Section 2 Host Communication

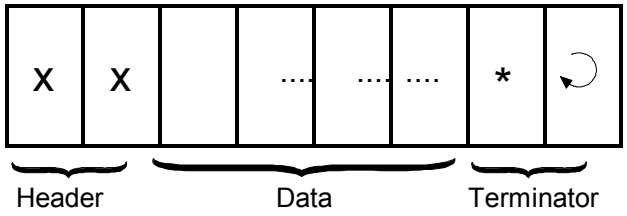
While a H-series programmable logic controller is running, a host computer may send commands in the form of ASCII strings to the controller to read or write to the inputs, outputs, relays, timers and counters. These ASCII commands are known as the "host-link commands" and are to be serially transmitted (via RS232C or RS485 port) to and from the controller. The standard serial port settings for communication are: 9600 baud, 8 data bit, 1 stop bit, no parity. Some models of PLC allow the serial port to be configured to another baud rate using the "BW" command described in Section 3.

All H-series PLCs support both point-to-point (one-to-one) and multi-point (one-to-many) communication protocols and hardware. Each protocol has a different command structure as described below:

1. Point-to-point Communication

In a point-to-point communication system, the host computer's RS232C serial port is connected to the PLC. At any one time, only one controller may be connected to the host computer. The host-link commands do not need to specify any controller ID code and are therefore of simpler format, as shown below:

Command/Response Block Format (Point to Point)



Each command block starts with a two-byte ASCII character header, followed by a number of ASCII data and ends with a terminator which comprises an '*' character and a carriage return (ASCII value = 13_{10}). The purpose of the command is denoted by the header, e.g. RI for Read Input, WO for Write Output, etc. The data are usually the hexadecimal representation of numeric data. Each byte of binary data is represented by two ASCII characters (00 to FF).

To begin a communication session, the host computer must first send one byte of ASCII character: Ctrl-E (=05Hex) via its serial port

to the controller. This informs the controller that the host computer wishes to send a (point-to-point) host-link command to it. Thereafter, the host computer must wait to receive an echo of the Ctrl-E character from the controller. Reception of the echoed Ctrl-E character indicates that the controller is ready to respond to the command from the host computer. At this moment, the host computer must immediately send the *command block* to the controller and then wait to receive the *response block* from the controller. The entire communication session is depicted in Figure 2-1.

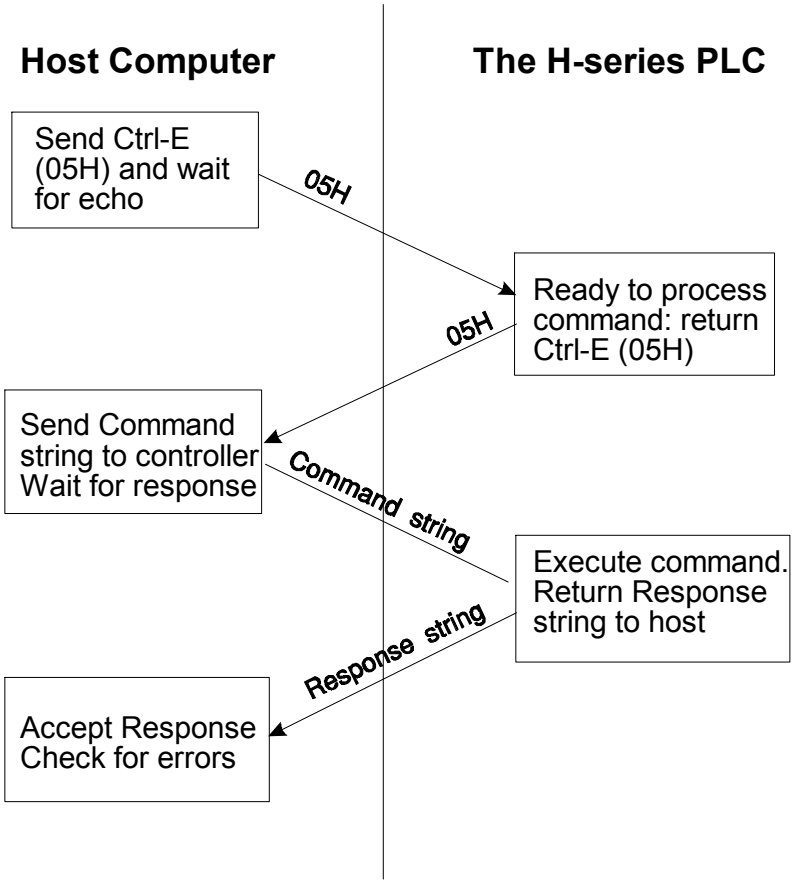
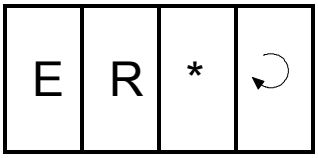


Figure 2-1

After the controller has received the command, it will send a response block back to the host computer and this completes the communication session. If the command is accepted by the controller, the response block will start with the same header as the command, followed by whatever information that is requested by the command and the terminator.

If an unknown command is received or if the command is illegal (such as access to an unavailable output or relay channel), the following **error response** will be received:

Error Response Format



The host computer program should always check the returned response for possibilities of errors in the command and take necessary actions.

2. Multipoint Communication system

In this system, one host computer may be connected to multiple H-series controllers on a RS485 network. The T28H, T44H and T64H controllers all have built-in RS485 ports. In the case of T20H a RS485 port is available on the expansion board TXP24, or available as an optional unit. T28H and T44H have only one serial port each, which can be connected either to the RS232C or the RS485, depending on the setting of their respective DIP switch SW1-2 (please refer to their respective Installation Guide).

In the case of T64H-Relay, two independent serial ports are available: one is connected to the RS232C while the other is connected to the RS485 port. Both serial ports may be accessed simultaneously by two computers, provided that only point-to-point commands are sent via the RS232C port and only multi-point commands are sent via the RS485 port.

2.1 Networking with RS485

The built-in RS-485 interface allows the H-series controllers to be networked to a host computer using very low cost twisted-pair cables. RS-485 allows up to 32 controllers (including the host computer node) to be networked together, at baud rates up to 38,400 bps. The twisted-pair cable goes from node to node

in a daisy chain fashion and should be terminated by 120 ohm resistors at each end as shown below.

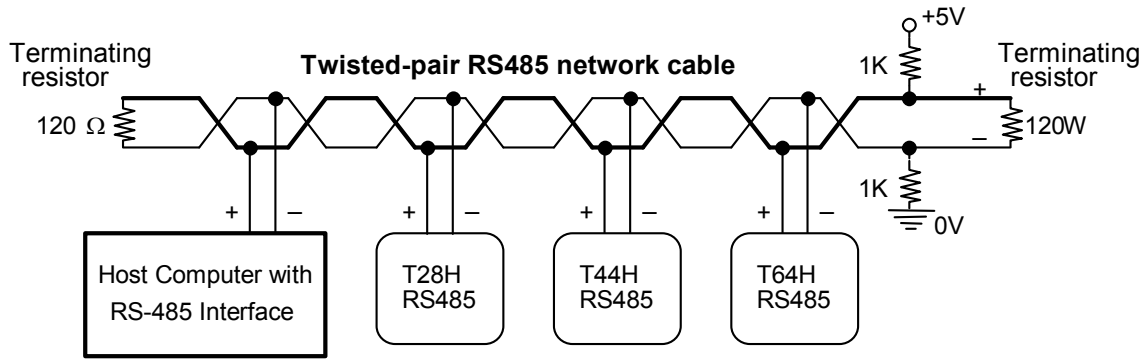


Figure 2-2

Note that the two wires are not interchangeable so they must be wired the same way to each controller. The maximum wire length should not be more than 1000 metres. RS-485 uses balanced or differential drivers and receivers, this means that the logic state of the transmitted signal depends on the differential voltage between the two wires and not on the voltage with respect to a common ground.

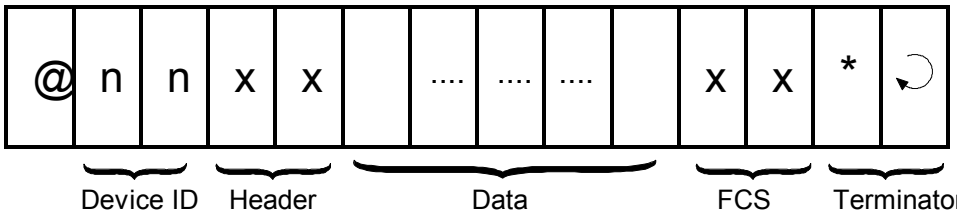
RS485 is a half-duplex network, i.e. the same two wires are used for both transmission of the command and reception of the response. Of course, at any one time, only one transmitter may be active. The H-series controllers implement master/slave network protocols. The network requires a master controller which is typically a microcomputer equipped with a RS485 interface. In the case of an IBMPC or AT, you can purchase a RS-485 adapter card or a RS232C-to-RS485 converter and connect it to the RS232C serial port.

All H-series controllers are slave devices. They do not initiate commands to each other or to the master. Only the master can issue commands to the slave controllers. To transmit a command, the master controller must first enable its RS-485 transmitter and send a multi-point command to the network of controllers (please refer to Section 3 for explanation on the command formats and protocols for multi-point commands). After the last stop bit has been sent, the master controller must relinquish the RS485 bus by disabling its RS485 transmitter and

enabling its receiver. At this point the master will wait for a response from the slave controller that is being addressed.
Since the command contains the ID of the target controller, only the controller with the correct ID would respond to the command by sending back a response string. For the network to function properly, it is obvious that no two nodes can have the same ID. Refer to Section 3 under the "IW" command on how to set the device ID. Also, all nodes must be configured to the same baud rate.

As there will be times when no transmitters are active (which leaves the wires in "floating" state), it is a good practice to ensure that the RS-485 receivers will indicate to the CPUs that there is no data to receive. In order to do this, we should hold the twisted pair in the logic '1' state by applying a differential bias to the lines using a pair of 1KΩ resistors connected to a +5V and 0V supply as shown in Figure 2-2. Otherwise, random noise on the pair could be falsely interpreted as data. Also, care should be taken to ensure that the power supplies for all the controllers are properly isolated from one another and from the main so that no large ground potential differences exist between any controllers on the network.

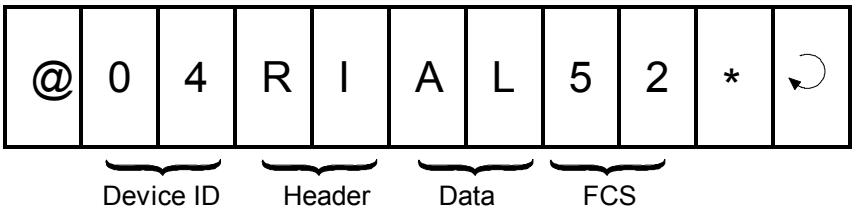
2.2 Command/Response Block Format (Multipoint)



Each command block starts with the character "@" and two-byte hexadecimal representation of the controller's ID (00 to FF), and ends with a two-byte "Frame Check Sequence" (FCS) and the terminator. FCS is provided for detecting communication errors in the serial bit-stream. If desired, the command block may omit calculating the FCS simply by putting the characters "00" in place of the FCS.

Calculation of FCS

The FCS is 8-bit data represented by two ASCII characters (00 to FF). It is a result of Exclusive OR sequentially performed on each character in the block, starting from @ in the device number to the last character in the data. An example is as follow:



@	0100 0000
	XOR
0	0011 0000
	XOR
4	0011 0100
	XOR
R	0101 0010
	XOR
I	0100 1001
	XOR
A	0100 0001
	XOR
L	0100 1100
	0101 0010 = 52_{16}

Value 52_{16} is then converted to ASCII characters '5' (0011 0101) and '2' (0011 0010) and placed in the FCS field.

FCS calculation program example

The following C function will compute and return the FCS for the "string" passed to it.

```
unsigned char compute_FCS(unsigned char *string)
{
   unsigned char result;
   result = *string++;     /*first byte of string*/
   while (*string)
      result ^= *string++; /* XOR operation */
```

```
    return (result);
}
```

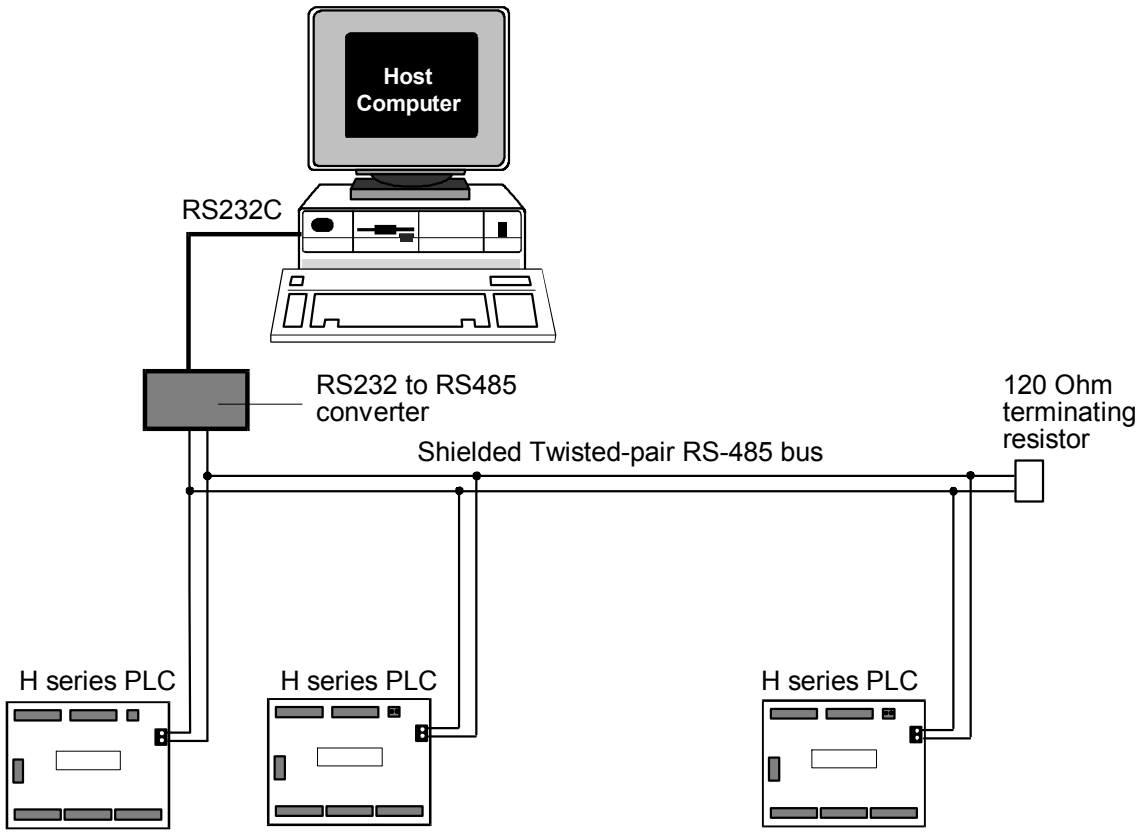


Figure 2-3 - A Multi-drop RS485 network

2.3 Communication Procedure

Unlike the point-to-point communication protocol, the host computer must NOT send the CTRL-E character before sending the command block. After the host computer has sent out the multi-point host link command block, only the controller with the correct device ID will respond. Hence it is essential to ensure that every controller on the RS485 network assumes a different ID. Otherwise contention may occur (i.e. two controllers simultaneously sending data on the receiver bus, resulting in garbage data being received by the host). On the other hand, if none of the controller IDs match that specified in the command block, then the host computer will receive no response at all.

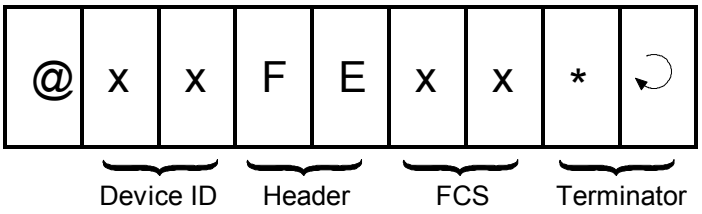
The PLC automatically recognizes the type of command protocols (point-to-point or multi-point) sent by the host

computer and it will respond accordingly. If a multi-point command is accepted by the controller, the response block will start with a character '@', followed by its device ID and the same header as the command, then the data requested by the command, a response block FCS and the terminator.

Framing Errors

When the controller receives a multi-point host link command block, it computes the FCS of the command and compares it with the FCS field received in the command block. If the two do not match, then a "framing error" has occurred. The controller will send the following Framing Error Response to the host:

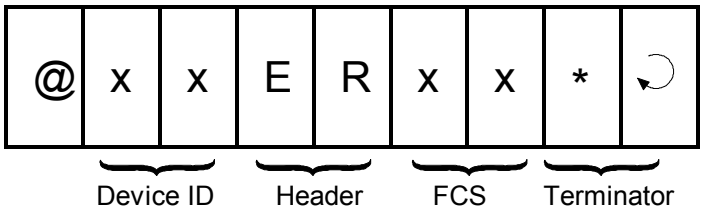
Framing Error Response Block (Multi-point only)



Command Errors

If an unknown command is received or if the command is illegal (such as an attempt access an unavailable output channel), the following **error response** will be received:

Error Response Format



The host computer program should always check the returned response for possibilities of errors in the command and take necessary action.

Sample Program

To help you get started, a sample networking program "HOST485.C" for the IBMPC has been written in Turbo C and is provided in the TRiLOGI Version 3.1/3.2 distribution diskette. This program works with an RS232-to-RS485 adapter attached to either COM1: or COM2: port of the PC. In this adapter, the direction of the RS485 transceiver is controlled by the /RTS line of the serial port. The RS485 transmitter is enabled only when the /RTS line is active, otherwise the transmitter is disabled and the receiver is enabled. The logic states of the /RTS line is determined by bit 1 of the "Modem control" register (at I/O address 3FC for COM1: and 2FC for COM2:) of the 8250 UART used in the PC serial adapter. You will need to modify some or all the following hardware-dependent functions if your RS485 adapter uses different UART and/or different method of controlling the transceiver's direction:

```
transmit485(), receive485(), serial_out(),
serial_status()  and serial_in().
```

3. USING NETWORK TRiLOGI

If you have connected the RS485 interface of a few H-series PLCs into a multi-drop network, you may download program or monitor the operation of any PLC from a single host computer connected to the network. Note that the host computer's RS485 adapter must be approved by Triangle Research to be compatible with the NETWORK-TRiLOGI.

The network version of TRiLOGI Version 3.2 is available as a separate file: `"TL32NET.EXE"` This program is almost identical to TL32.EXE, the exception being an additional command item `"Select Controller  Ctrl-I"` in the "Controller" main menu. For this program to function properly, each PLC on the network must assume a unique ID. Use the "Host Link Command" in the "Controller" menu and the command "BWxx" (xx = 00 to FF is the controller ID) to set the ID of each controller separately before connecting them to the network. When running the "TL32NET.EXE" program you can easily select any specific PLC to work with by specifying its ID.

After entering the device ID, Network TRiLOGI will automatically query the PLC of that particular ID for its source file name. It then searches the current directory of the PC's disk drive for a matching file. If found, it will prompt the user to confirm whether he/she wishes to open the source file. The selected controller is then available for program-downloading or On-line/Ladder monitoring as per the normal TRiLOGI operation. To switch to another PLC, simply press <Ctrl-I> and enter another ID. This program offers a quick way to test a new RS485 network.

If a communication errors occur, check to see if the PLC's ID has been properly defined. Next check for loose or incorrect wiring to the RS485 terminal. In the case of the T28H and T44H PLCs, ensure that the serial port selection DIP switch has been properly set. The 8-pin DIP IC - SN75176 provides the RS485 interface and it may be necessary to replace it if it is damaged during installation as a result of over voltage/current or prolonged short-circuit of its two terminals, etc.

4. Trouble-Shooting RS485 Network

a) Single faulty device

If a single device on the RS485 network becomes inaccessible, problems can be isolated to this particular device. Check out for loose or broken wiring or wrong DIP switch settings. Also double check the device ID using the host link command "IR*" sent via the RS232C port of the PLC. If all attempts fail, either replace the entire PLC or the SN75176 chip which handle the RS485 interfacing and try again.

b) Multiple faulty devices

If all the PLCs are inaccessible by the host computer, it may possibly be due to a faulty RS232C-to-RS485 converter at the PC, If this is the case, disconnect the RS485 converter from the network and check it using a single PLC. Also check to ensure that the converter has been properly configured with the correct DIP-switch settings. Replace the converter if it is confirmed to be faulty. Next check the wire from the converter to the begining of

the network. A broken wire here can lead to the failure of the entire network.

Since a RS485 network links many PLCs together electrically and in a daisy chain fashion, problems occurring along the RS485 network sometimes affect the operation of the entire network. For example, a broken wire at the terminal of one node may mean that all the PLCs connected after this node become inaccessible by the master. If the RS485 interface of one of the PLCs has short-circuited because of component failure, then the entire network goes down with it too. This is because no other nodes is able to assert proper signals on the two wires that are also common to the shorted device.

Hence when trouble-shooting a faulty RS485 network, it may be necessary to isolate all the PLCs from the network, then connect one PLC at a time back to the network, starting from the node nearest to the host computer. Use network TRiLOGI to check communication with each PLC until the faulty unit has been tracked down.