET</i> MODBUS RTU	<i>DirectNET</i> K-Sequence MODBUS RTU

### Base Configurations

The DL305 system currently offers two types of bases. Both types come in 5, 8, or 10 slot configurations. All DL305 CPUs will work in either type of base. The xxxxx-1 bases are designed to compliment the features of the DL350 CPU, however all other DL305 CPUs will work in these bases. You can also mix the bases in a system. By mixing the bases or by installing the DL350 in a conventional base, you will lose some of the features of the CPU. The DL350 will revert back to 8-bit addressing and will virtually function like a DL340 CPU. This section will focus on the xxxxx-1 bases using the DL350 CPU. If you will be using the DL350 in a conventional base or if you are mixing bases in a system, refer to Appendix F for base, I/O, and module placement information.

The xxxxx-1 bases support a 8 bit parallel bus that allows the use of intelligent modules when using the DL350 CPU. The addressing scheme is simplified and also extends the number of I/O points you can use. You will have a bigger power budget to work with due to the increase in the power supply capacity to 2.0A.

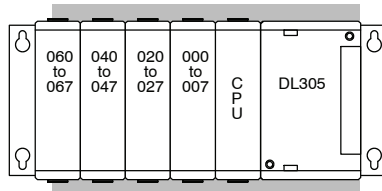
## Module Placement

### Slot Numbering

The DL305 bases each provide different numbers of slots for use with the I/O modules. You may notice the bases refer to 5-slot, 8-slot, etc. One of the slots is dedicated to the CPU, so you always have one less I/O slot. For example, you have four I/O slots with a 5-slot base. The I/O slots are numbered 0 - 3. The CPU slot always contains a CPU and is not numbered.

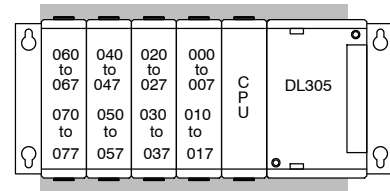
The examples below show the I/O numbering for a 5 slot local CPU base with 8 point I/O and a 5 slot local CPU base with 16 point I/O.

**5 Slot Base Using 8 Point I/O Modules**



**Slot Number: 3—2—1—0**

**5 Slot Base Using 16 Point I/O Modules**



**Slot Number: 3—2—1—0**

### I/O Module Placement Rules

There are some limitations that determine where you can place certain types of modules. Some modules require certain locations and may limit the number or placement of other modules. The table on pages 4-6 and 4-7 should clear up any gray areas in the explanation and you will probably find the configuration you intend to use in your installation.

In all of the configurations mentioned the number of slots from the CPU that are to be used can roll over into an expansion base if necessary. For example if a rule states a module must reside in one of the six slots adjacent to the CPU, and the system configuration is comprised of two 5 slot bases, slots 1 and 2 of the expansion base are valid locations.

The following table provides the general placement rules for the DL305 components.

Module	Restriction
CPU	The CPU must reside in the first slot of the local CPU base. The first slot is the closest slot to the power supply.
16 Point I/O Modules	Any slot.
Analog Modules	Any slot.
ASCII Basic Modules	Any slot.
High Speed Counter	The D3-350 CPU does not support a high speed counter module.

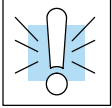
### I/O Configuration

I/O addresses use octal numbering, starting in the slot next to the CPU. The addresses are assigned in groups of 16 for each slot regardless of what module is in the slot. The discrete input and output modules can be mixed in any order, but there may be restrictions placed on some specialty modules.

## Calculating the Power Budget

### Managing your Power Resource

When you determine the types and quantity of I/O modules you will be using in the DL305 system it is important to remember there is a limited amount of power available from the power supply. We have provided a chart to help you easily see the amount of power available with each base. 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.



**WARNING:** It is *extremely* important to calculate the power budget. 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.

### Base Power Specifications

This chart shows the amount of current available for the three voltages supplied on the new xxxx-1 bases. Use these currents when calculating the power budget for your system.

Bases	5V Power Supplied in Amps	9V Power Supplied in Amps	24V Power Supplied in Amps	Auxiliary 24 VDC Output at Base Terminal
D3-05B-1	1.0A (50°C) 0.7A (60°C)	2.0	0.6	100mA max
D3-05BDC	1.4A (50°C) 0.7A (60°C)	0.8	0.6	None
D3-08B-1	1.0A (50°C) 0.7A (60°C)	2.0	0.6	100mA max
D3-10B-1	1.0A (50°C) 0.7A (60°C)	2.0	0.6	100mA max
D3-10BDC	1.4A (50°C) 0.7A (60°C)	1.7	0.6	None

**I/O Points Required for Each Module** Each type of module requires a certain number of I/O points. This is also true for the specialty modules, such as analog, counter interface, etc. The table on page 4-5 lists the number and type of I/O points required for each module.

**Module Power Requirements** The next three pages show the amount of maximum current required for each of the DL305 modules. The column labeled “External Power Source Required” is for module operation and is not for field wiring. Use these currents when calculating the power budget for your system. If 24 VDC is needed for external devices, the 24 VDC (100mA maximum) output at the base terminal strip may be used as long as the power budget is not exceeded.

	I/O Points Required	5V Power Required (mA)	9V Power Required in (A)	24V Power Required (mA)	External Power Source Required
<b>CPUs</b>					
D3-350		500	20	0	None
<b>DC Input Modules</b>					
D3-08ND2	8	0	10	112	None
D3-16ND2-1	16	0	25	224	None
D3-16ND2-2	16	0	24	209	None
D3-16ND2F	16	0	25	224	None
F3-16ND3F	16	0	148	68	None
<b>AC Input Modules</b>					
D3-08NA-1	8	0	10	0	None
D3-08NA-2	8	0	10	0	None
D3-16NA	16	0	100	0	None
<b>AC/DC Input Modules</b>					
D3-08NE3	8	0	10	0	None
D3-16NE3	16	0	130	0	None
<b>DC Output Modules</b>					
D3-08TD1	8	0	20	24	None
D3-08TD2	8	0	30	0	None
D3-16TD1-1	16	0	40	96	None
D3-16TD1-2	16	0	40	96	None
D3-16TD2	16	0	180	0	None
<b>AC Output Modules</b>					
D3-04TAS	8	0	12	0	None
F3-08TAS	8	0	80	0	None
F3-08TAS-1	8	0	25	0	None
D3-08TA-1	8	0	96	0	None
D3-08TA-2	8	0	160	0	None
F3-16TA-2	16	0	250	0	None
D3-16TA-2	16	0	400	0	None



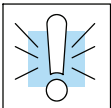
	I/O Point Required	5V Power Required in mA	9V Power Required in mA	24V Power Required in mA	External Power Source Required
<b>Relay Output Modules</b>					
D3-08TR	8	0	360	0	None
F3-08TRS-1	8	0	296	0	None
F3-08TRS-2	8	0	296	0	None
D3-16TR	16	0	480	0	None
<b>Analog</b>					
D3-04AD	16	0	55	0	24VDC @ 65mA max
F3-04ADS	16	0	183	50	None
F3-08AD	16	0	25	37	None
F3-08TEMP	16	0	25	37	None
F3-08THM-n	16	0	50	34	None
F3-16AD	16	0	33	47	None
D3-02DA	16	0	80	0	24VDC @ 170mA max
F3-04DA-1	16	0	144	108	None
F3-04DA-2	16	0	144	108	None
F3-04DAS	16	0	154	145	None
<b>Communications and Networking</b>					
FA-UNICON	0	0	0	0	(24 VDC or 5 VDC) @ 100mA
<b>ASCII BASIC Modules</b>					
F3-AB128-R	16	0	205	0	None
F3-AB128-T	16	0	205	0	None
F3-AB128	16	0	90	0	None
F3-AB64	16	0	90	0	None
<b>Specialty Modules</b>					
D3-08SIM	8	0	10	112	None
D3-HSC	16	0	70	0	None
<b>Programming</b>					
D2-HPP		200	50	0	Optional

### Power Budget Calculation Example

The following example shows how to calculate the power budget for the DL305 system.

Base #	Module Type	5 VDC (mA)	9 VDC (mA)	Auxiliary Power Source 24 VDC Output (mA)
<u>0</u>				
<b>Available Base Power</b>	D3-05B	<b>1000</b>	<b>2000</b>	<b>600</b>
<b>CPU Slot</b>	D3-350	+500	+ 120	
<b>Slot 0</b>	D3-16NE3	+ 0	+ 130	+ 0
<b>Slot 1</b>	D3-16NE3	+ 0	+ 130	+ 0
<b>Slot 2</b>	F3-16TA-2	+ 0	+ 250	+ 0
<b>Slot 3</b>	F3-16TA-2	+ 0	+ 250	+ 0
<b>Slot 4</b>				
<b>Slot 5</b>				+ 0
<b>Slot 6</b>				+ 0
<b>Slot 7</b>				+ 0
<b>Other</b>				
Handheld Prog	D2-HPP	+ 200	+ 200	+ 0
<b>Total Power Required</b>		<b>700</b>	<b>1080</b>	<b>0</b>
<b>Remaining Power Available</b>		1000-700=300	2000-1080=920	600 - 0 = <b>600</b>

1. Use the power budget table to fill in the power requirements for all the system components. First, enter the amount of power supplied by the base. Next, list the requirements for the CPU, any I/O modules, and any other devices, such as the Handheld Programmer or the DV-1000 operator interface. Remember, even though the Handheld or the DV-1000 are not installed in the base, they still obtain their power from the system. Also, make sure you obtain any *external* power requirements, such as the 24VDC power required by the analog modules.
2. Add the current columns starting with Slot 0 and put the total in the row labeled "**Total power required**".
3. Subtract the row labeled "**Total power required**" from the row labeled "**Available Base Power**". Place the difference in the row labeled "**Remaining Power Available**".
4. If "**Total Power Required**" is greater than the power available from the base, the power budget will be exceeded. It will be unsafe to use this configuration and you will need to restructure your I/O configuration.



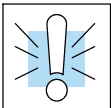
**WARNING:** It is *extremely* important to calculate the power budget. If you exceed the power budget, the system may operate in an unpredictable manner which may result in a risk of personal injury or equipment damage.

### Power Budget Calculation Worksheet

This blank chart is provided for you to copy and use in your power budget calculations.

Base #	Module Type	5 VDC (mA)	9 VDC (mA)	Auxiliary Power Source 24 VDC Output (mA)
0				
<b>Available Base Power</b>				
<b>CPU Slot</b>				
Slot 0				
Slot 1				
Slot 2				
Slot 3				
Slot 4				
Slot 5				
Slot 6				
Slot 7				
Other				
Handheld Prog	D2-HPP			
<b>Total Power Required</b>				
<b>Remaining Power Available</b>				

1. Use the power budget table to fill in the power requirements for all the system components. First, enter the amount of power supplied by the base. Next, list the requirements for the CPU, any I/O modules, and any other devices, such as the Handheld Programmer or the DV-1000 operator interface. Remember, even though the Handheld or the DV-1000 are not installed in the base, they still obtain their power from the system. Also, make sure you obtain any *external* power requirements, such as the 24VDC power required by the analog modules.
2. Add the current columns starting with Slot 0 and put the total in the row labeled "**Total power required**".
3. Subtract the row labeled "**Total power required**" from the row labeled "**Available Base Power**". Place the difference in the row labeled "**Remaining Power Available**".
4. If "**Total Power Required**" is greater than the power available from the base, the power budget will be exceeded. It will be unsafe to use this configuration and you will need to restructure your I/O configuration.



**WARNING:** It is *extremely* important to calculate the power budget. 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.

# Local I/O Expansion

## Base Uses Table

It is helpful to understand how you can use the various DL305 bases in your control system. The following table shows how the bases can be used.

Base Part #	Number of Slots	Can Be Used As A Local CPU Base	Can Be Used As An Expansion Base
D3-05B-1	5	Yes	Yes
D3-05BDC-1	5	Yes	Yes
D3-08B-1	8	Yes	Yes
D3-08BDC-1	8	Yes	Yes
D3-10B-1	10	Yes	Yes
D3-10BDC-1	10	Yes	Yes

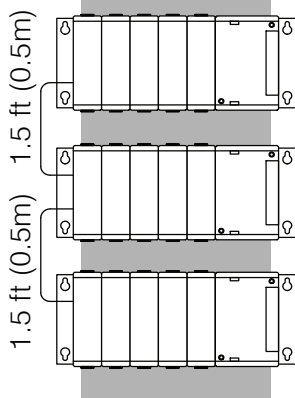
## Local/Expansion Connectivity

The configurations below show the valid combinations of local and expansion bases using the DL350 CPU.

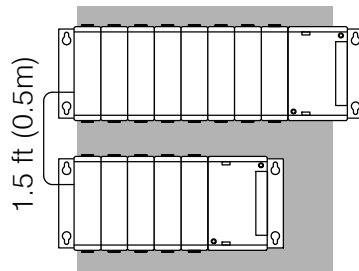


**NOTE:** You should use one of the configurations listed below when designing an expansion system. If you use a configuration not listed below the system will not function properly.

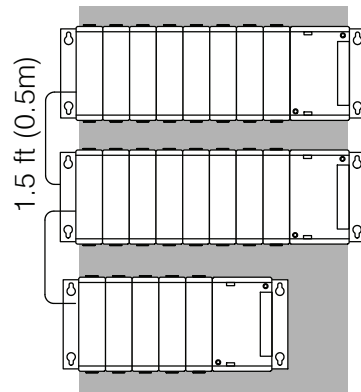
5 slot local CPU base with a maximum of two 5 slot expansion bases



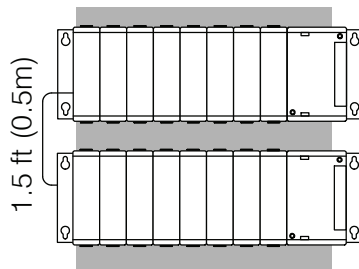
8 slot local CPU base with a 5 slot expansion base



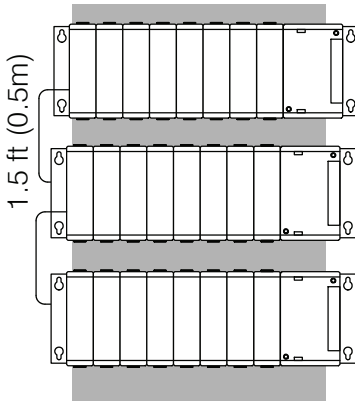
8 slot local CPU base with a 8 slot and 5 slot expansion base



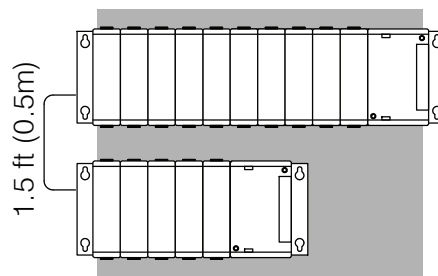
8 slot local CPU base with a 8 slot expansion base



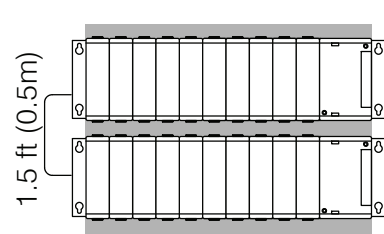
8 slot local CPU base with two 8 slot expansion bases



10 slot local CPU base with a 5 slot expansion base



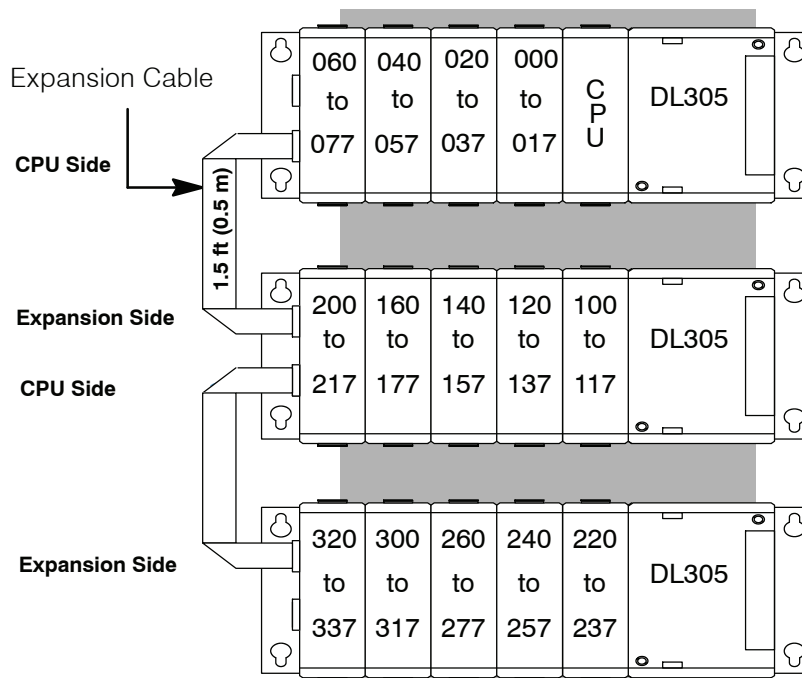
10 slot local CPU base with a 10 slot expansion base



### Connecting Expansion Bases

The local CPU base is connected to the expansion base using a 1.5 ft. cable (D3-EXCBL). The base must be connected as shown in the diagram below.

The top expansion connector on the base is the input from a previous base. The bottom expansion connector on the base is the output to an expansion base. The expansion cable is marked with "CPU Side" and "Expansion Side". The "CPU Side" of the cable is connected to the bottom port of the base and the "Expansion Side" of the cable is connected to the top port of the next base.

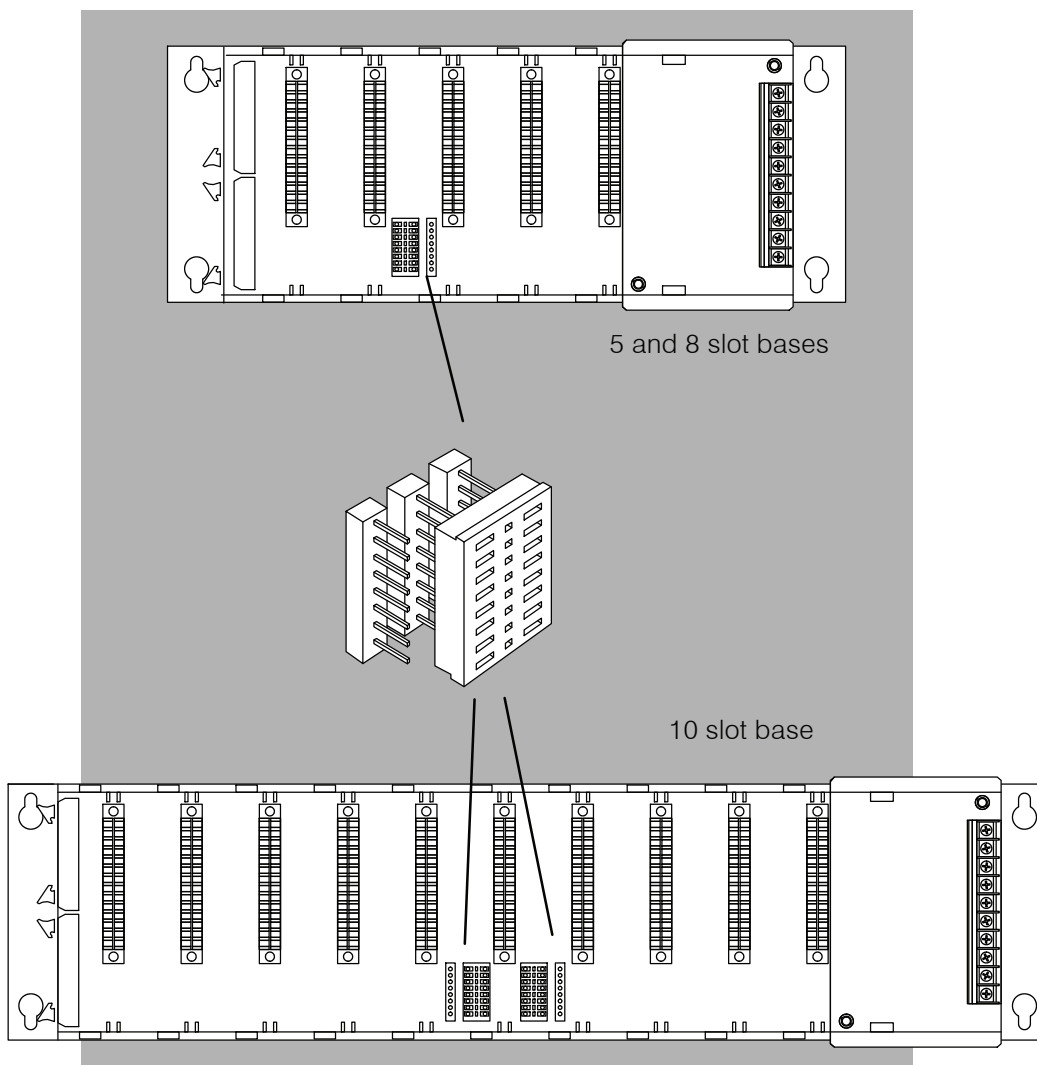


Note: Avoid placing the expansion cable in the same wiring tray as the I/O and power source wiring.

## Setting the Base Switches

### Jumper Switch

The 5, and 8 slot bases have a jumper switch between slot 3 and 4 used to set the base to local CPU base or expansion base. The 10 slot base has two jumpers, one is located between slots 4 and 5 and the other is located between slot 5 and 6. The second switch sets I/O addressing ranges for the DL330/340 CPUs. This switch should always be bridged to the right hand position for the DL350 CPU.



## I/O Configurations with a 5 Slot Local CPU Base

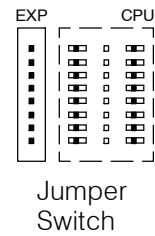
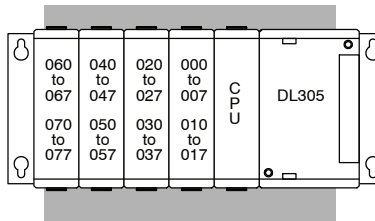
### Switch settings

The 5 slot base has a jumper switch on the inside of the base between slots 3 and 4 which allows you to select:

Type of Base	Switch Position
Local CPU	right side bridged
First Expansion	left side bridged
Last Expansion	right side bridged

### 5 Slot Base

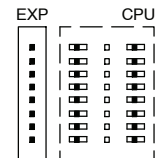
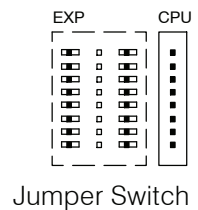
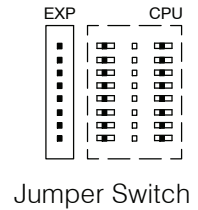
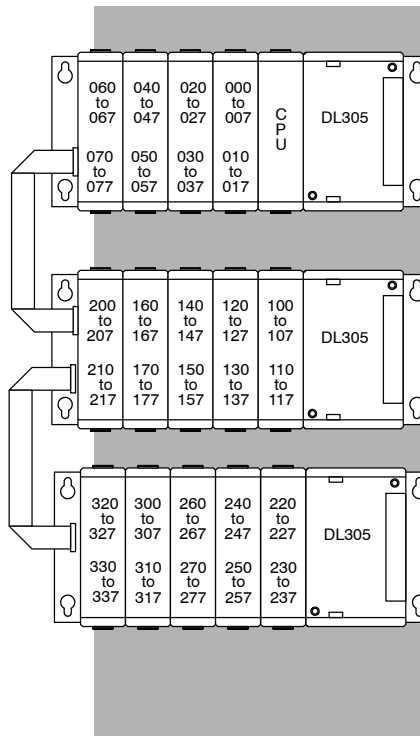
**Total I/O:**  
 8 pt. modules 32  
 16 pt. modules 64



### 5 Slot Base and up to two 5 Slot Expansion Bases

**Total I/O:**  
**1 Expansion base**  
 8 pt. modules - 72  
 16 pt. modules - 144

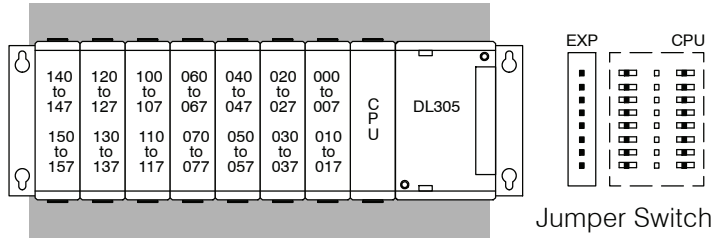
**2 Expansion Bases**  
 8 pt. modules - 112  
 16 pt. modules - 224



# I/O Configurations with an 8 Slot Local CPU Base

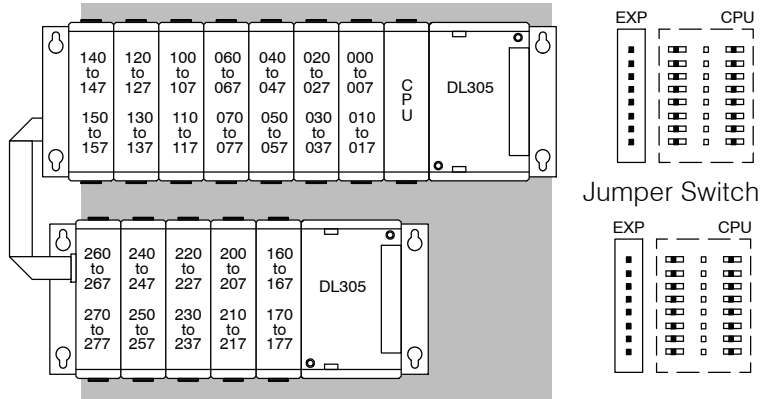
## 8 Slot Base

**Total I/O:**  
 8 pt. modules - 56  
 16 pt. modules - 112



## 8 Slot Base and 5 Slot Expansion Base

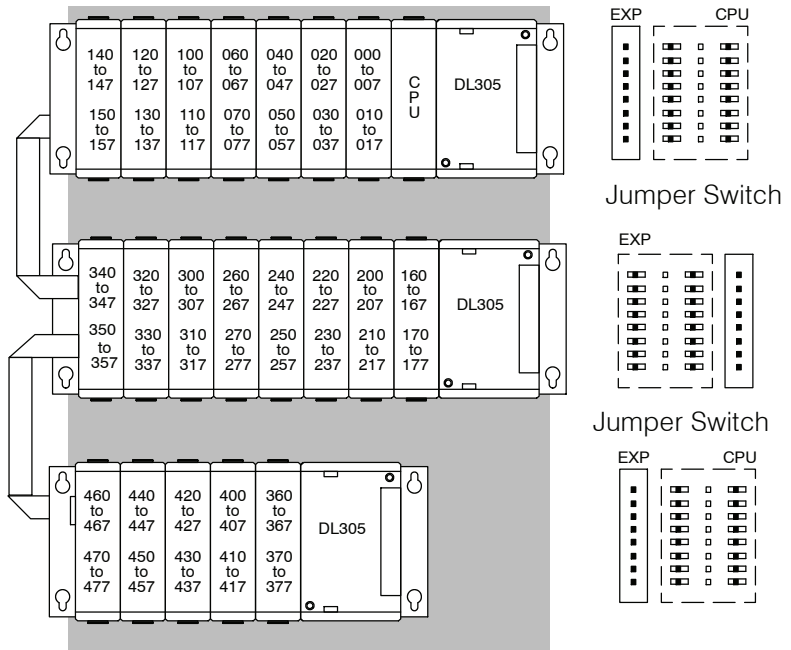
**Total I/O:**  
 8 pt modules - 96  
 16 pt modules - 192



## 8 Slot Base and One 8 slot and one 5 slot Expansion Bases

**Total I/O:**  
**1 Expansion Base**  
 8 pt modules - 120  
 16 pt modules - 240

**2 Expansion Bases**  
**1 - 8 slot 1 - 5 slot**  
 8 pt. modules - 160  
 16 pt. modules - 320

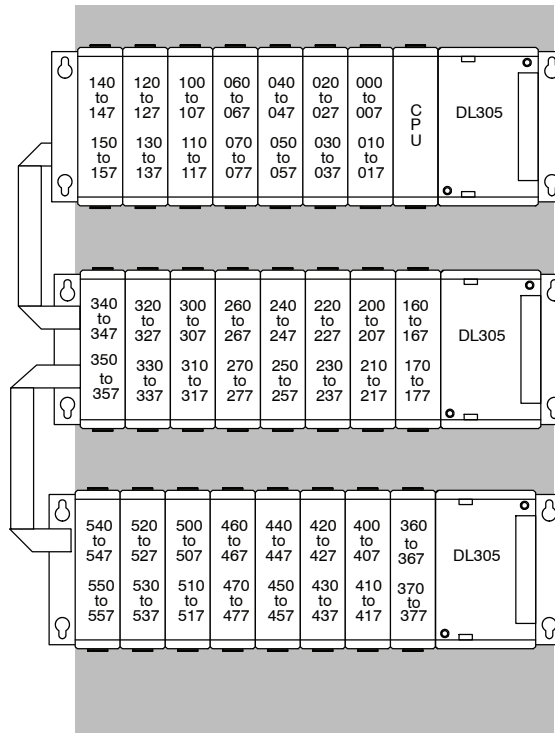




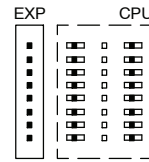
### 8 Slot Base and two 8 slot Expansion Bases

**Total I/O:**

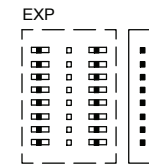
**2 Expansion Bases**  
**2 - 8 slot**  
 8 pt. modules - 184  
 16 pt. modules - 368



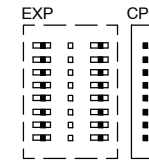
Jumper Switch



Jumper Switch



Jumper Switch

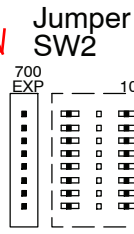


# I/O Configurations with a 10 Slot Local CPU Base

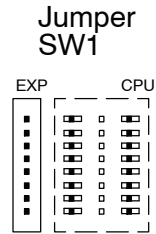
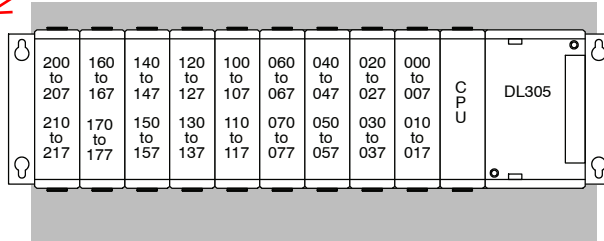
Should say 700, not 700 EXP

## 10 Slot Base

Total I/O:  
8 pt. modules - 72  
16 pt. modules - 144



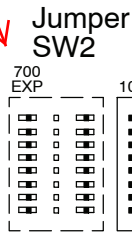
Should say 100 EXP, not 100



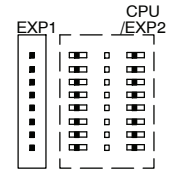
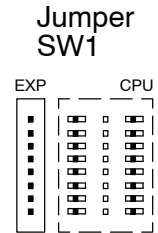
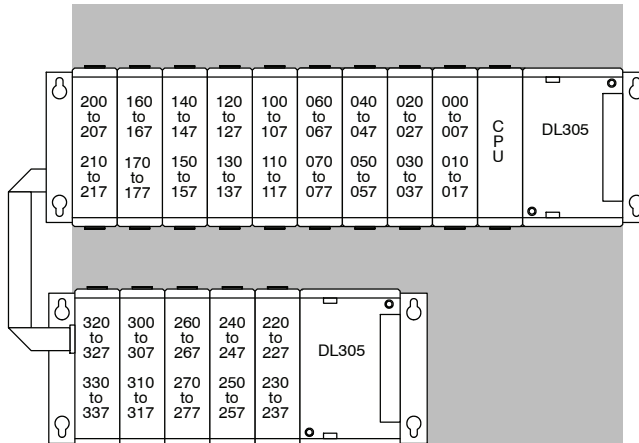
700, not 700 EXP

## 10 Slot Base and 5 Slot Expansion Base with 16 Point I/O

Total I/O:  
8 pt. modules - 112  
16 pt. modules - 224



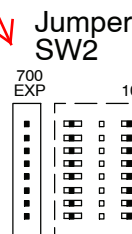
100 EXP



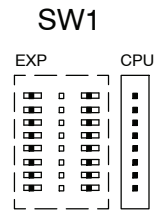
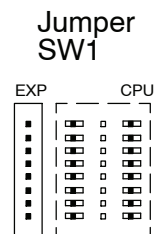
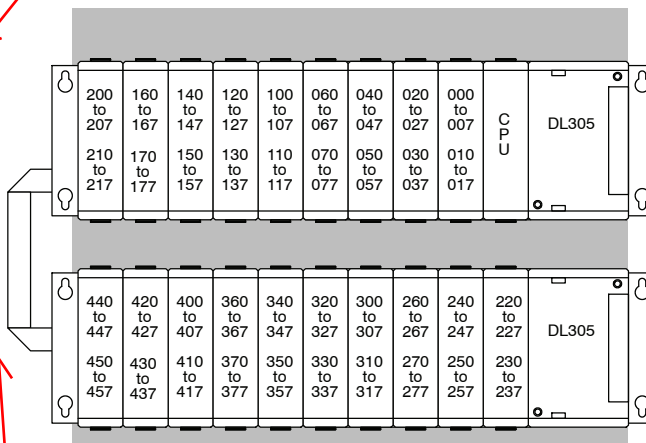
700, not 700 EXP

## 10 Slot Base and 10 Slot Expansion Base with 16 Point I/O

Total I/O:  
8 pt. modules - 152  
16 pt. modules - 304



100 EXP



700, not 700 EXP

100 EXP

System Design and Configuration

## Remote I/O Expansion

### How to Add Remote I/O Channels

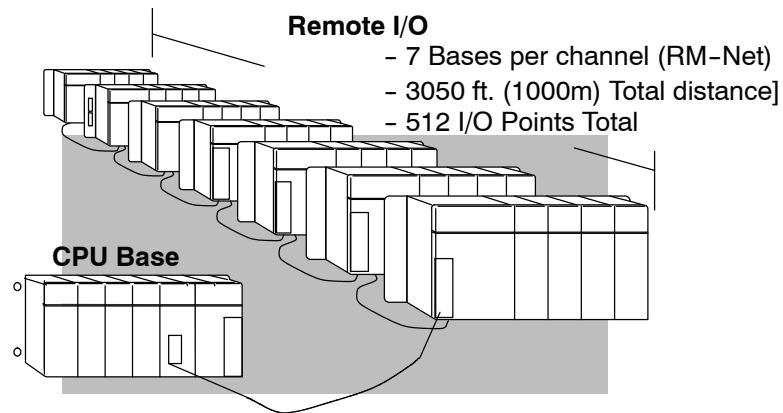
Remote I/O is useful for a system that has a sufficient number of sensors and other field devices located a relative long distance away (up to 1000 meters, or 3050 feet) from the more central location of the CPU. The DL350 supports a built-in Remote master, however the DL305 family does not have any Remote I/O modules. Therefore, you must use a DL205 or DL405 base for the slave channels. The methods of adding remote I/O are:

- **DL350 CPU:** The CPU's comm port 2 features a built-in Remote I/O channel.

	DL350
Maximum number of Remote Masters supported in the local CPU base (1 channel per Remote Master)	1
CPU built-in Remote I/O channels	1
Maximum I/O points supported by each channel	512
Maximum Remote I/O points supported	512
Maximum number of remote I/O bases per channel (RM-NET)	7

Remote I/O points map into different CPU memory locations, therefore it does not reduce the number of local I/O points. Refer to the DL205 Remote I/O manual for details on remote I/O configuration and numbering. Configuring the built-in remote I/O channel is described in the following section.

The following figure shows 1 CPU base with seven remote bases. The remote bases can be DL205 or DL405 bases.



**DL350 CPU Only  
RM-Net**

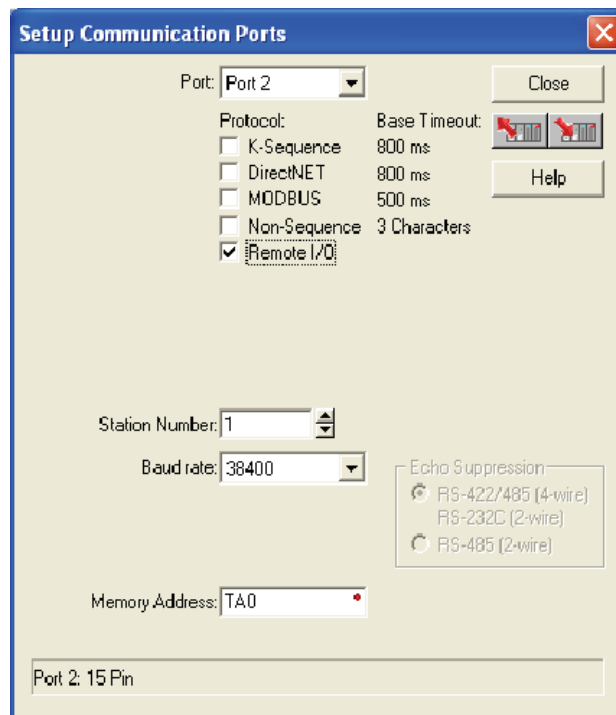
## Configuring the CPU's Remote I/O Channel

This section describes how to configure the DL350's built-in remote I/O channel. Additional information is in the Remote I/O manual, D2-REMIO-M, which you will need in configuring the Remote slave units on the network.

The DL350 CPU's built-in remote I/O channel has the same capability as the DL250 and DL450 CPUs. It can communicate with up to seven remote bases containing a maximum of 512 I/O points, at a maximum distance of 1000 meters.

You may recall from the CPU specifications in Chapter 3 that the DL350's Port 2 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 *DirectSOFT*, choose the PLC menu, then Setup, then Setup Secondary Comm Port...

- **Port:** From the port number list box at the top, choose "Port 2".
- **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 DL350 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 2.

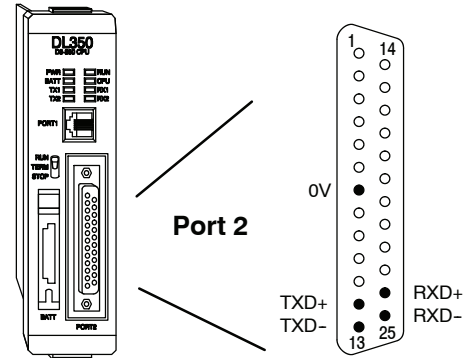


Then click the button indicated to send the Port 2 configuration to the CPU, and click Close.

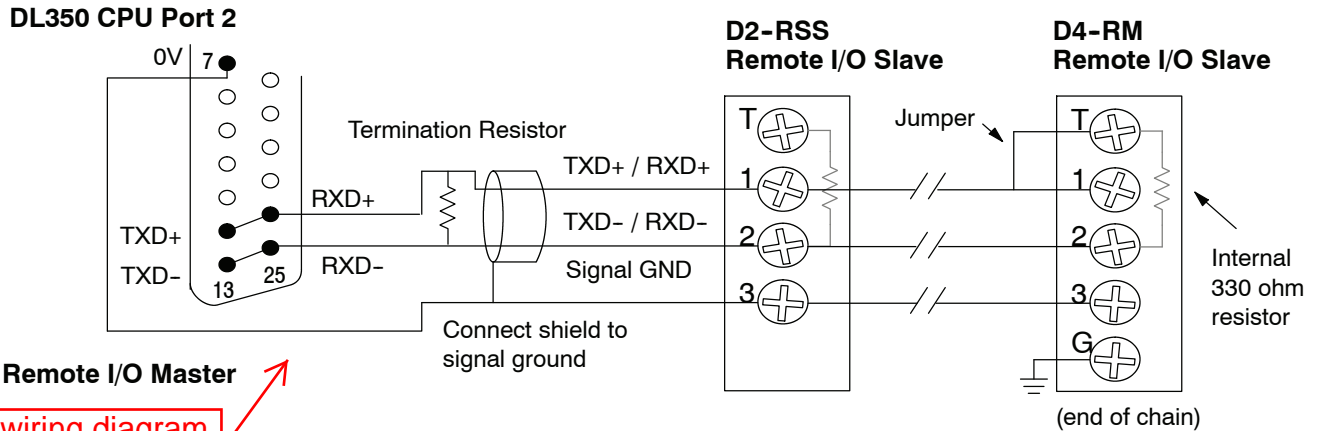
The next step is to make the connections between all devices on the Remote I/O link.

The location of the Port 2 on the DL350 is on the 25-pin connector, as pictured to the right.

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



Now we are ready to discuss wiring the DL350 to the remote slaves on the remote base(s). The remote I/O link is a 3-wire, half-duplex type. Since Port 2 of the DL350 CPU is a 5-wire port, we must jumper its transmit and receive lines together as shown below (converts it to 3-wire, half-duplex).



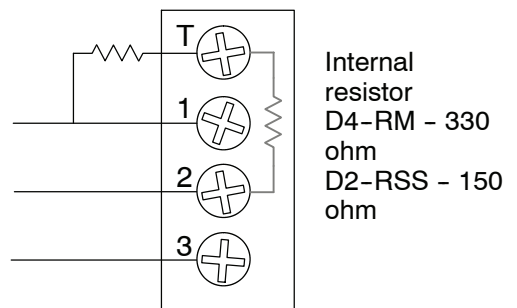
This wiring diagram is incorrect. Refer to the wiring diagram shown on the Errata Sheet on the first page of this file.

The twisted/shielded pair connects to the DL350 Port 2 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 will all three 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, parallel a matching resistance across the slave's pins 1 and 2 instead.

Remember to size the termination resistor at Port 2 to match the cables rated impedance. *The resistance values should be between 100 and 500 ohms.*

Add series external resistor



**Configure Remote I/O Slaves**

After configuring the DL350 CPU's Port 2 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 to match CPU's Port 2 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.

**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 2 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 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 (only 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 following page gives a short program example for one slave.

Memory Addr. Pointer	<b>37700</b>
----------------------	--------------

Remote I/O data

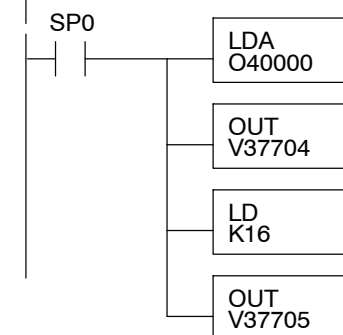
<b>Reserved</b>	<b>V37700</b>	xxxx
	V37701	xxxx
	V37702	xxxx
	V37703	xxxx

<b>Slave 1</b>	V37704	xxxx
	V37705	xxxx
	V37706	xxxx
	V37707	xxxx

•  
•  
•

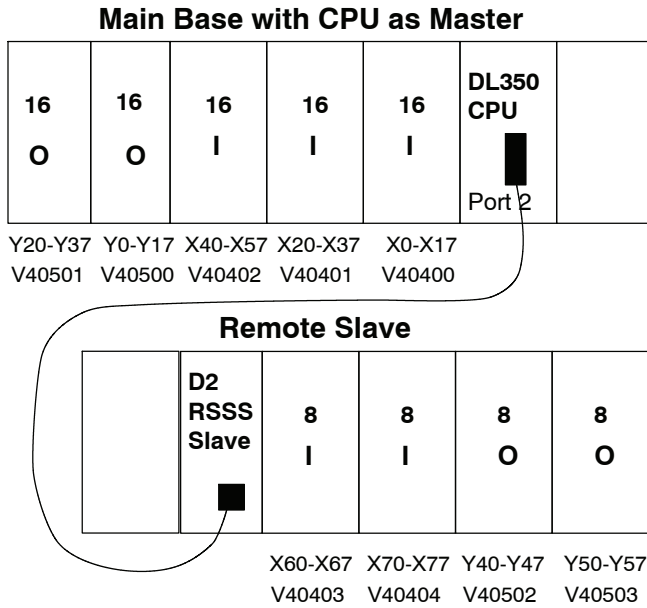
<b>Slave 7</b>	V37734	0000
	V37735	0000
	V37736	0000
	V37737	0000

DirectSOFT



Consider the simple system featuring Remote I/O shown below. The DL350'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. 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

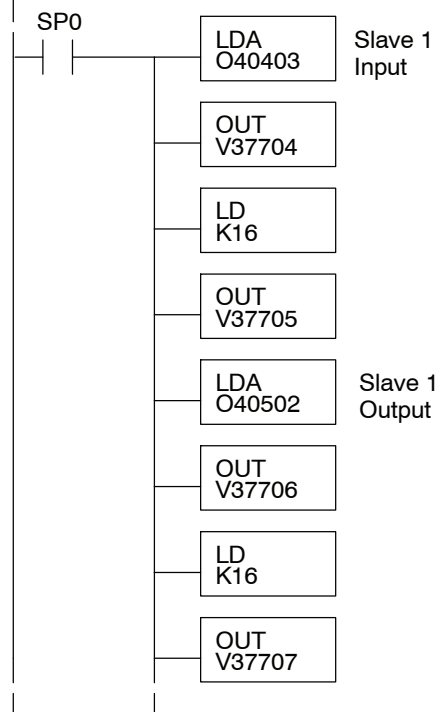
### Remote I/O Setup Program

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 the Remote I/O Manual). 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.

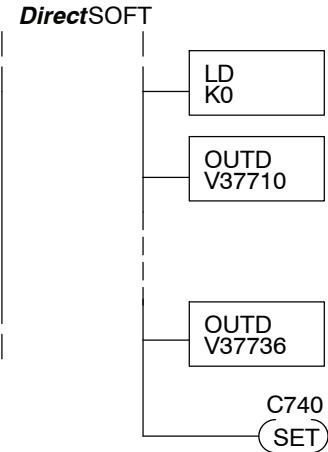
The next page covers the remainder of the required program to get this remote I/O link up and running.

### DirectSOFT



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.

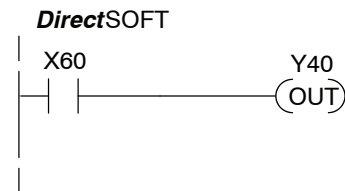
We continue our setup program from the previous page by adding a segment which fills the remainder of the table with zeros. The example to the right fills zeros for slave numbers 2-7, which do not exist in our example system.



On the last rung in the example program above, we set a special relay contact C740. This particular contact indicates to the CPU the ladder program has 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 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.



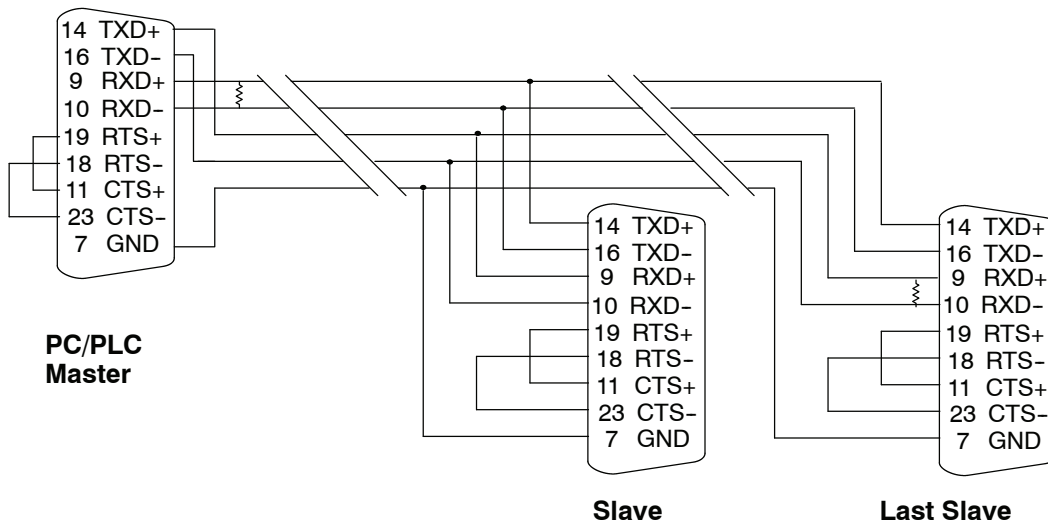


# Network Connections to MODBUS and *DirectNET*

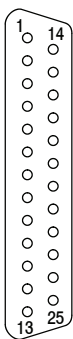
## Configuring the CPU's Comm Port

This section describes how to configure the CPU's built-in networking ports. for either MODBUS or *DirectNET*. This will allow you to connect the DL305 PLC system directly to MODBUS networks using the RTU protocol, or to other devices on a *DirectNET* 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 *DirectNET*, order our *DirectNET* manual, part number DA-DNET-M.

You will need to determine whether the network connection is a 3-wire RS-232 type, or a 5-wire RS-422 type. Normally, the RS-232 signals are used for shorter distances (15 meters max), for communications between two devices. RS-422 signals are for longer distances (1000 meters max.), and for multi-drop networks (from 2 to 247 devices). Use termination resistors at both ends of RS-422 network wiring, matching the impedance rating of the cable, for example, to match the termination resistance to Belden 9841 use a 120 ohm resistor. Resistors should be insatlled close to the end of the cable at the master and last slave connections.



The recommended cable for RS422 is Beldon 8102 or equivalent.



25-pin Female D Connector

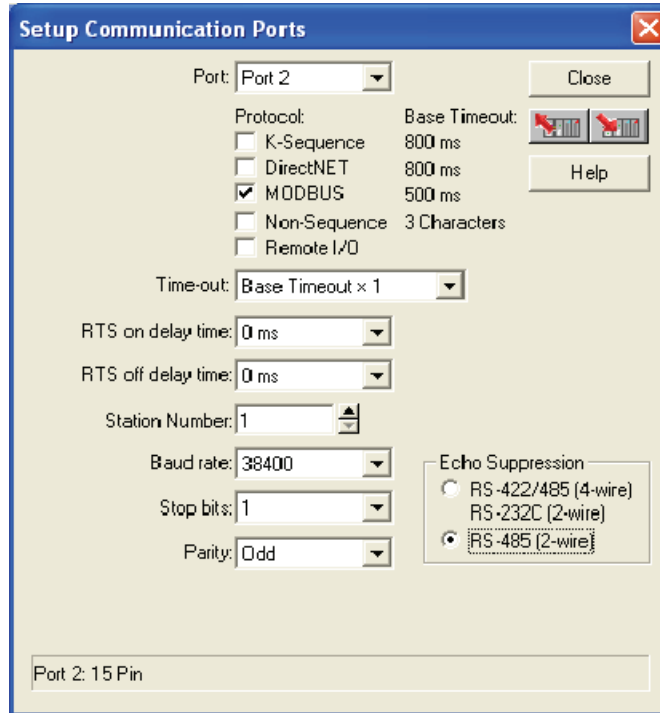
Port 2 Pin Descriptions (DL350 CPU)	
1	not used
2	TXD Transmit Data (RS232C)
3	RXD Receive Data (RS232C)
4	RTS Ready to Send (RS-232C)
5	CTS Clear to Send (RS-232C)
6	not used
7	0V Power (-) connection (GND)
8	0V Power (-) connection (GND)
9	RXD + Receive Data + (RS-422)
10	RXD - Receive Data (RS-422)
11	CTS + Clear to Send + (RS422)
12	TXD + Transmit Data + (REMIO)
13	TXD - Transmit Data - (REMIO)

Port 2 Pin Descriptions (Cont'd)	
14	TXD + Transmit Data + (RS-422)
15	not used
16	TXD - Transmit Data - (RS-422)
17	not used
18	RTS - Request to Send - (RS-422)
19	RTS + Request to Send - (RS-422)
20	not used
21	not used
22	not used
23	CTS - Clear to Send - (RS-422)
24	RXD + Receive Data + (REMIO)
25	RXD - Receive Data - (REMIO)

## MODBUS Port Configuration

In *DirectSOFT*, choose the PLC menu, then Setup, then “Secondary Comm Port”.

- **Port:** From the port number list box at the top, choose “Port 2”.
- **Protocol:** Click the check box to the left of “MODBUS” (use AUX 56 on the HPP, and select “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 between raising the RTS line and sending the data. This is for devices that do not use RTS/CTS handshaking. The RTS and CTS lines must be bridged together for the CPU to send any data.
- **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 DL350 network instructions used in Master mode 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 DL350 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.

### DirectNET Port Configuration

In *DirectSOFT*, choose the PLC menu, then Setup, then “Secondary Comm Port”.

- **Port:** From the port number list box, choose “Port 2”.
- **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 between raising the RTS line and sending the data. This is for devices that do not use RTS/CTS handshaking. The RTS and CTS lines must be bridged together for the CPU to send any data.
- **Station Number:** For making the CPU port a *DirectNET* master, choose “1”. The allowable range for *DirectNET* 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 DL350 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

This section describes how other devices on a network can communicate with a CPU port that you have configured as a **DirectNET** slave or MODBUS slave (DL350). A MODBUS host must use the MODBUS RTU protocol to communicate with the DL350 as a slave. The host software must send a MODBUS function code and MODBUS address to specify a PLC memory location the DL350 comprehends. The **DirectNET** host uses normal I/O addresses to access the applicable DL305 CPU and system. No CPU ladder logic is required to support either MODBUS slave or **DirectNET** slave operation.

### MODBUS Function Codes Supported

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 DL350 supports the MODBUS function codes described below.

MODBUS Function Code	Function	DL305 Data Types Available
01	Read a group of coils	Y, CR, T, CT
02	Read a group of inputs	X, SP
05	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	Write a value into a single register	V
16	Write a value into a group of registers	V

### 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 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 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.

DL350 Memory Type	QTY (Dec.)	PLC Range (Octal)	MODBUS Address Range (Decimal)	MODBUS Data Type
<b>For Discrete Data Types .... Convert PLC Addr. to Dec. + Start of Range + Data Type</b>				
Inputs (X)	512	X0 - X777	2048 - 2560	Input
Special Relays (SP)	512	SP0 - SP777	3072 - 3584	Input
Outputs (Y)	512	Y0 - Y777	2048 - 2560	Coil
Control Relays (CR)	1024	C0 - C1777	3072 - 4095	Coil
Timer Contacts (T)	256	T0 - T377	6144 - 6399	Coil
Counter Contacts (CT)	128	CT0 - CT177	6400 - 6271	Coil
Stage Status Bits (S)	1024	S0 - S1777	5120 - 6143	Coil
<b>For Word Data Types .... Convert PLC Addr. to Dec. + Data Type</b>				
Timer Current Values (V)	256	V0 - V377	0 - 255	Input Register
Counter Current Values (V)	128	V1000 - V1177	512 - 639	Input Register
V-Memory, user data (V)	3072 4096	V1400 - V7377 V10000 - V17777	768 - 3839 4096 - 8191	Holding Register
V-Memory, system (V)	256	V7400 - V7777	3480 - 3735	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 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, SP, Y, CR, S, T, C (contacts)
- Word - V, Timer current value, Counter current value

In either case, you basically 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.

DL350 Memory Type	QTY (Dec.)	PLC Range (Octal)	MODBUS Address Range (Decimal)	484 Mode Address	584/984 Mode Address	MODBUS Data Type
<b>For Discrete Data Types ... Convert PLC Addr. to Dec. + Start of Range + Appropriate Mode Address</b>						
Inputs (X)	512	X0 - X777	2048 - 2560	1001	10001	Input
Special Relays (SP)	512	SP0 - SP777	3072 - 3584	1001	10001	Input
Outputs (Y)	512	Y0 - Y777	2048 - 2560	1	1	Coil
Control Relays (CR)	1024	C0 - C3777	3072 - 4095	1	1	Coil
Timer Contacts (T)	256	T0 - T377	6144 - 6399	1	1	Coil
Counter Contacts (CT)	128	CT0 - CT177	6400 - 6527	1	1	Coil
Stage Status Bits (S)	1024	S0 - S1777	5120 - 6143	1	1	Coil
<b>For Word Data Types .... Convert PLC Addr. to Dec. + Appropriate Mode Address</b>						
Timer Current Values (V)	256	V0 - V377	0 - 255	3001	30001	Input Reg.
Counter Current Values (V)	128	V1000 - V1177	512 - 639	3001	30001	Input Reg.
V Memory, user data (V)	3072 4096	V1400 - V7377 V10000 - V17777	768 - 3839 4096 - 8192	4001	40001	Hold Reg.
V Memory, system (V)	256	V7400 - V7777	3840 - 3735	4001	40001	Hold Reg.

The following examples show how to generate the MODBUS addresses for hosts which require this format.

**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**

$$V2100 = 1088 \text{ decimal}$$

$$1088 + 40001 = \boxed{41089}$$

V Memory, system (V)	320	V700 - V777 V7400 - V7777	448 - 768 3840 - 3735	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 Addr. + Mode**

$$Y20 = 16 \text{ decimal}$$

$$16 + 2048 + 1 = \boxed{2065}$$

Outputs (Y)	1024	Y0 - Y1777	2048 - 3071	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**

$$T10 = 8 \text{ decimal}$$

$$8 + 3001 = \boxed{3009}$$

Timer Current Values (V)	256	V0 - V377	0 - 255	3001	30001	Input 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**

$$C54 = 44 \text{ decimal}$$

$$44 + 3072 + 1 = \boxed{3117}$$

Control Relays (CR)	2048	C0 - C3777	3072 - 5119	1	1	Coil
---------------------	------	------------	-------------	---	---	------

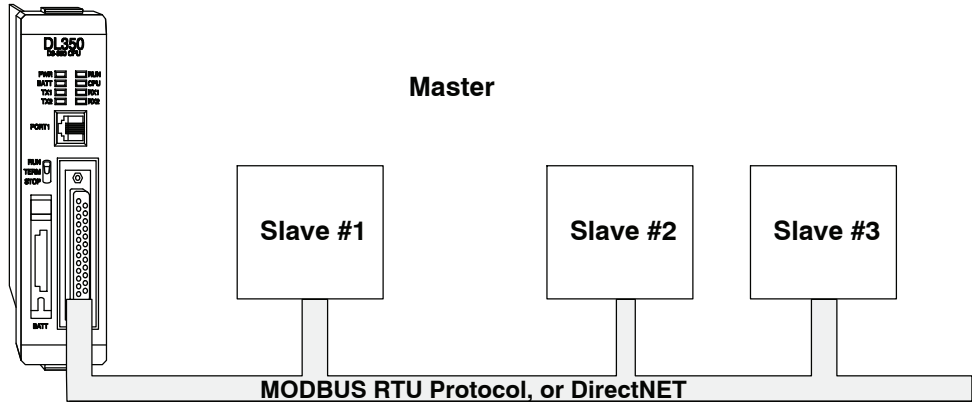
**Determining the DirectNET Address**

Addressing the memory types for **DirectNET** slaves is very easy. Use the ordinary native address of the slave device itself. To access a slave PLC's memory address V2000 via **DirectNET**, for example, the network master will request V2000 from the slave.

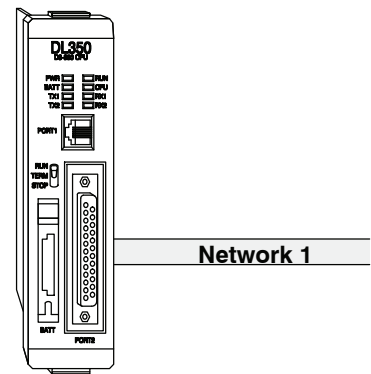
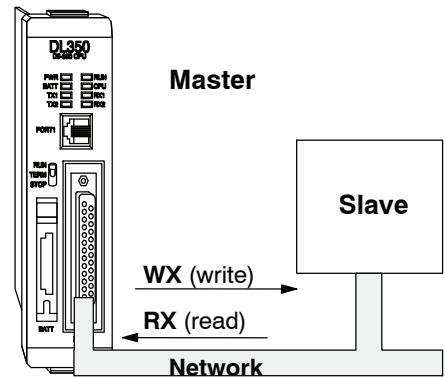


# Network Master Operation

This section describes how the DL350 can communicate on a MODBUS or *DirectNET* network as a master. For MODBUS networks, it uses the MODBUS RTU protocol, which must be interpreted by all the slaves on the network. Both MODBUS and *DirectNET* are single master/multiple slave networks. The master is the only member of the network that can initiate requests on the network. This section teaches you how to design the required ladder logic for network master operation.



When using the DL350 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.



The following step-by-step procedure will provide you the information necessary to set up your ladder program to receive data from a network slave.

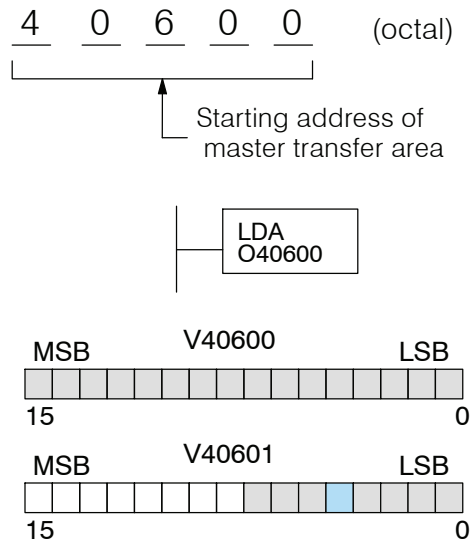


### Step 3: Specify Master Memory Area

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 DL350 CPU sends the number of bytes previously specified from its memory area beginning at the LDA address specified.

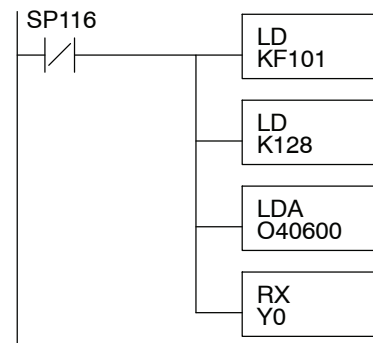
For an RX instruction, the DL350 CPU reads the number of bytes previously specified from the slave, placing the received data into its memory area beginning at the LDA address specified.



**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.



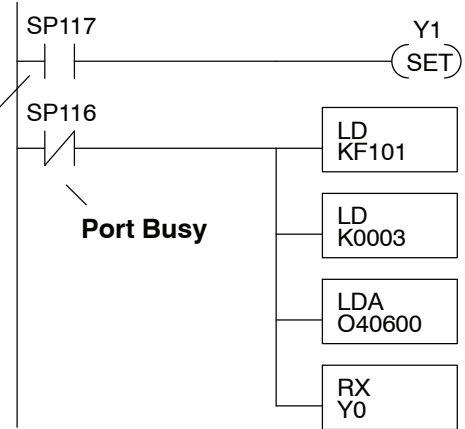
- **DirectNET** slaves - specify the same address in the WX and RX instruction as the slave's native I/O address
- MODBUS DL405, DL305 (DL350 CPU), or DL205 slaves - specify the same address in the WX and RX instruction as the slave's native I/O address
- MODBUS 305C (DL330/340 CPUs) slaves - use the following table to convert DL305 addresses to MODBUS addresses

DL305C (DL330/340 CPUs) Series CPU Memory Type-to-MODBUS Cross Reference					
PLC Memory type	PLC base address	MODBUS base addr.	PLC Memory Type	PLC base address	MODBUS 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**

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

**Port Communication Error**



The 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”(SP116), and the other indicates “Port Communication Error” (SP117). The example above shows the use of these contacts for a network master that only reads a device (RX). 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.

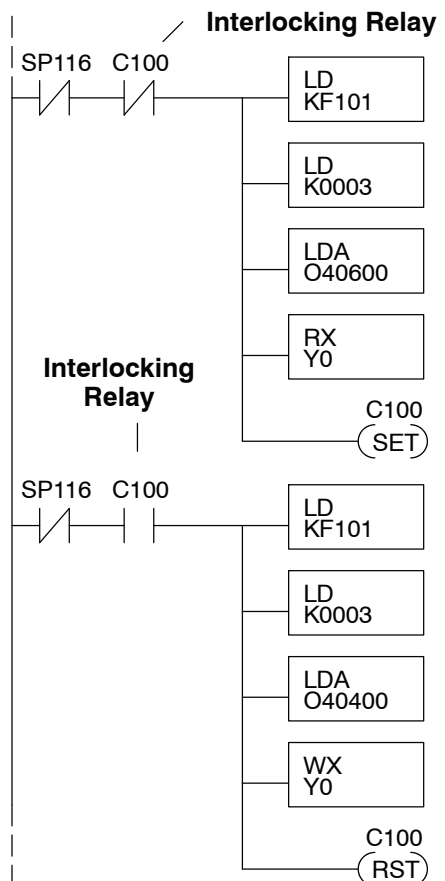
The “Port Communication Error” bit turns on when the PLC has detected an error. Use of this bit is optional. When used, it should be ahead of any network instruction boxes since the error bit is reset when an RX or WX instruction is executed.

**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, 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<sup>PLUS</sup> 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.



System Design and Configuration

