

Appendix D: HOST/SC Serial Protocol

The primary elements of the HCS/SS are the Supervisory Controller (SC) and the host IBM PC-compatible computer. The SC is responsible for overall system control while the host computer is used to edit and compile XPRESS programs, send the programs to the SC, set a new time in the SC, clear log memory, receive logged data and dump it to a disk file, and display system status. The program on the host computer used to communicate with the SC is called HOST.

The HOST/SC interface allows a great deal of control over the system from the PC side of the connection. The SC sends status information to the PC for display on the console at periodic intervals. What information is sent by the SC can be controlled from the PC. The PC may also send the SC a one-time request for the status of any system component and override that status or set a new value.

Not all elements of the interface are used by the stock HOST program. The interface is fully defined to allow third parties to develop custom PC interfaces to the HCS/SS to better fit individual installations.

All commands sent from the PC to the SC are of the form:

! <cmd> <data>

where <cmd> is the command to be executed and <data> is an optional data field of variable length, depending on the command.

Most replies are of the form:

\$ <cmd> <reply>

where <cmd> is the command generating the reply and <reply> is an optional data field of variable length, depending on the command. Two commands don't generate any reply, while one other uses a special reply sequence.

The following table describes the commands and their replies:

Command Summary

<u>cmd</u>	<u>Description</u>	<u>data</u>	<u>reply</u>	
00	Reset system (same as hardware reset)	none	—	no reply
01	Pause all status output until next command	none	none	
02	Request full system status information	none	—	no reply
03	Select time display frequency	a	none	a=0:off, 1:sec 2:min
04	Get time once	none	a-h	a-h=time data
05	Set new time	a-h	none	a-h=time data
06	Load new XPRESS program	prog	—	special timing used
07	Clear log memory	none	none	
08	Get size of logged data	none	a,b	a=low, b=high (0-4095)
09	Get logged data	none	block	terminated with FF,FF,FF
0A	Put data in log memory	a,b,c	none	a=ID, b=low, c=high
10	Select X-10 display	a,b	none	a=A-H, b=I-P
11	Get X-10 status once	a	a,b	a=house/mod, 0-F=A-P/1-16 b=0:off, 1:on
12	Set X-10 module	a,b,c	none	a=house/mod, b=0:alloff, 1:allon, 2:on, 3:off, 4:dim, 5:bright c=dim/br count
13	Select dig in display	a,b,c,d	none	a-d=bitmap
14	Get dig in status once	a	a,b	a=input, b=0:off, 1:on
15	Set dig in state	a,b	none	a=input, b=0:off, 1:on, 2:transparent
16	Select dig out display	a,b,c,d	none	a-d=bitmap
17	Get dig out status once	a	a,b	a=output, b=0:off, 1:on
18	Set dig out state	a,b	none	a=output, b=0:off, 1:on
19	Select ADC display	a,b,c	none	a-c=bitmap
1A	Get ADC value once	a	a,b,c	a=ADC, b=low, c=high

<u>cmd</u>	<u>Description</u>	<u>data</u>	<u>reply</u>	
1B	Set ADC value	a,b,c	none	a=ADC, b=low, c=high (0-4095,4096=transparent)
1C	Select DAC display	a	none	a=bitmap
1D	Get DAC value once	a	a,b	a=DAC, b=value
1E	Set DAC value	a,b	none	a=DAC, b=value
1F	Select net module display	a	none	a=bitmap
20	Get net module status once	a	a,b	a=mod/num, b=0:timeout, 1:active, 2:err
21	Select netbit display	a,b,c,d,e	none	a-e=bitmap
22	Get netbit status once	a,b	a,b,c	a=low, b=high, c=0:off, 1:on
23	Set netbit state	a,b,c	none	a=low, b=high, c=0:off, 1:on
24	Get variable value once	a	a,b,c	a=variable, b=low, c=high
25	Set variable value	a,b,c	none	a=variable, b=low, c=high
30	Send string to network	string	none	zero terminated
31	Send string to voice	string	none	zero terminated

The HCS/SS periodically sends status information to the PC. What information is sent depends on what has been requested by the PC using the commands described above. The format of the display packet is the same as the reply packet described above, namely:

\$ <cmd> <data>

where <cmd> indicates what kind of information is being sent and <data> contains the actual status information. The following table summarizes the information sent by the SC to the PC.

<u>cmd</u>	<u>Description</u>	<u>data</u>	
80	Current time	a-h	a-h=time data
81	X-10 modules	a,b,c	a=house, b=low, c=high
82	Digital inputs	a,b	a=group, b=data
83	Digital outputs	a,b	a=group, b=data
84	Analog inputs	a,b-q	a=group, b-q=data low, high
85	Analog outputs	a,b-e	a=group, b-e=data

Status Display Summary

<u>cmd</u>	<u>Description</u>	<u>data</u>	
86	Network modules	a,b	a=mod/num, b=0:timeout, 1:active, 2:error
87	Netbits	a,b	a=group, b=data
88	Console message	string	zero terminated

Data Formats

Data is formatted as efficiently as possible to minimize the number of bytes being sent over the interface. The following describes various data formats mentioned in the above tables:

Time Data All time references consist of eight BCD bytes in the order hundredths of second (0–99), second (0–59), minute (0–59), hour (0–23), day of week (1–7, 1=Sunday), day (1–31), month (1–12), year (00–99).

X-10 Data When described as *house/mod*, the module identifier is packed into a single byte. The upper nybble contains the house code (0–F = A–P) and the lower nybble contains the module number (0–F = 1–16). When selecting X-10 modules to display, the 16 house codes are bitmapped into two bytes. The low-order byte is first, with A in the least-significant bit.

In the status display, the house code being sent by the SC is indicated by an ASCII letter in the range of “A”–“P”. The 16 modules within the housecode are packed into two bytes—low-order byte first—with module 1 in the least-significant bit.

Digital Input Bitmap The digital inputs are grouped into blocks of eight. When selecting which inputs to display, those blocks are represented as four bitmapped bytes as follows:

56–63	48–55	40–47	32–39	24–31	16–23	8–15	0–7
7	6	5	4	3	2	1	0
120–127	112–119	104–111	96–103	88–95	80–87	72–79	64–71
15	14	13	12	11	10	9	8
184–191	176–183	168–175	160–167	152–159	144–151	136–143	128–135
23	22	21	20	19	18	17	16

248–255	240–247	232–239	224–231	216–223	208–215	200–207	192–199
31	30	29	28	27	26	25	24

When the status information is sent to the PC for display, the group number is sent first (as a single byte), followed by the status of the eight inputs in that group bitmapped into a single byte. The lowest numbered input in the group is represented in the least-significant bit of the status byte.

For example, to indicate the status of inputs 224–231, the SC sends a 29 followed by a byte containing the status of those eight inputs.

When selecting which inputs to display, the same four bitmapped bytes are used. For example, to select inputs 24–39 and 104–111 to be displayed, send the SC the byte sequence 18h, 20h, 00h, 00h.

Digital outputs are handled the same way as digital inputs.

Digital Output Bitmap

Analog input selection is handled the same way as digital inputs, but is limited to the first three bitmap bytes (i.e., analog channels 0–191 are supported by the bitmaps, though only 0–135 do anything).

Analog Input Bitmap

In the status packet sent to the PC, the group number is sent first, followed by 16 bytes of data (eight 16-bit channels, low byte first, low channel first).

Analog output selection is handled the same way as the digital inputs, but is limited to the first bitmap byte (i.e., analog channels 0–63 are supported by the bitmaps, though only 0–31 do anything).

Analog Output Bitmap

In the status packet sent to the PC, the group number is sent first, followed by 4 bytes of data (four 8-bit channels, low channel first).

Netbits are handled the same way as digital inputs, but add another bitmap byte as follows:

Netbits

312–319	304–311	296–303	288–295	280–287	272–279	264–271	256–263
39	38	37	36	35	34	33	32

Network Modules Network module types are always referenced by the following numbers:

0:PL	4:ADIO
1:MCIR	5:DIO+
2:LCD	6:reserved
3:DIO	7:reserved

When selecting which modules to display, the numbers correspond to bits in a byte. Setting a particular bit selects that module type for display. For example, sending a selection byte of 26h enables the display of the status of all MCIR-Links, LCD-Links, and DIO+-Links in the system.

When sending the network module status information, the SC includes the module type in the upper nybble of the first byte and the module number in the lower nybble. The actual status follows in the second byte. For example, to indicate that DIO-Link number 2 is active on the network, the SC sends 32h, 01h.

New Program Load

When sending a new XPRESS program to the SC, a special handshaking sequence is used. After the initial “!” 06h command is sent, the HOST program must wait for a “^” acknowledge character. This acknowledgement tells the PC that the SC has successfully shut down processing of the old XPRESS program and is ready to accept a new one. At this point, HOST sends the entire compiled XPRESS program. After the last byte has been sent, HOST waits for another acknowledgement character. This time the character indicates the final status of the program load as follows:

- ^ = Successful program load
- # = Firmware version doesn't match compiler version
- @ = Program too big to fit in SC memory

If either of the latter two errors occurs, the SC clears its memory so there is no XPRESS program there at all. There is no way to get the old program back after the command has been initiated.