
Subject: Re: Get Mouse Position using low level assembly language

Posted by Oz on Thu, 09 Oct 2014 11:42:52 GMT

[View Forum Message](#) <> [Reply to Message](#)

Dagen,

Quote:Obviously, they changed the way the mouse firmware was implemented on the ROM 3.

They have added the use of Numeric Pad keys to simulate the mouse move. I think it was for simplifying the usage of the Mouse by disable people.

So they had to change few things in the Mouse Firmware but the very low level (ie softswitchs) was not impacted.

Quote:Does this code you've posted work on all Apple IIgs machines? I assume the answer is yes.

This is the kind of code we were using in our games in the 90's, so I would say YES for the well known Apple IIgs models (Rom 1, Rom 3 & Mark Twain).

Quote:Does this code rely on interrupts? I.e.- If we disable interrupts, will the mouse data still update?

We don't use interrupts here. We just read softswitch values. The fact to disable interrupts at the 65c816 level (SEI / CLI) don't change anything about the updates of the softswitchs by the hardware. You can continue to read the Keyboard (\$C010), the Mouse (\$C024), the Joystick (\$C064/\$C065 + \$C025), Get VBL information (\$C02E)...

GS/OS uses the VBL interrupt to update the Mouse pointer on the screen when you are using the Apple Graphic Interface (this explain why, even if your program has crashed, you can still move the mouse on the screen), but here, because we want to handle the Mouse position in our way, we don't care about interrupts.

Quote:How often does the data in the registers change?

I think nothing change until you decide to move your mouse (or click the button) ! So this is why we exit immediately if we see the registers are not ready (you could wait forever if the GS has no mouse connected).

Once the move has been done, I think it needs few cycles for the ADB controller to update the softswitch with the value. The ADB has a 16 bits value for the Mouse move and has to copy it into a 8 Bit register (\$C024). This is why you have to read it twice (one for the DeltaX, one for the DeltaY). I have never really computed the number of cycles required to get the data ready because we were more in a 'not ready = no time to loose here = exit now' strategy.

Olivier
