Command Protocol
Love Controls Division

This document is intended to assist you in creating your own communications program to work with our 2600, 8600, 16A, and 32A Series of controls. The protocol can be used over any RS-232 or RS-485 serial communication line.

Wiring for the controls is found in the standard instruction manual provided with the control. The RS-485 is usually used in the half-duplex mode. This is a means of communication where the same two wires are used to both transmit and receive information. Only one device can communicate at a time. If you are using an auto switch RS-485 converter such as the Love 351, 352, or 356, you can write your program as though the controls were working in full duplex mode. If you are using the Love 350 or another hardware hand shaking type of converter, your software must control the hand shaking through the host computer.

The following conventions are used in this document:

1. **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 character format using 1 start bit, 8 data bits, no parity, and 1 stop bit (commonly referred to as 8N1).

3. The 'L' (ASCII 4C hex) is used as a filter character and address prefix. The filter characters are specific to Love Controls division. The filter character changes according to the **Instrument**'s address range.

"L" = **Instrument** addresses 01 through FF
"O" = **Instrument** addresses 101 through 1FF
"V" = **Instrument** addresses 201 through 2FF
"E" = **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 (converted to 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

```
<stx> L  | <addr>    <-- data --> |   <cksm>   <etx>
         |                        |
[02]  L  |   3 2       0  1  0  0 |    2  6     [03]
         |------- checksum -------|
         |33h+32h+30h+31h+30h+30h | = 126h (use only the low byte)
```

5. The checksum generated by the **Instrument** is obtained by adding all ASCII characters in hex, excluding the <stx> and the <ack>. To strip the result to the lowest 8 bits, perform a logical AND with the result and the value FFh. Convert the value to two ASCII characters and insert between the data field and the <ack>. Note that the filter character is included in the checksum calculation by **Instrument**.

Example: **Instrument** checksum calculation

```
                32h       010015h          D8h
   <stx>| L <addr><----- data ----->|<cksm>  <ack>
        |                          |
   [02] | L   3 2    0 1 0 0 1 5   |  D  8   [06]
        |------ checksum -----------|
        |4C+33+32+30+31+30+30+31+35 |= 1D8h(use only the low byte)
```

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**.

The most commonly used command is the READ STATUS command. This command returns the Process Variable from the addressed control and information in binary to indicate other information and status about the control. The READ STATUS command is 00.

```
Host command = <stx> L <addr> 00 <cksm> <etx>

Instrument reply   = <stx> L <addr> <data> <cksm> <ack>

                      [BINARY]
                  -------------------------------------
      1st CHAR.   bit 3 = Auto/Manual Mode,     0= Automatic,     1= Manual
                  bit 2 = Communication mode,   0= Local,         1= Remote
                  bit 1 = not used
                  bit 0 = Error present,        0= No Errors      1= Error
                              Read Full Status using command 05 to determine actual error.
                  -------------------------------------
      2nd CHAR.   bit 3 = Alarm #1  0 = De-Energized (Off)  1 = Alarm #1 Energized (On)
                  bit 2 = Alarm #2  0 = De-Energized (Off)  1 = Alarm #2 Energized (On)
                  bit 1 = Set Point selected.
                  bit 0 =  "   "       "
                              bit 1    bit 0  Set Point
                               0      0   = 1SP1
                               0      1   = 2SP1
                               1      0   = 3SP1
                               1      1   = 4SP1
                  -------------------------------------
      3rd CHAR.   bit 3 = No Activity Timer (nAt) 0= OK     1 = ERROR (Timed out)
                  bit 2 = not used
                  bit 1 = Decimal Point   bit 1 bit 0 Decimal Point position.
                  bit 0 =        "    "    0     0    =     0
                                           0     1    =   0.0
                                           1     0    =  0.00
                                           1     1    = 0.000
                  -------------------------------------
```

```
4th CHAR.   bit 3 = not used
            bit 2 = Engineering Units
            bit 1 =         "       "
                  bit 2  bit 1  Eng. Units
                    0       0   = none
                    0       1   = F
                    1       0   = C
            bit 0 = Process Variable Sign       0 = Positive      1 = Negative
            ------------------------------------
5th CHAR.   MSD   [VALUE]  Process Variable
6th CHAR.
7th CHAR.
8th CHAR.   LSD
            ------------------------------------
Data Example:     ASCII       [02] L 3 2 4 4 0 2 0 1 0 0 3 C [06]
                        Data Packet   -→|              |←---

                     [BIN]            [BIN]           [  VALUE  ]
 ACSII Data          4    4             0    2         0  1  0  0
BINARY Conversion 0100 0100          0000 0010
            Bit 3210 3210       Bit 3210 3210

From BINARY data
      1. Instrument is set in Remote mode.
      2. Alarm #2 is Energized (On)
      3. Value resolution is at 1 degree.
      4. Value is in degrees F and positive.

From ASCII data the PV is 100
```

This command may be sent as often as desired. If you are using hardware hand shaking, allow about 35 milliseconds for the control to process the return data packet. If the error bit is set, you may use the FULL STATUS command 05.

The FULL STATUS command allows you to look into many facets of the control.

```
Host command = <stx> L <addr> 05 <cksm> <etx>

Instrument reply   = <stx> L <addr> <data> <cksm> <ack>
```

```
                  [BINARY]
            ------------------------------------
      1st CHAR.   bit 3 Fail Test               0= OK       1= Fail Test Error
                  bit 2 not used
                  bit 1 Check Calibration       0= OK       1= CheC CAL Error
                  bit 0 Overflow Error          0= OK       1= OFL error
            ------------------------------------
      2nd CHAR.   bit 3 Underflow Error         0= OK       1= UFL error
                  bit 2 Bad Input               0= OK       1= bAd InP Error
                  bit 1 Open Input              0= OK       1= OPEn InPError
                  bit 0 Area                    0= OK       1= ArEA Error
            ------------------------------------
      3rd CHAR.   bit 3 Loop Break              0= OK       1= LOOP bAd Error
                  bit 2 Sensor Rate of Change   0= OK       1= SEnC Error
                  bit 1 not used
                  bit 0 not used
            ------------------------------------
```

```
4th CHAR.   bit 3 not used
            bit 2 not used
            bit 1 not used
            bit 0 not used
      ------------------------------------
5th CHAR.         not used
6th CHAR.         not used
7th CHAR.         not used
8th CHAR.         not used
9th CHAR.         not used
10th CHAR.        not used
```

Note: If any Error is present here, the "Error present" bit will be set in the Process Variable read command 00.

The next most common command is the command to read the set point. This is a 4 character command. The data packet is assembled as shown above. The command to read SP1 is 0100

Host command = <stx> L <addr> 0100 <cksm> <etx>

Instrument reply = <stx> L <addr> <data> <cksm> <ack>

The data field contains 6 ASCII characters, and are defined as follows.

```
            [BINARY]
      ------------------------------------
1st CHAR.   bit 3 – Not used
            bit 2 - Not used
            bit 1 = Decimal Point   bit 1 bit 0 Decimal Point position.
            bit 0 =        "    "    0     0    =     0
                                     0     1    =   0.0
                                     1     0    =  0.00
                                     1     1    = 0.000

      ------------------------------------

2nd CHAR.   bit 3 - Not used
            bit 2 = Engineering Units
            bit 1 =         "       "
                        bit 2  bit 1  Eng. Units
                          0       0   = none
                          0       1   = F
                          1       0   = C
      bit 0 = Process Variable Sign       0 = Positive      1 = Negative

      ------------------------------------

            [VALUE]
      ------------------------------------
3rd CHAR.   MSD
4th CHAR.
5th CHAR.
6th CHAR.   LSD
      ------------------------------------
```

Data Example:

ACSII data	2 2	0 1 5 0
BINARY Conversion	0010 0010	
Bit	3210 3210	

From bit data, note that decimal point format is 0.0 and Value is in degrees F. Actual set point one setting in this example is 15.0° F.

To Set the Set Point you use the write command 0200.

```
Host command = <stx> L <addr> <command> <  data  >  <cksm> <etx>
                                        Value Sign
                     = <stx> L <addr>   0200    0150  00    <cksm> <etx>


Instrument reply   = <stx> L <addr> <data> <cksm> <ack>
                             00  =Accepted
                             N03 =Error, out of range, bad command
```

The data field for the write command contains 6 ASCII characters, as follows:

1st CHAR.	MSD	[VALUE]
2nd CHAR.		
3rd CHAR.		
4th CHAR.	LSD	
		[SIGN]
5th CHAR.	Sign	[non-ZERO]=Negative
6th CHAR.	Sign	[ZERO]=Positive

Note:
1. Local/Remote (LOrE) must be set for Remote (rE) to Write. To instruct the control to switch from local to remote send the 0400 command:

```
Host command = <stx> L <addr> <command> <cksm> <etx>

                   = <stx> L <addr>   0400    <cksm> <etx>


Instrument reply   = <stx> L <addr> <data> <cksm> <ack>
                             00  =Accepted
                             N03 =Error, bad command
```

This command is fixed in its function, no data is required.

To return the control to local mode send the 0401 command:

```
Host command = <stx> L <addr> <command> <cksm> <etx>

                   = <stx> L <addr>   0401    <cksm> <etx>


Instrument reply   = <stx> L <addr> <data> <cksm> <ack>
                             00  =Accepted
                             N03 =Error, bad command
```

This command is fixed in its function, no data is required.

In the example above the SP1 value is set to 150 (decimal point is ignored).

Error Messages

The control can also send a number of messages to indicate difficulty in communication or to indicate the inability to perform a command. Possible error messages with their explanations follow:

Example: Checksum error detected by **Instrument** at address 32

Instrument responds to **Host**'s last command with Error message

<stx>	L	<addr>	N	<error code>	<ack>
[02]	L	3 2	N	0 2	[06]

(Note the absence of Checksum)

N = ASCII character used to denote Error Present

Table of Error Messages possible from **Instrument**

<error code>

00 = not used
01 = Undefined command. Command not within acceptable range.
02 = Check-sum error on received data from **Host**.
03 = Command not performed by **Instrument**. (option not enabled, restricted read/write menu, check msg.)
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 = Hardware fault. Return to Factory for service
09 = Hardware fault. Return to Factory for service
10 = Undefined command. Command not within acceptable range.