

다축 힘제어 시스템을 위한 웹기반 감시시스템 구현

이현철*, 남현도*, 강철구**
 단국대학교 전기공학과, 건국대학교 기계공학과

Implementation of Web Based Monitoring Systems for Multi-Axis Force Control Systems

Hyun-Chul Lee*, Hyun-Do Nam*, Chul-Goo Kang**

* Department Of Electrical Engineering, Dankook University, ** ME, Konkuk University

Abstract - In this paper, web based monitoring systems are implemented for multi-axis force control systems of an intelligence robot.

A web based monitoring system is implemented by porting Linux at embedded systems which include Xscale processors. A device driver is developed to receive data from multi-axis force sensors in Linux operation systems. To control this device driver, a socket program for web browser is also developed.

1. 서 론

오늘날, 정보 기술의 발달로 인하여 컴퓨터와 인터넷 사용자의 수가 급격하게 증가하고 있다. 컴퓨터와 인터넷 기술의 현저한 발달은 산업 현장에서의 인터넷 기반 제어와 감시를 가능하게 했다. 다축 힘 제어 센서를 사용한 자동화 제조 공정인 분포 제어 시스템과 로봇 힘 제어 시스템은 웹 기반 기술로 향상되었다[1,2].

본 논문에서는 지능형 로봇의 힘 제어 시스템을 위해 웹 기반 감시 시스템을 구현하였다. 스트레인지지를 응용한 다축 힘 센서에서 감지된 힘 정보는 오차를 수반하고 있으므로 신호라인에서 발생하는 노이즈의 처리를 위해 8bit A/D 컨버터 MX7828LN를 사용하여 이 데이터를 Xscale 프로세서에서 처리하는 시스템을 구현하였다[3]. 웹 기반 감시 시스템 구현을 위하여 Xscale 프로세서를 기반으로 한 임베디드 시스템에 리눅스를 포팅하여 Xscale 프로세서에 내장형 웹 서버를 구현하고 Xscale 프로세서에 연결된 다축 힘 센서의 출력을 LAN 망을 통한 범용 웹 브라우저에 의하여 감시하는 웹 기반 감시 시스템을 구현하였다. 리눅스 운영 시스템에서 다축 힘 센서의 데이터를 받기 위하여 디바이스 드라이버를 개발하고 이 디바이스 드라이버를 제어하기 위한 응용프로그램으로 웹 서버에서 동작 가능한 TCP/IP 소켓 I 프로그램을 사용하여 다축 힘 센서에서 받은 데이터를 웹 브라우저로 감시하는 시스템을 구현하였다.

2. 힘 센싱 시스템

다축 힘 센서는 대체로 센서 내부에 있는 탄성구조물에 여러 개의 스트레인지지를 부착하여 구조체에 힘이 가해졌을 때 그에 비례하는 스트레인지지의 변형량을 Wheatstone 브리지를 통해 미소 전압으로 바꾸고 이 미소 전압을 증폭기로 증폭시켜 나온 전압 값을 측정한다. 음, 이 데이터 값으로 탄성변형과 힘의 관계식으로부터, 센서에 작용하고 있는 세 방향의 힘과 세 방향의 토크 (또는 그것의 일부)를 구하는 것이다.

2.1 다축 힘 센서의 힘 감지 원리

탄성구조물의 거동이 재료의 탄성한계 이내에 있다고 하면, 내부 탄성구조물의 표면변형률과 센서에 작용하는 외력은 다음과 같은 선형적인 관계를 가진다.

$$Cf = \epsilon \tag{1}$$

여기서 f 는 센서에 작용하는 3 개의 힘성분과 3 개의 모멘트성분으로 구성된 $n \times 1$ 힘벡터이고, ϵ 은 탄성구조물의 6 지점에서 측정되는 $m \times 1$ 표면변형률벡터이고, C 는 이 두 벡터 사이의 관계를 표시하는 $m \times n$ 행렬로 컴플라이언스(compliance) 행렬 또는 보정행렬(calibration)로 불려진다. 컴플라이언스 행렬은 상수행렬로서, FEM 해석으로 얻을 수 있으나, 실험에 의하여 구하는 것이 힘감지오차를 줄일 수 있다. 일차독립인 6 개의 힘 f 에 대해 6 번의 실험을 수행하고, 이 때 얻어지는 ϵ 으로부터 C 를 얻을 수 있다. 그런 다음 임의의 힘이 주어지면 측정되는 ϵ 값과 C 의 역행렬로부터 식 (1)의 관계식을 이용하여 힘 f 를 구할 수 있다. 이 컴플라이언스행렬을 사용함으로써 탄성구조물에 필연적으로 존재하는 축간의 커플링효과를 제거할 수 있다.

이 때 측정되는 힘 f 는 ϵ 의 측정오차와 C 의 오차의 영향으로 구조적인 오차를 수반한다. 즉, 참값에 대한 측정된 힘의 상대오차 $\| \Delta f \| / \| f_0 \|$ 는 변형률 상대오차 $\| \Delta \epsilon \| / \| \epsilon_0 \|$ 과 컴플라이언스 행렬의 상대수 $c(C)$ 를 곱한 것만큼 증폭되어 나타난다는 것이 알려져 있다[1-3].

다축 힘센서의 힘정보 생성과정을 블록선도도 표시하면 그림 1과 같다.

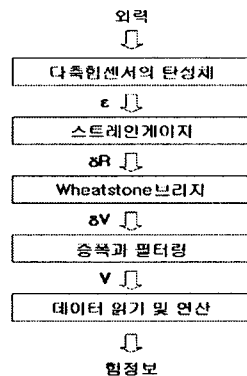


그림 1. 힘 감지 과정

2.2 힘 센서의 신호처리

탄성구조물에 발생하는 표면 변형률은 스트레인지지의 미소 저항변화로 바뀌고, 이 미소 저항변화는 Wheatstone 브리지를 통하여 미소 전압 변화로 감지된다. 이 작은 전압변화를 연산증폭기 회로를 이용하여 증폭한 뒤, 신호에 포함된 노이즈를 제거하기 위하여 저주파 통과필터를 통과시킨다. 본 논문에서는 신호 통과 대역에서 평탄 특성을 유지하면서 회로를 단순화하기 위하여 다음 그림 2와 같은 Sallen-Key Topology의 2차 Butterworth 저주파통과필터를 사용하였다. 그리고 일반

적으로 사용되는 기계시스템의 대역폭과 제어용 샘플링 시간을 고려하여 차단주파수를 102Hz로 선정하였다[3].

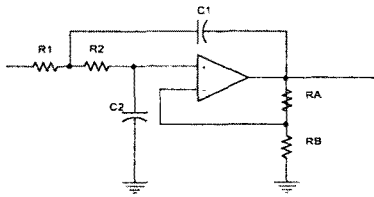


그림 2. 2차 Butterworth 저주파통과필터

이 저주파통과필터의 전달함수 $H(s)$ 는 다음과 같다.

$$H(s) = \frac{1/(R_1 R_2 C_1 C_2)}{s^2 + \frac{R_1 + R_2}{R_1 R_2 C_1} s + \frac{1}{R_1 R_2 C_1 C_2}} \quad (2)$$

본 논문에서 사용한 2차 Butterworth 저주파통과필터의 각 소자값으로 $R_1=R_2=110k\Omega$, $C_1=0.02\mu F$, $C_2=0.01\mu F$ 이고, $R_A=R_B=0$ 을 사용하였다.

2.3 A/D 컨버터

다축 힘 센서의 데이터를 디지털로 변환하는 A/D 컨버터로는 8 채널 아날로그 입력에 8비트의 분해능과 채널당 2.5 μ s 변환시간을 가지는 MAXIM MX7828LN을 사용하였다. 컨버터 내부에 멀티플렉서를 가지고 있어 Xscale 프로세서에서 A/D 컨버터의 A0, A1, A2핀 값을 제어함으로써 각 채널을 선택할 수 있다.

3. 임베디드 시스템

본 논문에서는 타겟 보드로는 한백전자사에서 제작한 Intel Xscale PAX255 Processor(400MHz), 128M Bytes SDRAM, 32M Bytes 플래시 메모리를 내장한 HBE-EMPOS II 보드를 사용하였다. PC를 개발 호스트로 사용하였고 사용 OS 및 프로그램은 Redhat Linux 8.0 (커널 ARM 패치), GNU Tool, Minicom 등이다[4-7].

3.1 커널 구성

커널은 운영체제의 가장 핵심적인 부분으로 프로세서와 시스템 메모리를 관리하여 시스템을 효율적이고 원활하게 동작시키는 기능을 수행한다.

본 논문에서는 커널은 버전 2.4.19를 사용했으며 ARM 패치, Xscale 패치를 하였다. 내장용 커널은 일반 리눅스에서의 커널을 타겟보드의 특성에 맞는 설정을 하고 의존성을 검사한 후에 이미지 파일을 만들 수 있다. 만들어진 이미지 파일은 BLOB을 이용하여 SDRAM에 다운로드하고 Jflash를 이용하여 플래시 메모리에 다운로드한다. SDRAM은 전원이 차단되면 저장된 파일을 보존하지 못하므로 전원을 인가할 때마다 커널 이미지파일을 다시 다운로드하는 것을 방지하기 위해 Jflash를 이용하여 플래시 메모리에 다운로드한다.

3.2 파일시스템

파일 시스템이란 운영체제가 파일을 시스템의 디스크 상에 구성하는 방식을 말한다. 운영체제는 시스템의 디스크나 파티션상에 파일들을 연속적이고 일정한 규칙을 가지고 저장하는데, 파일 시스템은 이러한 규칙들의 방식을 제시하는 역할을 한다. 또한 파일 시스템은 시스템 디스크나 파티션 그리고 파일 시스템의 형식을 말할 경우에도 쓰일 수 있다. 파티션과 파일 시스템은 다른 것이다. 파일 시스템은 파티션을 구성해 주는 역할을 한다.

파일 시스템을 포함하지 못한 파티션은 파일 시스템이 사용될 수 있도록 초기화되고 파일 정보를 기록하기 위한 형식을 만들어야 한다. 파일 시스템은 램 디스크, Jffs, Jffs2, Cramfs, Ramfs 중에서 타겟보드 사양에 맞게 여러 가지를 선택적으로 사용할 수 있다. 램 디스크란 별다른 물리적 장치를 지칭하는 것이 아니라, 메모리의 일부를 디스크로 인식시킨 것이다. 램 디스크를 Root 파일 시스템으로 사용하는 것이 내장형 리눅스에서 가장 일반적인 방법이다. 램 디스크는 램에서 동작하기 때문에, 속도가 빠를 뿐만 아니라 gzip 알고리즘으로 압축을 하기 때문에 용량을 줄일 수 있다는 장점이 있지만 휘발성이기 때문에 전원을 차단하면 데이터가 없어진다는 단점이 있다. 이를 보완하기 위해 플래시 파일 시스템(Jffs, Jffs2)을 같이 사용한다. 본 논문에서는 램 디스크에 웹 서버를 컴파일해서 추가하고 포팅했다.

4. 원격 제어를 위한 웹 기반 감시 시스템

인터넷을 통한 원격 제어 시스템이라 함은 인터넷을 통하여 인터넷에 연결되어 있는 다른 여러 개의 시스템에 쉽게 접속하여 제어함과 동시에 감시하는 것이라 할 수 있다. 이것은 대부분 PC에서 구현되고 있으며 최근에는 감시를 위하여 메일을 사용하는 경우가 많아졌다. 웹 서버란 웹 페이지가 들어있는 파일을 사용자들에게 제공하는 프로그램이나 이 프로그램이 설치되어 있는 기기를 말하기도 한다.

원격 제어 시스템을 위한 내장형 웹 서버는 주로 시스템에 연결된 특정 하드웨어를 제어하거나 그 정보를 전달하여 주는 역할을 하게 된다. 웹 서버는 사용자의 웹 브라우저 요청에 따라 HTTP를 통해 정보를 전달한다. 이를 위해 TCP/IP 소켓 프로그램을 사용하여 타겟 보드의 IP로 접근하여 데이터를 받아 올 수 있다.

4.1 내장형 웹 서버 구성

내장형 웹 서버는 정보를 교환할 수 있는 HTTP와 TCP/IP 지원 가능한 웹 서버 그리고 외부 기기를 제어하기 위한 디바이스 드라이버와 디바이스 드라이버를 제어하여 웹 서버에게 정보를 전달하는 TCP/IP 프로그램 등으로 구성된다. 본 논문에서는 Goahead 웹 서버를 사용하였다. 그림 3은 내장형 웹 서버의 구성을 보여준다.

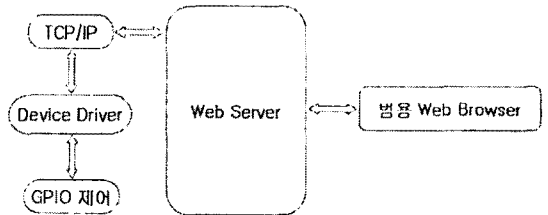


그림 3. 내장형 웹 서버 구성

4.2 디바이스 드라이버

범용 웹 브라우저를 통하여 프로세서의 포트를 제어하고 감시하기 위해서는 우선 포트를 제어할 수 있어야 하는데 리눅스 운영 시스템에서 하드웨어 주변장치를 제어하기 위해서는 디바이스 드라이버가 필요하다. 리눅스 운영 시스템은 하나의 장치를 파일로 인식하여 접근하며, 디바이스 드라이버를 통하여 응용 프로그램에서 하드웨어를 제어하게 된다.

디바이스 드라이버는 커널 모드에서 실행되며 메모리에 상주하고 이것을 통해 하드웨어 장치를 파일처럼 다룰 수 있게 된다. 따라서 프로세서의 내장된 주변 장치를 프로그램에서 사용하기 위하여 디바이스 드라이버를

제작하여 설치하여야 한다.

본 논문에서는 병렬포트를 웹 서버에 의하여 제어하는 것을 다루므로 병렬포트에 대한 디바이스 드라이버를 서술한다. Xscale 프로세서의 병렬포트는 버퍼를 통하지 않고 직접 읽고 쓸 수 있는 장치로서 캐릭터 디바이스라 하며 커널 모듈(커널의 일부분을 동적으로 로드 또는 언로드 할 수 있는 커널의 구성요소) 프로그램에 의하여 모듈로 사용할 수 있게 한다. 그림 4는 캐릭터 디바이스 드라이버의 구조를 보여준다.

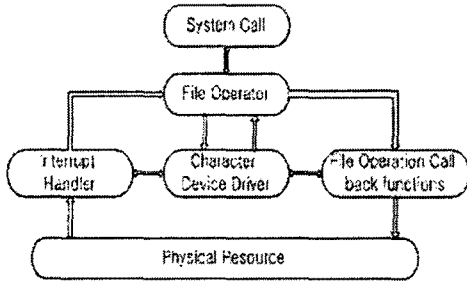


그림 4. 캐릭터 디바이스 드라이버 구조

커널 모듈을 만들 때는 반드시 초기화 함수와 삭제 함수가 정의 되어 있어야 하며 커널 버전이 정의 되어 있어야 한다. 제작된 ADC 디바이스 드라이버를 램디스크에 모듈화하여 실장함으로써 제어하고자 하는 프로그램에서 모듈을 열어 디바이스 드라이버를 사용할 수 있게 한다 [8].

Intel Xscale PXA255의 GPIO는 총 84비트가 있으며 양방향 입·출력이 가능하다. GPIO의 상태를 알기 위해서는 GPLR 레지스터의 값을 읽어오면 된다. 하지만 입력 값을 읽을 경우 반드시 각 핀의 입·출력 방향을 GPDR 레지스터에 정해 주어야 한다.

4.3 TCP/IP 소켓 프로그래밍

소켓(Socket)은 이를 통해 애플리케이션이 데이터를 보내고 받을 수 있는 추상적 개념으로서, 애플리케이션이 저장 장소에 데이터를 읽고 쓰는 과정을 open된 파일을 통해 처리할 수 있는 것과 유사하다. 소켓은 애플리케이션이 네트워크에 "플러그 인(plug in)"하여 동일한 네트워크에 마찬가지로 플러그 인된 다른 애플리케이션과 통신할 수 있도록 한다. 한 기계에서 애플리케이션에 의해 소켓에 쓰인 데이터는 다른 기계에서 동작하는 애플리케이션에 의해 읽혀질 수 있으며, 그 역도 성립한다[9].

본 논문에서는 C 언어를 사용하여 프로그래밍 하였다. 먼저, socket() 함수를 써서 소켓을 생성하고 -sockaddr 구조체를 정의해서 내장형 웹 서버로 접근하여 데이터를 받아올 수 있다.

4.4 자바 프로그래밍

소켓 프로그래밍을 통하여 타겟 보드의 내장형 웹 서버로 접근하여 Xscale 프로세서 내에 저장되어 있는 힘 정보를 받아 올 수 있었다. 이 데이터를 JSP를 이용하여 웹 브라우저에서 디스플레이 하였다. JSP(Java Server Page)는 Java 기반의 웹 스크립트 언어로서 HTML, Java Script 등의 클라이언트 측 스크립트 언어들과 연결된 동적 문서를 만들어 주는 자바 코드이다. 자바 프로그램의 구성은 웹 브라우저 창 내의 start 버튼을 클릭 하였을 때 이벤트를 발생시켜 힘 정보 데이터를 불러와 그래프로 디스플레이 되도록 하였다. 그림 5는 다축 힘 센서로부터 감지된 힘 정보 그래프를 보여준다.

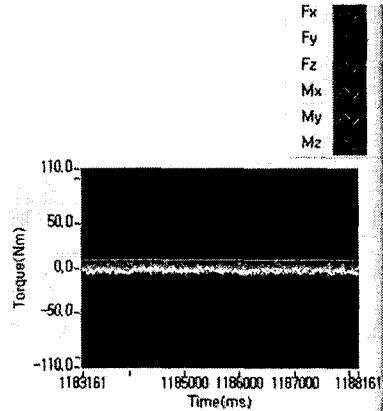


그림 5. 힘 정보 그래프

5. 결 론

본 논문에서는 다축 힘 센서의 성능을 향상시키기 위한 웹 기반 감시 시스템을 구현하였다. 이를 위해 내장형 시스템의 운영 시스템 중 여러 면에서 장점을 가지고 있어 최근 대두되고 있는 리눅스 운영 시스템을 Intel Xscale PAX255 Processor를 내장한 HBE-EMPOS II 보드에 실장하여 내장형 웹 서버를 구현하고 Xscale 프로세서에 연결된 힘 센서의 데이터를 받기 위하여 디바이스 드라이버를 제작하였으며 TCP/IP 소켓 프로그래밍을 이용하여 힘 센서의 데이터를 범용 웹 브라우저를 통하여 감시하는 웹 기반 감시 시스템을 구현하였다.

웹 기반의 하드웨어 제어가 가능한 내장형 웹 서버 시스템은 포스 시스템, 홈오도메이션, 공장자동화에 응용되고 있으며, 원격 디스플레이, 데이터 검출, 건물자동화나 사무자동화 등과 같은 수많은 전기, 전자 제품에 응용될 것이라고 기대된다.

본 연구는 한국과학재단 목적기초연구(R01-2002-000-00124-0(2002))지원으로 수행되었음.

[참 고 문 헌]

- [1] 강철구, "힘토크 센서의 힘감기 오차에 대한 해석," 대한기계학회논문집 A, 제22권, 제7호, pp.1250-1257,1998
- [2] 강철구, "다축 힘센서에서 힘감지 오차의 전파,"대한기계학회논문집 A, 제24권, 제11호, pp.2688-2695, 2000
- [3] 김용찬, "마이크로 컨터롤러와 Labview를 이용한 다축 힘 센서의 신호처리," 건국대학교 석사학위논문, 2003
- [4] 임홍식, "임베디드 시스템을 이용한 웹기반 감시시스템 구현," 단국대학교 석사학위논문, 2004
- [5] Michael Barr, Programming Embedded System In C And C++, O'Reilly, 1999
- [6] 박영환, 임베디드 시스템 임베디드 리눅스, 사이텍 미디어
- [7] 송태훈, Intel PXA255와 임베디드 리눅스 응용, 홍릉과학출판사, 2004
- [8] Alessandro Runibi, Linux Device Driver, O'Reilly, 2000
- [9] 박준철, TCP/IP 소켓 프로그래밍, 사이텍 미디어, 2001