- Host will be used to describe the computer operating as the originator of communications. Instrument will be used to describe the process control(s) using this protocol. In the following examples, only Host and Instrument will be used.
- 2. All data is sent and received in the ASCII HEX character format using 10 bits. 1 start bit, 8 data bits, no parity, and 1 stop bit.
- 3. The 'L' (ASCII 4C hex) is used as a filter character in the examples below. The filter characters are specific to Love Controls Corp. The filter character changes according to the **Instrument**'s address.
 - "L" (ASCII 4C hex) = **Instrument** addresses 01 through FF
 - "O" (ASCII 4F hex) = Instrument addresses 101 through 1FF
 - "V" (ASCII 56 hex) = **Instrument** addresses 201 through 2FF
 - "E" (ASCII 43 hex) = **Instrument** addresses 301 through 3FF

Addresses 00h, 100h, 200h, & 300H should not be used. These are for Factory Service use only.

4. The checksum generated by the **Host** is obtained by adding all ASCII characters in hex, excluding the <stx>, filter character, and the <etx>. Only the lowest 8 bits are used. The checksum is inserted into the message between the data field and the <etx>.

Example: **Host** checksum calculation

5. The checksum generated by the **Instrument** is obtained by adding all ASCII characters in hex, excluding the <stx> and the <ack>. Only the lowest 8 bits are used. The checksum is inserted between the data field and the <ack>. Note that the filter character is included in the checksum calculation by **Instrument**.

- 6. Checksums are generated for all **Host** commands
- 7. Checksums are generated for all normal **Instrument** responses. A checksum is not sent when errors are reported to the **Host**.

© 1992, Love Controls Division. All rights reserved. Rev. 12/94

- [nu]: " not used " at present time, (for future expansion) Undefined data, data returned should not be used.
- 2. <data> : These are ASCII characters shown in Hex notation. The upper case
 alpha character set is used for values A through F. (ASCII 46h = F, ASCII
 33h = 3, etc.) Lower case is not used.

3.	ASCII control character	ASCII in hex	Description					
	<stx></stx>	02	Start of Text					
	<etx></etx>	03	End of Text					
	<ack></ack>	06	Acknowledgment					
	L	4C	Filter character					

4. Data can be in either Hex, BCD (Binary coded decimal), or Binary, expressed in ASCII format.

ASCII (in hex)	Нех	ex BCD (decimal)			Ι	Binary					
30 31 32 33 34 35 36 37 38 39 41 42 43 44 45 46	0 1 2 3 4 5 6 7 8 9 A B C D E F	h h h h h h h h h h h h h h h h h h h		0 1 2 3 4 5 6 7 8 9	d d d d d d d d d d		0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1	0 0 0 0 1 1 1 0 0 0 1 1 1 1	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1	0 1 0 1 0 1 0 1 0 1 0 1 0 1	b b b b b b b b b b b b b

All numbers shown in examples are in hex, unless otherwise noted. Hex = 2Fh BCD (decimal) = 10d, Binary = 1101b. The h, d, b are symbols used in examples to indicate which number base is being used.

Example: Typical READ command sequence

Host command =

Host command transmitted in ASCII hex =

Instrument at address 32 replies with SP1 value of -15

Instrument reply =

Instrument reply transmitted in ASCII hex =
 02 4C 33 32 30 31 30 30 31 35

- Host starts command with start of text <stx>, and filter character.
- Next is the address of the Instrument desired.
- The Read data field is next. It contains the basic command for the Inst. The Read data field is four characters long.
- 4. The **Host** now adds the calculated checksum to the command frame. Finally the end of text <etx> is appended.
- 5. The **Instrument** responds when the <etx> has been received. If the data frame is correct, the **Instrument** will perform the received command. If the data field contains a bad command or bad checksum, an error message will be sent back to the **Host**. Details on errors will be explained later.
- 6. The **Instrument** starts its response with the start of text <stx> character, followed by the filter character.
- 7. Next is the **Instrument**'s address.
- 8. The Read data field is next. It contains the response to the command received. This field may be up to ten characters long.
- 9. The **Instrument** now adds the calculated checksum to the response frame. Finally the acknowledge <ack> is appended.
- 10. The Command and Response interchange is now complete.

Host wants to WRITE -15 to SP1 on Instrument at address 32

Host command =

<stx> L <addr> <----> data ----> <cksm> <etx> 32h 0200h 0015h 79h FFhcommand value siqn

Host command transmitted in ASCII hex =

Instrument at address 32 replies that SP1 has been accepted

Instrument reply =

<stx> L <addr> <data> <cksm> 32h 00h 11h

Instrument reply transmitted in ASCII hex =

- Host starts command with start of text <stx> and filter character. 1.
- 2. Next is the address of the Instrument desired.
- 3. The Write data field is next. It contains the basic command for the Instrument along with any data that may be required. The Write data field can be up to ten characters long. Write commands are used to modify the **Instrument**'s parameters.
- 4. The Host now adds the calculated checksum to the command frame. Finally the end of text <etx> is appended.
- The Instrument responds when the <etx> has been received. If the Data frame is 5. correct, the **Instrument** will perform the received command. if the data frame contains a bad command or a bad checksum, an error message will be sent back to the Host. An error will also occur if the value is not within the Instrument's range. Details on errors will be explained later.
- 6. Instrument starts its response with the start of text <stx> character, followed by the filter character.
- 7. Next is the **Instrument**'s address.
- 8. The Data field is next. It contains the response to the command received.
- 9. The Instrument now adds the calculated checksum to the response frame. Finally the acknowledge <ack> is appended.
- 10. Command and Response interchange is now complete.

General Notes:

- 1. Some commands have restricted access, such that they may not be entered if the menu item is not currently available. An example would be an attempt to read the P.I.D. values while the menu is set for the SELF TUNE mode of operation. An error will be reported to the **Host** if such a command is given, or if the command cannot be carried out by the **Instrument**.
- 2. Read commands will not change the **Instrument** display.
- 3. Some Write commands will cause the **Instrument** display to show the menu item to be entered, as though the "ENTER" key had been pressed.

Example: Checksum error detected by Instrument at address 32

Instrument responds to Host's last command with Error message

(Note the absence of Checksum)

N = ASCII character (4Eh) used to denote Error Present

<error code>

- 00 = not used
- 01 = Undefined command. Command not within acceptable range.
- 02 = Check-sum error on received data from Host.
- 04 = Illegal ASCII characters received in command. **Instrument** accepts only ASCII characters 0 through 9, A through F, and a through f in data the field.
- 05 = Data field error. Not enough, to many, or improper positioning of characters in data field.
- 06 = Undefined command. Command not within acceptable range.
- 07 = not used
- 08, 09 = Hardware fault. Return to Factory for service
- 10 = Undefined command. Command not within acceptable range.

READ COMMANDS ÄÄÄÄÄÄÄÄÄÄÄÄÄÄ	WRITE COMMANDS ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ
00 PROCESS VARIABLE VALUE W/STATUS 05 FULL STATUS	
01 00 SP1 VALUE 01 06 CY1 VALUE & S1Ot 01 07 SP1d VALUE 01 08 PUL1 VALUE 03 13 S1St dir/rE 01 12 S1OL VALUE 01 13 S1OH VALUE 03 14 S1LP O on/Ooff	02 00 SP1 VALUE 02 06 CY1 VALUE
03 35 SP1o Outb/OutA	
03 36 S2t AbS/dE 01 02 SP2 VALUE 01 09 CY2 VALUE & S2Ot 01 0A SP2d VALUE 01 0B PUL2 VALUE 03 15 S2St dir/rE 01 14 S2OL VALUE 01 15 S2OH VALUE 03 16 S2LP O on/Ooff	02 02 SP2 VALUE 02 07 CY2 VALUE
03 39 tune SELF/Pid/SLO/nor/FASt 03 12 Strt YES/no 03 38 LErn Cont/End 03 2D dFAC VALUE 01 0C Pb1 VALUE 01 0D Pb2 VALUE	04 03 tunE MODE = SELF 04 04 tunE MODE = FULL Pid 02 08 Pb1 VALUE 02 09 Pb2 VALUE
03 2C rES RESET AUTO/OFS (MANUAL) MODE 01 0E rES RESET VALUE	02 0A RESET VALUE AUTO MODE
01 OF rtE RATE VALUE 03 2E Pid2 On/OFF 03 2F ArUP On/OFF 01 25 ArtE VALUE	02 0B RESET VALUE OFFSET MODE 02 0C RATE VALUE
01 04 ALLO VALUE 01 05 ALHI VALUE 03 37 AL OFF/LO/Hi/HiLO 03 17 ALt AbS/dE 03 1B ALFE ONOF/Hold 03 1C ALPI ON/OFF 03 3A ALIH ON/OFF 03 18 ALST OPEN/CLOS 03 19 ALLP O ON/OFF	02 04 ALLO VALUE 02 05 ALHI VALUE

READ COMMANDS ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ	WRITE COMMANDS ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ					
03 34 SECr 1-4 SECURITY						
03 23 Inp INPUT TYPE 03 25 OSUP On/OFF (FOR IV) 03 10 Unit F/C (FOR T/C & RTD) 03 26 Unit F/C/nonE (FOR IV) 03 24 dPt DECIMAL POINTS POSITION 01 18 InPt INPUT FAULT TIMER VALUE 01 29 SENC VALUE 03 33 FiLt VALUE 01 24 InPC INPUT CORRECTION VALUE 01 28 LPbr VALUE	 					
01 16 SCAL VALUE 01 17 SCAH VALUE 01 10 SPL VALUE 01 11 SPH VALUE	 					
03 28 Auto On/OFF (OFF = MANUAL MODE) 01 1E AUTO/MAN SP1 MANUAL VALUE 01 2A AUTO/MAN SP2 MANUAL VALUE	04 06 Auto = OFF					
01 1A PEA PEAK VALUE 01 1B VAL VALLEY VALUE	04 07 PEAK RESET 04 08 VALLEY RESET					
01 1D SP1 & SP2 PERCENT OUTPUT VALUE 03 27 PctO On/OFF PERCENT OUTPUT	04 0B PctO = On 04 0C PctO = OFF					
03 30 Prog On/OFF 03 31 StAt On/OFF 01 26 1rt RAMP TIME 01 27 1St SOAK TIME 03 32 PEND OOFF/HOLD	 					
	02 0E CFSP COMM FAULT SP 04 00 LorE = rE (REMOTE) 04 01 LorE = LOC (LOCAL)					
	04 0D RESET "ENTER PRESSED					

04 0D RESET "ENTER PRESSED" FLAG TO "0"

ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ READ COMMAND DESCRIPTION ÄÄÄÄÄÄ 4-24-92 ÄÄ 8 ÄÄ COMMAND = 00READ "PV" PROCESS VARIABLE VALUE W/STATUS RECEIVED DATA = 1st CHAR (BINARY) 3 AUTO 3 COMM 3 ENTER KEY 3 ERROR 3 ON/OFF 3 REM/LOC 3 PRESSED? 3 PRESENT? 3 1=ON 3 1=REMOTE 3 1=YES 3 1=YES 3 O=NO 3 0=NO, OK 3 0=OFF, MANUAL 3 0=LOCAL 2nd CHAR (BINARY) 3 ALARM RELAY 3 [NU] 3 SP TYPE 3 [NU] 3 not used ³ not used 3 CFSV/LOCAL 3 3 3 1=ENERGIZED 3 1=CFSV 3 0=DEENERGIZED 3 3 0=LOCAL 3rd CHAR (BINARY) з [NU] з [NU] 3 [NU] 3 [NU] 3 not used 3 not used 3 ³ not used $^{\mathbf{3}}$ not used 3 3 3 4th CHAR (BINARY) 3 [NU] 3 NO ACTIVITY 3 PV SIGN 3 [NU] 3 not used 3 TIMER nAt 3 NEG/POS 3 not used 3 3 1=TIMEOUT 3 1=NEGATIVE 3 3 0=NORMAL,OK ³ 0=POSITIVE 3 5th CHAR MSD PROCESS VARIABLE VALUE 6th CHAR PROCESS VARIABLE VALUE 7th CHAR PROCESS VARIABLE VALUE 8th CHAR LSD PROCESS VARIABLE VALUE

NOTE:

IF ERROR PRESENT BIT = "1", ISSUE A FULL STATUS READ COMMAND "05" TO DETERMINE THE SPECIFIC ERROR.

NOTE:

BITS MARKED WITH AN "*" ARE ERROR BITS. ANY ONE OF THESE
THE "ERROR PRESENT BIT" IN THE "PV" PROCESS VARIABLE

STATUS (COMMAND 00).

```
        COMMAND = 0100
        READ "SP1 " VALUE

        COMMAND = 0102
        READ "SP2 " VALUE

        COMMAND = 0104
        READ "ALLO" VALUE

        COMMAND = 0105
        READ "ALHI" VALUE

        COMMAND = 0124
        READ "Inpc" INput Correction VALUE

        COMMAND = 0116
        READ "SCAL" VALUE

        COMMAND = 0117
        READ "SCAH" VALUE

        COMMAND = 0110
        READ "SPL " VALUE

        COMMAND = 0111
        READ "SPH " VALUE

        COMMAND = 011A
        READ "PEA " PEAK VALUE

        COMMAND = 011B
        READ "UAL " VALLEY VALUE

        COMMAND = 0121
        READ "CFSP" COMM FAULT SP VALUE

        DATA =
        DATA =

 COMMAND = 0100 READ "SP1 " VALUE
 DATA =
           1st CHAR SIGN 0 = POSITIVE, (NOT 0) = NEGATIVE 2nd CHAR SIGN 0 = POSITIVE, (NOT 0) = NEGATIVE
           3rd CHAR MSD VALUE
4th CHAR VALUE
5th CHAR VALUE
6th CHAR LSD VALUE
                                                                               BOTH 1st & 2nd CHARS
                                                                               MUST BE "0" FOR SIGN
                                                                                 TO BE POSITIVE.
- \Delta T \Delta =
           1st CHAR [NU] not used
2nd CHAR [NU] not used
3rd CHAR MSD VALUE
4th CHAR VALUE
5th CHAR VALUE
6th CHAR LSD VALUE
 COMMAND = 032D READ "dFAC" VALUE
COMMAND = 0333 READ "Filt" VALUE
COMMAND = 032B READ "nAt " NO ACTIVITY TIMER VALUE
 DATA =
            1st CHAR
                               MSD VALUE
            2nd CHAR
                                         LSD VALUE
```

```
ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ 11 Ä
                                    DATA = ^3NOT 0 ^3 0
                                               COMMAND = 0313 READ "S1St" 3 dir 3 rE 3
COMMAND = 0314 READ "S1LP" 3 0 on 3 0oFF 3 COMMAND = 0335 READ "SP10" 3 Outb 3 OutA 3 COMMAND = 0336 READ "S2t " 3 AbS 3 dE 3 COMMAND = 0315 READ "S2St" 3 dir 3 rE 3
COMMAND = 0316 READ "S2LP" 3 0 on 3 Ooff 3
COMMAND = 0312 READ "Strt" 3 YES 3 no
COMMAND = 0338 READ "LErn" <sup>3</sup> Cont <sup>3</sup> End <sup>3</sup>
COMMAND = 032C READ "rES " 3 AUTO 3 OFS 3 (OFS = OFFSET MODE)

COMMAND = 032E READ "Pid2" 3 On 3 OFF 3
COMMAND = 032E READ "PId2" 3 On 3 OFF 3 COMMAND = 0317 READ "ALLT" 3 AbS 3 dE 3 COMMAND = 0318 READ "ALF" 3 On 3 OFF 3 COMMAND = 031C READ "ALF" 3 On 3 OFF 3 COMMAND = 0318 READ "ALF" 3 On 3 OFF 3 COMMAND = 0318 READ "ALST" 3 OPEN 3 CLOS 3 COMMAND = 0319 READ "ALLT" 3 OPEN 3 CLOS 3 COMMAND = 0319 READ "ALLT" 3 OPEN 3 OPEN 3 CLOS 3 COMMAND = 0322 READ "ALLT" 3 OPEN 3 OPEN 3 OPEN 3
COMMAND = 0322 READ "ALbr" <sup>3</sup> On <sup>3</sup> OFF <sup>3</sup> COMMAND = 0325 READ "OSUP" <sup>3</sup> On <sup>3</sup> OFF <sup>3</sup> (For "Curr/Uolt" input)
COMMAND = 0310 READ "Unit" 3 F 3 C 3 (FOR T/C & RTD inputs 3 300 Series only)
                                                                    3 300 Series only)
COMMAND = 0328 READ "Auto" 3 On 3 OFF 3 (OFF = MANUAL MODE)

COMMAND = 0327 READ "Pcto" 3 On 3 OFF 3

COMMAND = 0330 READ "Prog" 3 On 3 OFF 3

COMMAND = 0331 READ "Stat" 3 On 3 OFF 3

COMMAND = 0332 READ "PENd" 3 OFF 3 Hold 3

COMMAND = 032A READ "Lore" 3 TE 3 LOC 3

COMMAND = 0329 READ "CFLt" 3 2 3 1 3 COMM FAULT MODE
                                               DATA =
                                               3NOT 0 3 0 3 BOTH 1st & 2nd
        1st CHAR
                                               3NOT 0 3 0 3 CHARS MUST BE "0"
        2nd CHAR
COMMAND = 0106 READ "CY1 " VALUE & "S10t" OUTPUT TYPE
COMMAND = 0109 READ "CY2 " VALUE & "S2Ot" OUTPUT TYPE
D\Delta T\Delta =
          1st CHAR & 10 = "OnOF" ON-OFF OUTPUT
                             08 = "PUL " PULSE OUTPUT
          2nd CHAR
                              04 = "Uolt" VOLTAGE OUTPUT
                              02 = "Curr" CURRENT OUTPUT
                              01 = "Ft " FAST TIME PROPORTIONING OUTPUT
                              00 = "CY " TIME PROPORTIONING OUTPUT
IF OUTPUT = "CY ", THEN 3rd & 4th CHARS = CYCLE RATE VALUE
         3rd CHAR
                           MSD VALUE OF CY#
                           LSD VALUE OF CY#
          4th CHAR
                           [NU] not used
          5th CHAR
          6th CHAR [NU] not used
```

DATA =

```
1st CHAR 0 = SP1 PCT OUT, NOT 0 = SP2 PCT OUT 2nd CHAR 0 = SP1 PCT OUT, NOT 0 = SP2 PCT OUT
```

3rd CHAR [NU] not used BOTH 1st & 2nd CHARS 4th CHAR MSD VALUE MUST BE "0" TO INDICATE

5th CHAR VALUE SP1 PCT OUT

6th CHAR LSD VALUE

NOTES:

- 1. THE VALUE IN THE 1st & 2nd CHARACTERS IS USED TO INDICATE THAT SP2 PERCENT OUTPUT VALUE IS BEING READ.
- 2. SP2 PERCENT OUTPUT WILL ONLY BE AVAILABLE IF **INSTRUMENT** IS CONFIGURED AS A DUAL (2 SET POINT) TYPE INSTRUMENT.
- 3. SP1 & SP2 PERCENT OUTPUTS ARE ALTERNATED AT A 1 SECOND RATE.

COMMAND = 0323 READ "InP" INPUT TYPE (1600 Series only)

DATA =

```
1st CHAR
          [NU] not used
          1 = J-IC
2nd CHAR
                           9 = b -
           2 = CA
                               A = C-
           3 = E -
                                B = P392 RTD
           4 = t -
                                C = n120 RTD
           5 = L -
                                D = P385 RTD
           6 = n-
                               E = Curr
            7 = r - 13
                          F = Uolt
```


COMMAND = 0324 READ "dPt" DECIMAL POINTS

DATA =

1st CHAR [NU] not used

2nd CHAR 0 = NONE 2 = 0.00 1 = 0.0 3 = 0.000

COMMAND = 0326 READ "Unit" UNITS (300 Series - FOR I/V INPUTS ONLY)

DATA =

1st CHAR [NU] not used See Command 0310 2nd CHAR 0 = NONE for reading T/C and RTD

COMMAND = 0334 READ "SECr" SECURITY MODE

DATA =

1st CHAR [NU] not used 2nd CHAR 0 = 1 AS VIEWED IN SECr

1 = 2 AS VIEWED IN SECr 2 = 3 AS VIEWED IN SECr 3 = 4 AS VIEWED IN SECr

2nd CHAR [NU] not used

3 = nor4 = FASt

```
COMMAND = 0200 [DATA] CHANGE "SP1 " VALUE
COMMAND = 0202 [DATA] CHANGE "SP2 " VALUE COMMAND = 0204 [DATA] CHANGE "ALLO" VALUE
COMMAND = 0205 [DATA] CHANGE "ALHi" VALUE
COMMAND = 020E [DATA] CHANGE "CFSP" VALUE (COMM FAULT SP)
DATA =
      1st CHAR
                      WRITE COMMAND WRITE COMMAND
       2nd CHAR
       3rd CHAR
                       WRITE COMMAND
       4th CHAR
                       WRITE COMMAND
DATA =
      5th CHAR MSD VALUE
       6th CHAR
                         VALUE
      7th CHAR VALUE
8th CHAR LSD VALUE
      9th CHAR SIGN 0 = POSITIVE, NOT 0 = NEGATIVE 10th CHAR SIGN 0 = POSITIVE, NOT 0 = NEGATIVE
                                BOTH 9th & 10th CHARS MUST BE "0"
                                FOR SIGN TO BE ACCEPTED AS POSITIVE.
```


	D = 0206 [DATA] CHANG D = 0207 [DATA] CHANG		-	_		-		
2nd 3rd	CHAR CHAR		WRITE WRITE	COM COM	MAND MAND			
	CHAR CHAR		[NU] [NU]					
_	CHAR CHAR		MSD V	_				
	CHAR n CHAF		[NU] [NU]		used, used,			

NOTES:

- 1. THE "CY" CYCLE RATE VALUE MUST BE AN EVEN NUMBER, I.E. 02, 04, 10, ETC. ODD NUMBERS OR OUT OF RANGE NUMBERS CHANGE WILL NOT BE ACCEPTED.
- 2. AN ERROR RESPONSE WILL BE GENERATED IF "CY" CHANGE IS ATTEMPTED AND THE OUTPUT TYPE "S10t" IS NOT SET TO "CY".

ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ WRITE COMMAND DESCRIPTION ÄÄÄÄÄ 4-24-92 ÄÄ 15 Ä COMMAND = 0208 [DATA] CHANGE "Pb1" VALUE COMMAND = 0209 [DATA] CHANGE "Pb2" VALUE COMMAND = 020A [DATA] CHANGE "rES" RESET VALUE & SET IN AUTO MODE

COMMAND = 020B [DATA] CHANGE "rES" RESET VALUE & SET IN OFFSET MODE

COMMAND = 020C [DATA] CHANGE "rtE" RATE VALUE

COMMAND = 020F [DATA] CHANGE SP1 AUTO/MAN "MANUAL" VALUE COMMAND = 0210 [DATA] CHANGE SP2 AUTO/MAN "MANUAL" VALUE

DATA =

1st CHAR WRITE COMMAND
2nd CHAR WRITE COMMAND
3rd CHAR WRITE COMMAND
4th CHAR WRITE COMMAND

DATA =

5th CHAR MSD VALUE
6th CHAR VALUE
7th CHAR VALUE
8th CHAR LSD VALUE

9th CHAR [NU] not used, set to 0

10th CHAR [NU] not used, set to 0

```
DATA =
          WRITE COMMAND
WRITE COMMAND
   1st CHAR
   2nd CHAR
   3rd CHAR
            WRITE COMMAND
   4th CHAR
            WRITE COMMAND
```

NOTE:

DATA IS NOT REQUIRED FOR THESE COMMANDS. THEY ARE SPECIFIC AS TO THEIR FUNCTION.