

uC/OS-II를 이용한 DC 모터의 실시간 분산 시스템 제어

이윤희*, 김태강, 이건영
 광운대학교 전기공학과

A Implementation of Distributed Microsystem for DC Servo using uC/OS-II

Youn Hee Lee, Tae Kang Kim and Keon Young Yi
 Dept. of Electrical Engineering, Kwangwoon University

Abstract - In this paper, we discuss about the real-time distributed system for the control of two motors using uCOS-II kernel. Real-time kernel needs to process the tasks for two motors in desired time to synchronize motion. We used both semaphore and message mail box for synchronization. In the previous study, we used step motors for the actuator of two axes robot, but could not draw a clear line because of motor resolution and synchronizing step pulse. To resolve the problem we rebuilt the transfer robot with DC motors and the dedicated position servor which had built in out lab. Moreover we developed the PC based graphic user interface for generating planar drawing image control. Experimental results also presented to show the proposed control system is useful.

모터 컨트롤러로 사용되는 87c196CA 보드, 반송로봇의 기구부로 구성된다.

이전 실험에서는 Slave에 해당하는 Am188ES 보드에서 스텝 모터 컨트롤과 동기화를 수행하고, Master PC에서는 uC/OS-II 기반에서 기구학 계산하였다. uC/OS-II는 도스 상에서 실행시킬 경우, 실시간 OS로 동작하기 위해 커널을 실행 할 때 시스템의 인터럽트 벡터를 수정하여 실시간을 보장받는다. 그러나 Windows 2000 이상의 OS의 경우 도스 자체를 에뮬레이션하고 있으므로 인터럽트 벡터를 수정한다 할지라도 실시간을 보장 할 수 없다. 따라서, 이번 실험에서는 Master 측에서는 uC/OS-II를 사용하지 않고 기구학을 계산하였으며 사용자 인터페이스를 위하여 Visual Basic으로 작성한 GUI 환경의 사용자 인터페이스를 제작하였다.

1. 서 론

실시간 커널은 전체 응용 작업을 각각의 태스크로 분류하여 정해진 시간 범위 동안 프로세서의 사용권을 갖도록 조절하는 멀티태스킹이 이루어진다. 전체 시스템을 태스크로 나누어 관리하게 되면 복잡한 실시간 응용 프로그램을 쉽게 다룰 수 있고, 다양한 서비스를 제공하여 CPU의 사용을 용이하게 한다. 반면에 적절한 태스크의 설계가 시스템의 전체 성능에 큰 영향을 미치게 된다^[1].

앞선 연구에서는 2축 반송로봇의 구동부로 스텝 모터를 사용하였으나 스텝 모터의 정밀도와 펄스의 비동기 문제로 인하여 매끄러운 직선을 구현하지 못하였다^[2]. 본 논문에서는 이러한 문제를 해결 하기 위하여 스텝 모터 대신 DC 서보 모터를 사용하였고 Am188ES 보드에서 DC 모터 제어까지 담당하기에는 무리가 있어 87c196CA를 사용한 DC 모터 컨트롤 보드를 추가로 사용하였고 대신 Am188ES 보드에서 다른 상용 컨트롤러 보드에서도 모터 동기화 문제를 해결할 수 있도록 PC에서 받은 이동 각도를 몇 개의 작은 각도로 분할하여 전송할 수 있는 기능을 추가 하였다.

uC/OS-II는 커널 크기가 작고 선점형 커널로서 비용이 매우 저가라는 특징을 가지고 있어 태스크간의 동기화 작업과 시간 개념을 연구할 수 있다^[3]. PC 측은 Visual Basic을 사용하여 사용자 인터페이스를 구성하였다. 이러한 시스템은 실시간 운영체제를 사용한 소규모 분산제어 시스템의 연구 및 반송 로봇의 모션 컨트롤 연구에 적용할 수 있다.

2. 본 론

2.1 시스템 구성.

시스템의 구성은 그림 1에서와 같이 Master로 사용되는 PC와 Slave로 사용되는 Am188ES 보드, 서보

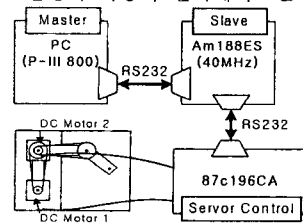


그림 1. 실시간 마이크로 분산 제어 시스템의 구성도

Fig. 1. Diagram of real-time micro distributed control system

각 보드는 RS-232 방식을 통하여 19200[bps]의 속도로 명령을 주고받는다. 기구부는 각 관절이 1:5의 기어 비를 가진다.

2.2 uC/OS-II와 태스크의 설계

Am188ES보드에 포팅하여 사용한 실시간 커널은 uC/OS-II이며 교육용으로 사용 제한이 없는 저가의 상용 실시간 커널이므로 실험에서와 같은 작은 분산시스템을 구축하는데 적합하다.

표 1. Am188ES의 uC/OS-II 응용 태스크

Table 1. uC/OS-II application task of Am188ES

Control Task	<ul style="list-style-type: none"> 이동 명령을 n개로 재분할하고 87C196CA 보드가 다음 명령을 전송 받을 준비가 되었는지 확인하여 명령을 전송한다. Monitor Task의 세마포어를 관리하여 동기화시킨다.
Receive Task	<ul style="list-style-type: none"> 시리얼 통신의 받는 부분을 관리. 명령을 해석하여 이동 명령일 경우, Control Task로 MailBox를 전송
Monitor Task	<ul style="list-style-type: none"> 87C196CA 보드로 다음 명령을 전송하기 위해 준비가 되었는지를 확인하기 위한 명령 'Q'를 500ms 마다 전송한다.

uC/OS-II 커널은 멀티타스킹, 세마포어 관리, 타스크 관리, 메모리 관리, 시간 관리, 메일박스, 큐 등의 서비스를 제공하여 CPU를 좀 더 편리하게 사용할 수 있게 해준다⁽¹⁾. uC/OS-II에서 다양한 서비스를 제공하여 CPU의 사용을 쉽게 해 주는 반면에 실시간 OS에서는 타스크의 효율적인 설계에 의해 시스템의 성능이 좌우된다⁽⁴⁾. 본 실험에서는 시리얼 통신을 관리하고, 명령의 분석 및 모터의 각도 분할 등의 작업을 위하여 표1과 같이 3개의 타스크를 설계하였다. 각 타스크의 구조는 그림 2와 같다.

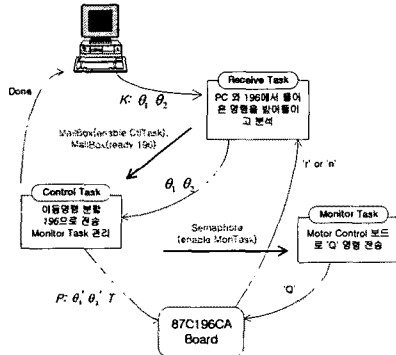


그림 2. uC/OS-II 응용 타스크의 구성도

Fig 2. Diagram of uC/OS-II application task

uC/OS-II에서는 동기화를 위해서 세마포어 및 메일박스, 큐 등의 서비스를 사용할 수 있고 본 실험에서는 1개의 세마포어와 2개의 메일 박스를 사용하였다. 추가로 시리얼 포트의 공유 사용을 위해 2개의 세마포어가 사용되었으며, 디버그를 위한 스텝 명령 전송 및 타스크의 브레이크 신호로 메일 박스가 1개 사용된다.

메일박스 및 세마포어의 사용 용도는 표2와 같다.

표 2. 응용 타스크의 메일 박스와 세마포어

Table 2. Mail-box and Semaphore of application task

MboxCtlTask	컨트롤 타스크를 동기시킴
MboxRdy196	컨트롤 타스크에 196보드
SemMonTask	Monitor Task의 동기화
SemComm1	시리얼 포트 1
SemComm2	시리얼 포트 2
MboxBreak	스텝 명령 구현 (디버그용)

2.3 PC 제어 소프트웨어

사용자 인터페이스는 그림3과 같이 구성하였다.

사용자가 원하는 좌표에 도형을 그리면 다음 장에 소개되는 알고리즘에 의하여 각 좌표를 관절의 이동 각도로 변환한다. 변환된 이동 각도는 RS-232 통신을 통하여 Am188ES 보드로 전달되며 모터가 주어진 이동 각도만큼 이동하고 나면 다음 이동 각도를 자동으로 보낸다.

PC 제어 소프트웨어는 4개의 Child 품으로 구성되며, 각 품은 다음과 같은 기능을 갖는다.

- **Draw window** : 기구부에서 그려야 할 도형을 그리는 윈도우이며 사용자가 마우스로 드래그 하여 사각형을 그릴 수 있다.

- **Status window** : 전송해야 할 버퍼의 내용 및 시리얼 통신으로 받은 데이터 및 연결 상태 등을 표시하는 창

- **Tool window** : Draw window에서 작업을 선택하여 선 그리기, 사각형 그리기, 지우기 등을 선택 할 수 있는 윈도우

- **Communication window** : 모터 on/off 명령 및 전송 시작과 전송 취소, 현재 설정 확인, Kinematics 변환 명령에 관련된 윈도우.

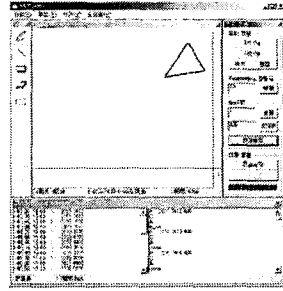


그림 3. PC 제어 소프트웨어

Fig 3. PC control software

2.4 반송 로봇 기구학

두 축의 길이가 일정하므로 P점의 좌표를 알면 Q점의 좌표를 알 수 있다. 식 (1)과 식 (2)는 Q점을 계산하기 위한 방정식이다.

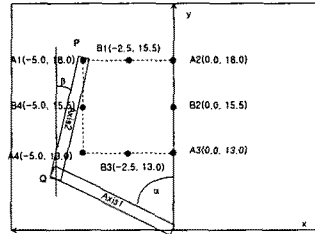


그림 4 사각형 경우 지점

Fig. 4. Passing point to draw a quadrangle

$$C^2 + D^2 = 144 \quad (1)$$

$$(A - C)^2 + (B - D)^2 = 100 \quad (2)$$

정리하면,

$$D = \frac{1}{2B}(A^2 + B^2 + 144 - 100) - \frac{AC}{B} = a + bC \quad (3)$$

$$(p^2 + 1)C^2 + 2pqC + q^2 = 0 \quad (4)$$

와 같이 되고 근의 공식을 사용하여 C의 좌표를 계산할 수 있다. 실험에서는 C의 좌표 중 왼쪽에 있는 값을 사용하였다. 양 축의 끝의 좌표에서 x축과 y축과의 각도 (α, β) 를 계산할 수 있다.

표 3. 사각형을 그리기 위한 각 이동 좌표와 회전각, 명령

Table 3. Coordinates, rotation angle and command to draw a quadrangle

	P점	Q점	Axis1 회전각	Axis2 회전각	명령형식
출발	(0.0, 22.0)	(0.0, 12.0)	0.00	0.00	
A1	(-5.0, 18.0)	(-8.38, 8.59)	-44.30	64.06	K: -2215 3203
B1	(-2.5, 18.0)	(-7.52, 9.35)	-38.81	68.94	K: -1940 3447
A2	(0.0, 18.0)	(-6.29, 10.22)	-31.59	70.53	K: -1579 3526
B2	(0.0, 15.5)	(-7.74, 9.17)	-40.17	90.90	K: -2009 4545
A3	(0.0, 13.0)	(-8.77, 8.19)	-46.95	108.21	K: -2347 5411
B3	(-2.5, 13.0)	(-10.09, 6.49)	-57.25	106.65	K: -2862 5332
A4	(-5.0, 13.0)	(-10.93, 4.95)	-65.64	102.02	K: -3282 5101
B4	(-3.0, 15.5)	(-9.90, 6.78)	-55.58	84.92	K: -2779 4246
A1	(-5.0, 18.0)	(-8.38, 8.59)	-44.30	64.06	K: -2215 3203

각 축이 이동해야 할 각도는 과거에 이동한 각도를 현재 계산된 각도에서 빼면 된다. 2번째 축은 중심축이 회전할 때 첫 번째 축과 같이 회전하므로 다시 과거에 중심축이 이동한 각을 더 빼주어야 한다. 그림4는 4각형

을 그릴 경우 P점의 경로를 표시한 것으로 양 끝점과 각 변의 중간 지점의 좌표를 함께 나타낸 것이다. 좌표는 중심축과의 실제 길이로 각 P점을 경유해 가면 4각형을 그릴 수 있게 된다. 표3에 그림4의 표시된 지점을 경유하기 위한 양축의 이동 좌표와 회전각 및 Am188ES보드로 전송되는 명령형식을 나타내었다.

2.5 실험

본 실험에서는 앞선 실험에서 사용한 스텝 모터용 반송로봇을 개선하여 DC 서보 모터로 동작을 시킨다. 그림 5는 본 실험 장치의 사진이다.

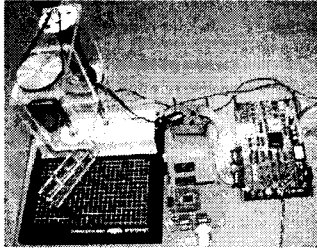


그림 5 2축 DC 모터 제어 시스템
Fig 5. Control system of two axes DC servo motor

그림 6은 위의 표 3의 좌표를 이전 실험에서 스텝 모터를 이용하여 그린 것이다.

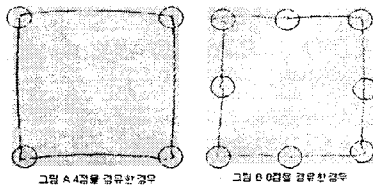


그림 6. 2축 스텝 모터로 그린 사각형
Fig 6. Quadrangle drew by two axes step motor

스텝 모터의 0.18도의 정밀도와 펄스 방식의 구동으로 인해 매끄러운 직선을 그리지 못하였다. 사각형의 중간점을 거쳐 가는 방식으로 그린 그림B에서도 각 점에서 직선이 흔들리는 현상이 나타나고 있다.

표 4. 사각형과 삼각형을 그리기 위한 87c196CA 보드 명령
Table 4. 87c196CA board command to draw a quadrangle and a triangle

사각형을 그린 경우		삼각형을 그린 경우	
그림 7-A	그림 7-B	그림 8-A	그림 8-B
P: -2215 3203 4000	P: -2215 3203 2000	P: -2003 3412 4000	P: -2003 3412 2000
	P: -1940 3447 2000		P: -2718 4303 2000
P: -1579 3526 4000	P: -1579 3526 2000	P: -3417 4968 4000	P: -3417 4968 2000
	P: -2009 4545 2000		P: -2955 5095 2000
P: -2347 5411 4000	P: -2347 5411 2000	P: -2420 5065 4000	P: -2420 5065 2000
	P: -2862 5332 2000		P: -2258 4309 2000
P: -3282 5101 4000	P: -3282 5101 2000	P: -2003 3412 4000	P: -2003 3412 2000
	P: -2779 4246 2000		
P: -2215 3203 4000	P: -2215 3202 2000		

본 실험에서는 이러한 문제를 해결하기 위하여 기구부를 DC 모터로 교체하여 0.02의 정밀도를 갖게 하였다. DC 모터로 개선 후 표 3의 좌표를 그린 그림은 그림 7과 같다. PC에서 표 3의 명령 형식과 같이 Am188ES 보드로 전송하면 Am188ES는

P: $\theta_1 \theta_2$ STEP

과 같은 형식으로 87c196CA 보드로 전송한다. 이때,

θ_1, θ_2 는 모터의 이동 각도를 나타내며, STEP는 모터의 이동시간이다. 다음의 표 4는 그림 7의 A, B와 그림 8의 A, B를 그릴 때의 Am188ES 보드에서 87c196CA 보드로 전송하는 명령의 형식을 나타내었다. 또한, 기구학을 정밀히 풀어 더 많은 좌표를 거쳐가는 경우에도 한 번을 그리는 데 걸리는 시간을 4000 스텝으로 동일하게 그렸다.

실험 결과 사진에서 나타나는 바와 같이 매끄러운 직선으로 나타났으며 사각형의 중점을 거쳐 갈 경우 더욱 직선에 가까워지고 각 점에서 흔들리는 현상도 사라졌다.

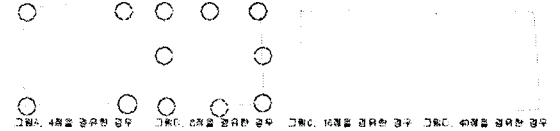


그림 7. 2축 DC 모터로 그린 사각형

Fig7. Quadrangle drew by two axes DC servo motor

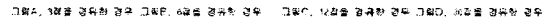


그림 8. 2축 DC 모터로 그린 삼각형

Fig 8. Triangle drew by two axes DC servo motor

그림 8은 기구학의 정밀도에 의해 더욱 매끄러운 직선이 그려짐을 보이고 있다. 그림 8의 좌측에서 우측의 그림으로 갈수록 더욱 깔끔한 직선이 그려져 각 변을 10번의 좌표로 나누어서 그린 그림D의 경우는 거의 완벽한 직선에 가까운 그림이 그려져 만족할 만한 결과를 얻었다.

3. 결 론

본 논문에서는 PC를 Master로 사용하고 실시간 커널을 포팅한 Am188ES를 Slave로, 87c196CA 보드를 서보 컨트롤러로 이용하여 마이크로 분산시스템을 구성하였다. 또한, 이전 실험에서 매끄러운 직선이 그려지지 못한 문제점이 스텝 모터의 정밀도와 동기화에 있다고 보고 기구부를 DC 서보 모터로 개선하였다. 실시간 커널은 작은 코드 크기로 여러 마이크로 컨트롤러에 사용되는 uC/OS-II를 사용하였다.

이전 실험과의 비교를 위해 이전 실험에서 그렸던 4각형을 같은 좌표에서 그려 보았고, 매끄럽지 못하고 각 점에서 흔들리는 현상을 개선하였다. 또한, 정밀도를 높였을 경우 직선에 가까운 매끄러운 선을 얻을 수 있었다.

또한, PC기반의 GUI 프로그램을 추가하여 사용자의 편의를 도모하였으며, Am188ES 측에서는 각도를 분할하여 전송하는 기능을 추가하여 상용 서보 컨트롤러에서도 쉽게 동기화를 맞출 수 있게 하였다.

(참 고 문 헌)

[1] Jean J. Labrosse, "MicroC/OS-II The Real-Time Kernel", R&B Books, 2002
 [2] 김종걸, "실시간 커널에 의한 스텝 전동기의 마이크로 분산 제어 시스템의 구현", 광운대학교 석사학위 청구논문, 2001
 [3] Jean J. Labrosse, "Embedded Systems Building Blocks", R&B Books, 2002
 [4] Ian D. McCammon, Stephen C. Jacobsen, "Communication and Control for Distributed Microsystems", IEEE Control Systems Magazine, pp. 48-50, 1990