

Apple II Technical Notes



Developer Technical Support

Mouse

#1: Interrupt Environment with the Mouse

Revised by:	Matt	Deatherage
November 1988		
Revised by:	Rilla	Reynolds
November 1985		

This Technical Note describes the interrupt environment one should take into account when programming mouse-based applications on the Apple II family of computers.

Software developers who are writing mouse-based programs in assembly language need to be concerned about the computer's interrupt environment, even if they are using the mouse in passive mode. Listed below are several conditions which assembly language programmers should take into account if their programs are to run on the Apple II family of computers.

- Do not disable interrupts unless absolutely necessary. If you disable them, be sure to re-enable them.
- Disable interrupts when calling any mouse routine. Always use `PHP` and `SEI` to disable interrupts, then use `PLP` to re-enable them. This method preserves the state of interrupts (enabled or disabled).
- Do not re-enable interrupts (`PLP`) after a call to `ReadMouse` until X and Y data have been removed from the screen holes.
- Disable interrupts (`PHP` and `SEI`) before placing position

information in the screen holes (`PosMouse` or `ClampMouse`).

- Enter all mouse routines (except `ServeMouse`) with the X register set to `$Cn` and Y register set to `$n0`, where `n` = the slot number.
- Some programs need to disable interrupts for purposes other than reading the mouse. If interrupts are disabled then re-enabled, the first call to `ReadMouse` could return incorrect values; subsequent calls to `ReadMouse` will return correct values until interrupts are disabled and re-enabled again. Disabling interrupts for mouse calls does not create this problem. If you watch numbers from the mouse while moving it in a direction which would increase values, you would see something similar to: 6, 7, 8, 9, 8, 9, 10. In practice, this momentary “glitch” in the stream of data has little importance. If you feel you must avoid this glitch altogether, do not disable interrupts for more than 40 microseconds or make sure that at least one mouse interrupt takes place after re-enabling interrupts.