Sequential Event Recorder

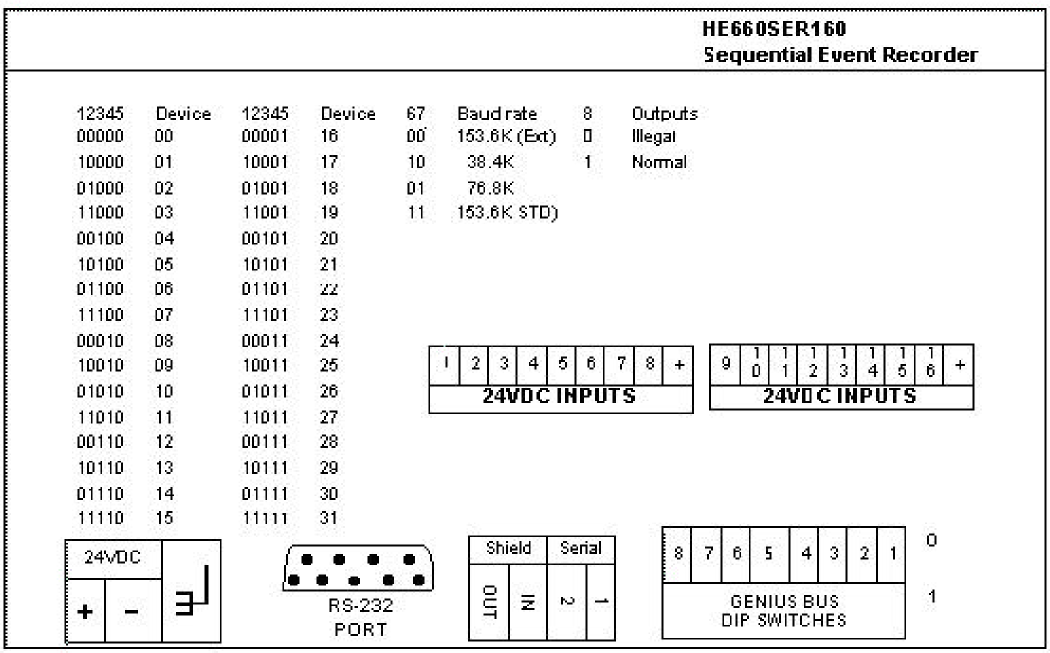
Product Specifications and Installation Data

DESCRIPTION

The Sequential Event Recorder (SER) is an intelligent Genius I/0 block that resides directly on the Genius LAN. The primary function of the Sequential Event Recorder is to monitor the status of digital inputs and record state changes to an internal buffer. Once the state changes have been recorded, the data can be recalled from the SER's memory over the Genius LAN through the use of datagrams or via the SER's RS-232 port. The SER features a dip switch that allows configuration of the Genius baud rate and bus addresss.

There are two versions of the Sequential Event Recorder: HE660SER160 and the HE660SER320. The HE660SER160 is a sixteen input module with a voltage level of 24VDC, a resolution of 1 millisecond, and a maximum network baud rate of 153.6K. The event buffer can accept 1 to 2,048 events with 8 words per event. Data acquisition is handled by either datagrams or RS-232. The HE660SER320 is equivalent to the HE660SER160 but has an additional sixteen inputs for a total of thirty-two inputs.

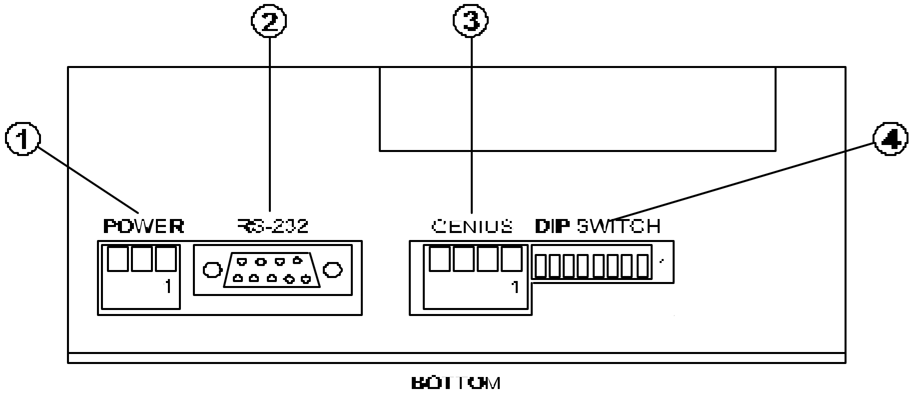
ILLUSTRATION



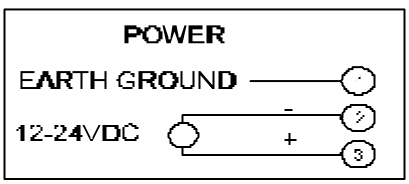
SPECIFICATIONS

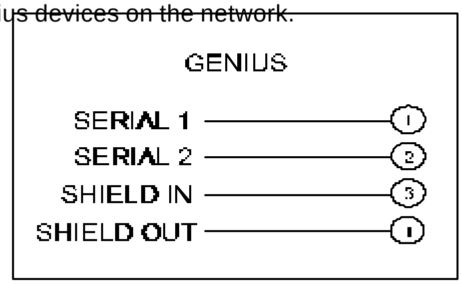
Specification	HE660SER160	HE660SER320
Number of Inputs	16	32
Voltage Level	3-30VDC (24VDC)	3-30VDC (24VDC)
Genius Baud Rate	up to 153.6K	up to 153.6K
Number of Nodes	32 (0 - 31)	32 (0 - 31)
Event Buffer	1 to 2048 events	1 to 2048 events
Event Data	8 words per event	8 words per event
Data Acquisition	Datagram or RS-232	Datagram or RS-232
Time Stamp Resolution	1 millisecond	1 millisecond
Unit Power	24VDC	24VDC
Battery-backed clock/calendar	Yes	Yes
Time Setting	Datagram or RS-232	Datagram or RS-232

POWER and GENIUS CONNECTIONS



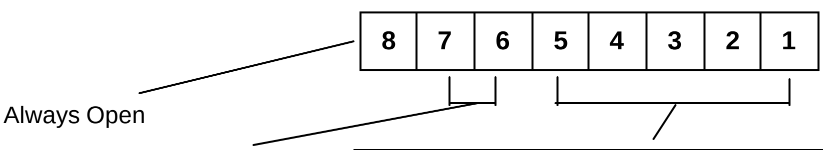
Physical Layer. The HE660SER160 and HE660SER320 modules are equipped with several connectors. The diagram above depicts the power connector (1), RS-232 port (2), Genius network connector (3), and the Genius dip switch (4). The module ***must*** be powered by an external 24VDC supply, physically connected to the Genius LAN, and have the correct Genius address (0-31) and baud rate. The SER node address must not conflict with any other Genius node. The baud rate must also be set to the same rate as the other Genius devices on the network.





Power and Genius Connections. The tables above show the proper 24VDC power connections as well as the physical connections for the SER on the Genius LAN.

GENIUS CONFIGURATION

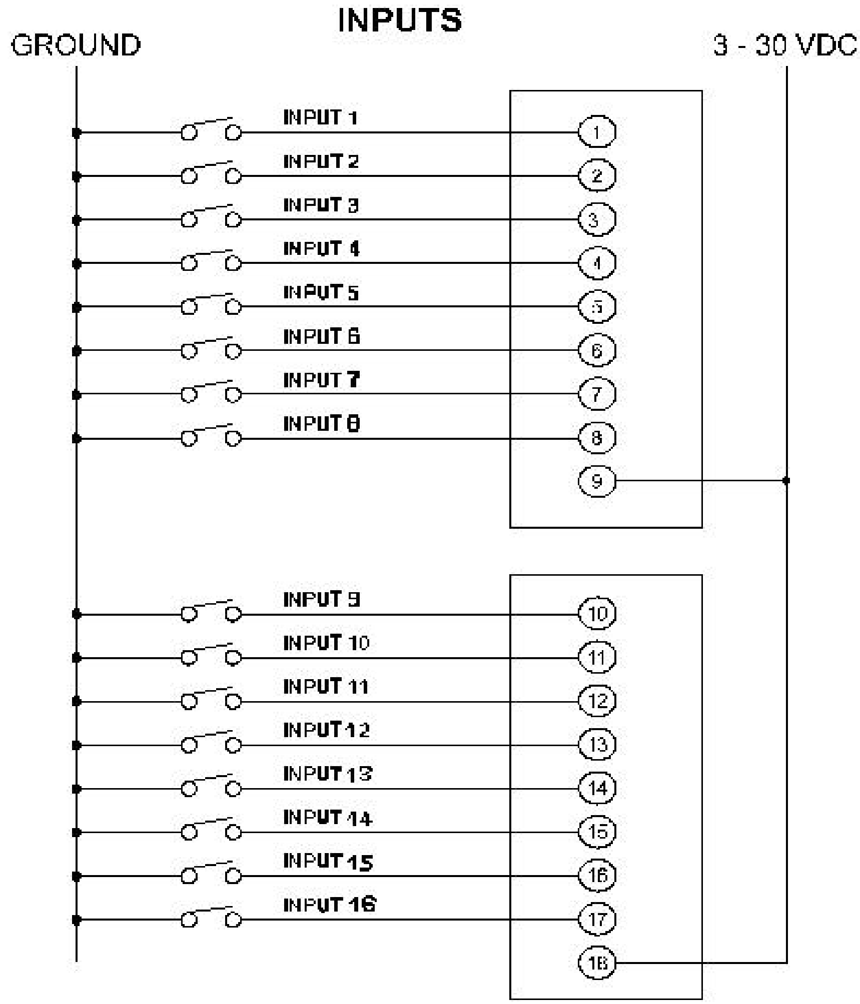


7	6	baud rate
CLOSD	CLOSD	153.6K extended
CLOSD	OPEN	38.4K
OPEN	CLOSD	76.8K
OPEN	OPEN	153.6K standard

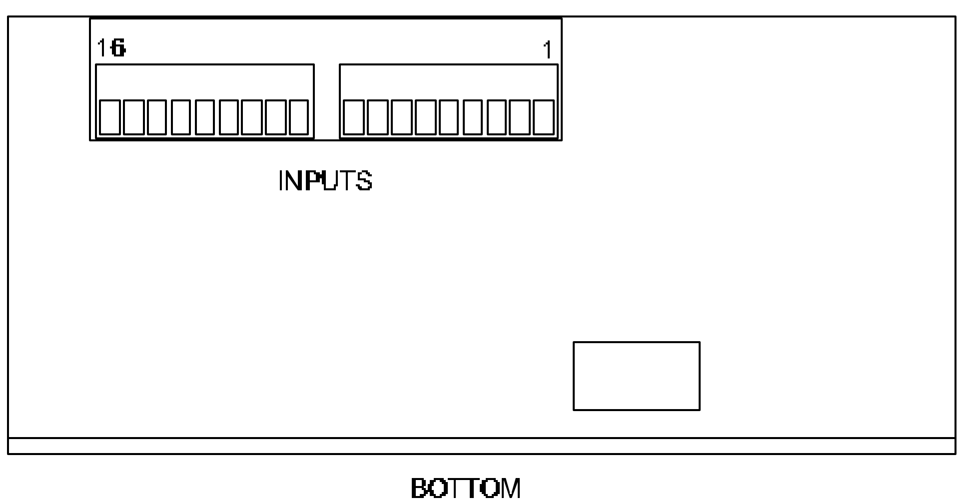
5	4	3	2	1	address	5	4	3	2	1	address
CLOSD	CLOSD	CLOSD	CLOSD	CLOSD	0	OPEN	CLOSD	CLOSD	CLOSD	CLOSD	16
CLOSD	CLOSD	CLOSD	CLOSD	OPEN	1	OPEN	CLOSD	CLOSD	CLOSD	OPEN	17
CLOSD	CLOSD	CLOSD	OPEN	CLOSD	2	OPEN	CLOSD	CLOSD	OPEN	CLOSD	18
CLOSD	CLOSD	CLOSD	OPEN	OPEN	3	OPEN	CLOSD	CLOSD	OPEN	OPEN	19
CLOSD	CLOSD	OPEN	CLOSD	CLOSD	4	OPEN	CLOSD	OPEN	CLOSD	CLOSD	20
CLOSD	CLOSD	OPEN	CLOSD	OPEN	5	OPEN	CLOSD	OPEN	CLOSD	OPEN	21
CLOSD	CLOSD	OPEN	OPEN	CLOSD	6	OPEN	CLOSD	OPEN	OPEN	CLOSD	22
CLOSD	CLOSD	OPEN	OPEN	OPEN	7	OPEN	CLOSD	OPEN	OPEN	OPEN	23
CLOSD	OPEN	CLOSD	CLOSD	CLOSD	8	OPEN	OPEN	CLOSD	CLOSD	CLOSD	24
CLOSD	OPEN	CLOSD	CLOSD	OPEN	9	OPEN	OPEN	CLOSD	CLOSD	OPEN	25
CLOSD	OPEN	CLOSD	OPEN	CLOSD	10	OPEN	OPEN	CLOSD	OPEN	CLOSD	26
CLOSD	OPEN	CLOSD	OPEN	OPEN	11	OPEN	OPEN	CLOSD	OPEN	OPEN	27
CLOSD	OPEN	OPEN	CLOSD	CLOSD	12	OPEN	OPEN	OPEN	CLOSD	CLOSD	28
CLOSD	OPEN	OPEN	CLOSD	OPEN	13	OPEN	OPEN	OPEN	CLOSD	OPEN	29
CLOSD	OPEN	OPEN	OPEN	CLOSD	14	OPEN	OPEN	OPEN	OPEN	CLOSD	30
CLOSD	OPEN	OPEN	OPEN	OPEN	15	OPEN	OPEN	OPEN	OPEN	OPEN	31

DIP Switch Settings. The SER160 and SER320 are equipped with a bank of 8 DIP switches which are used to configure the Genius port and baud rate.

The two 9-pin I/O connectors on the SER are the 24VDC monitoring inputs. These connectors, when viewed from the wire opening with the screw at the top, are grouped and numbered from 1 to 9 and 10 to 18 (left to right). Pins 1 through 8 and 10 through 17 are grounded (or sinking) while pin 9 and 18 are the positive common supplies (3-30VDC). When 3VDC or greater is applied to the positive common of one of the inputs, it is considered to be ON or "high". See the figures below to for illustrations.

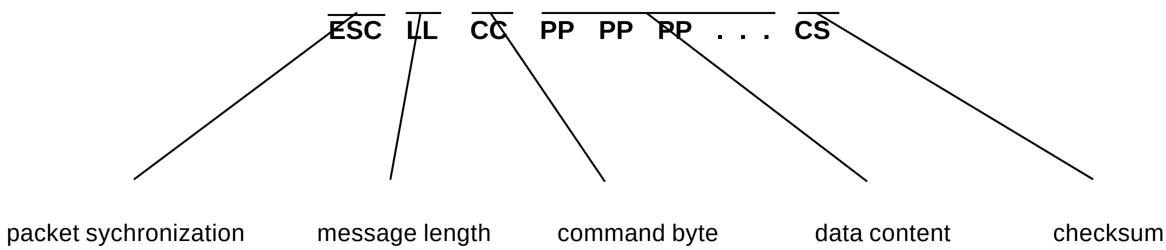


I/O Configuration. The diagram above illustrates the I/O connections for the SER160. Please note that only Pins 9 and 18 are connected to the positive supply (3-30 VDC). The diagram below shows the input pin configuration and layout on the module.



The Sequential Event Recorder (SER) communications protocol has been defined to allow reception of commands from either Genius datagrams or from the RS-232 serial port. The command structure is identical, regardless of the medium. However, the way the packets are structured differs slightly within the two methods.

RS-232 Serial Access. The RS-232 serial protocol employs a special packetization and error-checking scheme. Each message begins with an **ESC** character (1Bh), and with that single exception all other characters are HEX-ASCII 0-9 and A-F. The starting **ESC** byte is used for synchronization. The single-byte **LL** represents the 8-bit message length and includes all bytes between and including the command (**CC**) and the last parameter byte (all **PP**s), but *not* the checksum (**CS**). If no parameter bytes are sent, the message length will be 1. The message is formatted as follows:



The last byte of each message is the checksum (**CS**). It is calculated simply by adding the command byte and each of the data content bytes. The checksum contains ***only*** the lower 8 bits of the calculated checksum value.

For example, if 1B 07 **43** **04 00 06 02 01 00** 50

then the checksum (**CS**) would equal **50**h. The actual value calculated should be *0050*h but since only the lower 8 bits are used the checksum is **50**h.

Each of the command blocks is described in detail on the following pages.

Command bytes. The command bytes (**CC**) are message commands sent either *to* or *from* a Host device (master), another device on the network, or the SER. Listed below are the command bytes and a description of their uses.

CC Byte (Command)		
Hexadecimal Command	Character Equivalent	Meaning
43h	C	Config Unit
45h	E	Erase Buffer
4Ch	L	Last Broadcast Command
4Dh	M	Mode and Status
50h	P	Print Entry
52h	R	Read Buffer
53h	S	Set Clock

These command bytes are explained in greater detail on the following pages.

Configuration Unit. The SER's RS-232 port is configurable using the command message described below. This message is sent *from* the Host device *to* the SER. The SER will send a reply with the same format that contains the current SER configuration. When configured to Auto Print or Auto Report, the unit will automatically print the buffered information to a serial printer or will automatically send a READ REPLY message to the specified device whenever an event occurs.

Configuration Command		
Byte Number	Value (must be in HEX)	Meaning
1	C (43h)	Configure Command (C)
2 - 3	0 to 2048	Desired Buffer Size (0 for no change). Must be converted to hexadecimal value.
4	0 = No change 1 = 300 2 = 600 3 = 1200 4 = 2400 5 = 4800 6 = 9600 7 = 19200	Desired baud rate
5	0, 1, 2, or 3	**AUTO REPORT** (See the *Auto Report Table* below)
6	0 to 31	SER Report Address (0 to 1Fh)
7	0 to 1	**AUTO PRINT** 0 = Disable, 1 = Enable

Auto Report Byte 5 of Configuration Unit			
Binary		Decimal Equivalent	Definition
0	0	0	Disable
0	1	1	Disable
1	0	2	Genius
1	1	3	RS-232

<u>Command from Host to SER</u>

Configuration Command example: 1B 07 43 04 00 06 02 01 00 50

1Bh	=	Sychronization command (ESC)
07h	=	8-bit message length, 7 bytes in above example
43h	=	Byte Number 1, Configuration Unit (C)
04 00h	=	Byte Numbers 2-3, Buffer size, 1024 equals 0400h
06h	=	Byte Number 4, 9600 baud rate
02h	=	Byte Number 5, Auto Report *Enabled* using Genius
01h	=	SER Report Address of SER set to 1
00h	=	Auto Print *Disabled*
50h	=	Checksum (CS) equals to 50h in the above example (lower 8 bits only)

See the following page for the ***Configuration*** reply example.

Reply from SER to Host

Configuration reply example: 1B 07 43 04 00 06 02 01 00 50

1Bh	=	Sychronization command (ESC)
07h	=	8-bit message length, 7 bytes in above example
43h	=	Byte Number 1, Configuration Unit (C)
04 00h	=	Byte Numbers 2-3, Buffer size, 1024 equals 0400h
06h	=	Byte Number 4, 9600 baud rate
02h	=	Byte Number 5, Auto Report *Enabled* using Genius
01h	=	SER Report Address of SER set to 1
00h	=	Auto Print *Disabled*
50h	=	Checksum (CS) equals to 50h in the above example (lower 8 bits only)

Erase Buffer. The SER unit will erase the entire entry buffer. A single-byte command (E) with no parameters is sent *by* the Host *to* the SER. A single-byte reply (E) is echoed back *to* the Host *from* the SER as an acknowledgment.

Erase Buffer Command		
Byte Number	Command Variable	Meaning
1	E (45h)	Mode and Status Command

Command from Host to SER

Erase Buffer command example: 1B 01 45 45

1Bh	=	Synchronization command (ESC)
01h	=	8-bit message length, 1 byte in the above example
45h	=	Command byte, Erase Buffer
45h	=	Checksum (CS) equals to 45h in the above example (lower 8 bits only)

Reply from SER to Host

Erase Buffer reply example: 1B 01 45 45

1Bh	=	Synchronization command (ESC)
01h	=	8-bit message length, 1 byte in the above example
45h	=	Command byte, Erase Buffer
45h	=	Checksum (CS) equals to 45h in the above example (lower 8 bits only)

Last Broadcast Command and ***Last Broadcast Reply.*** The Last Broadcast Command is used to obtain the command code and time-stamp of the last broadcast command received *by* the SER *from* the Host. The Last Broadcast Command is a single-byte command (L) with no parameters. The Last Broadcast Reply is the reply message sent *by* the SER *to* the requesting device.

Last Broadcast Command		
Byte Number	Command Variable	Meaning
1	L (4Ch)	Last Broadcast Command

Last Broadcast Reply		
Byte Number	Value (must be in hexadecimal)	Meaning
1	4Ch	Mode and Status Command (L)
2	X	X = Last Broadcast Command Code
3	0 - 23	Hour of Broadcast Reception
4	0 - 59	Minute of Broadcast Reception
5	0 - 59	Second of Broadcast Reception
6 - 7	0 - 999	Millisecond of Broadcast Reception

<u>Command from Host to SER</u>

Last Broadcast Command example: 1B 01 4C 4C

1Bh	=	Synchronization command (ESC)
01h	=	8-bit message length, 1 byte in the above example
4Ch	=	Command byte, Last Broadcast Command (L)
4Ch	=	Checksum (CS) equals to 4Ch in the above example (lower 8 bits only)

<u>Reply from SER to Host</u>

Last Broadcast Reply example: 1B 07 4C 45 0E 1E 32 00 FA 55

1Bh	=	Sychronization command (ESC)
07h	=	8-bit message length, 7 bytes in the above example
4Ch	=	Byte Number 1, Last Broadcast Command (L)
45h	=	Byte Number 2, Code of the last broadcasted command, 45H (Erase Buffer)
0Eh	=	Byte Number 3, Hour of broadcast reception, 14:00 hours (2:00 pm) equals 0Eh
1Eh	=	Byte Number 4, Minute of broadcast reception, 30 minutes equals 1Eh
32h	=	Byte Number 5, Second of broadcast reception, 50 seconds equals 32h
00 FAh	=	Byte Numbers 6-7, Millisecond of broadcast reception, 250ms equals 00FAh
55h	=	Checksum (CS) equals to 55h in the above example (lower 8 bits only)

Mode and Status. This command is used to control the SER recording mode (start/stop) and to obtain the firmware version number of the unit. The Mode and Status command is sent *from* the Host device *to* the SER.

MODE AND STATUS		
Byte Number	Values (must be in hexadecimal format)	Meaning
1	M (4Dh)	Mode and Status command
2	0 to 1	0 = stop, 1 = start recording
3	0 to 255	Reply only (version number, 100 = 1.00)

<u>Command from Host to SER</u>

Mode and Status command example: 1B 03 4D 01 7D CB

1Bh	=	Sychronization command (ESC)
03h	=	8-bit message length, 3 bytes in the above example
4Dh	=	Byte Number 1, Mode and Status Command (M)
01h	=	Byte Number 2, start recording information, 1 equals 01h
7Dh	=	Byte Number 3, 200 replies with version number 2.00 (200 = 7Dh)
CBh	=	Checksum (CS) equals to CBh in the above example (lower 8 bits only)

<u>Reply from SER to Host</u>

Mode and Status reply example: 1B 03 4D 01 7D CB

1Bh	=	Sychronization command (ESC)
03h	=	8-bit message length, 3 bytes in the above example
4Dh	=	Byte Number 1, Mode and Status Command (M)
01h	=	Byte Number 2, start recording information, 1 equals 01h
7Dh	=	Byte Number 3, 200 replies with version number 2.00 (200 = 7Dh)
CBh	=	Checksum (CS) equals to CBh in the above example (lower 8 bits only)

Print Entry. The SER unit will print the specified entries through the serial port. This command is sent *from* the Host device *to* the SER. A single-byte reply (P) will be returned in acknowledgment *from* the SER.

PRINT ENTRY		
Byte Number	Value (must be in hexadecimal format)	Meaning
1	P (50h)	Print configuration command
2	1 to 2048	Starting print numbers
3	1 to 2048	Number of entries to print

Command from Host to SER

Print Entry command example: 1B 03 50 32 64 E6

1Bh	=	Sychronization command (ESC)
03h	=	8-bit message length, 3 bytes in the above example
50h	=	Byte Number 1, Print Configuration Command (P)
32h	=	Byte Number 2, Starting print numbers at 50 (32h)
64h	=	Byte Number 3, Number of entries to print equals 100 or 64h
E6h	=	Checksum (CS) equals to E6h in the above example (lower 8 bits only)

Reply from SER to Host

Print Entry reply example: 1B 03 50 32 64 E6

1Bh	=	Sychronization command (ESC)
03h	=	8-bit message length, 3 bytes in the above example
50h	=	Byte Number 1, Print Configuration Command (P)
32h	=	Byte Number 2, Starting print numbers at 50 (32h)
64h	=	Byte Number 3, Number of entries to print equals 100 or 64h
E6h	=	Checksum (CS) equals to E6h in the above example (lower 8 bits only)

Read Entry. The Read Entry command is sent *from* the host *to* the SER. This command polls the SER for information on current I/O states. The Read Reply is the corresponding reply message sent *from* the SER *to* the Host. See the following pages for a complete description on the Read Reply message.

READ ENTRY COMMAND		
Byte Number	Value (must be in hexadecimal format)	Meaning
1	R (52h)	Read Entry command
2 -3	1 to 2048	Starting entry number
4	1 to 10	Number of entries to report

Command from Host to SER

Read Entry example: 1B 03 01 F4 05 49

1Bh	=	Sychronization command (ESC)
03h	=	8-bit message length, 3 bytes in the above example
52h	=	Byte Number 1, Read Configuration command (R)
01 F4h	=	Byte Numbers 2-3, Starting entry number equals 500 or 01F4h
05h	=	Byte Number 4, 5 entries will be reported (05h)
49h	=	Checksum (CS) equals to 49h in the above example (lower 8 bits only)

Read Reply. The Read Reply is the reply from the SER stating the information called for in the Read Entry command. This reply is sent *from* the SER *to* the Host device.

READ REPLY		
Byte Number	Value (must be in hexadecimal format)	Meaning
1	R (52h)	Read configuration command
2 - 3	1 to 2048	Starting report number
4	1 to 10	Number of entries being reported (maximum of 10)
5 - 6	1 to 64K	Total number of entries since clear
7 - 8	0 to 2047	Next entry slot

READ REPLY ENTRIES		
First Reported Entry		
Byte Number	Value (must be in hexadecimal format)	Meaning
9 - 10	BIT-MAPPED	Inputs that have changed
11 - 12	BIT-MAPPED	Current input state
13	00 to 99	Year of change
14	1 to 12	Month of change
15	1 to 31	Date of change
16	0 to 23	Hour of change
17	0 to 59	Minute of change
18	0 to 59	Second of change
19 - 20	0 to 999	Millisecond of change
The next reported Entries all follow the above format (up to 10)		

Note: The ***Read Reply*** command is the reply message from the SER to the Host device. Because the Read Reply is different in structure from the other commands, it is explained and illustrated in greater detail.

The following page contains the Read Reply example.

<u>Reply from SER to Host</u>

Read Reply example: 1B 14 52 01 F4 05 7D 00 07 D0 00 01 00 01 60 08 0F 0E 1E 32 FA 71

1Bh	=	Sychronization command (ESC)
14h	=	8-bit message length, 20 bytes in the above example equals 14h
52h	=	Byte Number 1, Read Reply message (R)
01 F4h	=	Byte Numbers 2-3, Starting report number, 01F4h, Erase Buffer
05h	=	Byte Number 4, Number of entries being reported, 05h is 5 entries being reported
7D 00h	=	Byte Numbers 5-6, Total number of entries since CLEAR, 32,000 equals 7D00h
07 D0h	=	Byte Numbers 7-8, Next slot entry, 2000 equals 07D0h
00 01h	=	Byte Numbers 9-10, Bit-mapped inputs that have changed, 1 input equals 0001h
00 01h	=	Byte Numbers 11-12, Bit-mapped current input states, 1 input equals 0001h
60h	=	Byte Number 13, Year of change, 96 equals 60h
08h	=	Byte Number 14, Month of change, August is 08h
0Fh	=	Byte Number 15, Day of change, 15th day of August is 0Fh
0Eh	=	Byte Number 16, Hour of change, 14:00 equals 2:00 pm is 0Eh
1Eh	=	Byte Number 17, Minute of change, 30 equals 1Eh
32h	=	Byte Number 18, Second of change, 50 seconds equals 32h
FAh	=	Byte Number 19, Millisecond of change, 250 ms equals FAh
71h	=	Checksum equals to 71h in the above example (lower 8 bits only)

Set Clock. The Set Clock command is the command message that will set the clock on the SER. This message is sent *from* the Host device *to* the SER. A single-byte reply (S) is returned from the SER to the requesting device.

SET CLOCK		
Byte Number	Value (must be in hexadecimal format)	Meaning
1	S (53h)	Set Clock command
2	00 to 99	Current Year
3	1 to 12	Current Month
4	1 to 31	Current Day
5	0 to 23	Current Hour
6	0 to 59	Current Minute
7	0 to 59	Current Second
8 - 9	0 to 999	Current Millisecond

See the following page for ***Set Clock*** examples.

Command from host to SER See the following page for ***Print Entry*** examples.

Set Clock command example: 1B 09 53 60 0B 15 16 1E 32 00 FA 33

1Bh	=	Sychronization command (ESC)
09h	=	8-bit message length, 9 bytes in the above example
53h	=	Byte Number 1, Set Clock Command (S)
60h	=	Byte Number 2, Current year, 96 equals 60h
0Bh	=	Byte Number 3, Current month, 11 (November) equals 0Bh
15h	=	Byte Number 4, Current day, 21st day of the month equals 15h
16h	=	Byte Number 5, Current hour, 22:00 pm equals 16h
1Eh	=	Byte Number 6, Current minute, 30 minutes equals 1Eh
32h	=	Byte Number 7, Current second, 50 seconds 32h
00 FAh	=	Byte Numbers 8-9, Current millisecond, 250ms equals 00FAh
33h	=	Checksum (CS) equals to 33 bytes in the above example (lower 8 bits only)

Reply from SER to host

Set Clock reply example: 1B 01 53 45

1Bh	=	Synchronization command (ESC)
01h	=	8-bit message length, 1 byte in the above example
53h	=	Command byte, Set Clock (S)
53h	=	Checksum (CS) equals to 53h in the above example (lower 8 bits only)

There are two means of data transfer on the Genius network, *global data* and *datagrams*.

Global data. Global data is data broadcast by Genius devices blindly onto the network. The data is not directed to any particular Genius device, but is available to all devices on the network with the ability to read and process the data. Any device (such as a Genius bus controller) with the ability to read and interpret global data then has the ability to detect the pressing of one or more function keys.

Datagrams. Datagrams are a directed message from one Genius device to another. Whenever the SER needs to know the status of a register not broadcast as global data, the SER will send a datagram to the Genius Bus Controller requesting this information. The bus controller then processes the message and sends a datagram response to the SER which includes the requested data.

The Genius datagram commands must use the HORNER function code (40H), the SER subfunction code (50H), and the proper datagram length. The actual datagram message contains the binary command and optional parameters as described below.

A DATAGRAM REQUEST, or **COMM_REQ,** is a communications request initiated by one device to another that allows such functions as reads, write, clears, etc. The logic block follows the structure below:

Word 0:	Command Length, [(Datagram length + 1) / 2] + 7
Word 1:	No wait (0)
Word 2:	Memory type for status word (usually 8 for %R)
Word 3:	Memory offset for status word (-1)
Word 4:	Idle timeout (0)
Word 5:	Communication time (0)
Word 6:	Command number (13 for DEQUEUE Datagram, 14 for SEND Datagram)

A **DEQUEUE** (READ) **DATAGRAM** transfers an incoming datagram from the Bus Controller to the CPU. This command follows the structure below:

Word 1-6:	COMM_REQ
Word 7:	Maximum data to receive
Word 8:	Memory type for data received (usually 8 %R)
Word 9:	Memory offset for data received (-1)
Word 10:	Function code (40H)
Word 11:	Subfunction code (50H)
Word 12:	Device number (0-31, or 0FFH for global message)

A **SEND** DATAGRAM sends a datagram to a device which then sends a reply datagram. This command follows the structure below:

Word 1-6:	COMM_REQ
Word 7:	Genius bus address of target device (0 - 31)
Word 8:	Function code (40H)
Word 9:	Subfunction code (50H)
Word 10:	Priority (0)
Word 11:	Datagram length (in bytes)
Word 12 - command size:	Datagram data (high/low bytes are swapped)

For Word 12, the size of the datagram is dependent upon the specific command being used, such as Set Clock, Erase Buffer, etc.

For more information regarding transmission and reception of datagrams with the Series 90-70 Genius Bus Controller or any other Genius device, see the *Series 90-70 Bus Controller User's Manual, GFK-0398A.*