SAC HOST PORT COMMUNICATIONS

MicroComm DXI

DXI Host Communication Protocol

1. Intent & Scope

The MicroComm DXI quickly and reliably exchanges command and status messages with external computer systems using a serial protocol specified in this document. Command messages are messages sent from the host to the DXI system (to tell the DXI system to do something) while status messages are messages sent from the DXI system to the host (to indicate that something has happened on the DXI system).

2. Serial Protocol

The communication protocol is a simple ASCII character based one. Command/status messages consist of up to four characters followed by a several numbers, each separated by one or more spaces. Each message is terminated with an ASCII carriage return. The messages are not case sensitive. All numeric parameters are sent as ASCII decimal strings (i.e., as a sequence of the characters 0-9). All MicroComm DXI system ID numbers are limited to the range 0-65535 and the value zero is reserved for use as a null ID. These command/status messages are described in another document.

When a command/status message is sent from the host to the DXI system, or vice versa, the message is sent as a string of characters followed by an ASCII carriage return character (13 decimal or 0D hex). As all messages are terminated with the carriage return, the user should wait until the entire message has been received before decoding it. The receiver does not have to send any response to the sender (as in more complicated protocols) to indicate that the message has been received. The maximum command/status message length is 40 characters.

For example, if there is a call made from master number 1 to station number 2, then the DXI system will send a status message represented by the following string of characters:

Ical 1 2<CR>	*string of characters sent in ASCII*
73 99 97 108 32 49 32 50 13	*string of characters sent in decimal*
49 63 61 6C 20 31 20 32 0D	*string of characters sent in hex*

When writing the program code to decode this string of characters, the user should be careful to write their code so that any non-recognizable status strings are ignored. As all status messages are terminated with a carriage return, it is easy for the user to read in the message, and then decide whether to act on or ignore the message. Once the carriage return has been received, all subsequent characters are the start of the next message.