C

Host Master Example

In This Chapter. . . .

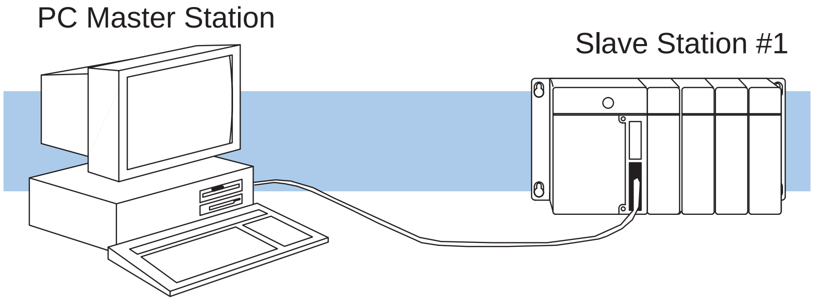
— The Example
— Remember these Four Steps!
— Step 1: Design the Network
— Step 2: Select the Communication Settings
— Step 3: Create the Communications Program
— Step 4: Start the Network.
— What should I do if it isn’t working correctly?

The Example

A hosted network utilizes an intelligent device, such as a personal computer, to act as the network master station. Slave stations can be DCMs or ***Direct***LOGIC™ PLCs with built-in ***Direct***Net ports. The host device initiates all communication requests and the slaves respond to the requests for data. This example is not intended to show you how to build ***Direct***Link programs. You should review your programming language manuals and Chapter 6 of this manual before you attempt to build your own ***Direct***Link programs.

Example Equipment

This chapter provides an example that allows you to quickly and easily set up a hosted network.



Master Station

You need a personal computer for the master station.

Slave Station #1

The following equipment is needed slave station #1.

- DL405 Base (4, 6, or 8 slot)
- DL430 or DL440 CPU
- I/O modules, including at least one I/O simulator and one output module

Install the Equipment

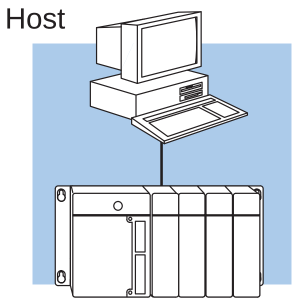
Normally, you can install the modules in any manner. However, we'd like to keep our examples consistent between the different configurations. Install the equipment in the following order.

WARNING: To minimize the risk of electrical shock, personal injury, or equipment damage, always disconnect the system power before installing or removing any system component.

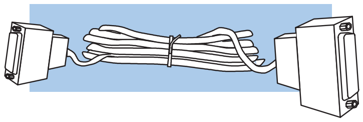
1. Install the CPUs as described in the DL405 User's Manual. You can connect the power wiring now (if it's not already connected), but don't connect the source power yet.
2. Install the I/O Simulator in slot 0 of slave station #1, next to the CPU.
3. Install the Output module in slot 1 of slave station #1.

Remember these Four Steps!

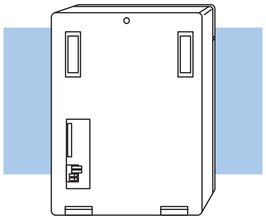
Use these steps to build your example network. The remainder of this chapter provides detailed explanations and examples of these steps.



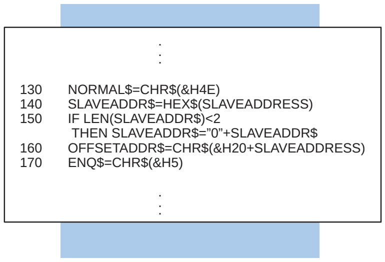
1. Design the network by:
   - Selecting the configuration (this is a Host as Master configuration)
   - Building the communication cables.



2. Select the communication settings by:
   - Setting the station switches

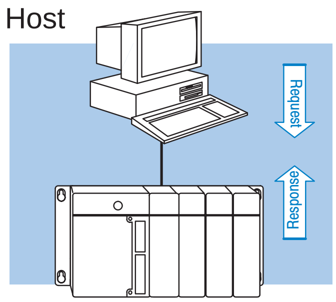


3. Write the communication control program.
   - ***Direct***NET program is used with Host master configurations.



```
                    :
                    :
130    NORMAL$=CHR$(&H4E)
140    SLAVEADDR$=HEX$(SLAVEADDRESS)
150    IF LEN(SLAVEADDR$)<2
        THEN SLAVEADDR$="0"+SLAVEADDR$
160    OFFSETADDR$=CHR$(&H20+SLAVEADDRESS)
170    ENQ$=CHR$(&H5)
                    :
                    :
```

4. Start the network operation.



Step 1: Design the Network

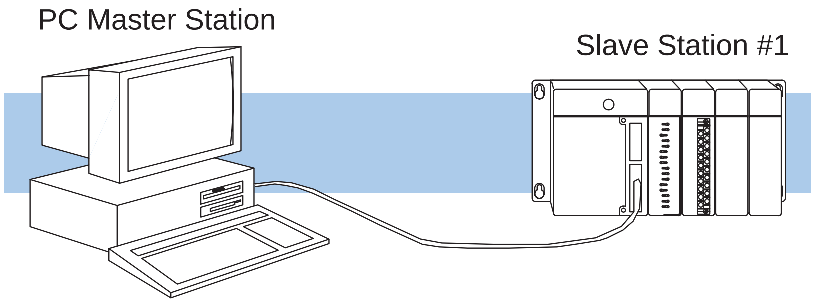
The Example Configuration

In this chapter we'll use the following example configuration to create a simple Host master network.

WARNING: These examples are for illustration purposes only and are not intended for use in actual applications. This is because there may be many aspects of your system safety precautions that are not addressed in the examples. If you use these examples in actual applications, you are increasing the risk of personal injury and/or property damage.

Goal:

1. Read a bit pattern from an input module (X0 – X7) in Slave #1
2. Write a bit pattern to an output module (Y0 – Y7) in Slave #1.

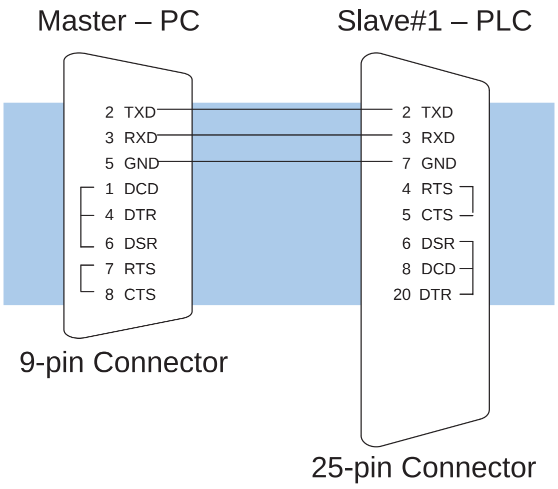


The Example Cable In our example configuration we have:

- A personal computer as the master station
- A DL440 PLC with a built-in ***Direct***Net port as slave station #1

Since this is a simple point-to-point network, we can use RS232C communications.

RS232C



Step 2: Select the Communication Settings

Set the Personal Computer as the Master Station

The next step is to set the personal computer communication parameters (in the BASIC program). The parameters are typically set with software on the personal computer. The best place to find this information is in the documentation that came with your computer. It is important the communication settings for both the master and the slave(s) are the same. We will use the following settings in our example.

- Baud Rate — 9600 baud
- Parity — None

Set the PLC Switches for the Slave Station

Slave station #1 is a DL405 PLC. If you look at the back of the DL405 CPU you will notice a small bank of switches. Switches 2, 3, and 4 are used to set the communication parameters for the bottom communication port.

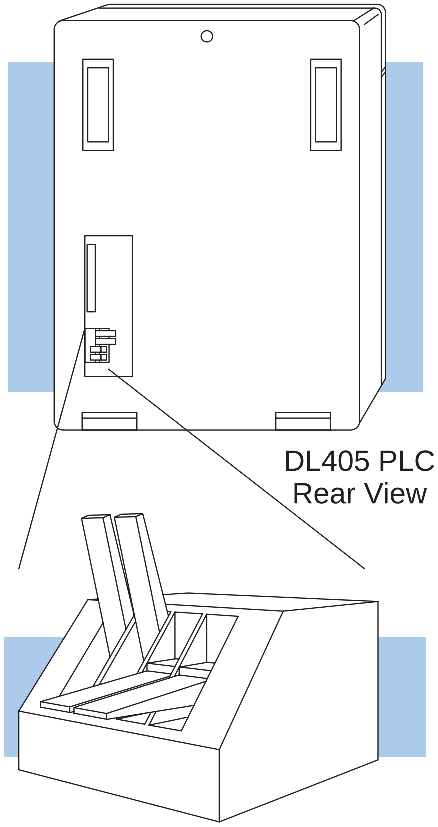
Switch 2 — This switch selects the CPU slave address. If this switch is On, then an address of 1 is used. If the switch is Off, then you can use a programming device to set the address.

Switch 3 & 4 — These switches select the baud rate for the bottom port.

Baud	3	4
300	OFF	OFF
1200	OFF	ON
9600	ON	OFF
19200	ON	ON

Set switch 2 to OFF – Address 1.
Set switch 3 ON and 4 OFF – 9600 Baud

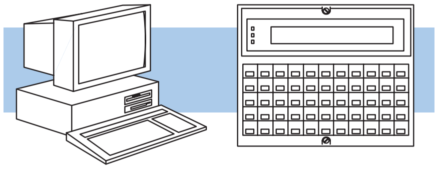
Switch 1 does not apply to the networking example. It is used to select the battery low indictor operation.


DL405 PLC Rear View

Set the Station Address for the Slave Station

With ***Direct***SOFT, use AUX 56 from the Auxiliary functions menu to set the port parameters.

With the DL405 Handheld Programmer, use AUX 56 to set the port parameters. The following example shows how to use the handheld programmer to set the address.



DirectSOFT

DL405 Handheld Programmer

NOTE: The PLC port address is set in decimal, not hexadecimal.

Select AUX 56

CLR CLR AUX 5 6 ENT ENT

```
AUX 5* CPU CFG
AUX 56 CPU N/W ADDRESS
```

Enter the Address (in decimal)

0 1 ENT

Set Address to 1 (decimal)

```
AUX 56 CPU N/W ADDRESS
N/W  #  01    ▯
```

Select HEX or ASCII

To change the mode press .... →

then press enter. ENT

Select ASCII mode for data transfer.

```
AUX 56 CPU N/W ADDRESS
HEX / ASCII
```

Select the Parity Option

To change the parity press .... →

then press enter. ENT

Set Parity to none.

```
AUX 56 CPU N/W ADDRESS
NONE / ODD
```

Clear the Display

To clear the display press.... CLR

```

OK
```

Step 3: Create the Communications Program

_Direct_NET Programs

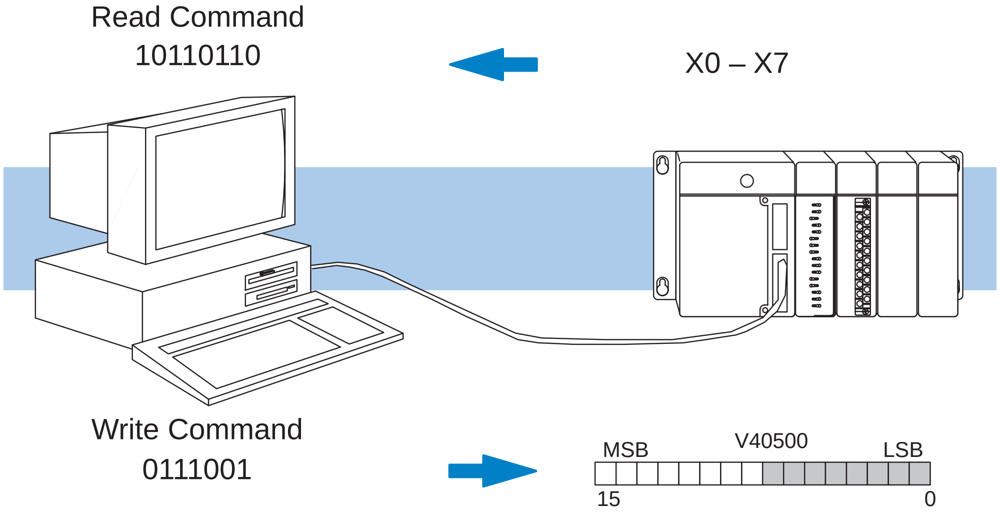
The communications program used with a hosted network is more complex than the simple RLL instructions used with the other configurations, but the concept is the same. The host is the ***Direct*NET** master and must use a ***Direct*NET** protocol communications program to initiate all network requests to read or write data. These communication programs can be written in many different languages, such as BASIC, C, etc. and must include the appropriate ***Direct*NET** protocol commands.

Example Program

We've provided two host programs for this example. The programs are written in BASIC, which is provided with many personal computers.

Goal:

1. Read a bit pattern from an input module (X0 – X7).
2. Write a bit pattern to the first output module (Y0–Y7). (We'll actually use V40500, which is the V-memory location that corresponds to Y0–Y7).



Entering the Program

You can use most any file editor to enter the program. Make sure your editor does not embed formatting codes. For example, most word processing packages and/or desktop publishing packages embed codes that describe the page layout. These types of packages are generally not suitable for entering programs.

Slave Station Programs

The slave station does not require any programming in this example.

NOTE: The DL405 CPU must be in the ASCII mode for the following BASIC programs to work.

Example Program to Read Data

The following program will read X0–X7 from Slave Station #1.

```
10    REM Program to read X0–X7 from a 405 PLC
20    REM
30    REM Define all variables
40    REM
50    REM Change the slave address in HEX at line 60 if required.
60    SLAVEADDRESS=&H1
70    DATATYPE$=CHR$(&H32)
80    DATAADDR$=CHR$(&H30)+CHR$(&H31)+CHR$(&H30)+CHR$(&H31)
90    COMPLETEBLK$=CHR$(&H30)+CHR$(&H30)
100   PARTBLK$=CHR$(&H30)+CHR$(&H32)
110   MASTERADDR$=CHR$(&H30)+CHR$(&H30)
120   NORMAL$=CHR$(&H4E)
130   SLAVEADDR$=HEX$(SLAVEADDRESS)
140   IF LEN(SLAVEADDR$)<2 THEN SLAVEADDR$=”0”+SLAVEADDR$
150   OFFSETADDR$=CHR$(&H20+SLAVEADDRESS)
160   ENQ$=CHR$(&H5)
170   ACK$=CHR$(&H6)
180   SOH$=CHR$(&H1)
190   STX$=CHR$(&H2)
200   ETB$=CHR$(&H17)
210   NAK$=CHR$(&H15)
220   ETX$=CHR$(&H3)
230   EOT$=CHR$(&H4)
240   RED$=CHR$(&H30)
250   DATA$=””
260   REM
270   REM Build enquiry and header
280   REM
290   SLAVEENQ$=NORMAL$+OFFSETADDR$+ENQ$
300   HEADER$=SOH$+SLAVEADDR$+RED$+DATATYPE$+DATAADDR$
310   HEADER$=HEADER$+COMPLETEBLK$+PARTBLK$+MASTERADDR$+ETB$
320   REM
330   REM Begin active program
340   REM
350   CLS
360   OPEN ”COM1:9600,N,8,1,RS,DS” AS #1
370   LRC=0
380   GOSUB 770
390   RETRY=0
400   REM              Do Enquiry
410   PRINT #1,SLAVEENQ$;
420   INPUT #1,ENQRES$:PRINT ”Enquire Response = ”;ENQRES$
430   IF MID$(ENQRES$,3,1)=ACK$ THEN GOTO 480
440   RETRY=RETRY+1
450   IF RETRY>2 THEN GOTO 760
460   GOTO 410
```

```
470   REM            Do Header
480   RETRY=0
490   PRINT #1,HEADER$;LRC$;
500   INPUT #1,HEADERRESP$:PRINT ”Header Response = ”;HEADERRESP$
510   IF MID$(HEADERRESP$,1,1)=ACK$ THEN GOTO 560
520   RETRY=RETRY+1
530   IF RETRY>2 THEN GOTO 760
540   GOTO 490
550   REM            Get the data
560   RETRY=0
570   INPUT #1,DAT$
580   GOSUB 850
590   IF VALUELRC$=DATLRC$ THEN GOTO 640
600   PRINT #1,NAK$;
610   RETRY=RETRY+1
620   IF RETRY>2 THEN GOTO 760
630   GOTO 570
640   REM            Print the data to the screen
650   PRINT ”Data = ”;VALUE$
660   REM            Do the ACK
670   RETRY=0
680   PRINT #1,ACK$;
690   INPUT #1,ACKRESP$:PRINT ”EOT = ”;ACKRESP$
700   IF MID$(ACKRESP$,1,1)=EOT$ THEN GOTO 750
710   RETRY=RETRY+1
720   IF RETRY>2 THEN GOTO 760
730   GOTO 680
740   REM            Do the EOT
750   PRINT #1,EOT$;
760   CLEAR:CLOSE:END
770   REM
780   REM Routine to calculate LRC
790   REM
800   FOR I=2 TO (LEN(HEADER$)–1)
810   LRC=LRC XOR ASC(MID$(HEADER$,I,1))
820   NEXT I
830   LRC$=HEX$(LRC):IF LEN(LRC$)<2 THEN LRC$=”0”+LRC$
840   RETURN
850   REM
860   REM Routine to remove data from data packet, Get LRC for check in main
870   REM
880   VALUELRC=0
890   DATLRC$=””
900   FOR I=2 TO (LEN(DAT$)–3)
910   VALUE$=VALUE$+MID$(DAT$,I,1)
920   VALUELRC=VALUELRC XOR ASC(MID$(DAT$,I,1))
930   NEXT I
940   DATLRC$=MID$(DAT$,(LEN(DAT$)–1),2)
950   VALUELRC$=HEX$(VALUELRC)
960   IF LEN(VALUELRC$)<2 THEN VALUELRC$=CHR$(&H30)+VALUELRC$
970   RETURN
```

Example Program to Write Data

The following program will write a value to Y0 – Y7 in Slave Station #1.

```
10     REM Program to write X0–X7 to a 405 PLC
20     REM
30     REM Define all variables
40     REM
50     REM Change the slave address in HEX at line 60 if required
60     SLAVEADDRESS=&H1
70     DATATYPE$=CHR$(&H33)
80     DATAADDR$=CHR$(&H30)+CHR$(&H31)+CHR$(&H30)+CHR$(&H31)
90     COMPLETEBLK$=CHR$(&H30)+CHR$(&H30)
100    PARTBLK$=CHR$(&H30)+CHR$(&H32)
110    MASTERADDR$=CHR$(&H30)+CHR$(&H30)
120    REM
130    NORMAL$=CHR$(&H4E)
140    SLAVEADDR$=HEX$(SLAVEADDRESS)
150    IF LEN(SLAVEADDR$)<2 THEN SLAVEADDR$=”0”+SLAVEADDR$
160    OFFSETADDR$=CHR$(&H20+SLAVEADDRESS)
170    ENQ$=CHR$(&H5)
180    ACK$=CHR$(&H6)
190    SOH$=CHR$(&H1)
200    STX$=CHR$(&H2)
210    ETB$=CHR$(&H17)
220    NAK$=CHR$(&H15)
230    ETX$=CHR$(&H3)
240    EOT$=CHR$(&H4)
250    RITE$=CHR$(&H38)
260    DATA$=””
270    REM
280    REM Build enquiry and header
290    REM
300    SLAVEENQ$=NORMAL$+OFFSETADDR$+ENQ$
310     HEADER$=SOH$+SLAVEADDR$+RITE$+DATATYPE$+DATAADDR$
320    HEADER$=HEADER$+COMPLETEBLK$+PARTBLK$+MASTERADDR$+ETB$
330    REM
```

```
340   REM Begin active program
350   REM
360   CLS
370   OPEN ”COM1:9600,N,8,1,RS,DS” AS #1
380   LRC=0:DATLRC=0
390   INPUT ”ENTER DATA STRING (2 DIGITS, 0–F) ” , VALUE$
400   GOSUB 650
410   GOSUB 730
420   RETRY=0
430   PRINT #1,SLAVEENQ$;
435   INPUT #1,ENQRESP$:PRINT ”Enquire Response = ”;ENQRESP$
440   IF MID$(ENQRESP$,3,1)=ACK$ THEN GOTO 490
450   RETRY=RETRY+1
460   IF RETRY>2 THEN GOTO 640
470   GOTO 430
490   RETRY=0
500   PRINT #1,HEADER$;LRC$;
510   INPUT #1,HEADERRESP$:PRINT ”Header Response = ”;HEADERRESP$
520   IF MID$(HEADERRESP$,1,1)=ACK$ THEN GOTO 560
530   RETRY=RETRY+1
540   IF RETRY>2 THEN GOTO 640
550   GOTO 500
560   RETRY=0
570   PRINT #1,DAT$;DATLRC$;
580   INPUT #1,DATRESP$:PRINT ”Data Response = ”; DATRESP$
590   IF MID$(DATRESP$,1,1)=ACK$ THEN GOTO 630
600   RETRY=RETRY+1
610   IF RETRY>2 THEN GOTO 640
620   GOTO 570
630   PRINT #1,EOT$;
640   CLEAR:CLOSE:END
650   REM
660   REM Routine to calculate LRC
670   REM
680   FOR I=2 TO (LEN(HEADER$)–1)
690   LRC=LRC XOR ASC(MID$(HEADER$,I,1))
700   NEXT I
710   LRC$=HEX$(LRC):IF LEN(LRC$)<2 THEN LRC$=”0”+LRC$
720   RETURN
730   REM
740   REM Routine to put data into packet and calulate data LRC
750   REM
760   DAT$=CHR$(&H2)+VALUE$+CHR$(&H3)
770   FOR I=2 TO (LEN(DAT$)–1)
780   DATLRC=DATLRC XOR ASC(MID$(DAT$,I,1))
790   NEXT I
800   DATLRC$=HEX$(DATLRC):IF LEN(DATLRC$)<2 THEN DATLRC$=”0”+DATLRC$
810   RETURN
```

Step 4: Start the Network.

Execute the PC Master Program

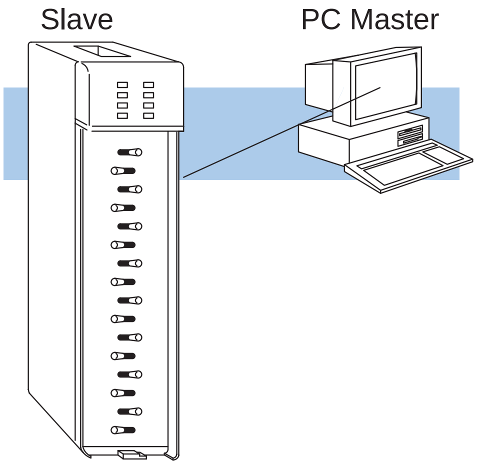
This program was designed to work with GWBasic. Complete the following steps to execute the example program.

1. Start GWBasic.
2. Load the file you used to create the read example program.

Verify the Read Command

Now you can change the I/O simulator switch settings and verify the communications.

1. Set every other switch to the ON position .
2. Execute the program. Look at the screen to verify the data matches the I/O simulator switch settings.
3. Change the I/O simulator switch settings at random and execute the program again. Notice how the data changes.

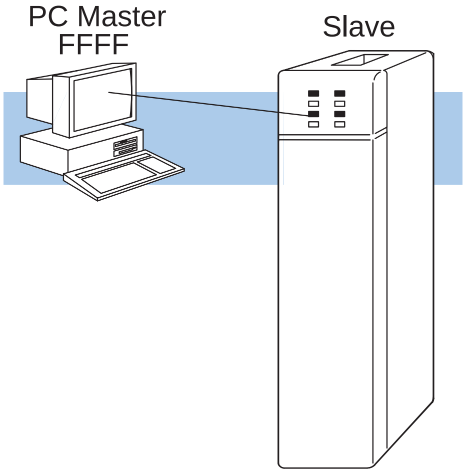


Verify the Write Command

You can change the bit pattern for the output module to verify it is working correctly. For example, enter FFFF. This should turn on the first eight output points in slave station #2.

Notice the numbers are entered in hexadecimal format. Here's a cross reference for a few patterns.

- FFFF – all on
- 0000 – all off



What should I do if it isn't working correctly?

Troubleshooting Steps

If the network does not seem to be working correctly, check the following items.

1. Cable and connections. Incorrectly wired cables and loose connectors cause the majority of problems. Verify you've selected the proper cable configuration and check to see the cable is wired correctly.
2. Dipswitch settings. Make sure you've set the slave stations to match the communication parameters required by the master station (operator interface or personal computer).
3. Communications program. Verify the program has been entered as shown in the example.
4. If the network still doesn't work correctly go to Chapter 7, Network Operation and Troubleshooting, and use the troubleshooting charts.