RLL Communications Programs

5

In This Chapter. . . .

— Why do you need a communications program?
— Identifying the Master and Slave
— Specifying the Amount of Data
— Designating the Master Station Memory Area
— Identifying the Slave Station Memory Area
— Controlling the Communications
— What part of the manual should you use next?

Why do you need a communications program?

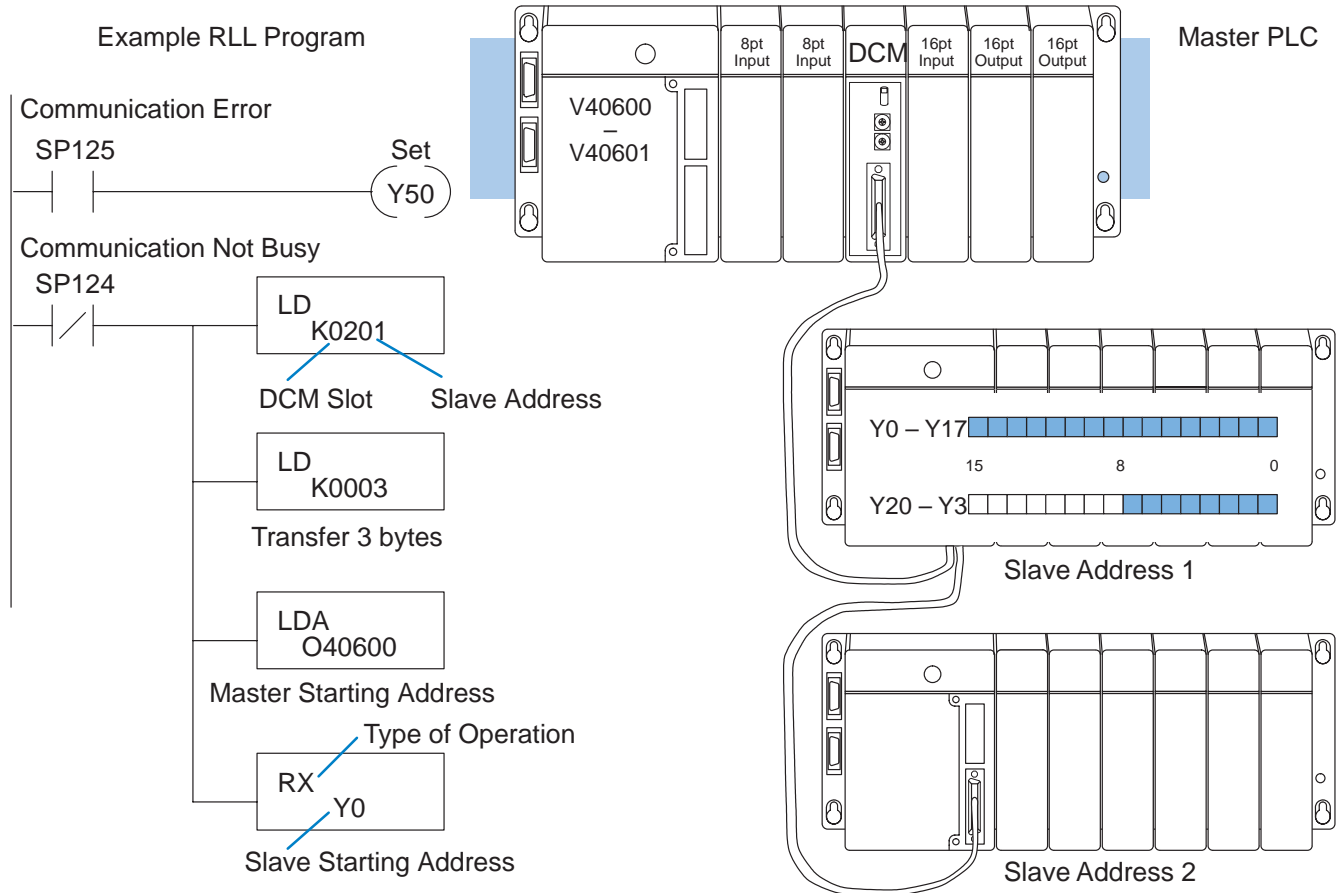**The Master Initiates Requests**   *Direct*NET is a master / slave network, therefore; the master station must initiate requests for network data transfers. If you're using a PLC as the master station, you use simple RLL instructions to initiate the requests.
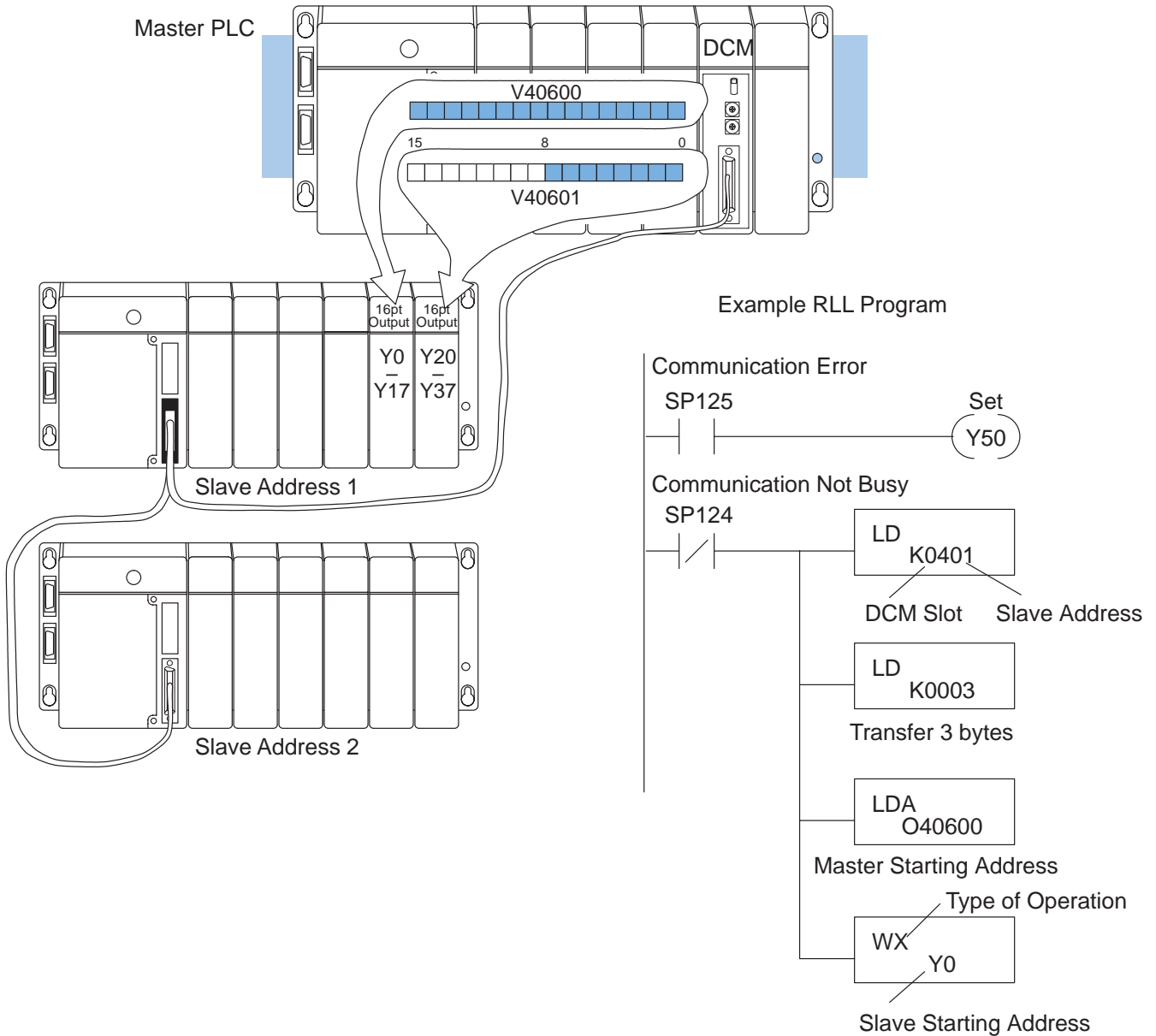
Why Ladder Logic?   Since the DCM (network interface) does not contain any RLL, you have to use the PLC to issue the commands telling the DCM where to read or write data. The DCM takes the PLC commands and converts them into the appropriate *Direct*NET commands. The RLL instructions identify the following items.

1. Slot location of the DCM master and the slave station address. (using the LD instruction)
2. Amount of data (in bytes) you want to transfer. (using the LD instruction)
3. Area of memory to be used by the master. (using the LDA instruction)
4. Area of memory to be used by the slave, and whether it is a read or write operation. (using either the RX or WX instruction)
5. 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).

Example RLL Program

Communication Error
SP125   Set
                                     Y50

Communication Not Busy
SP124
            LD
               K0201

         DCM Slot   Slave Address

            LD
               K0003

         Transfer 3 bytes

            LDA
               O40600

         Master Starting Address

                    Type of Operation
            RX
               Y0

         Slave Starting Address

Master PLC

V40600
–
V40601

Y0 – Y17
15   8   0
Y20 – Y3

Slave Address 1

Slave Address 2

This example writes 3 bytes of data from the Master Station (starting at V40600) to Y0 – Y37 in Slave Station #1.

Master PLC

DCM

V40600

15   8   0

V40601

16pt Output | 16pt Output

Y0 – Y17 | Y20 – Y37

Slave Address 1

Slave Address 2

Example RLL Program

Communication Error

SP125   Set

                                              (Y50)

Communication Not Busy

SP124

LD
    K0401

DCM Slot   Slave Address

LD
    K0003

Transfer 3 bytes

LDA
    O40600

Master Starting Address

Type of Operation

WX
    Y0

Slave Starting Address

The following paragraphs explain each operation and provide some helpful hints to make your programs simple and easy to follow.
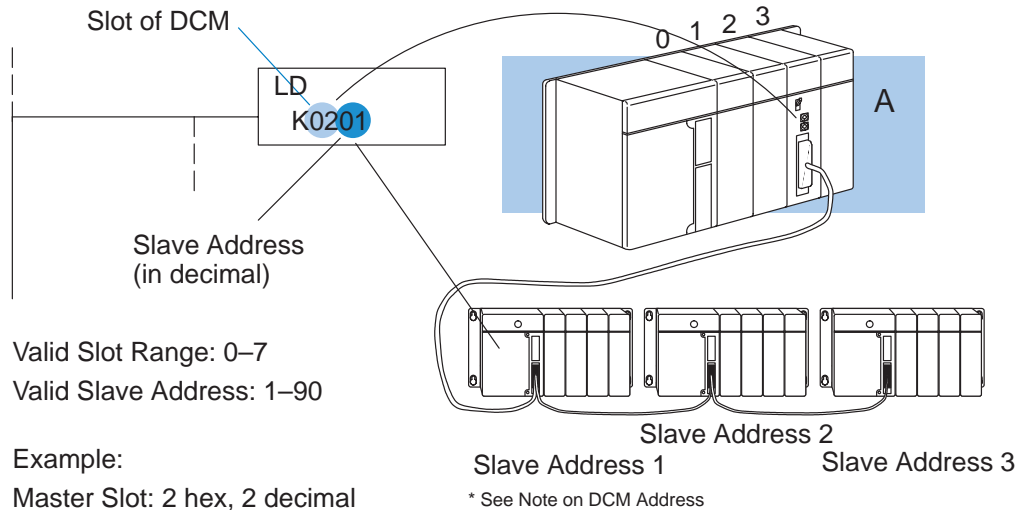
# Identifying the Master and Slave

Location of Master and Slave

Whenever the WX or RX instructions are used, they must be preceded with the necessary LD and LDA instructions to set up the communication addresses, data to transfer and the location for the transference. The first Load (LD) instruction identifies the slot location of the DCM master and the slave station address. (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 DCM master location and the second two digits specify the slave station location.

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

Valid Slot Range: 0–7

Valid Slave Address: 1–90

Example:

Master Slot: 2 hex, 2 decimal

Slave Address: 1 hex, 1 decimal

* See Note on DCM Address

NOTE: The LD instruction K value is entered in decimal, but the DCM master and slave addresses are in HEX (The DCU, DL340, DL430 and DL440 all have their addresses set in decimal.) You have to convert the HEX addresses for the DCM to their decimal equivalent for the LD instruction. It's easy to convert from HEX to decimal.
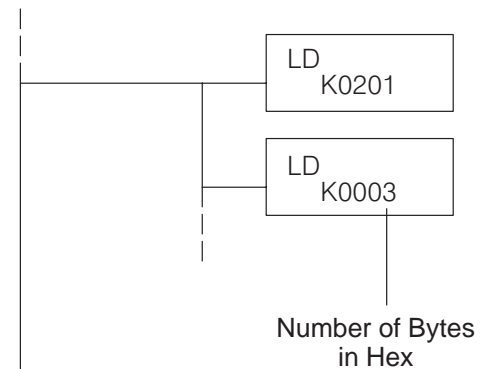
HEX Format

0 1 2 3 4 5 6 7 8 9 A B C D E F

10 11 12 13 14 15

HEX 3C

$3 \times 16 = 48 + C = 12 = 60$ decimal
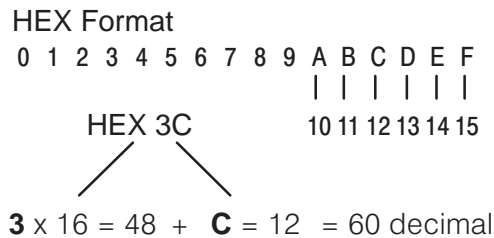
Specifying the Amount of Data

Number of Bytes to Transfer

The second LD instruction indicates the amount of data (in bytes) that needs to be transferred. You have to specify the amount of data in complete bytes. For example, Y0 – Y27 would be three bytes of data.

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. Appendices D, E, F provide complete listings of the DL405, DL305 and DL205 memory types.

```
          ┌──────────┐
          │ LD       │
          │   K0201  │
          └──────────┘
          ┌──────────┐
          │ LD       │
          │   K0003  │
          └──────────┘
```

Number of Bytes in Hex

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
Scratch Pad Memory	8	1
Diagnostic Status	8	1

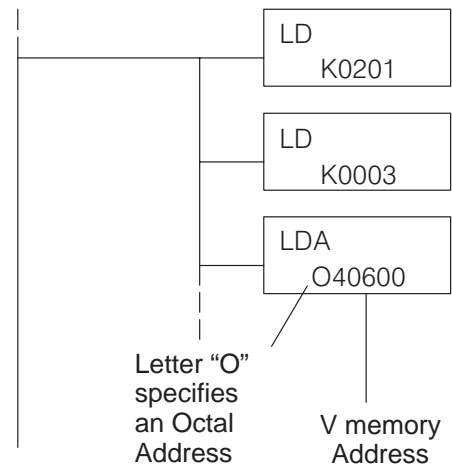| DL305 Memory	Bits per unit	Number of bytes
Data registers	8	1
T / C accumulator	16	2
I/O, internal relays, shift register bits, T/C bits, stage bits	1	1
Scratch Pad Memory	8	2
Diagnostic Status (5 word R/W)	16	10

Designating the Master Station Memory Area

Memory Area of Master

The Load Address (LDA) instruction specifies the V memory area of the master that will be used. This is the starting address (in octal). 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.

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.


```
LD
    K0201

LD
    K0003

LDA
    O40600
```

Letter "O" specifies an Octal Address

V memory Address

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.


MSB   V40600   LSB

15   0

MSB   V40601   LSB

15   0
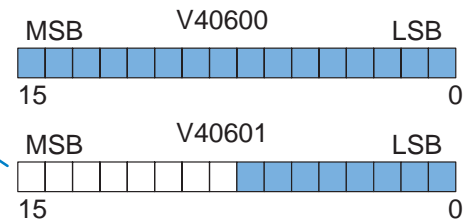
Identifying the Slave Station Memory Area

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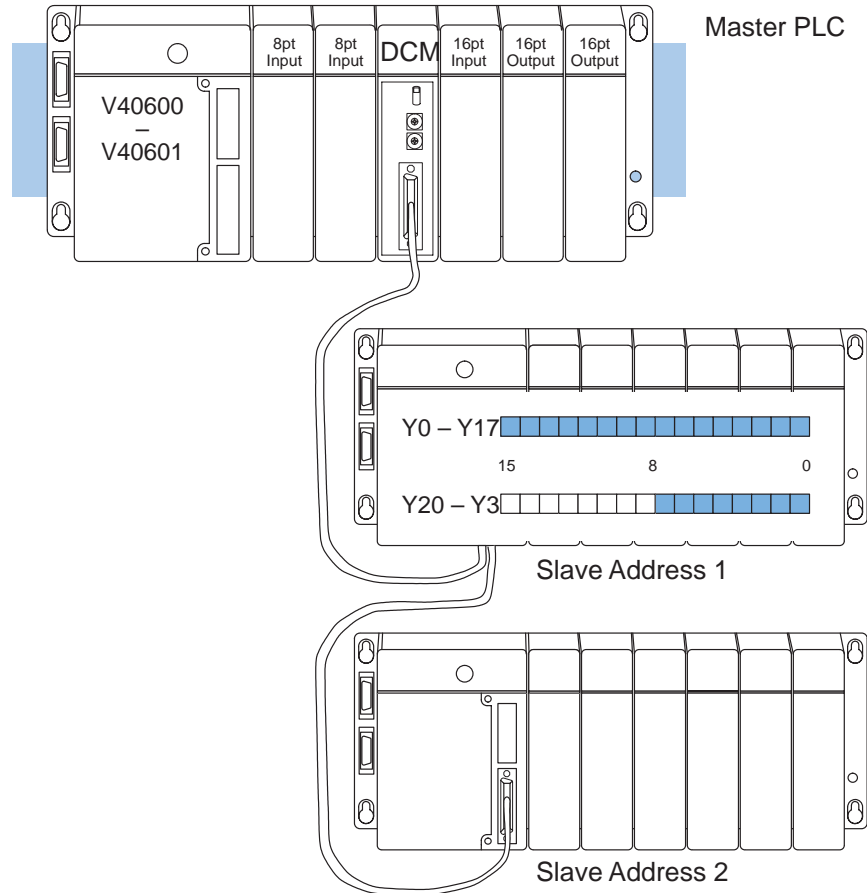
You have to specify the data type and the starting address (in octal) for the slave. (Remember, you have to specify a data type that will work correctly with the number of bytes specified.)

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.

LD	K0201
LD	K0003
LDA	O40600
RX	Y0

Data Type and Address
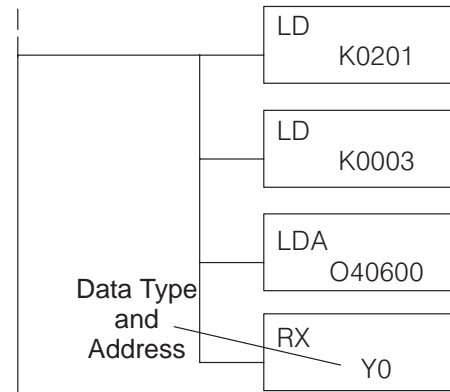
Example:
Read from slave starting at Y0.

NOTE: If you are using an RLL communications program to transfer data to or from a DL305 slave station, the data type is slightly different. For example, the DL305 I/O points are accessed with the GY data type. AppendixE provides a complete listing of the memory types and cross references for the DL305 family.

Master PLC

V40600 – V40601

8pt Input | 8pt Input | DCM | 16pt Input | 16pt Output | 16pt Output

Y0 – Y17
15   8   0

Y20 – Y3

Slave Address 1

Slave Address 2

Controlling the Communications

Communications Special Relays

When you execute communication with a DCM, chances are good the communication may take longer than the actual PLC scan. If the DCM is busy, you shouldn't initiate another request until it is finished. Fortunately, there's an easy solution for this.

There are two SPs (special relays) for each slot in the CPU base which are used to provide communication status. For example, slot 0 has SP120 and SP121. SP120 is the Busy relay and, when turned on, indicates the DCM is busy. SP121 indicates there is a communication error for slot 0.

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

The communication error SP is optional, but it's 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.
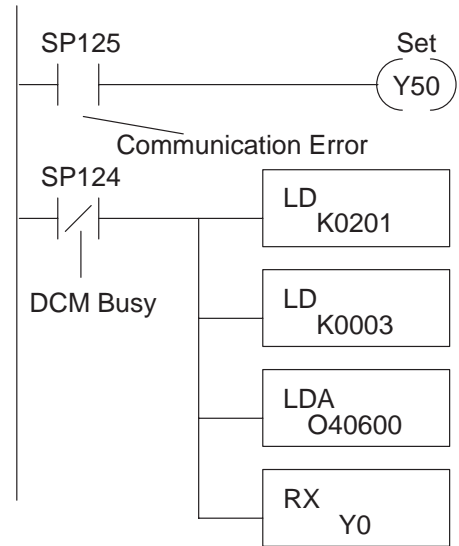
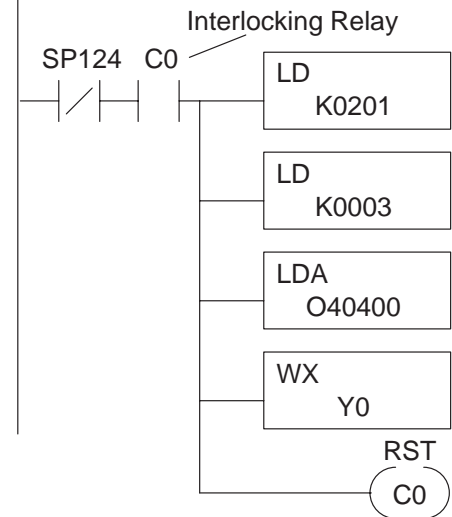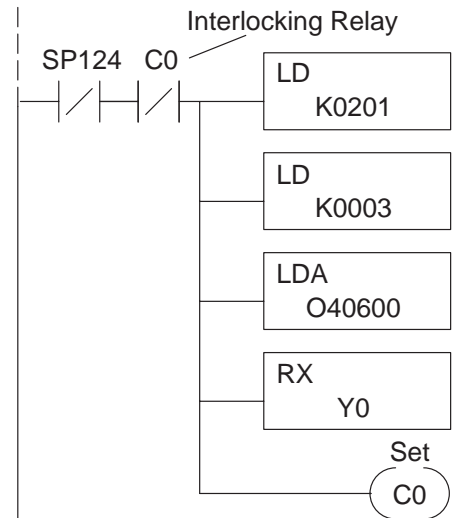| Special Purpose Communication Relays								
Communication Busy	SP120	SP122	SP124	SP126	SP130	SP132	SP134	SP136
Communication Error	SP121	SP123	SP125	SP127	SP131	SP133	SP135	SP137
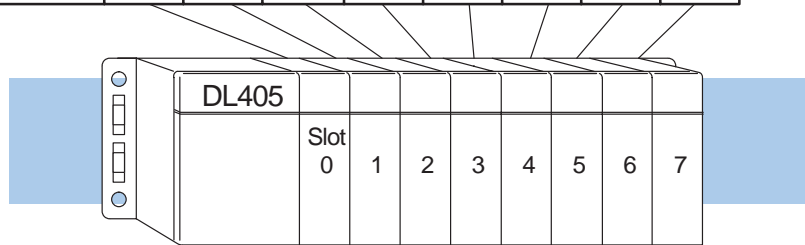
Multiple Read and Write Interlocks

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 the DCM can only handle one routine at a time.

In the example, once the RX instruction is executed, C0 is set. When the 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} is a much more efficient way to create automation program.

Appendix A provides a master / slave example with both RLL and RLL^{Plus} program descriptions.

```
              Interlocking Relay
  SP124  C0
   ─┤/├──┤/├─┐   ┌──────────┐
             ├───│ LD       │
             │   │    K0201 │
             │   └──────────┘
             │   ┌──────────┐
             ├───│ LD       │
             │   │    K0003 │
             │   └──────────┘
             │   ┌──────────┐
             ├───│ LDA      │
             │   │   O40600 │
             │   └──────────┘
             │   ┌──────────┐
             ├───│ RX       │
             │   │      Y0  │
             │   └──────────┘
             │            Set
             └──────────( C0 )
```

```
              Interlocking Relay
  SP124  C0
   ─┤/├──┤ ├─┐   ┌──────────┐
             ├───│ LD       │
             │   │    K0201 │
             │   └──────────┘
             │   ┌──────────┐
             ├───│ LD       │
             │   │    K0003 │
             │   └──────────┘
             │   ┌──────────┐
             ├───│ LDA      │
             │   │   O40400 │
             │   └──────────┘
             │   ┌──────────┐
             ├───│ WX       │
             │   │      Y0  │
             │   └──────────┘
             │            RST
             └──────────( C0 )
```

What part of the manual should you use next?

Start the Network   Once you've created the communications program, you can start the network. Chapter 7 provides information concerning network operation and troubleshooting. Appendices A and B provide examples of PLC master and PLC Peer networks. Both of these networks use RLL instructions for the communications program.