IMPLEMENTING THE HAWLEY MOUSE MODEL X063X AND RANDOM ACCESS INCORPORATED MU-2 SERIAL INTERFACE

by

Roy William Richard

B.R.E. Chicago Technical College, 1952

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1985

Approved by:

David A. Gustafson

Dr. David Gustafson

CONTENTS

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

This paper describes an interface between a mouse and screen oriented user programs using the UNIX operating system.

This project originated as a five week summer semester project. The goal of the project is to provide a set of programs which will allow a person, when running screen oriented programs, to use a mouse at a remote terminal. The mouse controller is to be placed in series between the terminal and the RS-232 communication line. It is intended to provide the capability to move the cursor and enter up to three three character commands as desired using the capabilities of the terminal and software without restriction. The addition of the mouse will be transparent to the system and application programs. Using the mouse will increase the speed and accuracy of cursor movement. Depending on the needs of the user three commands requiring multiple keystrokes will be reduced to one push button each.

Input devices in general have been developed to meet a specific need using the currently existing technology or the materials available at the time. The light pen, data tablet, keyboard, and mouse have undergone a number of improvements since their inception. In many cases the improvements have resulted in enhanced performance as well as widening the applications for the input device.

Dr. Douglas Engelbart, then affiliated with the Stanford Research Institute, developed the mouse as part of his efforts to develop a system which is designed to augment the Human Intellect. DR. Engelbart's goals were as follows: (1) to find the factors that limit the effectiveness of the individuals basic information handling capabilities in meeting the various needs of society for problem solving in its most general sense; and (2) to develop new techniques, procedures, and systems that will better adapt these basic capabilities to the needs, problems, and progress of society (Engelbart 63). The mouse was developed as part of the overall concept.

The first mouse was a two by three by four inch block of wood with three buttons on top. Its bottom was hollowed out to make room for two wheels, each of which was attached to the shaft of a separate potentiometer. (Engelbart 68)

"The mouse is a hand held device with rollers on its base. Moving the mouse across a flat surface causes the rollers to turn and potentiometers coupled to one roller sense the relative movements in two orthogonal directions the motion is converted into digital values and is used to determine the direction and magnitude of the mouse's movement. A set of experiments comparing the relative speed and accuracy obtained with this and other selection devices showed the mouse to be better then a light pen or a joy stick(English 67)."

In the early 1970s, Xerox's Palo Alto Research Center (PARC) set out to develop powerful personal computers for use in OFFICE Automation Research. The computer developed by PARC, the ALTO, was the next major system to use a mouse(Card 83). From 1972 to about 1975 Engelbart's design of right angled wheels were used. It was soon replaced by a mouse which used a large ball bearing and was manufactured by Jack Hawley who had become president of the newly formed company called the Mouse House.

This new model used the ball bearing to smooth out the drag caused by one wheel being dragged against its normal axis of rotation It still retained the two wheels but they now were in contact with the ball bearing instead of the surface that the mouse was being moved across. Since its introduction the mouse has become the most successful and widely used pointing device. W. K. Card has conclusively demonstrated not only that the mouse is significantly faster then joy stick and keyboard controls, but also that the speed of the cursor manipulation with a mouse approaches minimum positioning time achievable (Card 78). The major limitation of the mouse appears to be in the reliability of the tracking ball mechanism.

The mouse is only one of a group of input devices that are currently available. Depending on the main thrust of your interest be it graphics, office automation, factory automation,robotics ,or systems development, the choice of which input device is best can be effectively debated from each persons point of view It is clear however that the mouse is currently enjoying a certain amount of popularity.

1.1 Other Input Devices

Light pens have been used for many years in CAD and in special tracking systems used by the military and the Federal Aviation Agency. The resolution of a light pen is poor in comparison to the resolution of a tablet and cursor. This is the combined result of the light pen's relatively large field of view and the display screens limited image sharpness. This presents a problem when the light pen must discriminate among several tightly spaced elements in the image (Scott 82). Although light pens provide an excellent pointer mechanism, their use in an integrated programming environment would require the specification of replacement text and thus would require a secondary interface such as a keyboard. The resulting combination would result in constant switching from keyboard to light pen and back again. With experienced users the mouse has been shown to be faster at text manipulation than the light pen (English 67).

The touch sensitive screen, where the human finger is used as the pointer, has been tried but the resolution required for text manipulation has not been attained. The touch sensitive screen is being used in applications where it is necessary only to make selections from a menu as in a menu driven testing sequence used to test a piece of electronic gear. An example of the touch screen using menu selection is the Fluke 2450 measurement and control system, which is made up of a Fluke 2400 data acquisition system with the addition of a 1720 processor. The 1720 processor has a five inch touch screen and a floppy disk drive to store programs and data from measurements (Guteri 83).

The data tablet has the potential to replace the mouse as the favorite pointing device since it has the capability to be used as a multi mode input device. While the mouse has been shown to be more reliable in retaining a cursor location and in ease of movement the input of a signature or the entry of characters is not practical. As a pointing device and/or a picking device the mouse has been shown to out perform the data tablet. Data tablets are currently used for menu selection, upper case alphanumerics recognition, signature verification, and graphics input in addition to the pointer function.

Voice data entry systems have a great deal of promise and they attracted a great deal of attention several years ago, Due to the large memory requirements and limited vocabulary, it is primarily used in specialized environments where one word commands are normally used. Voice data entry systems are programmed or trained to recognize voice patterns of an individual. The current state of voice recognition is such that if an individual using the voice entry system has a cold the system will make a great number of input errors.

Given the current state of the art a comparison between the mouse, light pen, touch sensitive screen, data tablet, and voice data entry leads to only one conclusion. The use of the mouse for cursor movement in an environment where text editing is heavily used is the best choice. The light pen requires constant motion to the CRT and back to the keyboard while the mouse need only be released. The touch sensitive screens resolution is not currently fine enough to allow accurate placement of the cursor. The data tablet requires additional hand and eye coordination beyond that required by the mouse. The need for training the system and limited command structure currently available places the voice entry system at a disadvantage when compared to the mouse.

1.2 Interfacing A Mouse To Your Computer

There are several types of interfaces possible when adding a mouse to your system some of which are: analog to digital converter, RS-232 port in series with a communication line, RS-232 port, keyboard interface, and bus interface.

The first type of interface is to use the analog to digital converter in the form of the Personal Computer game adapter (figure 1). When the game adapter is used in conjunction with a mouse, the mouse substitutes for the joy stick. The big disadvantage to this approach is the programmer must continually poll the game adapter. The adapter does not have the ability to interrupt the program and say when one of the analog signals have changed (Foth 84). Since application programs must be arranged to poll the game adapter any programs not specifically written for use with the mouse are unable to use the mouse.

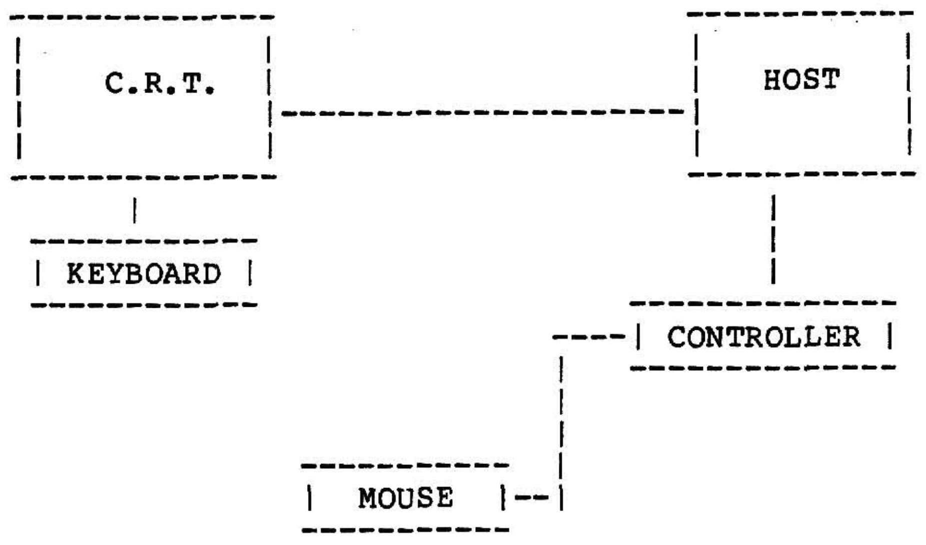

Figure 1. GAME ADAPTER INTERFACE

The second type of interface is to use the mouse controller in series with the RS-232 communication line. Using this method the mouse controller is placed in series with a communication link, such as the Mu-2 controller used in this implementation (figure 2). When the mouse controller is in series with the RS-232 communication link the controller passes information between the terminal and host computer in a transparent mode. With the mouse controller in series with the communication link the application program has no awareness of the mouse and any characters generated by the mouse appear to be coming from the keyboard. As a result any screen oriented programs may be used with the mouse. Programs can also be written to communicate with the mouse and take full advantage of the mouse's capabilities.

```
--------------                               ------------
|            |                               |          |
|   C.R.T.   |      --------------           |   HOST   |
|            |-----| CONTROLLER |-----|          |
|            |      --------------           |          |
--------------            |                  ------------
      |                   |
 ------------             |
 | KEYBOARD |             |
 ------------             |
                          |
                          |
            -----------   |
            |  MOUSE  |---|
            -----------
```

Figure 2. RS-232 SERIES INTERFACE

The third type of interface is to have the mouse controller connected to an existing RS-232 port at the host and communication between the controller and host would be handled like other peripheral devices (figure 3). This method is probably the most popular choice. Connecting the controller to an unused RS-232 port is the approach used by the Mouse Systems, Summagraphics, USI, and by Microsoft and Random Access Mu-1 controller (when used with the Hawley Mouse). Using this interface the computer software communicates with the microprocessor inside the mouse by issuing commands to and reading responses from the mouse over one of the programable communication devices.

There is no standard mouse language each mouse has its own command and response syntax. Some mice however can emulate other positioning devices such as bit pads. The advantage of this approach is that most Personal Computers already have at least one RS-232 port (Foth 84). The software must be made to interrogate the RS-232 port,as a result existing programs are not able to have the advantage of using the mouse. In many cases the use of an RS-232 port would require additional hardware to be added.

```
--------------                    ------------
|            |                    |          |
|   C.R.T.   |                    |   HOST   |
|            |--------------------|          |
|            |                    |          |
--------------                    ------------
      |                                 |
 ------------                           |
 | KEYBOARD |                           |
 ------------                    --------------
                             ----| CONTROLLER |
                             |   --------------
                             |
                             |
              -----------    |
              |  MOUSE  |----
              -----------
```

Figure 3. RS-232 INTERFACE

The fourth type of interface is to use the keyboard in this arrangement the mouse interface is plugged into the keyboard socket and the keyboard plugs into the mouse controller (figure 4). The advantage of course is that the computer does not see the mouse and all characters produced by the mouse are recognized as coming from the keyboard (Foth 84). Since the host cannot tell the difference between a character typed at the keyboard and one it receives from the mouse interface, programs need only to read from the keyboard to detect mouse movement. Just like characters produced at the keyboard, characters produced by the mouse and delivered through this interface can generate interrupts. The disadvantage of this arrangement is that there is no standard keyboard socket requiring the controller to be arranged for a specific keyboard. User programs are not able to communicate with the mouse requiring all communication with the mouse to originate at the keyboard.

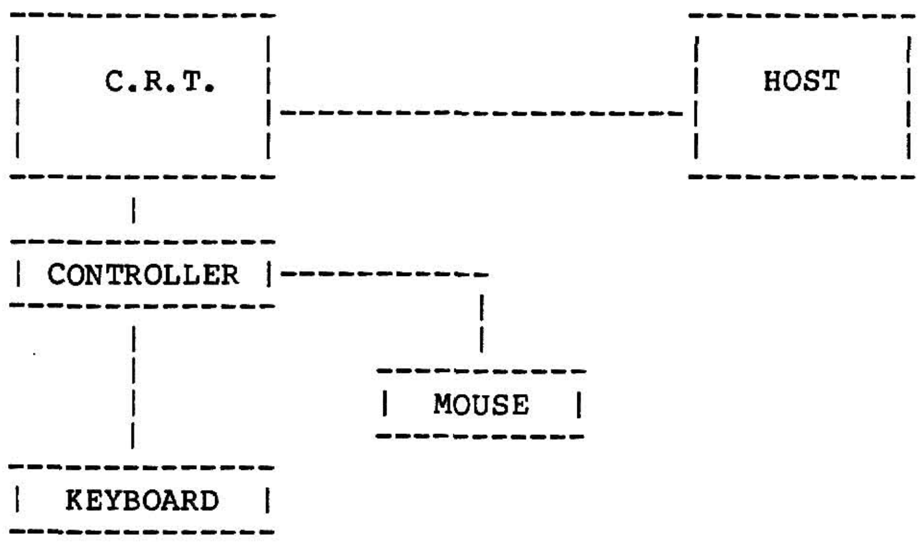

Figure 4. KEYBOARD INTERFACE

The last type of interface is the dedicated board which is directly attached to the bus (figure 5) of the host. The bus interface requires that the software must be arranged to communicate with the mouse. the mouse uses this awareness to take full advantage of the capabilities of the mouse controller. One major advantage of this arrangement is speed since it does not go through the keyboard or an RS-232 interface (Foth 84).

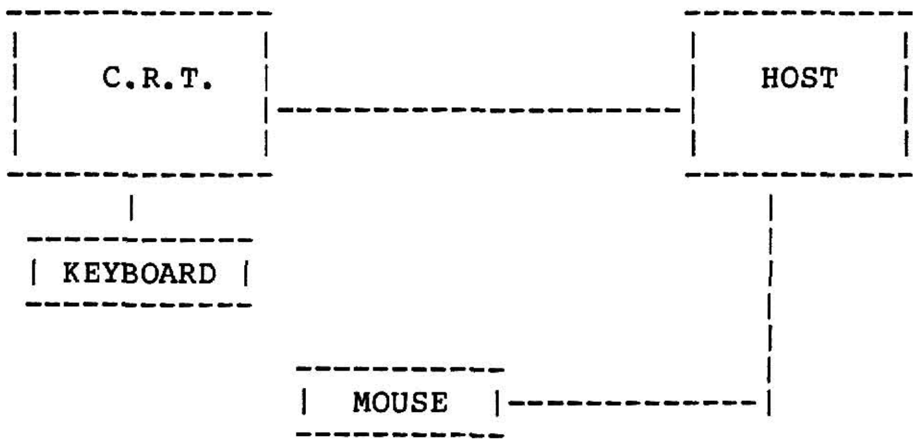

Figure 5. BUS INTERFACE

The disadvantage of this interface is the fact that it is limited to that particular computer. Software not arranged for use with the mouse cannot use the mouse as was the case with the RS-232 series interface and keyboard interface. One more disadvantage is if the application programs arranged for use with the mouse are not provided with an alternate means of cursor movement and selection a mouse malfunction renders the program useless.

Of all the interfaces placing the mouse controller in series with a communication line provides the most flexibility allowing the use of existing software as well as being able to take full advantage of the mouse controllers capabilities when using programs specifically written for use with a mouse in this configuration.

Once a mouse has been physically connected to the computer we now need the software interface. There are three levels of software interface for the mouse: system managed interfaces,transparent interfaces, and application managed interfaces.

1.3 <u>System Managed Interfaces</u>

Digital Research's Concurrent CP/M-86 with Windows,Microsoft's Windows, Quantum Softwares System's QNX and Visi Corp's VisiOn are examples of system managed interfaces. In these environments the mouse is an integral part of the operating system, and applications that run in these environments can make use of the mouse through standard operating system calls. Some of these systems provide mouse to cursor translations for applications that understand cursor movements but are not fully integrated into the operating system. These systems are arranged to work with specific brands of mice. VisiOn interrogates the mouse for a serial number and if the mouse does not present the serial number encoded into the software the software does not run (Foth 84). Programs developed for specific operating systems prior to implementation of windows or the use of the mouse may run without problems or may require some modification.

1.4 Transparent Interfaces

This type of interface uses software in the mouse controller in order to intercept the command characters and not allow them to reach the host computer or CRT.

At the transparent level of interfaces there are Trillian's VisiALL, Quarterdeck's Desq, Microsoft's Makemenu, also mice interfaced through the keyboard such as Comco Concept's Keyboard mouse, Corman Custom Electronic's PC Mouse Trap with the Hawley mouse, and Logitech's Logimouse provide a high degree of transparency. When used with this software or these mice, the applications are not aware that a mouse exists on the system. This software or mice appear to the system as if someone were typing at the keyboard. As with system managed software interfaces, only certain vendor's mice work with certain transparent interface software (Foth 84). This type of interface does not interfere with the running of programs developed prior to the installation of the mouse controller. Our implementation is an example of the transparent level of interface.

1.5 Application Managed Interfaces

At the lowest level are those software interfaces that require the application software to communicate with the mouse specifically, either through a set of subroutines or through a program arranged to translate the commands for the mouse into controller readable code. The major disadvantage of this type of interface is that it can not be used with packages that have no awareness of the mouse. Countering that disadvantage is the fact that for new applications this kind of interface provides the most control over the mouse (Foth 84).

For our purposes the transparent interface is the interface that best meets our requirements of being able to utilize existing software with a mouse.

The mouse that is being used is a HAWLEY mouse from the *MOUSE* HOUSE. The mouse controller is from RANDOM ACCESS INCORPORATED and is designed to be used with a variety of RS-232 compatible data terminals. The MU-2 mouse controller is inserted in series with the RS-232 line from the host to the terminal and will transmit mouse position and button information in several formats without interfering with the normal data transmission. For the purposes of this project the mouse controller is being configured to provide information to the user programs as to the desired location of the cursor. This will be presented to the user programs as if the information is being provided by the cursor control keys.

This implementation is specific to the Mu-2 mouse controller from Random Access Incorporated and the Hawley mouse which the controller is designed to work with. If it is desired to use another manufacturer's controller and mouse associated with the other controller, changing the control information to agree with the requirements of the replacement controller should enable this program to continue to function. The replacement controller should be of the same type as the Mu-2 controller which is to say that it must be placed in series with the terminal and host computer.

CHAPTER TWO

REQUIREMENTS

This is to provide a program to enable software to be used with a mouse connected in series with an RS-232 serial interface. The user will call the mouse program in order to view a menu of programs available for use with the mouse. The user will be able to select a program, initialize a program for use with the mouse, or exit. In order to use the mouse with a screen oriented program, the user will be requested to answer a series of questions designed to enable a file to be created which will be specific for that user program. After a program has been set up for use it will be listed in a menu. To select a program the user will be asked to position the cursor next to the program to be used and press a button on the mouse. After selection of the user program it will be necessary to add any files as required by the user program. The user will be returned to the menu after a program has been initialized or run.

2.1 Inputs

The inputs to the mouse program consist of the user interface, text file of programs, and text files of control characters.

The user interfaces with the mouse program by providing the four cursor control characters and the optional characters for the three buttons. After a program has been initialized requesting that program is only a matter of using the mouse to select the program.

The text file of programs is a list of available programs which have been initialized for use with the mouse controller. The file of programs is user readable and is updated whenever a new program is initialized for use with the mouse. The name of the file of programs is mselist.

The text file of control characters consist of the requirements specified by the manufacturer of the MU-2 controller Random Access Incorporated, along with the user dependent character information. There will be a file for each program initialized. These files will in some cases contain characters that are unprintable on some terminals. The required control characters that are needed for each file consist of:

1. Enter command input mode	- control-A
2. Leave command input mode	- control-B
3. Enable mouse data transmission	- control-R

Each text file of control characters will be named with the application programs name with an extension of .mse.

2.2 Functions

The primary goal is to provide an easy method of setting up a mouse controller for use with a variety of programs which require different control information. Creating the various control files is to be as simple as possible with the user entering only information which is specific to the program being initialized. The user will be directed to answer a series of questions designed to retrieve the characters which are to move the cursor. An opportunity to assign up to three characters per button will also be provided to the user at the same time.

The mouse set up is achieved by the user specifying the mouse program or may be accomplished without using the program if he is aware of the existence of the various control programs.

The calling program is arranged in such a manner as to allow the user to view a menu of programs which are available for use with the mouse. At this point the user will have several options. The first option is to select a program which is on the menu of available programs. The software will return the user to the menu screen after completion of the selected program. The second option is to initialize a screen oriented program for use with the mouse after failure to find it listed in the menu. The software will return the user to the menu screen after initialization allowing the user to verify that the control file has been created and is ready for use. The third option is for the user to select exit which quits the program.

2.3 Output Files

The output files consist of a group of files each of which is specific to a particular program. these files contain the control characters for the mouse controller. This file of control characters, which is user-program specific, is sent to the mouse controller. The mouse controller in turn is set up to return to the host the specified characters for cursor movement when the mouse is moved in the desired direction. The optional mouse buttons are also activated at this time, if the control information was originally provided.

The basis of our approach to this implementation is to provide as much flexibility as possible by providing several short C and shell routines. The shell routines use only basic shell commands without any involved constructs. The C routines are simple with the hope of providing portability to other UNIX systems.

```
            MOUSE
           CONTROLS
              v
         ------------
 USER-->-| ACTIVATE |-->-
         |  MOUSE   |   |
         ------------   |
       MOUSE ^          |
    CONTROLLER   --------
                 |     FILE OF
                 |     PROGRAMS
                 |     .MSE  v
                 |     -----------
           PLIST ----->-| DISPLAY |->-
                       |  MENU   |  |
                       -----------  |
                  MOUSE ^           |
               CONTROLLER    --------
                             |   -------------
                   USER MENU -->-| PERFORM   |--A]   ------
                   SELECTION     | SELECTION |->----|EXIT|
                                 |           |--B]   ------
          PROGRAM                -------------
           MOUSE                 MOUSE ^
          CONTROLS               CONTROLLER
             v
           -----------
     A]-->-|   RUN   |-->-PROGRAM
PROGRAM    | PROGRAM |    RESULTS
 NAME      -----------
             ^    ^ MOUSE
 APPLICATION      CONTROLLER        SET_CONT
   PROGRAM                            v
                              --------------
                     B]---->--| INITIALIZE |--->-MOUSE
                  INITIALIZE  |  MOUSE     |     CONTROL
                              | CONTROLS   |     FILE
                     USER-->- |            |
                 RESPONSES    --------------
```

Figure 6. PROGRAM FLOW

It is intended to provide user friendly screens for initialization. A menu of available programs will be provided that allows a quick reference as to the programs that are currently available. If a user is familiar with the system and knows that the program he wants to use with the mouse is already available, standard UNIX commands will allow him to call the mouse without going through the menu selection routine.

The initialization screen will provide one question at a time to the user. As each question is answered the following question will be presented without removing the preceding question. This will allow the user to review his answers and reinitialize the program if revisions are required.

Interfacing a mouse using an RS-232 interface between the host computer and terminal will require the mouse controller be initialized in accordance with the manufacturers requirements. The RANDOM Access Incorporated mouse controller is capable of being arranged to provide four distinct formats depending on the requirements of the user program. This implementation will be using format three which provides selected characters for cursor movement. The mouse controller also provides for sending up to three characters to the host for each of the three keys on the mouse. This option is being provided but will not need to be selected by the user. The baud rate and frame rate are switch selectable and must agree with the host computer and terminal.

CHAPTER THREE

DESIGN AND IMPLEMENTATION

Our goals are to provide the necessary control software to allow a user at a remote terminal to use the HAWLEY mouse when running screen oriented programs. It is intended that the user will be able to activate the mouse whenever desired. Since there are three buttons which are user programable the files for the control of the mouse should be maintained in the user's directory. This will allow the user to program the keys for commands that would be normally used in his applications.

If one of our concerns had not been flexibility, some other means of coding could have been used to provide a more efficient program. It is thought that by giving up some efficiency we retain the flexibility for future expansion. In our attempt to allow for future expansion the routine for initialization is a complete module and may be replaced by a similar module arranged for another mouse controller. If it is desired additional initialization modules may be added with minimal modification to other routines. It is also our hope that portability between UNIX and UNIX like operating systems will be achieved.

Common shell commands have been used without using intricate constructs. The c program used in this implementation used common C instructions and the standard library. This was done to allow the program to be easily moved to non-unix operating systems. Most C compilers that operate with other then UNIX operating systems have implemented the standard library. As an example the non-unix compilers implement scanf, strcpy, and strcat.

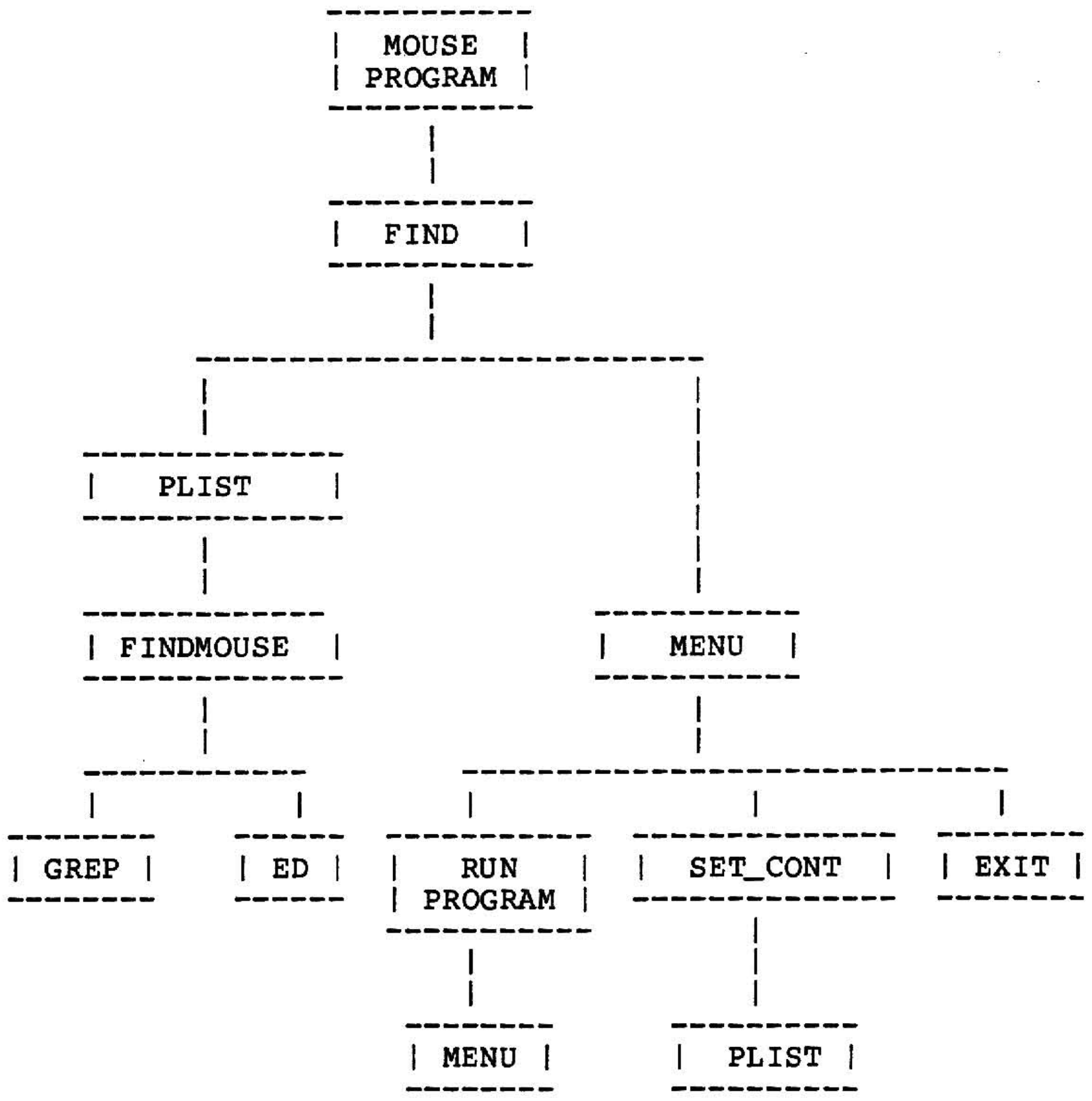

Figure 7. PROGRAM HIERARCHY

The C routines are to be used for locating the cursor and selecting the proper program for use with the mouse. The file of available programs is created by searching the current directory for file names with the extension of .mse. The extension is removed by using an editor and this list is saved in the users current directory as mselist and is the file used for printing out the list of available programs.

The files with the extension of .mse are the control files created by the initialization program. The control files are made up of the mouse controllers start and end control characters with the required user program control information provided by the user from the requirements of the screen oriented user program. The user programs will be able to use the mouse for cursor control and optionally use the three buttons to provide up to three characters for each button as desired.

The control files will consist of the following data records;

1. control-A v w x y control-B
2. control-AL abc control-B
3. control-AM abc control-B
4. control-AR abc control-B
5. control-R

The letters v w x y represent any ascii character but null.

The letters abc represent any ascii character but control-B.

The program uses common shell routines and simple C constructs for a program of approximately 170 lines of code. The mouse implementation consists of a C program, which uses some existing shell routines where practical.

The user wishing to view what programs are available or initiate a program calls in the routine **mouse.**

The **mouse** programs main function is a continuous while loop which calls the routine **find.** The routine **find** activates the mouse and calls the routine **plist. Plist** displays the menu on the crt and returns control back to **find.** The routine **find** now provides selection instructions and waits for the user to choose an application program, initialize a program, or exit from the **mouse** program. Depending on the selection **find** will exit, call the routine **set_cont** to initialize an application program, or call an application program. When the application program is completed control is returned to **find** which returns to **main.**

The routine **set_cont** provides a question and answer session in which the user is prompted to provide the necessary control characters for the program being initialized. After the information has been collected a file of control characters is created using the name of the program being initialized and adding the extension of .mse The routine **set_cont** returns to

find and **find** returns to **main.**

The routine **plist** calls the shell program **findmouse** which checks the current directory for files with the extension of .mse. The names of these files are placed into a file called mselist with the extension removed. **Findmouse** then returns to **plist. Plist** then displays the file of program names and returns to **find.**

The **mouse** *program* calls in **find, find** calls in **plist,** which calls in **findmouse. Findmouse** is a shell routine which uses the grep routine to find the files with an extension of .mse in the current directory. The extension is stripped off using the (Ed) editor and the file names are placed in the file mselist in the current directory for later use, returning to **plist.**

plist then displays the available programs for use with the mouse and returns to **find.** The user at this point may choose a program, initiate a new program, or exit. The user uses the mouse to make his program selection exit or enter the initialization mode.

If the user chooses to initiate a new program **set_cont** is called in which there is a question and answer session stepping the user through a series of questions designed to obtain the required information for the use of the mouse. Upon completion of the question and answer session the user is returned to **find.**

If the user chooses to use one of the programs that are available for use with the mouse, the user inputs any additional control information and/or files as the user would normally enter when not using the mouse. Upon a normal exit from the program, the user is provided with the menu for further selection or exit.

Testing of this program was done on a step by step basis. As each routine was developed it was tested by verifying that the routine would function properly at its extremes and at its boundary points. As an example the routine to find the files of the programs that have been initialized was tested when no files were present and again with one file in existence. The mouse was used by several fellow students when the mouse was activated after program selection. They found the mouse to move the cursor around the screen much more accurately then using the keys of the keyboard.

CHAPTER FOUR

CONCLUSIONS AND EXTENSIONS

Since this implementation had only a five week period to develop and test the programs using the mouse there was only a limited amount of time actually used in testing people's reactions to using the mouse. The overall reaction was good and some comments were made as to the possibility of obtaining a mouse for their own use. This implementation has provided one format for the Hawley Mouse. This format provides the cursor movement in any direction based upon the requirements of the user program. In addition to cursor movement the mouse is provided with three buttons which can be programmed to provide up to three characters to the user program on an optional basis.

The controller by Random Access Inc. is capable of furnishing four different formats to the host computer depending on program requirements. If programs are available or are being developed which would be suitable for use with the mouse and one of the formats would be ideal there appears to be no reason not to provide the capability using this method. The calling program can easily be arranged to provide for the setting up of the additional formats. It would then be a matter of designing additional screens to ask the required questions for each format being used. It appears that the addition of icons on the screen would be a desirable and useful extension.

BIBLIOGRAPHY

[1] Card, S.K., English, W.K. ,and Burr B.J. "Evaluation of Mouse, Rate Controlled Isometric Joystick, Step Keys, and Text Keys for Text Selection on a CRT" Ergonomics, 21 (8), pp 601-613, August 1978

[2] Card, S.K. "The evaluation of pointing Devices" Digest of Papers COMPCON 83, pp 179, Spring IEEE

[3] Engelbart, D.C. "A Conceptual Framework For The Augmentation Of Man's Intellect" in Vistas in Information Handling Vol 1,pp 1-29 D. w. Howerton and D. C. Weeks eds. Sparton Books Washington D. C. 1963

[4] Engelbart, D.C. ,English, W.K. "A Research Center For Augmenting Human Intellect", Proceedings 1968 AFIPS FJCC 33 No 1, pp 395-410, (Fall)

[5] English, W. K. Englebart, D. C. ,and Berman, M. L. , "Display-Selection Techniques for Text Manipulation", IEEE Transactions on Human Factors in Electronics VOL HFE 8 NO 1, pp 5-15, March 1967

[6] Foley, J. D. and Van Dam, Andries, Fundamentals of Interactive Computer Graphics, Addison Wesley Publishing Co., Reading Ma., 1982

[7] Foth, T. "The Origin, Anatomy, and Varieties of Mus Computeralis" Softalk Vol.2, pp 88-96, April 1984.

[8] Goodwin, N. C. , "Cursor Positioning on an Electronic Display Using Lightpen, Lightgun, or Keyboard for Three Basic Tasks", Human Factors 17(3), pp 289-295, June 1975

[9] Guteri, F., "Supersmart Equipment" , IEEE Spectrum, pp 48-50 Jan, 1983

[10] Liscano, G. "Voice Data Entry for Computers" Man Machine Interfaces for Industrial Control, Technical Publishing Co. 1980.

[11] Mehr, M. H. , and Mehr, E. "Manual Digital Positioning in 2 Axis A Comparison of Joystick and Track Ball Controls", Proceedings of the Sixteenth Annual Meeting of The Human Factors Society 1972

[12] Scott,J. E., Introduction to Interactive Computer Graphics, John Wiley and Sons Publisher, New York N. Y.,

1982.

[13] Thornburg, D.D. "The Computer as a Bandwidth Diode The Challenge and Promise of Low-cost Alternatives to the Keyboard" , Digest of Papers COMPCON 83, pp 172-174, Spring IEEE

APPENDIX A

<u>USER'S MANUAL</u>

This instruction covers the use of the programs which implement the operation of the HAWLEY mouse on a computer using a unix operating system . It will be necessary for a user to have access to the programs "mouse" and "findmouse".

Instructions for the physical connection of the HAWLEY mouse and RANDOM ACCESS controller are provided in the instruction manual provided by RANDOM ACCESS INCORPORATED.

Once physically connected a user need only to call the program mouse to activate the HAWLEY mouse. The mouse will become active immediately. The mouse program will provide a list of programs that are available for use with the mouse. Initially the mouse program is arranged to move the cursor down for any direction of mouse movement. Using the mouse you need only to move the cursor to the selected program ,or to initiate to initialize a new program for use with the mouse, or exit to quit the program. Once the mouse has been activated it may be disabled by typing a control-T(DC-4). To enable the mouse a control-R(DC-2) is required to be typed in.

To initiate a screen oriented program for use with the mouse it will be necessary for the user to move the cursor to initiate and push the left button on the mouse. The following is a sample of a program being arranged for use with the mouse.

APPENDIX A

```
THE FOLLOWING PROGRAMS MAY BE USED WITH THE MOUSE.
vi
initialize
exit
MAKE SELECTION BY MOVING CURSOR TO DESIRED PROGRAM.
THEN PRESS LEFT BUTTON ON MOUSE.
TO EXIT MOVE CURSOR TO EXIT THEN PRESS LEFT BUTTON ON MOUSE
TO INITIALIZE A PROGRAM FOR USE WITH THE MOUSE.
MOVE CURSOR TO INITIALIZE THEN PRESS LEFT BUTTON ON MOUSE.

PROVIDE NAME OF PROGRAM TO BE USED WITH A MOUSE.
NAME = atest
WHAT CHARACTER IS USED FOR MOVING CURSOR UP.
UP = k
WHAT CHARACTER IS USED FOR MOVING CURSOR DOWN.
DOWN = j
WHAT CHARACTER IS USED FOR MOVING CURSOR LEFT.
LEFT = h
WHAT CHARACTER IS USED FOR MOVING CURSOR RIGHT.
RIGHT = l
BUTTONS (L)EFT (M)IDDLE (R)IGHT CAN BE DESIGNATED TO SEND
UP TO THREE CHARACTERS FOR EACH BUTTON WHEN PRESSED.
WHAT CHARACTERS ARE TO BE USED FOR THE LEFT BUTTON
LEFT BUTTON =
WHAT CHARACTERS ARE TO BE USED FOR THE MIDDLE BUTTON
MIDDLE BUTTON =
WHAT CHARACTERS ARE TO BE USED FOR THE RIGHT BUTTON
RIGHT BUTTON =
```

APPENDIX A

```
THE FOLLOWING PROGRAMS MAY BE USED WITH THE MOUSE.
atest
vi
initialize
exit
MAKE SELECTION BY MOVING CURSOR TO DESIRED PROGRAM.
THEN PRESS LEFT BUTTON ON MOUSE.
TO EXIT MOVE CURSOR TO EXIT THEN PRESS LEFT BUTTON ON MOUSE
TO INITIALIZE A PROGRAM FOR USE WITH THE MOUSE.
MOVE CURSOR TO INITIALIZE THEN PRESS LEFT BUTTON ON MOUSE.


program complete
```

APPENDIX A

Failure to provide a character for each direction of motion or more then one character will cause an error message and the question will be repeated. Since the buttons are used on an optional basis and may have from one to three characters an error message is provided when more then three characters are selected. An example of the error messages on a typical screen follows:

APPENDIX A

```
THE FOLLOWING PROGRAMS MAY BE USED WITH THE MOUSE.
vi
initialize
exit
MAKE SELECTION BY MOVING CURSOR TO DESIRED PROGRAM.
THEN PRESS LEFT BUTTON ON MOUSE.
TO EXIT MOVE CURSOR TO EXIT THEN PRESS LEFT BUTTON ON MOUSE
TO INITIALIZE A PROGRAM FOR USE WITH THE MOUSE.
MOVE CURSOR TO INITIALIZE THEN PRESS LEFT BUTTON ON MOUSE.

PROVIDE NAME OF PROGRAM TO BE USED WITH A MOUSE.
NAME =
PROVIDE 1 TO 15 CHARACTERS FOR PROGRAM NAME.
PROVIDE NAME OF PROGRAM TO BE USED WITH A MOUSE.
NAME = atest
WHAT CHARACTER IS USED FOR MOVING CURSOR UP.
UP =
EACH DIRECTION OF CURSOR MOVEMENT MUST BE PROVIDED.
USING ONE CHARACTER PER DIRECTION.
WHAT CHARACTER IS USED FOR MOVING CURSOR UP.
UP = k
WHAT CHARACTER IS USED FOR MOVING CURSOR DOWN.
DOWN =jg
EACH DIRECTION OF CURSOR MOVEMENT MUST BE PROVIDED.
USING ONE CHARACTER PER DIRECTION.
WHAT CHARACTER IS USED FOR MOVING CURSOR DOWN.
DOWN = j
WHAT CHARACTER IS USED FOR MOVING CURSOR LEFT.
LEFT = h
WHAT CHARACTER IS USED FOR MOVING CURSOR RIGHT.
RIGHT = l
BUTTONS (L)EFT (M)IDDLE (R)IGHT CAN BE DESIGNATED TO SEND
UP TO THREE CHARACTERS FOR EACH BUTTON WHEN PRESSED.
WHAT CHARACTERS ARE TO BE USED FOR THE LEFT BUTTON
LEFT BUTTON =bcde
NO MORE THAN THREE CHARACTERS PER BUTTON.
WHAT CHARACTERS ARE TO BE USED FOR THE LEFT BUTTON
LEFT BUTTON =c
WHAT CHARACTERS ARE TO BE USED FOR THE MIDDLE BUTTON
MIDDLE BUTTON =
WHAT CHARACTERS ARE TO BE USED FOR THE RIGHT BUTTON
RIGHT BUTTON =
```

APPENDIX A

To utilize the mouse with a program which has been previously initialized it is only necessary to position the cursor on the line of the desired program. You will be prompted to provide any additional files as required for the program you choose. After successful completion of the program you are returned to the menu screen The following is a sample screen of a typical selection.

APPENDIX A

```
THE FOLLOWING PROGRAMS MAY BE USED WITH THE MOUSE.
atest
vi
initialize
exit
MAKE SELECTION BY MOVING CURSOR TO DESIRED PROGRAM.
THEN PRESS LEFT BUTTON ON MOUSE.
TO EXIT MOVE CURSOR TO EXIT THEN PRESS LEFT BUTTON ON MOUSE
TO INITIALIZE A PROGRAM FOR USE WITH THE MOUSE.
MOVE CURSOR TO INITIALIZE THEN PRESS LEFT BUTTON ON MOUSE.







PROVIDE ADDITIONAL FILES AS REQUIRED.atest
```

The selected program runs and any screen operations set up on the mouse function unless the mouse is deactivated by a CONTROL-T. After completion of the program the menu screen is shown.

```
THE FOLLOWING PROGRAMS MAY BE USED WITH THE MOUSE.
atest
vi
initialize
exit
MAKE SELECTION BY MOVING CURSOR TO DESIRED PROGRAM.
THEN PRESS LEFT BUTTON ON MOUSE.
TO EXIT MOVE CURSOR TO EXIT THEN PRESS LEFT BUTTON ON MOUSE
TO INITIALIZE A PROGRAM FOR USE WITH THE MOUSE.
MOVE CURSOR TO INITIALIZE THEN PRESS LEFT BUTTON ON MOUSE.






program complete
```

APPENDIX B

PROGRAM DISCRIPTION

Findmouse

This shell program looks for files with a .mse extension in the current directory strips off the extension and creates a file called mselist. This program uses grep, sort, ls, and the ed editor.

Input- current directory file

Output- mselist file

Program Mouse

Sets up the HAWLEY mouse to be used with an RS-232 communication line in series with a mouse controller from RANDOM ACCESS INCORPORATED. This is the main controll program under which all other routines are run.

APPENDIX B

Plist

This routine calls findmouse after completion the file mselist is printed on the CRT.

Input- file mselist

Output- none

Error Message

Error messages are provided for setting up cursor motion control characters, and the three buttons on the mouse.

Input- none

Output- none

Set_Cont

This program provides the questions for seting up the mouse controller. It also creates the control file giving it the name of the program being initialized and adding the extension of .mse.

Input- control characters

Output- (program name).mse

APPENDIX B

Find

This routine activates the mouse by sending the control characters for cursor movement to the mouse controller. Routine plist is called and instructions on the selection of a program are printed. The mouse is then activated in accordance with the specified program or a program is initialized. A return to the menu is provided before exit.

Input- none

Output- file setmse, file (program name).mse

Main

Calls routine find forever.

Input- none

Output- none

APPENDIX C

SOURCE CODE

```
# findmouse
# This program looks for files with a .mse extension
# in the current directory strips off the extension
# and creates a file called mselist.
#       Input- file current directory
#       Output- file mselist
ls| grep '.mse' | sort > mselist
ed  - mselist <<!
        H
        g/.mse/s///
        w
        q
!
        echo "initialize" >> mselist
        echo "exit" >> mselist
```

```
/* Program mouse - Sets up the HAWLEY mouse */
/* to be used with an RS-232 communication line */
/* in series with a mouse controller from */
/* RANDOM ACCESS INCORPORATED. */
#include <stdio.h>
#define TRUE 1
#define homescrn "H"
#define clrscrn "2J"
#define setmse "........3\n\n\n\n.......La\n"
#define findmouse "findmouse"
#define cat "cat "
#define mse ".mse"
char name[15],names[15][15],up[3],down[3],left[3],right[3];
char leftbut[4],rightbut[4],middlebut[4];
char stre[256], other[256-15];
char temp[15];
FILE *fp,*fp2;
int  i,c,count;
int  eoln = '\n';
int lncount = Ø;
plist()
/* This routine calls findmouse after completion the file */
/* mselist is printed on the CRT. */
/*      Input- file mselist     */
/*      Output- none            */
{
system(findmouse);
if ((fp = fopen("mselist", "r")) == NULL)
{
printf("can not open file");
exit(Ø);
}
printf("%s%s", clrscrn, homescrn );
printf("THE FOLLOWING PROGRAMS ");
printf("MAY BE USED WITH THE MOUSE.\n");
for (count = Ø;fscanf(fp,"%s",names[count]) != EOF;count++)
printf("%s\n",names[count]);
fclose(fp);
lncount = Ø;
}
error()
/* Error message for setting up cursor motion control */
/* characters.*/
/*      Input- none     */
/*      Output- none    */
{
printf("EACH DIRECTION OF CURSOR ");
printf("MOVEMENT MUST BE PROVIDED.\n");
printf("USING ONE CHARACTER PER DIRECTION.\n");
}
error1()
/* Error message for setting up control characters */
/* for the three buttons on the mouse. */
/*      Input- none      */
/*      Output- none     */
{
printf("NO MORE THAN THREE CHARACTERS PER BUTTON.\n");
}
set_cont()
/* This program provides the questions for setting up the */
/* mouse controller. It also creates the control file*/
/* giving it the name of the program being initialized*/
/* and adding the extension of .mse.*/
/*      Input- control characters         */
/*      Output- (program name).mse        */

{
printf("PROVIDE NAME OF PROGRAM TO BE USED WITH A MOUSE.\n");
printf("NAME = ");
while ((i = strlen(gets(name)))< 1 || i > 15 ){
printf("USE FROM 1 TO 15 CHARACTERS FOR PROGRAM NAME.\n");
printf("PROVIDE NAME OF PROGRAM TO BE USED WITH A MOUSE.\n");
printf("NAME = ");
}
printf("WHAT CHARACTER IS USED FOR MOVING CURSOR UP.\n");
printf("UP = ");
while (i = strlen(gets(up)) != 1) {error();
printf("WHAT CHARACTER IS USED FOR MOVING CURSOR UP.\n");
printf("UP = ");
}
printf("WHAT CHARACTER IS USED FOR MOVING CURSOR DOWN.\n");
printf("DOWN = ");
while (i = strlen(gets(down)) != 1) {error();
printf("WHAT CHARACTER IS USED FOR MOVING CURSOR DOWN.\n");
printf("DOWN = ");
}
printf("WHAT CHARACTER IS USED FOR MOVING CURSOR LEFT.\n");
printf("LEFT = ");
while (i = strlen(gets(left)) != 1) {error();
printf("WHAT CHARACTER IS USED FOR MOVING CURSOR LEFT.\n");
printf("LEFT = ");
}
printf("WHAT CHARACTER IS USED FOR MOVING CURSOR RIGHT.\n");
printf("RIGHT = ");
while (i = strlen(gets(right)) != 1) {error();
printf("WHAT CHARACTER IS USED FOR MOVING CURSOR RIGHT.\n");
printf("RIGHT = ");
}
printf("BUTTONS (L)EFT (M)IDDLE (R)IGHT CAN BE ");
printf("DESIGNATED TO SEND \nUP TO THREE ");
printf("CHARACTERS FOR EACH BUTTON WHEN PRESSED.\n");
printf("WHAT CHARACTERS ARE TO BE ");
printf("USED FOR THE LEFT BUTTON\n");
printf("LEFT BUTTON = ");
```

```
while (i = strlen(gets(leftbut)) > 3) {error1();
printf("WHAT CHARACTERS ARE TO BE ");
printf("USED FOR THE LEFT BUTTON\n");
printf("LEFT BUTTON = ");
}
printf("WHAT CHARACTERS ARE TO BE ");
printf("USED FOR THE MIDDLE BUTTON\n");
printf("MIDDLE BUTTON = ");
while (i = strlen(gets(middlebut)) > 3) {error1();
printf("WHAT CHARACTERS ARE TO BE ");
printf("USED FOR THE MIDDLE BUTTON\n");
printf("MIDDLE BUTTON = ");
}
printf("WHAT CHARACTERS ARE TO BE ");
printf("USED FOR THE RIGHT BUTTON\n");
printf("RIGHT BUTTON = ");
while (i = strlen(gets(rightbut)) > 3) {error1();
printf("WHAT CHARACTERS ARE TO BE ");
printf("USED FOR THE RIGHT BUTTON\n");
printf("RIGHT BUTTON = ");
}
strcpy(names,name);
strcat(names,mse);
if (( fp2 = fopen( names, "w")) != NULL)
fprintf( fp2, ".3%s %s %s %s ", up, down, right, left);
if (*leftbut != NULL)
        fprintf( fp2, ".L %s ", leftbut);
if (*middlebut != NULL)
        fprintf( fp2, ".M %s ", middlebut);
if (*rightbut != NULL)
        fprintf( fp2, ".R %s ", rightbut);

        fprintf( fp2, "\n" );
        fclose(fp2);
}
find()
/* This routine activates the mouse by sending the control */
/* characters for cursor movement to the mouse controller. */
/* Routine plist is called and instructions on the */
/* selection of a program are printed. */
/* The mouse is then activated in accordance with the */
/* specified program or a program is initialized. A */
/* return to the menu is provided before exit. */
/*      Input- none */
/*      Output- file setmse, file (program name).mse    */
{
printf("%s", setmse);
plist();
printf("MAKE SELECTION BY MOVING CURSOR ");
printf("TO DESIRED PROGRAM.\n");
printf("THEN PRESS LEFT BUTTON ON MOUSE.\n");
printf("TO EXIT MOVE CURSOR TO EXIT THEN PRESS ");
printf("LEFT BUTTON ON MOUSE\n");
printf("TO INITIALIZE A PROGRAM FOR USE WITH THE MOUSE.\n");
printf("MOVE CURSOR TO INITIALIZE ");
printf("THEN PRESS LEFT BUTTON ON MOUSE.\n");
printf("%s\n", homescrn);
while ((c = getchar()) == eoln)
}
++lncount;
if (lncount > (count - 1))
{
printf("%s\n", homescrn);
lncount = Ø;
}
}
if (strcmp(names[lncount], "exit") == Ø )
{
printf("\n\n\n\n\n\n\nprogram complete \n");
        exit(Ø);
}
else if (strcmp(names[lncount], "initialize") == Ø)
{
c = getchar();
printf("%s%s", clrscrn, homescrn);
set_cont();
}
else
{
c = getchar();
for (i = Ø; (( count + 5) - lncount) > i; i++)
printf("\n");
printf("PROVIDE ADDITIONAL FILES AS REQUIRED.");
other[Ø] = ' ';
i = 1;
while (( c = getchar())!= eoln)
        other[i++] = c;
strcpy(temp, cat);
strcat(temp, names[lncount]);
strcat(temp, mse);
strcpy(stre, names[lncount]);
strcat(stre, other);
system(temp);
system(stre);
}
}
main()
/* Calls routine find forever. */
/*      Input- none         */
/*      Output- none        */
{
while (TRUE)
{
find();
```

```
}
}
```

IMPLEMENTING THE HAWLEY MOUSE MODEL XØ63X AND RANDOM ACCESS INCORPORATED MU-2 SERIAL INTERFACE

by

Roy William Richard

B.R.E. Chicago Technical College, 1952

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1985

ABSTRACT

The implementation of a program designed to allow the use of a mouse with a remote terminal. The mouse is to be used with existing screen oriented application programs.

A history of the mouse is briefly discussed with comparisons made against the lightpen, data tablet, keyboard, and voice data entry. A discussion is provided of the various hardware interfaces and software control.

The particular configuration chosen is shown to be the most viable solution in meeting the need to provide a user at a remote terminal with a mouse controller. Application programs in existence and new programs written to use the full capabilities of the mouse controller are able to be used in this configuration.