



SLC™ Programmable Controller Communications Protocol

| | | |
|-----------------|---|--------|
| Contents | SLC Communications Protocol | Page 1 |
| | Format of Transmission | 1 |
| | Specific Commands | 4 |
| | The Memory Map | 8 |
| | User Address to Controller Address Table | 10 |
| | User Program Addresses | 10 |
| | Connector Pin Configuration | 12 |
| | SLC Personal Computer Interface Converter | 12 |

IMPORTANT INFORMATION

The SLC communications protocol provides the necessary information to develop computer software which can access or change selected registers or variables stored in the SLC controller's memory. The protocol is compatible with SLC 100 controller firmware revision number (FRN) 5, and succeeding revisions. It is compatible with all firmware versions of the SLC 150 controller.

Any person using this information to develop software assumes sole responsibility for its application and safety. In no event will the Allen-Bradley Company be responsible or liable for indirect or consequential damages resulting from the use of this protocol information.

SLC Communications Protocol

The SLC communications protocol was designed as a single device system, with one control device per SLC controller. The controllers are not addressable for use on a multi-drop communications system.

Communications between the SLC controller and external devices is via a serial RS-422 communications link. The protocol uses ASCII characters, even parity, and operates at 9600 baud. All numbers are in hexadecimal coded binary format. The controller ignores any transmission not in this format. Words are structured as follows:
7 – data bits, 1 – parity bit, and 1 – stop bit.

Format of Transmission

The following command format is used in the communications between an external device and the controller in the SLC family.

Commands are sent in the following format:

<leading character> <byte count> <command code>
<parameters> [ID] [CRC] <cr>

The various fields are defined as follows:

<leading character>
One ascii character. # is used for all commands.

AB Spares

**Format of
Transmission**
(continued)

<byte count >

Two ascii hex characters representing the total sum of the bytes within the <command code > and <parameters > fields.

<command code >

Two ascii hex characters as detailed in the specific sections below.

<parameters >

Ascii hex characters as required per the command code.

[ID]

Two digit ascii hexadecimal characters identifying the message number. The programming device must increment this number by one for each transmission to the controller, with the exception of transmission of messages containing ' ; Z '. The controller will verify that sequential order is maintained on all transmissions not containing ' ; Z '.

An otherwise valid message with an [ID] of 00 sent to the controller is always accepted, and restarts the message identifier numbering sequence, with the next message to have an expected [ID] of 01. The identifier will roll over from 127D to 01 to eliminate unintended resynchronization caused by a message ID of zero. Upon power up, the controller expects an [ID] of 00. If a message contains an unexpected [ID], the controller will respond with:

; I {id} [CRC] <cr >.

where {id} is the last valid [ID].

[CRC]

Cyclical redundancy check. Two ascii hex characters.

[CRC] is a 2 digit hexadecimal number representing the Cyclical Redundancy Check for the message, derived from the <byte count > <command code > <parameters > and [ID] fields using the polynomial

$$X^{**8} + X^{**6} + X^{**5} + X^{**4} + 1$$

The following BASIC program [IBM BASIC (A)] may be used to compute the CRC for a string of characters:

```

10 DIM G(16)
20 FOR I=0 TO 15
30 READ G(I)
40 NEXT I
50 INPUT "Enter data string" ;A$
60 CHECKSUM=0
70 FOR COUNT=1 TO LEN(A$)
80 CHAR=ASC (MID$(A$, COUNT, 1))
90 FOR J=1 TO 2
100 INDEX=(CHECKSUM XOR CHAR) AND &HF
110 CHECKSUM=INT (CHECKSUM / 16)
120 CHECKSUM=CHECKSUM XOR G (INDEX)
130 CHAR=INT (CHAR / 16)
140 NEXT J
150 NEXT COUNT
160 PRINT "The SLC CRC checksum byte is " ; HEX$(CHECKSUM)
170 GOTO 50
180 DATA 0, 23, 46, 57, 92, 75, 114, 101, 184
190 DATA 175,150,129,228,243,202,221

```

**Format of
Transmission**
(continued)

Note that the start character ;, ,, or # is not sent through the CRC computation. Also the CRC computation is done on the character either before parity is applied on transmission or after it has been stripped off on reception.

<cr>
carriage return.

The controller responds as follows:

The SLC does not respond to any message which is received with a parity, CRC, or incorrect format error.

Certain commands and/or <parameters> must meet mode and/or range conditions. If a command or <parameters> is illegal, the SLC will respond with a message as follows:

;E [ID] [CRC] <cr>

If the SLC detects a fault, it will respond with a message as follows:

;T [ID] [CRC] <cr>

Since this message is not sent in response to a stimulus from the user, the ID is meaningless. Upon receiving this message the user should read the bytes at addresses 84H and 85H to find out the reason for this message.

The format of the error word is:

| | | | | | | | | |
|---------|----|----|----|----|---|----|---|----|
| Address | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 84H | 62 | 4 | 6 | 3 | 8 | 5 | 2 | 1 |
| 85H | | 26 | 25 | 24 | 9 | 51 | 7 | 51 |

The numbers represent the error number displayed by the programmer for the corresponding error bit set.

If a response from the SLC is missing or garbled (e.g. parity, format, or CRC errors) the user can request a retransmission via the command:

;Z [ID] [CRC] <cr>

The SLC ignores the ID in this message and retransmits its most recent acknowledge or data message with the same ID as was sent previously. This allows the user to determine if the last message was received by the SLC since the message ID will not match if it was garbled. In this case the command may be safely repeated. Normally the SLC will respond within 0.3 seconds to any transmission. The exceptions are edit commands and mode changes.

Specific Commands

The following paragraphs describe the various commands sent by the user and the responses of the SLC controller.

**Upload and Download
– Program Mode**

Once the user begins to upload or download a user program, the SLC will not allow a mode or load direction change until the entire user program (all 7BH blocks for SLC 100, all A5H blocks for SLC 150) has been transferred. The user can reset the procedure by re-entering the Program mode. However, if a download had been started, then all SLC NVRAM, containing the user program, will be cleared.

Uploading valid only in the Program mode:

The user sends:

```
#0100 [ID] [CRC] <cr>
```

meaning upload the user program in SLC NVRAM to the user's terminal. This command is only valid in the program mode. (See the note following the DOWNLOAD command)

SLC responds with the command string:

```
:<block #> <15 bytes> [ID] [CRC] <cr>
```

where <block #> is two characters of ascii hex between 00H and 7AH for SLC 100 and 00H and A4H for SLC 150. The user must initiate the transfer block by sending the above command after each successive block is received from the SLC. This process must be incremented sequentially.

Downloading valid only in the Program mode:

The user sends:

```
#1101 <block number> <15 bytes> [ID] [CRC] <cr>
```

meaning download a program block from the user's terminal to the SLC NVRAM. This command is valid only in the program mode.

<block number> is two characters ascii hex, 00H through 7AH for SLC 100 and 00H through A4H for SLC 150, and must be sent in sequential order.

SLC responds after each transmission, if valid:

```
;R [ID] [CRC] <cr>
```

If a power down cycle occurs during download, the download process can be reinitiated as described in the note below.

Note: Once the SLC begins to upload or download a user program, a mode change or a load direction change will not be allowed until the entire transfer is complete. The user can reset the procedure by re-entering the Program mode via the communication protocol; however, if a download had been started, then all of the SLC NVRAM will be cleared. All blocks of data must be transferred, regardless of the length of the user's ladder diagram. Data will be loaded as follows:

Upload and Download
- Program Mode
 (continued)

| | | <u>Block Number</u> | <u>Data</u> |
|-------|-------|--|-------------|
| SLC | SLC | | |
| 100 | 150 | | |
| 00H | 00H | First 15 bytes of user program memory | |
| 01H | 01H | Next 15 bytes of user program memory | |
| | | | |
| | | | |
| 75H | 9FH | Last 15 bytes of user program memory | |
| 76H | A0H | Three bytes containing end of program information plus first 12 bytes of preset table. | |
| 77H | A1H | Next 15 bytes of preset table | |
| 78H | A2H | Next 15 bytes of preset table | |
| 79H | A3H | Next 15 bytes of preset table | |
| 7AH | A4H | Last 7 bytes of preset table plus 8 bytes of fill. | |

(User ignores fill on upload and must send zero fill on download)

Mode Selection
- All Modes

The user sends:

#04030101 <data byte> [ID][CRC] <cr>

This command is allowed in any mode. <data byte> is two characters ascii hex and is limited to:

80H - Run mode

8CH - Save to EEPROM mode

82H - Continuous test mode

06H - Program mode for upload and download; also see "Upload and Download -Program Mode" above.

SLC responds, if valid:

;R[ID][CRC] <cr>

Read Device FRN ID Tag
- All Modes

The user sends:

#0108 [ID][CRC] <cr>

This command is allowed in any mode.

SLC responds, if valid:

:021FFE <device type> <FRN #> [ID][CRC] <cr>

where <device type> is a masked (fixed at factory) code, two characters ascii hex, indicating the controller type (e.g. 10 for SLC ; 15 for SLC 150; etc.), and <FRN #> is two characters ascii hex indicating the FRN release number (e.g. 05 for FRN 5; 47 for FRN 47; etc.).

**Reading Data
– Run or Continuous
Test Modes**

The user sends:

#0502 <number of bytes> <address> [ID] [CRC] <cr>
<number of bytes> is four characters ascii hex, and limited to either 0001 or 0002. <address> is four characters ascii hex, and should be limited to:

| | |
|------------------|--|
| 0000H thru 003EH | (timer/counter/sequencer accumulator) |
| 0040H thru 007EH | (timer/counter presets)* |
| 0084H thru 0085H | (error bits) |
| 0086H thru 0095H | (force select table) |
| 0096H thru 009CH | (output state table) |
| 009DH thru 00A5H | (input state table) |
| 00A6H thru 00BAH | (internal relay table) |
| 00BBH | (fine timer bits, auto/man switch) |
| 00BCH thru 00DBH | (sequencer step counters) |
| 00DCH thru 00DFH | (timer/counter/sequencer output table) |
| 00E0H thru 00E3H | (timer/counter/sequencer overflow table) |
| 0102H thru 0111H | (force state table) |
| 0112H | (access code byte) |

SLC responds, if valid:

:<number of bytes> <address> <data byte(s)> [ID] [CRC] <cr>

*Note: When reading timer/counter preset data, Bit 7 of the most significant byte (protect bit) should be masked out so that it does not effect data value.

**Writing Data Words
– Run or Continuous
Test Modes**

The user sends:

#0503 <address> <data word> [ID] [CRC] <cr>
<data word> is four characters ascii hex and limited to be less than 2710H (10,000D). <address> is four characters ascii hex, must be even, and is limited to:

| | |
|------------------|---------------------------------------|
| 0000H thru 003EH | (timer/counter/sequencer accumulator) |
| 0040H thru 007EH | (timer/counter presets) |

Note: Before the data is written, the most significant bit of the data word being overwritten must be 0 (the device must not be protected).

SLC responds, if valid:

;R [ID] [CRC] <cr>

**Writing Data Bytes
– Run or Continuous
Test Modes**

The user sends:

#0403 <address> <data byte> [ID] [CRC] <cr>
<data byte> is two characters ascii hex and <address> is four characters ascii hex, and is limited to:

00A6H thru 00B4H (internal relay table - 701 thru 820 only).

SLC responds, if valid:

;R [ID] [CRC] <cr>

**Reading Sequencer
Preset
– Run or Continuous
Test Modes**

The user sends:

#0404 < rung number > < step number > [ID] [CRC] < cr >
< rung number > is four characters ascii hex. < step number > is two characters ascii hex. The selected rung is verified to contain a sequencer and the step number is tested for validity in the sequencer.

SLC responds, if valid:

:02 < user address > < preset word > [ID] [CRC] < cr >

Note: When reading sequencer preset data, Bit 7 of the most significant byte (protect bit) should be masked out so that it does not effect data value.

**Reading Reset
Accumulator
– Run or Continuous
Test Modes**

The user sends:

#0305 < rung number > [ID] [CRC] < cr >
< rung number > is four characters ascii hex. The selected rung is verified to contain a reset instruction.

SLC responds, if valid:

:02 < user address > < accumulator word > [ID] [CRC] < cr >

Note: When reading reset accumulator data, Bit 7 of the most significant byte (protect bit) should be masked out so that it does not effect data value.

**Writing Sequencer Preset
– Run or Continuous
Test Modes**

The user sends:

#0606 < rung number > < step number > < preset word > [ID] [CRC]
< cr >
< rung number > is four characters ascii hex. < step number > is two characters ascii hex. The selected rung is verified to contain a sequencer and the step number is tested for validity in the sequencer. < preset word > is four characters ascii hex, and checked to be less than 2710H (10,000D) and non-zero. Before the new preset data is written, the most significant bit of the preset being overwritten must be 0 (the device must not be protected).

SLC responds, if valid:

;R [ID] [CRC] < cr >

**Writing Reset
Accumulator
– Run or Continuous
Test Modes**

The user sends:

#0507 < rung number > < accumulator word > [ID] [CRC] < cr >
< rung number > is four characters ascii hex. The selected rung is verified to contain a reset instruction. < accumulator word > is four characters ascii hex, and is checked to be less than 2710H (10,000D). Before the new accumulator data is written the most significant bit of the accumulator being overwritten must be 0 (the device must not be protected).

SLC responds, if valid:

;R [ID] [CRC] < cr >

The Memory Map

| Bytes Used | Hex Address | Bit | | | | | | | |
|------------|-----------------|--|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 64 | 0 ↓ 3F | The 32 timer/counter/sequencer accumulator values, MSB (Most Significant Byte) first, LSB (Least Significant Byte) second. | | | | | | | |
| 64 | 40 ↓ 7F | The 32 timer/counter preset values, MSB first, LSB second. Bit 7 of MSB equals 1 if preset is protected and equals 0 if preset in unprotected. | | | | | | | |
| 2 | 80** 81** | Check sum of last ladder diagram executed. | | | | | | | |
| 2 | 82* 83* | Current rung number. | | | | | | | |
| 2 | 84 85 | Controller error mode bits. | | | | | | | |
| 16 | 86 ↓ 95 | The force select table. | | | | | | | |
| 7 | 96* ↓ 9C* | The output state table. | | | | | | | |
| 9 | 9D* ↓ A5* | The input state table. | | | | | | | |
| 21 | A6* ↓ BA* | The internal relays. | | | | | | | |
| 1 | BB* | Fine timer bits, AUTO/MAN switch. | | | | | | | |
| 32 | BC* ↓ DB* | The 32 sequencer step counters. | | | | | | | |
| 4 | DC* ↓ DF* | Timer/counter/sequencer output bits. | | | | | | | |
| 4 | E0* ↓ E3* | Timer/counter/sequencer overflow bits. | | | | | | | |
| 4 | E4* ↓ E7* | Timer/counter/sequencer immediate rung bits. | | | | | | | |

* **WARNING:** Memory cells indicated with * should not be changed by the controlling device and should be read only. Memory cells indicated with ** should not be accessed in any way. Changing cells indicated with * or ** could result in unpredictable processor operation.

The Memory Map (continued)

| Bytes Used | Hex Address | Bit | | | | | | | |
|--|-------------------|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 4 | E8* ↓ EB* | Reset instruction immediate rung state bits. | | | | | | | |
| 4 | EC** ↓ EF** | Up counter/timer/sequencer last state bits. | | | | | | | |
| 4 | F0** ↓ F3** | Down counter last state bits. | | | | | | | |
| 4 | F4** ↓ F7** | RTF instruction start up (trigger) bits. | | | | | | | |
| 4 | F8** ↓ FB** | Flags to identify sequencer instructions. | | | | | | | |
| 4 | FC** ↓ FF** | RTF instruction identifier bits. | | | | | | | |
| 1 | 100* | RAM size, new check sum flag, watchdog fault. | | | | | | | |
| 1 | 101 | Controller mode byte. | | | | | | | |
| 16 | 102 ↓ 111 | The force state table. | | | | | | | |
| 1 | 112 | Access code byte. | | | | | | | |
| 1770 for SLC 100 2400 for SLC 150 | | User program. | | | | | | | |
| 3 | | End of user program. | | | | | | | |

* **WARNING:** Memory cells indicated with * should not be changed by the controlling device and should be read only. Memory cells indicated with ** should not be accessed in any way. Changing cells indicated with * or ** could result in unpredictable processor operation.

Communications Protocol

User Address to Controller Address Table

Addresses for Examine ON, Examine OFF, Output Energize, Latch, and Unlatch instructions are grouped 8 to a byte in the controller. Bits are arranged in the byte from the lowest user address at the least significant bit, to the highest user address at the most significant bit. Note that for Hex Addresses A4 and A5, the bits are not consecutive.

User Program Addresses

| | Hex Address | Bit | | | | | | | | Seq Group |
|--------------------|----------------|-----|-----|-----|-----|-----|-----|-----|-----|--------------|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| Outputs | 96 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 0 |
| | 97 | 118 | 117 | 116 | 115 | 114 | 113 | 112 | 111 | 1 |
| | 98 | 218 | 217 | 216 | 215 | 214 | 213 | 212 | 211 | 2 |
| | 99 | 318 | 317 | 316 | 315 | 314 | 313 | 312 | 311 | 3 |
| | 9A | 418 | 417 | 416 | 415 | 414 | 413 | 412 | 411 | 4 |
| | 9B | 518 | 517 | 516 | 515 | 514 | 513 | 512 | 511 | 5 |
| | 9C | 618 | 617 | 616 | 615 | 614 | 613 | 612 | 611 | 6 |
| Inputs | 9D | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 7 |
| | 9E | 108 | 107 | 106 | 105 | 104 | 103 | 102 | 101 | 8 |
| | 9F | 208 | 207 | 206 | 205 | 204 | 203 | 202 | 201 | 9 |
| | A0 | 308 | 307 | 306 | 305 | 304 | 303 | 302 | 301 | 10 |
| | A1 | 408 | 407 | 406 | 405 | 404 | 403 | 402 | 401 | 11 |
| | A2 | 508 | 507 | 506 | 505 | 504 | 503 | 502 | 501 | 12 |
| | A3 | 608 | 607 | 606 | 605 | 604 | 603 | 602 | 601 | 13 |
| | A4 | 210 | 209 | 110 | 109 | 10 | 9 | X | X | 14 |
| A5 | 610 | 609 | 510 | 509 | 410 | 409 | 310 | 309 | 15 | |
| Internal Relays | A6 | 708 | 707 | 706 | 705 | 704 | 703 | 702 | 701 | 16 |
| | A7 | 716 | 715 | 714 | 713 | 712 | 711 | 710 | 709 | 17 |
| | A8 | 724 | 723 | 722 | 721 | 720 | 719 | 718 | 717 | 18 |
| | A9 | 732 | 731 | 730 | 729 | 728 | 727 | 726 | 725 | 19 |
| | AA | 740 | 739 | 738 | 737 | 736 | 735 | 734 | 733 | 20 |
| | AB | 748 | 747 | 746 | 745 | 744 | 743 | 742 | 741 | 21 |
| | AC | 756 | 755 | 754 | 753 | 752 | 751 | 750 | 749 | 22 |
| | AD | 764 | 763 | 762 | 761 | 760 | 759 | 758 | 757 | 23 |
| | AE | 772 | 771 | 770 | 769 | 768 | 767 | 766 | 765 | 24 |
| | AF | 780 | 779 | 778 | 777 | 776 | 775 | 774 | 773 | 25 |
| | B0 | 788 | 787 | 786 | 785 | 784 | 783 | 782 | 781 | 26 |
| | B1 | 796 | 795 | 794 | 793 | 792 | 791 | 790 | 789 | 27 |
| | B2 | 804 | 803 | 802 | 801 | 800 | 799 | 798 | 797 | 28 |
| | B3 | 812 | 811 | 810 | 809 | 808 | 807 | 806 | 805 | 29 |
| | B4 | 820 | 819 | 818 | 817 | 816 | 815 | 814 | 813 | 30 |
| | B5 | 828 | 827 | 826 | 825 | 824 | 823 | 822 | 821 | 31 |
| | B6 | 836 | 835 | 834 | 833 | 832 | 831 | 830 | 829 | 32 |
| | B7 | 844 | 843 | 842 | 841 | 840 | 839 | 838 | 837 | 33 |
| B8 | 852 | 851 | 850 | 849 | 848 | 847 | 846 | 845 | 34 | |
| B9 | 860 | 859 | 858 | 857 | 856 | 855 | 854 | 853 | 35 | |
| BA | 868 | 867 | 866 | 865 | 864 | 863 | 862 | 861 | 36 | |

X indicates unused.

**User Program
Addresses**
(continued)

| | Hex Address | Bit | | | | | | | | Seq Grp |
|---|----------------|--|-----|-----|-----|-----|-----|-----|-----|------------|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| Fine Timer Bits Auto/Man Switch | BB | 876 | 875 | 874 | 873 | 872 | 871 | 870 | 869 | 37 |
| | BC-DB | Sequencer Step Counters 901-932 (1byte each) | | | | | | | | 38-69 |
| Timer Counter Sequencer Outputs | DC | 908 | 907 | 906 | 905 | 904 | 903 | 902 | 901 | |
| | DD | 916 | 915 | 914 | 913 | 912 | 911 | 910 | 909 | |
| | DE | 924 | 923 | 922 | 921 | 920 | 919 | 918 | 917 | |
| | DF | 932 | 931 | 930 | 929 | 928 | 927 | 926 | 925 | |
| Timer Counter Sequencer Overflow | E0 | 958 | 957 | 956 | 955 | 954 | 953 | 952 | 951 | |
| | E1 | 966 | 965 | 964 | 963 | 962 | 961 | 960 | 959 | |
| | E2 | 974 | 973 | 972 | 971 | 970 | 969 | 968 | 967 | |
| | E3 | 982 | 981 | 980 | 979 | 978 | 977 | 976 | 975 | |
| Timer Counter Sequencer Rung State | E4 | 908 | 907 | 906 | 905 | 904 | 903 | 902 | 901 | |
| | E5 | 916 | 915 | 914 | 913 | 912 | 911 | 910 | 909 | |
| | E6 | 924 | 923 | 922 | 921 | 920 | 919 | 918 | 917 | |
| | E7 | 932 | 931 | 930 | 929 | 928 | 927 | 926 | 925 | |
| Reset Instruction Rung State | E8 | 908 | 907 | 906 | 905 | 904 | 903 | 902 | 901 | |
| | E9 | 916 | 915 | 914 | 913 | 912 | 911 | 910 | 909 | |
| | EA | 924 | 923 | 922 | 921 | 920 | 919 | 918 | 917 | |
| | EB | 932 | 931 | 930 | 929 | 928 | 927 | 926 | 925 | |

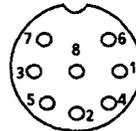
AB Spares

Connector Pin Configuration

Use a "DIN" 8-pin male plug with 360° pin pattern (O-type).
 Manufacturer's Part No: Calrad #30-503, or equivalent.

| Pin No. | Function |
|---------|---|
| 1 | $\overline{\text{RXD}}$ (from remote device to SLC) |
| 2 | RXD (from remote device to SLC) |
| 3 | TXD (from SLC to remote device) |
| 4* | GND |
| 5* | +12VDC to +30VDC |
| 6* | +12VDC to +30VDC |
| 7* | GND |
| 8 | $\overline{\text{TXD}}$ (from SLC to remote device) |

* These pins provide power from the SLC to the pocket programmer. They are not used for communication.



Female connector on the SLC.

CAUTION: Improper termination could result in damage to the SLC.

Appendix**SLC Personal Computer Interface Converter**

(Catalog Number 1745-PCC)

The SLC Programmable Controller can be directly connected to any RS-232-C port via the SLC Personal Computer Interface Converter. Both local and remote communication are possible with the converter. A six-foot cable is provided for local communication. For distances beyond six feet, and up to 4000 feet, a remote communication power supply (Catalog number 1745-PCP) must be used. Remote communication cables must be assembled by the user. **Note:** SLC 100, Series A processors are limited to a maximum communication distance of 100 ft (30.5 meters).

The interface converter optically isolates the SLC from the personal computer. Computer software must set the RS-232-C DATA TERMINAL READY (pin #20) to the SPACING condition (positive voltage) and the REQUEST TO SEND (pin #4) to the MARKING condition (negative voltage) in order for communication to occur. The SLC personal computer software, Catalog No. 1745-PCD will set these pins to their proper condition.



ALLEN-BRADLEY
 A ROCKWELL INTERNATIONAL COMPANY

Industrial Control Group
 Milwaukee, Wisconsin 53204

