FROM: Ma●McClung, 76336,3110
TO: Dave Brown, 72103,2235
DATE: 11/10/94 10:29 PM

Re: WOSA/XMC, Draft 2, ITimer

Hello Dave,

NDA - WIN6400.DLL ------------- I received the signed NDA for the WIN6400.DLL source as well as your addendum to the NDA. Bob has a copy of the NDA and will work out any additional matters with you.

Meanwhile, I would like to plow through sections 8 and 9 of the WOSA/XMC specification. This correspondence will concentrate on the ITimer interface.

ITimer Interface -------------- Every 6000 product has 1 timer. On stepper products the timer resolution is fixed at 2ms. On servo products the timer resolution is dependent on the system update period (see SSFR command). In either case a Set- Resolution() method would not be appropriate.

To time events, the 6000 timer can be started (TIMST1), stopped (TIMSTP) and reset (TIMST0). The current timer value can be obtained from all 6000 products via a timer variable ([TIM]). On bus-based products the current timer value can also be obtained more efficiently via the "fast status" registers. I think there should be a StartTimer(), StopTimer(), ResetTimer() and GetElapsedTime() methods added to the ITimer interface.
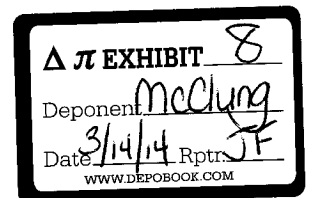
To implement time delays, every 6000 product has the time delay command (T). I wasn't sure if the GetTimer() and SetTimer() methods applied to this. If so, maybe they should be renamed SetTimeDelay() and GetTimeDelay().

Bus-based 6000 products have the ability to interrupt the PC when a timer setpoint has been reached (see INTHW and TIMINT commands). There should be a method called SetTimerInterrupt(). It's still not clear to me what AttachInterrupt() and DetachInterrupt() do?

Also, I don't understand what GetErrorStatus() and Enable() do?

Conclusion --------- Dave, I would like to proceed with evaluating the remaining interfaces in this manner. I realize that you are now working with the 6000 hardware/software and will discover most of this for yourself. Let me know if you find this kind of information helpful.