An Evaluation of the Use of a Cathode Ray Tube Light Pen Option Within the Deep Space Station Monitor and Control Subsystem

J. Thomson
DSN Data Systems Development Section

A routine has been written and tested to provide a run-time environment to evaluate the feasibility of using a cathode ray tube light pen option to simplify the operator interface with the Deep Space Station Monitor and Control Subsystem. This routine introduces two new concepts within the scope of the operator/program interface. First is the idea of using a selection list to aid the operator in executing the available commands. Second is the concept of using a tree-like hierarchy of information display for improved fault detection and isolation.

I. Introduction

During the previous year, a routine was written and tested to provide an environment to evaluate the feasibility of using a light pen option within the DSN Monitor and Control Subsystem. The light pen, along with the current cathode ray tube (CRT) terminal, would be used to simplify the operator interface with the Station Monitor and Control Subsystem. The routine was written to operate within the station Monitor and Control Subsystem Operational Program, for purposes of the evaluation.

II. Hardware Description

The Monitor Program resides in a Xerox Data Systems (XDS) 920 computer contained within the Digital Instrumentation Subsystem. This subsystem is connected to the other Deep Space Station subsystems via several

types of communication links. The current operator/program interface uses a CRT station to display the status of the DSS and to allow input of control messages (commands).

The light pen option can be added to the CRT station and consists of the following:

(1) Two circuit boards fitting within the CRT controller.

(2) A light pen holder assembly plugged into the controller and connected to the light pen.

(3) A light pen; a small hand-held device with a photo sensor at its end.

III. Hardware Operation

The light pen system functions as follows: When the pen is held up to a character displayed on the CRT, its photo sensor receives an impulse every time the character is refreshed. This impulse causes the light pen controller to copy the contents of the CRT refresh address register into the light pen address register. Thus, the location of the character selected by the pen can be saved. If the address remains constant, the light pen controller causes a blotter character to flash at the selected location. At this point, the light pen switch can be pressed causing the CRT controller to enter the interrupt mode.

IV. Monitor Input and Data Display

The Monitor Program uses the CRT as an interactive terminal for the input of commands and to display the current status of a Deep Space Station in the form of a grid of 20 parameters. Each of the 110 available parameters gives the status of some subsystem or value by displaying the parameter's name and its value. The value's color is used to give parameter status. For example, a value displayed in red indicates an alarm condition, and a value displayed in yellow indicates a warning condition. Normal or good values are displayed in green or blue. Commands may be keyed on line 0 of the CRT screen and executed by pressing the interrupt key. The commands are used to control operational features of the Monitor and may be used to select a single parameter or a set of 20 parameters, called a format, for CRT display.

V. Purpose

The task can be divided into two main goals:

(1) To develop the essential software for communication between the light pen and the XDS 920 computer, and to test the available hardware for any defects and for ease of physical manipulation.

(2) To develop the software needed to provide a run-time environment for evaluating the use of the light pen to simplify the operator interface with the Monitor Program.

VI. Program Description

This routine introduces two new concepts within the scope of the operator interface. First is the idea of using a selection list to aid the operator in executing the available commands. Second is the concept of using a tree-like hierarchy of information display for improved fault detection and isolation.

Each time the light pen switch is pressed, two pieces of information must be obtained: first, if necessary, the source of the interrupt can be determined by requesting the "status word" which contains a 1-bit flag if the interrupt was from the light pen rather than the keyboard; second, the light pen address must be requested—once translated, the address yields the row and column of the selected character.

Once this information is obtained, the program can be made to react in any given way. This program is actually made up of two separate subprograms. The first has been written as a means of creating the commands used by the monitor with a minimal use of the keyboard. The commands can be divided into two or three fields. Since there are a limited number of choices for field-1, they can be displayed within a "selection list." After field-1 is selected, the corresponding field-2 choices can be listed, etc. Once the command has been determined (and created on line 0), it can be processed and executed by the Monitor software.

The second subprogram has been provided as an alternative to the formatted display of parameters. All of the approximately 110 parameters displayed by the Monitor program have been separated into several groups—and some groups have been separated into smaller subgroups. Each group has a "header" parameter associated with it. The color of the header is determined by worst status of any of the parameters within the group. For example, if one parameter turns yellow and one turns red, the header will turn red, indicating an alarm condition within the group. The headers are grouped together in a similar fashion so that all parameters together form a

"tree." If any single parameter turns red, the root of the tree will also turn red, indicating trouble. Thus, only one parameter needs to be displayed.

A "page" of information about any subtree (a header and its group) can be accessed by using the light pen to call the subprogram and then advancing and/or retracing through the tree to the proper subtree. Since this was a sample program, the pages were simple and consisted of a title, a list of parameters or headers within the group, some information messages pertaining to the group, and a return list of the previous pages that could be recalled. The user could advance to a subtree by selecting a header parameter displayed on the current page and retrace to a previous page by selecting a page title within the return list.

VII. Results

There were several problems which involved the particular hardware used. The light pen lacked sensitivity to certain characters and could only respond to characters in blue. This tended to slow down an inexperienced user. Another difficulty was that wherever messages were being output onto the CRT screen, both the light pen and keyboard were disabled. This tended to cause less difficulty with the light pen than with the keyboard, since keyed-in commands with a dropped character resulted in an error.

Pressing the light pen switch twice in close succession sometimes caused the command to be erased. Minor improvements in the light pen hardware and the CRT routine should help to alleviate these problems.

The basic light pen operating system required to find the light pen row and column used 160 24-bit locations in addition to the CRT routine. The page handling routines used 270 locations. Most of the 2000 locations were used to create the Monitor commands. With this much core usage, it became necessary to delete part of the Monitor Program to make space available. New computer hardware with increased core memory and a disk operating system would alleviate this problem. This would allow room for a complete monitoring and diagnostic system using the page display subprogram. This program would also be ideal for a disk operating system, since only a small amount of data is needed in core for the displayed page of information.

VIII. Conclusions

The concept of a selection list helped to make the user aware of the full scope of commands available and simplified their execution. The page display subprogram produced successively higher degrees of diagnostic information and thus provided a fast and accurate means of pinpointing a problem area in the system. The simple tree structure also helped an unfamiliar user to understand the structure of the DSS and its subsystems. These simplications in the operator interface could be used to lower the level of training required for operational personnel and thus could provide a savings in operational costs for the DSN.