

AAAAAAAAAAAAAAAAAAAAAAAA COMMAND INTRODUCTION AAAAAAA 4-24-92 AAA 1 AA  
 LoveLink Protocol, Love Controls Division, Wheeling, Illinois

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 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  
 32h 0100h 26h  
 <stx> L 3 <addr> <-- data --> 3 <cksm> <etx>  
 3  
 02 4C 3 33 32 30 31 30 30 3 32 36 03  
 3<AAAAAA cksm AAAAAA>3  
 3 33+32+30+31+30+30 3 = 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>. 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**.  
 Example: **Instrument** checksum calculation  
 32h 010015h D8h  
 <stx> 3 L <addr><----- data -----> 3<cksm> <ack>  
 3  
 02 34C 33 32 30 31 30 30 31 35 3 44 38 06  
 3<AAAAAAAA cksm AAAAAAAAA>3  
 34C+33+32+30+31+30+30+31+35 3= 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**.



# COMMAND INTRODUCTION 4-24-92 3

## Field Position and Description of Data Frame

Example: Typical READ command sequence

Host wants to READ SP1 value from Instrument at address 32

Host command =

<stx>	L	<addr>	<-- data -->	<cksm>	<etx>
		32h	0100h	26h	
			command		

Host command transmitted in ASCII hex =

02	4C	33 32	30 31 30 30	32 36	03
----	----	-------	-------------	-------	----

Instrument at address 32 replies with SP1 value of -15

Instrument reply =

<stx>	L	<addr>	<----- data ----->	<cksm>	<ack>
		32h	01h	0015d	D8h
			sign	value	

Instrument reply transmitted in ASCII hex =

02	4C	33 32	30 31	30 30 31 35	44 38	06
----	----	-------	-------	-------------	-------	----

- 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.
- 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 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.
- The Instrument starts its response with the start of text <stx> character, followed by the filter character.
- Next is the Instrument's address.
- The Read data field is next. It contains the response to the command received. This field may be up to ten characters long.
- The Instrument now adds the calculated checksum to the response frame. Finally the acknowledge <ack> is appended.
- The Command and Response interchange is now complete.

Field Position and Description of Data Frame

Example: Typical WRITE command sequence

```

Host wants to WRITE -15 to SP1 on Instrument at address 32
Host command =
<stx> L <addr>  <----- data -----> <cksm> <etx>
                32h      0200h      0015h      FFh      79h
                  command      value      sign

```

```

Host command transmitted in ASCII hex =
02 4C 33 32 30 32 30 30 30 30 31 35 46 46 37 39 03
Instrument at address 32 replies that SP1 has been accepted

```

```

Instrument reply =
<stx>  L  <addr>  <data>  <cksm>  <ack>
                32h      00h      11h
Instrument reply transmitted in ASCII hex =
02 4C 33 32 30 30 31 31 06

```

1. Host starts command with start of text <stx> and filter character.
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.
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 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. 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 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.

## Errors Messages Reported

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>
		32h	N	02h	

02	4C	33	32	4E	30	32	06
----	----	----	----	----	----	----	----

(Note the absence of Checksum)

N = ASCII character (4Eh) 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 Inst. (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, 09 = Hardware fault. Return to Factory for service
- 10 = Undefined command. Command not within acceptable range.

COMMAND SUMMARY 4-24-92 6

COMMAND SUMMARY 4-24-92 7

READ COMMANDS  
AAAAAAAAAAAAAAAA

**WRITE COMMANDS**

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)	04 05 Auto = On
			04 06 Auto = OFF
01 1E	AUTO/MAN	SP1 MANUAL VALUE	02 0F SP1 MANUAL VALUE
01 2A	AUTO/MAN	SP2 MANUAL VALUE	02 10 SP2 MANUAL VALUE
01 1A	PEA	PEAK VALUE	04 07 PEAK RESET
01 1B	VAL	VALLEY VALUE	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	-----
01 21	CFSP	COMM FAULT SP	02 0E CFSP COMM FAULT SP
03 2A	LorE	COMM rE/LOC	04 00 LorE = rE (REMOTE)
			04 01 LorE = LOC (LOCAL)
03 2B	nAt	NO ACTIVITY TIMER VALUE	-----
03 29	CFLt	COMM FAULT MODE 2/1	-----
			04 0D RESET "ENTER PRESSED"
			FLAG TO "0"

[illegible]







AAAAAAAAAAAAAAAAAAAA READ COMMAND DESCRIPTION AAAAA 4-24-92 AA 10 A

DATA =			
1st CHAR	SIGN	0 = POSITIVE,	(NOT 0) = NEGATIVE
2nd CHAR	SIGN	0 = POSITIVE,	(NOT 0) = NEGATIVE
3rd CHAR	MSD VALUE		BOTH 1st & 2nd CHARS
4th CHAR	VALUE		MUST BE "0" FOR SIGN
5th CHAR	VALUE		TO BE POSITIVE.

COMMAND = 0107	READ "SP1d" VALUE
COMMAND = 0108	READ "PUL1" VALUE
COMMAND = 0112	READ "S1OL" VALUE
COMMAND = 0113	READ "S1OH" VALUE
COMMAND = 010A	READ "SP2d" VALUE
COMMAND = 010B	READ "PUL2" VALUE
COMMAND = 0114	READ "S2OL" VALUE
COMMAND = 0115	READ "S2OH" VALUE
COMMAND = 010C	READ "Pb1 " VALUE
COMMAND = 010D	READ "Pb2 " VALUE
COMMAND = 010E	READ "rES " RESET VALUE
COMMAND = 010F	READ "rtE " RATE VALUE
COMMAND = 0125	READ "ArTE" VALUE
COMMAND = 0118	READ "InPt" INPUT FAULT TIMER VALUE
COMMAND = 0129	READ "SEnC" VALUE
COMMAND = 0128	READ "LPbr" VALUE
COMMAND = 011E	READ SP1 MANUAL VALUE
COMMAND = 012A	READ SP2 MANUAL VALUE
COMMAND = 0126	READ "lrt " RAMP TIME
COMMAND = 0127	READ "lSt " SOAK TIME

[illegible]

DATA =	
1st CHAR	MSD VALUE
2nd CHAR	LSD VALUE







AAAAAAAAAAAAAAAAAAAA WRITE COMMAND DESCRIPTION AAAAA 4-24-92 AA 14 A

```
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
	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	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.

[illegible]

```
COMMAND = 0206 [DATA]    CHANGE "CY1" CYCLE RATE VALUE FOR SP1
COMMAND = 0207 [DATA]    CHANGE "CY2" CYCLE RATE VALUE FOR SP2
```

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

DATA =	
5th CHAR	[NU] not used, set to 0
6th CHAR	[NU] not used, set to 0
7th CHAR	MSD VALUE
8th CHAR	LSD VALUE
9th CHAR	[NU] not used, set to 0
10th CHAR	[NU] not used, set to 0

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 "Slot" IS NOT SET TO "CY".

[illegible]

AAAAAAAAAAAAAAAAAAAAAA WRITE COMMAND DESCRIPTION AAAAA 4-24-92 AA 15 A

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

AAAAAAAAAAAAAAAAAAAAAA

COMMAND = 0403 CHANGE "tunE" MODE = SELF  
COMMAND = 0404 CHANGE "tunE" MODE = Pid (FULL)  
COMMAND = 0402 CHANGE ALARM ACK  
COMMAND = 0405 CHANGE "Auto" = On  
COMMAND = 0406 CHANGE "Auto" = OFF (in MANUAL)  
COMMAND = 0407 CHANGE PEAK RESET  
COMMAND = 0408 CHANGE VALLEY RESET  
COMMAND = 040B CHANGE "PctO" = On  
COMMAND = 040C CHANGE "PctO" = OFF  
COMMAND = 0400 CHANGE "LorE" = rE (REMOTE)  
COMMAND = 0401 CHANGE "LorE" = LOC (LOCAL)  
COMMAND = 040D CHANGE RESET "ENTER PRESSED" FLAG TO "0"

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

NOTE:

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

AAAAAAAAAAAAAAAAAAAAAA