THE SOFTWARE

The driver included with this documentation is the result of about 30 to 40 hours of work. It has several components: Status byte handler and synchronizer, X coordinate adjuster, Y coordinate adjuster, and two supervisor calls.

The interrupt driven routine will keep track of the mouse within a 1024 by 1024 area. The programmer needs to do nothing to update the current cursor position of the mouse since this is done with interrupts.

Two SVC's are provided to read information from the mouse. They are as follows:

SVC 120 - Read mouse position and status

On entry:
A = 120
On exit:
HL = X coordinate
DE = Y coordinate
B = Button status when last pushed (bits 4 & 5)
C = Current button status (bits 4 & 5)

* Bit 4 is for the right button, bit 5 is for the left.

When a button is pushed, the current button status is saved and can be recalled in register B. This status byte is NOT reset by the interrupt routine, but must be reset by the programmer (using SVC 121). The status returned in register C is the last status byte that was read.

You can find out if a button has been pushed, and if it is still being held down by compairing B and C. If bit 4 in B is set, then the right button has been pushed. If bit 4 is set in C, then the right button is still being held down, otherwise it has been released.

SVC 121 - Change Coordinates and button status

On entry:
A = 121
HL = X coordinate
DE = Y coordinate
B = What you want the button status to be.

In most cases, you will probably want to set B to 0 to clear the status of the buttons.

You now have a way to write programs using a mouse. The routine is very accurate, but I have noticed that the mouse can sometimes make unusual jumps.

THE DRIVER

The routine to monitor the mouse is located in high memory from FF00 to FFFF (hex). No other drivers or routines should be loaded in high memory when you load the mouse driver! It is NOT a relocatable driver. You should NOT load any drivers dealing with the RS-232 port with this driver installed. This includes COM/DVR and terminal programs such as XT4. If you want to use any of these, please re-boot first.

I have also noticed that several programs will not work once this driver has been loaded. AllWrite is one of these, it must do something will the RS-232 interrupts.

Another note: Do not load the driver after exiting XT4! I believe that XT4 forgets to reset some interrupt addresses when it finishes executing, and your computer will crash easily!

If you have any questions or comments, please E-mail them to me.

Scott McBurney
Genie ID: S.MCBURNEY

MOUSE/CMD

INTRODUCTION

Welcome to the wonderful world of mice! I congratulate you on your choice of mouse software (mine) and offer you these instructions and suggestions to help you in writing software for the mouse.

THE MOUSE

The driver software is written specifically for the Tandy Serial Mouse. (catalog #25-1040, $49.95) You will also need to purchase their 9 pin to 25 pin adapter to plug the mouse in (or make your own). I think the price of it is $7.95. The software *should* work with any serial mouse that is Microsoft Serial Mouse compatible! Other mice may use different data packet formats, and therefore will not work with this driver.

The serial mouse communicates to its host in a very simple way. Whenever something occurs (movement, button pushed, or button released) it sends a 3 byte packet to the host. The first byte is a status byte. The second byte is the amount of movement in the X direction, the third is movement in the Y direction. I won't go into the specifics of each bit in each byte here, because it would take up too much room.