B

RLL Communications Programs

Why are networking instructions needed in your RLL?

The Master Initiates Requests

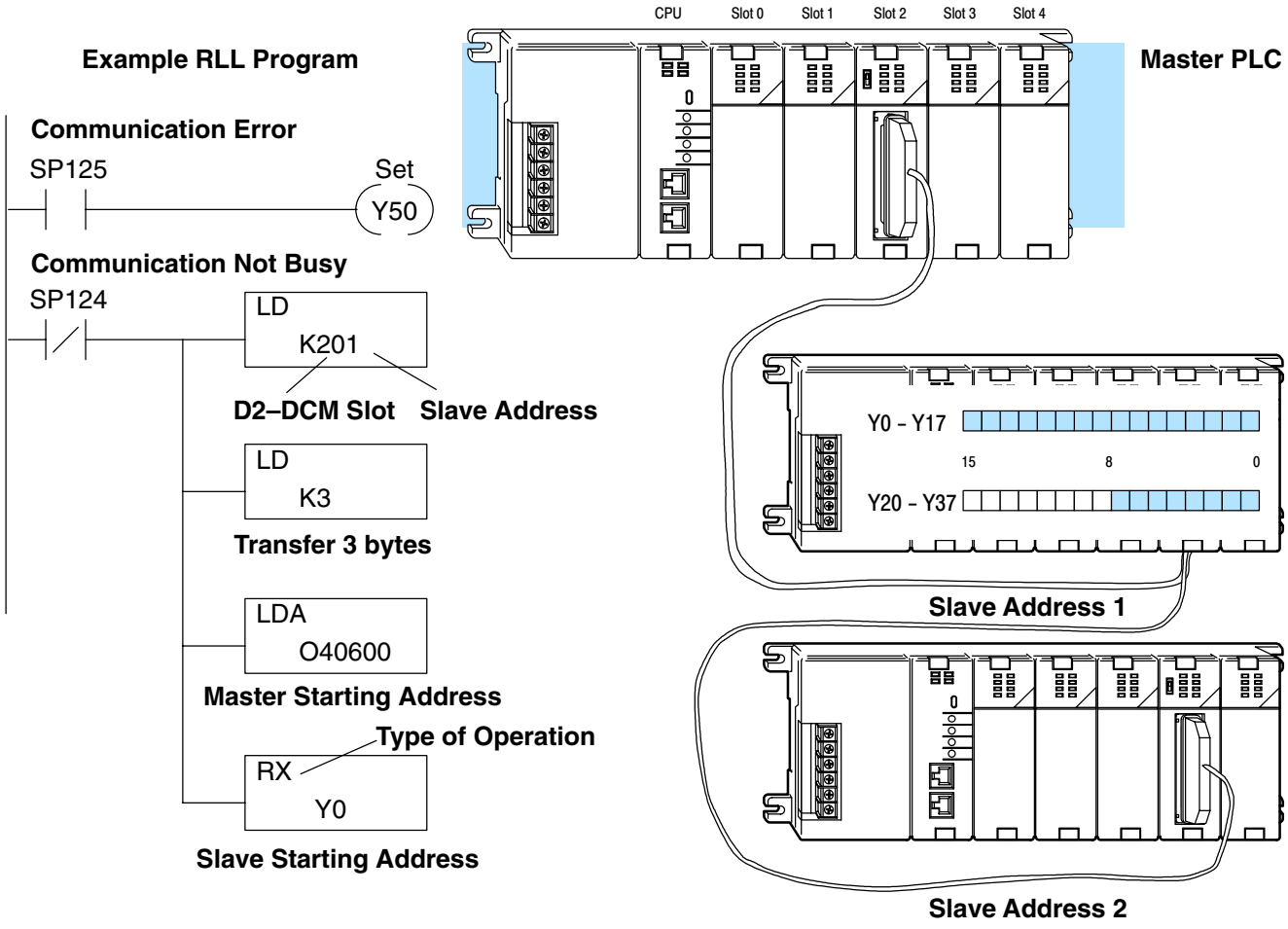
Since ***Direct***NET is a master/slave network, the master station must initiate requests for network data transfers. If you're using a PLC as the master station, you use simple RLL instructions to initiate the requests.

Why Ladder Logic?

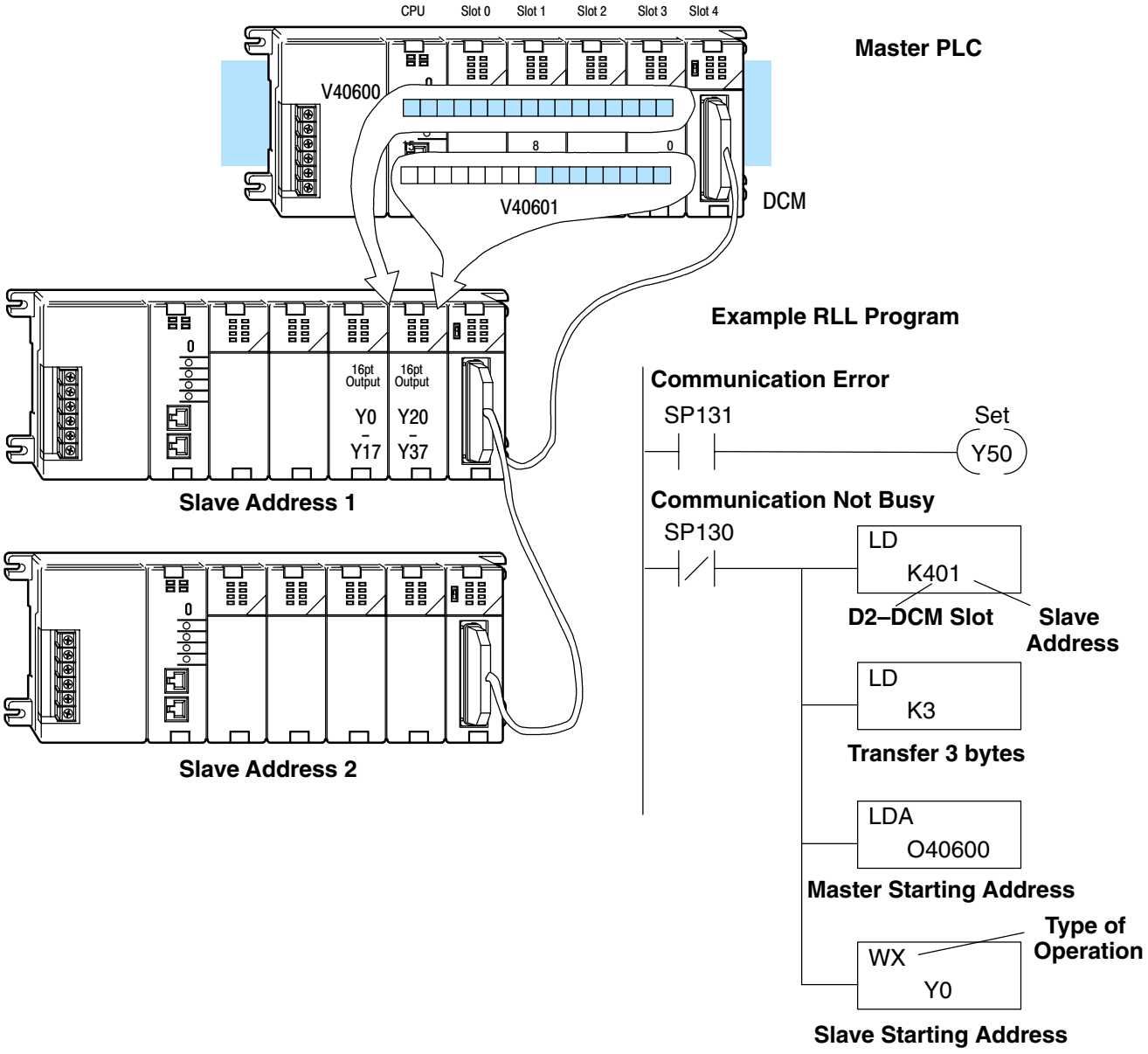
Since the D2–DCM network interface does not contain a program, you have to use the PLC to issue the commands to tell the D2–DCM where to read or write data. The D2–DCM gets information from the PLC and then converts the information into the appropriate ***Direct***NET commands. The RLL instructions use or identify the following items.

1. Uses the special relays assigned to the slot to control the communications.
2. Slot location of the D2–DCM master and the slave station address. (LD instruction)
3. Amount of data (in bytes, decimal) you want to transfer. (LD instruction)
4. Area of memory to be used by the master. (LDA instruction, see the DL205 User Manual for a detailed memory map.)
5. Area of memory to be used by the slave, and whether it is a read or write operation. (RX or WX instruction)
6. Interlocks for communication timing and multiple RX and WX routines.

This example reads 3 bytes of data from Slave Address #1,(starting at Y0), into the Master PLC starting at V40600 (Control Relays).



This example writes 3 bytes of data from the Master Station (starting at V40600) to Y0 – Y27 in Slave Station #1.



The following paragraphs explain each operation and provide some helpful hints to make your programs simple and easy to follow.

Identifying the master and slave locations & addresses

The first Load (LD) instruction identifies the slot location of the D2–DCM master and the address of the slave station. (Remember, the slot numbers start at 0.)

The constant (K) portion of the instruction actually contains two pieces (bytes) of information. The first two digits specify the D2–DCM master location and the second two digits specify the slave station address.

It is necessary to specify both the master slot location and slave address because you can have more than one D2–DCM master in the base and you can have up to 90 slave stations for each master.

Conversion Hints!

Valid Slot Range: 0–7

Valid Slave Address: 1–90

Example

Master in Slot: 2

Slave Address: 3C HEX (60 decimal)

Convert the HEX address to decimal

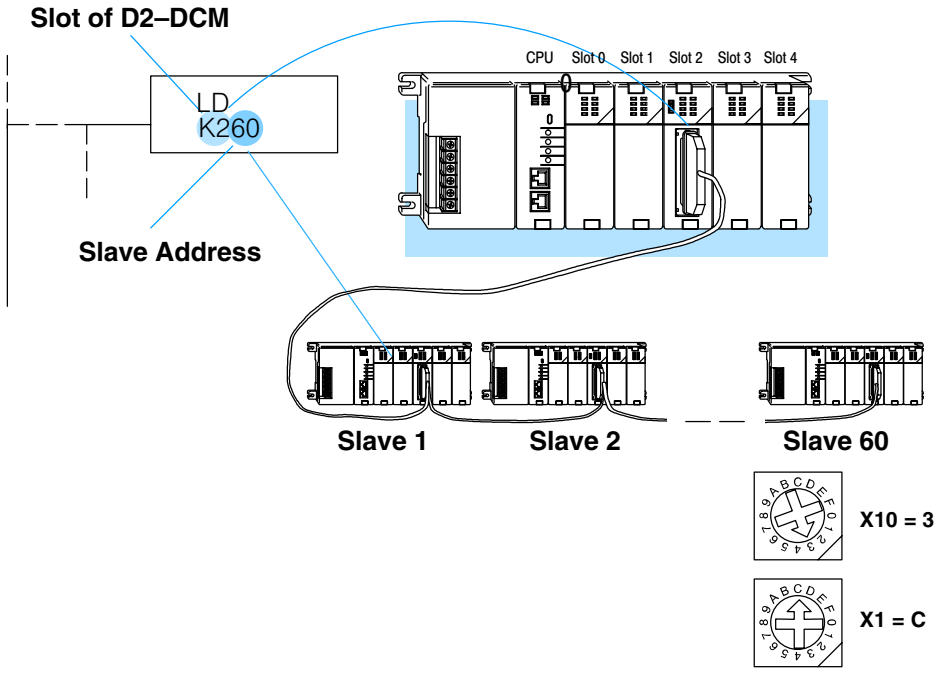
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	HEX
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	DEC

HEX 3C

3 x 16 = 48 + **C** = 12 = 60 decimal

NOTE: The LD instruction K value is entered in decimal, but the D2–DCM master and slave addresses are in HEX. The HEX addresses must be converted to their decimal equivalent for this instruction. See the conversion hints above.

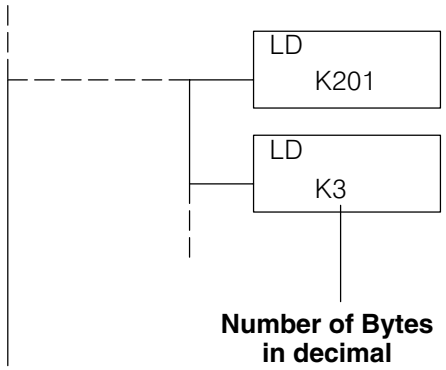
This example is showing three slaves. In this case the address conversions are simple. Check the Conversion Hints shown above for a more complex example.



Specifying the amount of data to transfer

The second LD instruction indicates the amount of data that needs to be transferred (in bytes, 128 maximum). You have to specify the amount of data in complete bytes. For example, Y0 – Y27 would be three bytes of data. There are 24 bits for the output range of Y0–Y27 (these I/O addresses are in octal). From the charts below we see that we can obtain 8 bits per byte for this type of memory. Therefore, 24 bits yields 3 bytes, of 8 bits each.

The charts below can be very helpful. Notice that the different PLC families do not always use the same types of memory or the same byte boundaries. For example, the DL305 does not use a separate data type for input and output points.



Example:

3 bytes of data to be transferred

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
Diagnostic Status	8	1

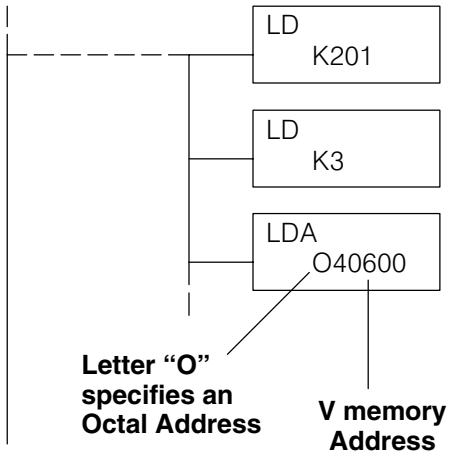
DL305 Memory	Bits per unit	Number of bytes
Data registers	8	1
T / C accumulator	16	2
I/O, internal relays, shift register bits, T/C bits, stage bits	1	1
Diagnostic Status (5 word R/W)	16	10

Designating the master station memory area

The Load Address (LDA) instruction specifies the V memory area of the master that will be used. This is the starting address. Additional sequential locations may be used, depending on the number of bytes that are being transferred. Since all DL405 data is mapped into V memory, you can easily access the data you need.

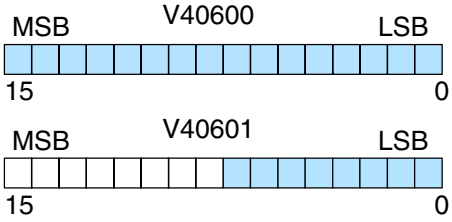
If you are reading information from the slave station, this is the destination area, or, the area where the master will store the information.

If you are writing information to the slave station, this is the source area, or, the area where the master will obtain the information that will be transferred to the slave.



Example:

V memory location 40600 will be the starting point of the data transfer area for the master. The following locations will be used to store the data.



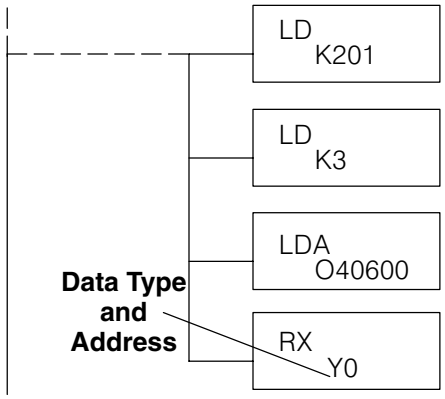
NOTE: Since V memory words are always 16 bits, you may not always use the whole word. For example, if you only specify 3 bytes and you are reading Y outputs from the slave, you will only get 24 bits of data. In this case, only the 8 least significant bits of the last word location will be modified. The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

Identifying the slave station memory area to read or write

The Read Network (RX) or Write Network (WX) is the last instruction in the routine. Use the RX if you want to read data from the slave, or use the WX instruction if you want to write data to the slave.

You have to specify the data type and the starting address for the slave. (Remember, you have to specify a data type that will work correctly with the number of bytes specified.)

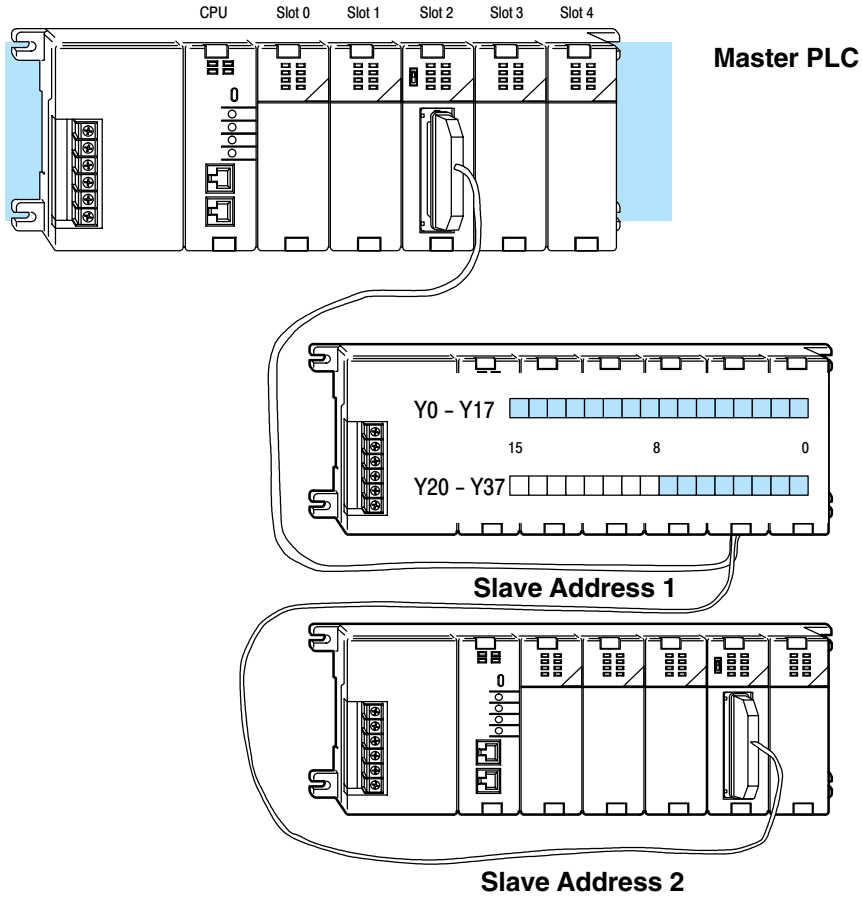
If you use the RX instruction, the data will be read from the slave starting at the address specified. If you use the WX instruction, the data will be written to the slave starting at the address specified.



Example:

Read from slave starting at Y0.

NOTE: If you are exchanging data with a DL305 system, it is important to understand how to reference the DL305 memory locations. For example, the DL305 I/O points are accessed with the V data type or the GY data type, even though the DL305 does not actually have those data types present in the CPU. The table on the next page provides a detailed cross reference.



D3–330 / D3–340 CPUs					
To get ... TMR/CNT Current Values	**use...**	**TMR / CNT Status Bits**	**use...**	**Data Registers**	**use...**
R600	V0	CT600	GY600[1]	R401, R400[2]	V100
R601	V1	CT601	GY601[1]	R403, R402[2]	V101
— — —	— — —	— — —	— — —	— — —	— — —
R677	V77	CT677	GY677[1]	R777, R776[2]	V237
To get ... I/O Points	**use...**	**Control Relays**	**use...**	**Shift Registers**	**use...**
IO 000	GY0[1]	CR160	GY160[1]	SR400	GY400
IO 001	GY1[1]	CR161	GY161[1]	SR401	GY401
— — —	— — —	— — —	— — —	— — —	— — —
IO 157	GY157[1]	CR377	GY377[1]	SR577	GY577

D3–330P CPUs					
To get ... TMR/CNT Current Values	**use...**	**TMR / CNT Status Bits**	**use...**	**Data Registers**	**use...**
R600	V0	CT600	GY600[1]	R401, R400[2]	V100
R601	V1	CT601	GY601[1]	R403, R402[2]	V101
— — —	— — —	— — —	— — —	— — —	— — —
R677	V77	CT677	GY677[1]	R777, R776[2]	V237
To get ... I/O Points	**use...**	**Control Relays**	**use...**	**Shift Registers**	**use...**
IO 000	GY0[1]	CR160	GY160[1]	SR200	GY400
IO 001	GY1[1]	CR161	GY161[1]	SR201	GY401
— — —	— — —	— — —	— — —	— — —	— — —
IO 157	GY157[1]	CR277	GY277[1]	SR277	GY477

To get ... Stage Status Bits	**use...**
S0	GY200[1]
S1	GY201[1]
— — —	— — —
S177	GY277[1]

1 . You must have CPU firmware V1.9 or greater to use the GY data type in the RX/WX instructions.

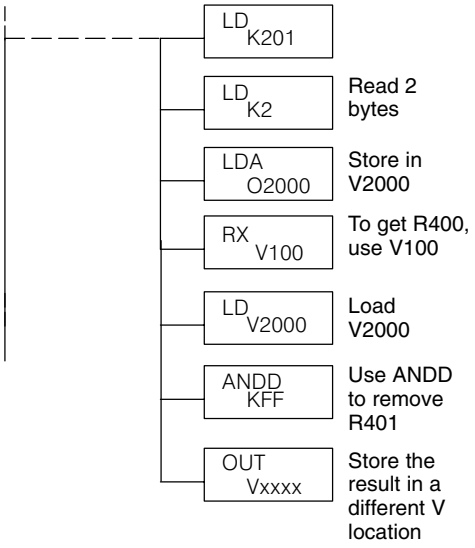
2 . Two bytes of DL305 register data are returned with one DL205 V memory location.

Example:

Read current value from R400 into memory location V2000.

If you're just obtaining I/O or Timer/Counter values, the task is fairly simple. But when you work with data registers, it's a bit more involved. Here's why.

To get R400, you examine the table and find that you must use reference V0. You will also notice that you always get at least 2 registers! So you get R400 *and* R401. Since you only want the contents of R400, you have to add some ladder logic to get rid of the data from R401.



Controlling the communications

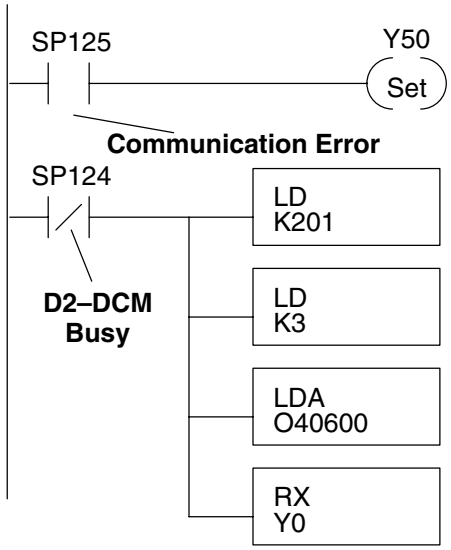
Communications Special Relays

Whenever communication is executed with a D2–DCM, chances are the communication will take longer than the actual PLC scan. If the D2–DCM is busy, another request should not be initiated until it is finished. Fortunately, there is an easy solution for this.

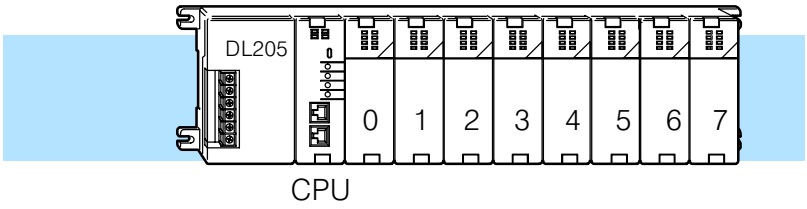
There are two SPs for each slot in the CPU base which are used only with the D2–DCM. For example, slot 0 has SP120 and SP121. SP120 is the D2–DCM Busy relay and, when turned on, indicates the D2–DCM is busy. SP121 indicates there is a communication error for slot 0.

You should always use the D2–DCM Busy SP in your RLL programs to ensure the D2–DCM is ready.

The communication error SP is optional, but it is a good way to monitor the communication status in the RLL program. If you use the communication error SP, make sure you place it at the beginning of your communication routines. This is because the communication error relay is always reset (turned off) whenever an RX or WX instruction is executed.



Special Purpose Communication Relays								
I/O Slot Location	0	1	2	3	4	5	6	7
Communication Busy	N/A	SP122	SP124	SP126	SP130	SP132	SP134	SP136
Communication Error	N/A	SP123	SP125	SP127	SP131	SP133	SP135	SP137



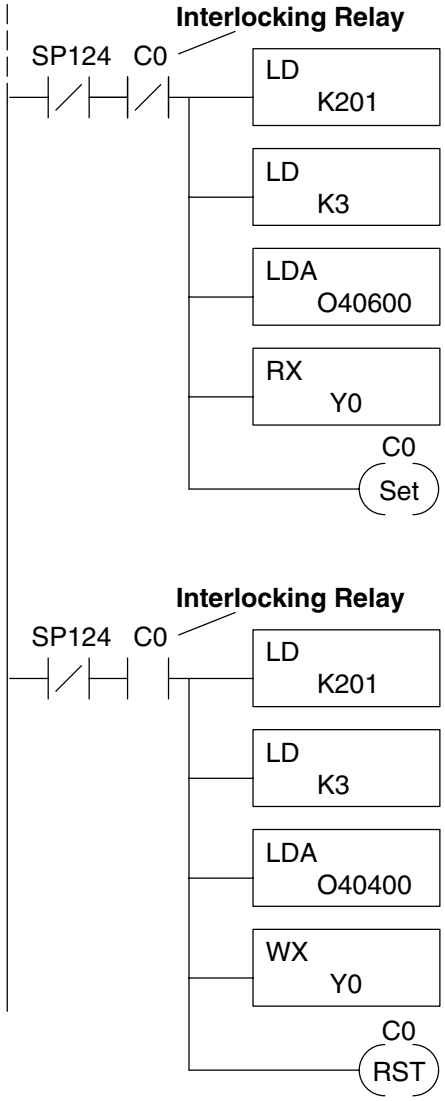
Multiple Read and Write Interlocks

If you're using multiple reads and writes in the RLL program, the routines need to be interlocked to be certain that all the routines are executed. If the interlocks are not used, then the CPU will only execute the first routine. This is because the D2–DCM can only handle one transaction at a time.

In the example, once the RX instruction is executed, C0 is set. When the D2–DCM has finished the communication task, the second routine is executed and C0 is reset.

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

The ***Direct***NET manual provides a master / slave example with both RLL and RLL*PLUS* program descriptions.



Multiple Read and Write Interlocks

This example is showing three slaves. In this case the address conversions are very simple. Check the Conversion Hints shown above for a more complex example.