How to Successfully Communicate Between a DECsystem-10 and Another Computer Over an Asynchronous Phone Line

Author: Peter M. Hurley
Date: October 27,1972

This is a technical paper aimed at setting down the rules for successful data transmission between a DECsystem-10 and another computer over an asynchronous telephone line. The approach described in this paper has been tried and has proven to be successful. All other schemes will be considered by the author to be unworkable unless specifically demonstrated to be viable.

1) All communications to the asynchronous line must be done by a DECsystem-10 user program. The Monitor is incapable by itself of doing any error dectection or correction on asynchronous lines.

2) The DECsystem-10 user program must perform all input and output to the asynchronous line in IMAGE mode. This is necessary to insure that the Monitor does not attempt to interpret any characters as special characters. This eliminates problems caused by receiving Control-C or Control-O characters.

3) Data must be sent in blocks of less than 80 characters. Following the transmission of a block of data the DECsystem-10 program must wait for an acknowlegement that the data was received in the other computer. The reason for keeping the blocks of data smaller than 80 characters in length is that the Monitor buffer for incoming or outgoing characters is only 80 characters long. Messages longer than 80 characters could get truncated.

4) Each block of data must include a header and a trailer. The message format chosen by the author is as follows:

SOH	Start of Header (001)
MESSAGE #	0-17 octal
CHARACTER COUNT	Count of data characters
STX	Start of Text (002)
.	
.	
.	
DATA	N characters as specified
.	by character count
.	
.	
ETX	End of Text (203)
BCC1	16 bit CRC generated from
BCC2	all previous characters in
	the message

This message protocal is a subset of the DEC Standard Message Protocal for Synchronous Lines.

5) For each message received the DECsystem-10 program must send an Acknowlege message to the other computer. The format of an Acknowlege message is as follows:

SOH	Start of Header (001)
ACK/NAK,Mess #	Acknowlegement character
BCC1	16 bit CRC for the two
BCC2	previous characters

The Acknowlegement character is made up of a 4-bit ACK (12) or a 4-bit NAK (05) concatenated with the 4-bit message number to make a 8-bit character. If a message is received that agrees completely with the specified message format, then an ACK message is sent out. If a message is received where the correct message format was not used or where the CRC received and the CRC calculated did not match then a NAK message is sent out. A NAK message is also sent out if a complete message is not received within a specified amount of time.

6) Upon receiving an ACK message the sending program can then send out the next message. However, if a NAK message is received then the sending program must repeat the sending of the current message

7) When sending a message (either data messages or Acknologement messages) the sending program send the entire message with a single OUTPUT command. This insures that the transmission of the message will be continuous for the duration of the message. If a

message is sent out in several parts (more than one OUTPUT UUO) the sending program might be swapped out between parts and could cause the receiving computer to timeout and erronously think that the remainder of the message had been lost in transmission.

The approach described in this paper has one drawback. This is a simple half-duplex system. It will not handle Full-duplex modes of operation. The programming restrictions placed on the user programs require that the two programs agree at all times which computer is going to send and which is going to receive. If both programs ever tried to send messages at the same time to the other program, an interlock would occur where both programs would never be able to complete a message.