

## PCNC Interface Board를 위한 가상 장치 드라이버의 개발

조인호\*, 편영식\*\*

The Development of Virtual Device Driver for PCNC Interface Board

In Ho Cho\*, Young Shik Pyoun\*\*

\* 선문대 대학원, \*\* 선문대 기계 및 제어공학부

### 1. 서론

수치제어장치(NC)는 컴퓨터기술의 발달과 더불어 발전하여 왔고, 과거 공구 경로를 제어하기 위하여 또한 그 외의 제어 장치들을 동시에 제어하기 위하여 여러개의 CPU가 필요하였지만 현재는 하나의 CPU를 가지고 모든 제어작업을 할 수 있게 되었다. [1][2]

두 개 이상의 CPU를 가진 PC-NC 시스템에서는 사용자 Interface와 같은 실시간 처리가 필요하지 않은 부분과 필요한 부분을 구분하여 서로 다른 CPU와 다른 운영체제를 채택하였다. 그러나 one CPU PC-NC 장치를 구현할 때는 사용자 Interface부분과 NC 제어 부분이 같은 운영체제기반위에 구현되어야 한다. 현재 NC 장치 기술 경향이 사용자 중심의 system을 지향하고 있어서 사용자 Interface부분은 Windows 95나 Windows NT 운영체제를 기반으로 하여 구현되고 있다. 따라서 Windows 운영체제를 채택하는 one CPU PC-NC 장치의 구현에서는 운영체제가 Real Time Processing을 지원하지 못하기 때문에 NC 제어부분에 Real Time Processing를 보장해 주는 것이 중요한 문제로 대두되었다.[5]

현재 대부분의 NC 업체에서는 보다 사용이 용이한 사용자 Interface를 지원하는 동시에 실시간 처리를 지원하기 위하여 GUI(Graphic User Interface)환경과 Multitasking을 지원하는 Real Time Operating System을 이용하고 있다. 그러나 NC에 이러한 상용 Operating System이

탑재되면서 현재 많은 Windows 운영체제 기반의 유틸리티들과의 호환성과 유지 보수의 용이성이 저하되고 있다. 따라서, Windows 기반의 운영체제위에서 Real time Processing을 지원하는 방법이 연구되기 시작하였다.

Windows 95또는 Windows NT 운영체제가 Real Time Processing을 지원하지 못하는 문제점을 해결하기 위한 방법에는 그 정도는 다르지만 두가지로 나눌 수 있다. 하나는 Real Time Extension을 이용하는 방법이다. 현재 Windows NT기반의 Real Time Extension은 상품화되어져 이미 몇몇 업체에서는 이를 응용하여 One CPU PC-NC를 생산하고 있다. 그러나 이 방법은 상용화된 Extension을 이용하기 때문에 이로 인한 생산가의 상승을 피할 수 없다.

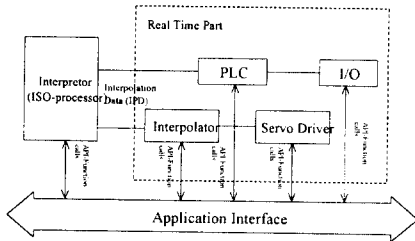
다른 하나의 방법은 Windows 운영체제를 위한 근본적인 드라이버를 구현하는 것이다. 이러한 드라이버는 간접적인 방법을 통하여 I/O port나 Interrupt를 다룰 수 있다. 이 방법은 운영체제가 가지는 제한들을 완전하게 극복할 수는 없지만 상용 Extension을 이용하지 않기 때문에 생산가를 낮출 수 있는 장점을 가지고 있다.

아직 Windows 운영체제 기반에서 시스템 드라이버를 이용하여 PC-NC system를 구축한 사례는 없다. 따라서 본 연구에서는 가상 장치 드라이버를 이용하여 NC Interface Board을 제어함으로써 NC 제어부의 시간 응답성 및 제어속도의 향상을 목적으로 한다.

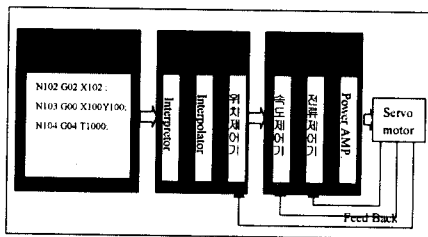
## 2. PC-NC Interface Board용 가상 장치 드라이버

### 2.1 PC-NC에서 Real Time Processing지원부

PC-NC에서 Real Time Processing이 지원되어야 하는 부분은 <그림 1>과 같이 Interpolator, PLC, Servo 제어부이다. Servo Motor 제어 경로는 <그림 2>와 같이 Interpreter에 의해 경로 데이터가 생성되고, Interpolator가 상세한 공구데이터를 다시 생성하여, Servo 제어부로 전달된다.



<그림 1> PC-NC의 Real Time Part

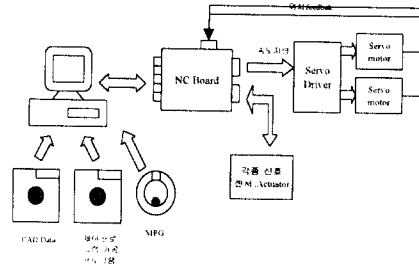


<그림 2> 공구 경로 제어 Block Diagram

### 2.2 PC-NC Interface Board의 역할

<그림 3>은 PC-NC 구성도이다. 이 그림과 같이 PC-NC Interface Board의 역할은 공구경로의 제어 명령을 Servo Driver에 전달하고 Servo Motor로부터 들어오는 feedback정보를 NC 제어부로 전달한다. 또

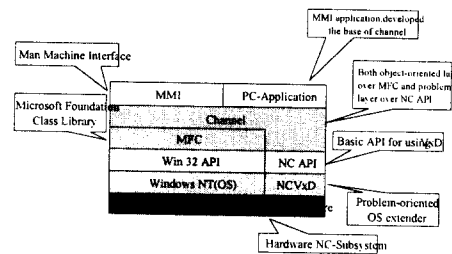
한 PLC제어를 위하여 일정주기마다, 각종 Sensor로부터의 상태신호를 NC system에 전달하고 NC system의 논리연산 결과인 제어출력신호를 각 제어기에 전달한다.



<그림 3> PC-NC 시스템 구성도

### 2.3 가상 장치 드라이버의 역할

PC-NC system의 프로그램의 계층 구조는 <그림 4>와 같다. 가상장치 드라이버 (VxD)는 NC Hardware와 NC API의 중간에 위치하여 API로부터의 제어명령을 NC Hardware에 전달하고 NC Hardware로부터 응용프로그램에 전달되는 정보를 API에 전달한다. VxD는 결국 <그림 3>의 PC-NC Interface Board를 제어하게 된다.



<그림 4> PC-NC의 계층구조

VxD의 집합체는 Windows 운영체제에서 가장 핵심부분이며, privilege level의 최상위인 ring 0 프로그램이다. 따라서 VxD는 운영체제안에 포함되어 VMM(Virtual Machine Manager)에 의하여 관리 제어됨으로서 다른 응용프로그램보다 실행 처리의 우선권을 가지게 되며 Interrupt시 수행속도도

빠르게 된다.[3]

## 2.4 PC-NC Interface Board용 가상장치 드라이버의 기능

가상 장치 드라이버(VxD)는 PC-NC Interface Board의 제어를 위하여 다음 두가지의 기능을 가져야 한다.

### 2.4.1. Interrupt Handle

PC-NC Interface Board는 일정한 시간주 기마다 Servo Motor의 위치를 응용프로그램에 전달하고 또 속도 명령을 Servo Driver에 전달해야 한다. 또한 PLC제어를 위하여 Sensor로 부터오는 신호를 받아서 연산후 제어기로 제어출력하여야 한다. 따라서 VxD는 이러한 동작을 Interrupt를 통하여 일정한 시간주기로 반복해야하므로 Interrupt를 다루는 기능을 가져야 한다.

VxD는 VPICD\_Virtualize\_IRQ message를 발생시킴으로서 IRQ의 default handling을 다시 작성할 수 있다. 특별히 IRQ를 가상화시키는 것은 ring 0 privilege에서 하드웨어 장치에 대한 완벽한 handling을 얻기위해서이다.

VPICD는 Windows 95에서 모든 하드웨어 interrupt를 다룬다. 장치의 interrupt에 대하여 수행되는 interrupt vector를 직접적으로 다루는 것 대신에 VxD가 VPICD call를 통하여 IRQ를 가상화한다.[3]

### 2.4.2 I/O control

VxD는 PC-NC Interface Board의 I/O address에 있는 정보를 읽거나 I/O address에 정보를 쓰기위하여 I/O port를 제어하여야 한다. PC-NC Interface Board에서는 I/O address대하여 그 제어 동작이 일정하다. 따라서 VxD에서는 I/O port마다 일정한 callback procedure를 만들어 I/O address로

의 고유한 제어동작을 수행하도록 한다. 이러한 것을 I/O trapping이라고 한다.[3]

## 2.5 가상 장치 드라이버 구현

PC-NC Interface Board을 제어하는 가상장치 드라이버가 가져야 하는 기능을 크게 다음 두 부분으로 나누어 구현한다. VxD 동작 제어 알고리즘은 <그림 5>와 같다.

### 2.5.1 Interrupt Handler

PC-NC Interface Board로부터 오는 Interrupt Request(IRQ) 신호를 받기위하여 VPICD\_Virtualize\_IRQ Message를 이용하여 다음과 같이 프로그램을 작성한다.

```

;; -----
;; Initialize & Virtualize IRQ
;; -----
BeginProc _Virt_IRQ
    push edi
    mov edi, OFFSET32 _VIRQdat
    VxDcall VPICD_Virtualize_IRQ
    jnc VIRQEXIT
    mov eax, 0fffffffh ;set ERR if appro
VIRQEXIT:
    pop edi
    ret
EndProc _Virt_IRQ

;;-----
;; VPICD IRQ Descriptor Block
;;-----VxD
_LOCKED_DATA_SEG
_VIRQdat dw PNC_IRQ_Used ;; IRQ#
          dw 0
          dd OFFSET32_ISR_PNC3
                               ;; ISR ADDRESS
          dd 0
          dd 0

```

```

dd 0
dd 0
dd 500
dd 0
dd 0
dd 0

```

VxD\_LOCKED\_DATA\_ENDS

### 2.5.2 I/O trapping

VxD에 전체 I/O port를 포함 시키지 않고 실험에 필요한 부분만을 작성하였다. 필요한 부분은 3개의 Encoder Pulse, 3개의 D/A converter 출력 이다.

```

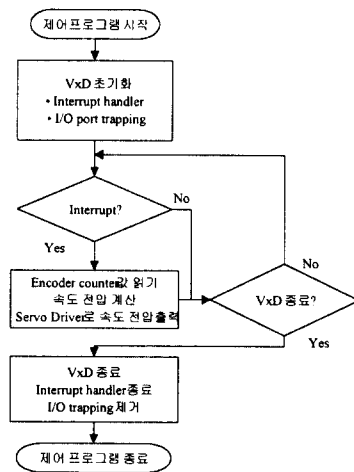
;;-----
;; I/O trap
;;-----
BeginProc _Set_IO
    move di, offset32 PNC_IO_Table
    VMCall Install_Mult_IO_Handlers
EndProc _Set_IO
.....

;;-----
;; VxD I/O Table
;;-----
Begin_VxD_IO_Table PNC_IO_Table
    220h, Encoder1
    224h, Encoder2
    240h, Encoder3
    280h, X
    282h, Y
    284h, Z
End_VxD_IO_Table PNC_IO_Table
.....

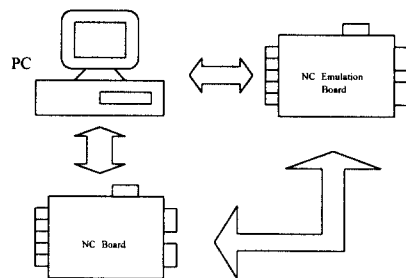
```

### 3.1 NC Emulator

NC Emulator는 Servo Driver로의 입출력 신호, 외부 I/O의 입출력 신호, 각 제어축의 Encoder Pulse값을 발생시켜 주는 장비로 통일중공업(주)에서 자사 제품의 Interface Card를 점검하고 응용 프로그램을 시험해 보는 장비이다. 이 Emulator의 구성은 <그림 6>과 같다. Emulator는 자체 가지고 있는 제어 Register들을 통해서 Interrupt 주기, Encoder 회전당 Pulse수, 정회전/역회전, 회전속도등을 지정할 수 있다.



<그림 5> VxD 동작 알고리즘



<그림 6> NC Emulator 구성도

### 3. 가상 장치 드라이버 성능평가

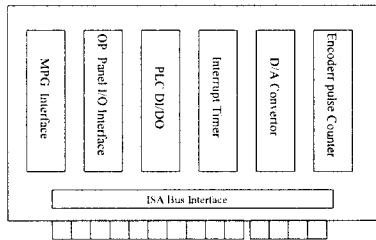
가상 장치 드라이버의 성능 평가를 위하여 다음 장비들을 이용하였다.

### 3.2 PC-NC Interface Board

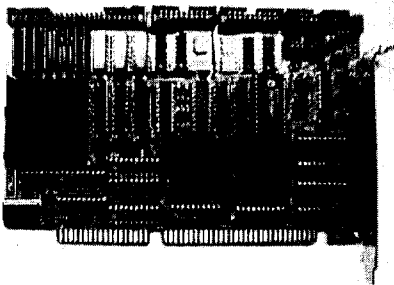
PC-NC Interface Board는 상용화된 제품으로서 <그림 7>와 같은 구성이다. NC Board를 Servo Driver로의 전압 출력 Port,

Encoder 입력 Port, I/O Port, 각종 상태 및 제어 Register들을 가지고 있다. 이 Board는 동시 3축을 제어할 수 있고, I/O는 각 40점씩을 가지고 있다. 내부에 16 bit D/A Converter가 있어 속도값을 속도 전압으로 바꾸어 Servo Driver로 출력한다.

NC Board에서는 자체 Timer를 통하여 원하는 시간 주기마다 Interrupt를 발생시킨다. Encoder pulse수는 계속 Encoder 16 bit counter가 세고 있으며 Interrupt가 발생되면 Encoder counter의 값을 Latch한다. 따라서 반드시 일정한 시간 주기마다 한번씩 Servo Motor의 회전량을 측정함으로써, Servo 제어를 할 수 있다. <그림 8>은 평가에 이용한 통일중공업(주)의 PC-NC Interface Board의 실제 모습이다.



<그림 7> PC-NC Interface Board 구성도



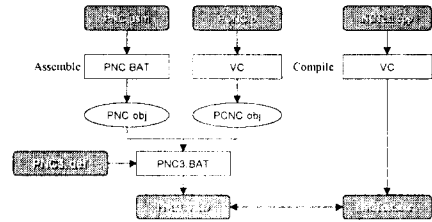
<그림 8> PC-NC Interface Board의 실제 모습

#### 4.3 가상 장치 드라이버 만들기

NC Board를 제어하기 위한 Software에는 장치 드라이버를 구현하고, 구현된 장치 드

라이버를 응용하는 프로그램이 필요하다. 장치 드라이버의 구현은 Windows 95 DDK(Device Driver Development Kit)과 Microsoft Assembler 6.1을 이용하였고, 응용 프로그램은 Visual C++ 5.0을 이용하여 작성하였다.

가상 장치 드라이버의 프로그램 구성은 <그림 9>과 같다.



<그림 9> NC Board용 source program 구성도

#### 4.4 Test program 만들기

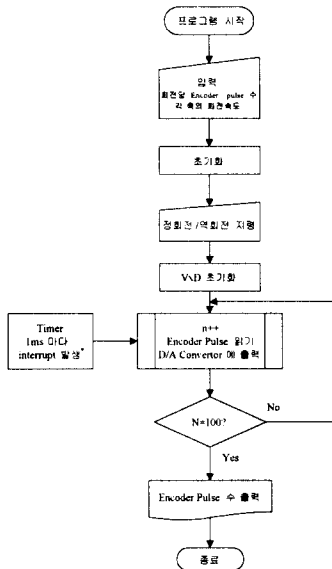
VxD의 성능 평가를 위한 Test program의 Flow chart <그림 4-4>와 같다. 먼저 프로그램은 NC Emulator에 필요한 parameter를 지정한다. Encoder 회전당 pulse수를 입력하고, 각 축당 Servo Motor 회전속도를 지정한다. 다음으로 입력한 parameter를 가지고 Emulator를 초기화하면 Emulator에서는 지정된 속도에 해당하는 Encoder pulse를 출력한다. 정회전/역회전을 선택하면 프로그램에서 가상적으로 정회전/역회전을 Encoder A상과 B상의 인식순서를 변경함으로써 구현한다. PC-NC Interface Board에서는 2ms마다 Encoder counter의 pulse수를 Latch하고 PC에 hardware interrupt request신호를 발생시켜준다. 모든 Interrupt의 처리 및 Board에서 PC로 또는 PC에서 Board로 전달되는 데이터는 장치 드라이버에 의하여 처리된다.

이 실험에서는 Interrupt 후의 I/O port를 통하여 정보가 교환되기 까지의 시간을 측정하였다.

#### 4.5 성능 평가 결과

구현된 가상 장치 드라이버 성능평가 방법에 따라 제어 수행 속도를 측정하였다. 측정된 데이터의 비교를 위하여 가상 장치 드라이버(VxD)를 이용하여 PC-NC Interface Board를 제어한 경우와 응용프로그램에서 직접 제어한 경우 두가지를 실험 비교하였다.

응용프로그램에서 직접 제어한 경우 얻은 수행 시간은  $100\mu s \sim 5\text{ ms}$ 였다. 가상 장치 드라이버(VxD)를 이용하여 제어한 경우 얻은 수행시간은  $50\mu s \sim 500\mu s$ 였다.



<그림 10> Test program flow chart

#### 5. 결론

본 연구에서는 Windows 운영체제 기반의 PC-NC 구현을 위하여 Real time processing 지원 방법 중의 하나인 Windows 95 운영체제를 위한 드라이버를 제작하여 NC 제어부분의 응답시간 및 처리속도를 줄이려고 하였다.

장치 드라이버를 이용한 실험과 이용하지

않은 실험을 수행한 결과 장치 드라이버를 사용하는 경우가 interrupt 발생후 지정된 작업을 마칠 때까지의 최대 제어 수행 시간이 5 ms에서  $500\mu s$ 로 개선되었다. 이 결과는 Windows 운영체제에 장치 제어를 위한 real time processing을 지원하기 위한 한 가지의 가능성을 제시하고 있다.

가상 장치 드라이버를 이용하여 Windows 운영체제의 취약한 실시간 처리를 개선하면 산업현장에서도 범용화된 운영체제를 사용할 수 있다. 특히 하나의 CPU를 가진 PC-NC에서 Windows 95와 같은 범용화된 운영프로그램을 이용하는 것이 생산단가를 줄이고 또한 시스템의 빠른 현장 적응력과 작업자의 빠른 숙련이 가능한 장점을 가지게 될 것이다.[4]

#### 참고 문헌

- [1] 무토 카즈오, "NC의 Open화(1)", 일본 M&E 11월호, pp102-113, 1996
- [2] 青木一信, "퍼스컴 .NC", 일본 M&E 11월호, pp114-119, 1996
- [3] Walter Oney, "System Programing for Windows 95", Microsoft Press, pp11-102, 377-424, 1996
- [4] 강화식, "최신 PC\_BASED CNC 기술", 기계기술, V25, N8, pp18-64
- [5] 조인호, "PC-NC Interface Board를 위한 가상 장치 드라이버의 개발", 선문대학교 석사학위 논문, pp1-5, 1999