IEEE-1394 네트웍 기반 분산형 로봇 제어기의 소프트웨어 구조에 관한 연구

윤기중[O] 박재현 김홍석
인하대학교 정보통신 대학원, 인하대학교 정보통신 대학원, 생산기술 연구원
gjyun90@resl.inha.ac.kr [O] jhyun@inha.ac.kr hskim@chaos.kitech.re.kr

Software architecture for Robot control system Based on IEEE-1394 Network

Gi-Jung Yun [O] Jae-Hyun Park Hong-seok Kim
Dept. of Information and Communication Engineering, inha University, Inchon, Korea
Dept. of Information and Communication Engineering, inha University, Inchon, Korea
Korea Institute of Industrial Technology, ChungNam, Korea

요 약

현재 대부분의 로봇 제어용 네트웍으로는 FieldBus 등이 사용되고 있다. 그러나 로봇 시스템의 고성능화와 다양한 기능으로 인하여 기존의 FieldBus가 제공하던 기능으로는 대역폭에서나 성능면에서 부족한 점이 나타나고 있다. IEEE1394는 이러한 로봇 제어용 네트웍에 매우 적합한 특성을 갖고 있다. 본 논문에서는 IEEE1394가 로봇 제어용 네트웍에 사용될 때 가질 수 있는 실시간성과 신뢰성 특징에 대해 분석해보고, IEEE1394의 특성을 잘 살릴 수 있는 제어용 소프트웨어 구조에 대해 연구하고 이를 구현한다. 실시간성 데이터를 위해서는 우선순위 큐를 이용한 패킷 전송방법을, 주기적 데이터를 위해서 등시성 전송방법을 이용한다.

1. 서 론

기존의 제어 시스템은 그 구성이 중앙 집중적인 방법들을 사용해왔다. 각 디바이스들은 중앙의 제어용 컴퓨터와 일대일로 연결이 되어 있으며, 그 프로토콜도 각 회사마다 독립된 형태를 따르고 있었다. 이러한 시스템에서는 설치 비용이나 유지비등이 많이 소모되었으며, 확장성이 부족하였다. 이러한 문제점을 개선하기 위한 방법으로 FieldBus[5]를 이용하게 되었다. FieldBus를 통하여 디바이스들은 하나의 버스를 통하여 데이터를 주고 받을 수 있게 되었고, 그 결과로 중앙 집중적인 구조를 벗어나 분산 제어가 가능하게 되었다. French standard Factory Instrumentation Protocol(FIP)와 Controller Area Network(CAN)등이 대표적인 FieldBus 이다. 그러나 이러한 FIP 와 CAN 은 현재의 로봇 제어 시스템이 요구하는 성능을 모두 수용하기에 부족한 실정이다[3].

IEEE1394는 디지털 캠코더, 디지털 TV, 퍼스널 컴퓨터를 포함한 가정내외 디지털 네트웍을 구성하기 위한 대표적인 인터페이스이다[6]. 최근에는 기존의 멀티미디어 데이터를 지원하는 장비의 인터페이스뿐만 아니라 실시간성이 요구되는 제어 시스템에 적용이 되고 있다[4].

본 논문에서는 IEEE1394가 로봇 제어용 네트웍으로 사용될 수 있는 가능성에 대해 분석해보고, 그 특징을 잘 살릴 수 있는 제어용 소프트웨어 구조를 제안한다. 제 2장에서는 IEEE1394의 특징과 로봇 제어용 네트웍에 적합한가에 대해 알아보고, 3장에서는 로봇 제어용 네트웍으로서 IEEE1394를 사용할 때 그 성능을 잘 살릴 수 있는 소프트웨어 구조를 제안한다, 4장에서는 제안된 소프트웨어 구조의 구현과 이를 통하여 성능을 평가하여 결과를 보여주고, 5장에서 결론을 맺는다.

본 연구는 산업자원부지원 차세대 신기술 개발사업인 퍼스널 로봇 기반기술개발 과제로 수행되었습니다.

2. 기본 개념 및 관련 연구

2.1 IEEE1394

IEEE1394는 100Mbps, 200Mbps 및 400Mbps 의 전송속도를 지원한다. 이는 전송속도가 서로 다른 노드들 사이에서도 중재를 통해 하나의 버스를 통하여 전송이 가능하며, 서로간의 효율적인 전송이 보장되도록 버스 초기화 과정에서 버스의 대역폭을 조절한다. 버스는 Control and Status Register(CSR) architecture (IEEE1212 표준)[2]를 따르며, 64bit 의 어드레스 구조를 갖는다.

IEEE1394는 서로 다른 두 가지 데이터 전송방식을 지원한다. 비동기 전송(Asynchronous transfer)은 제어와 셋업을 목적으로 하는 짧은 메시지에 주로 사용되며, 버스 대역폭의 최소 20%이상이 사용 가능하다. 데이터 교환은 요청(request) 과 응답(response)의 구조로 이루어 지며 패킷을 받은 수신자는 ACK 패킷을 바로 전송하게 되어 시간보다 신뢰성이 중요한 곳에 사용된다. 비동기 전송은 트랜잭션 계층에서 발생된다. 등시성 전송(Isochronous transfer)은 고정된, 또는 보장된 대역폭으로 다량의 데이터를 전송할 때 사용되며, 재 전송이나 ACK 패킷이 없다. 등시성 송수신은 트랜잭션 계층을 거치지 않고 응용프로그램에서 직접 링크 계층을 통해 이루어진다.

2.2 IEEE1394의 로봇제어 네트웍으로의 사용 가능성

제어 시스템에서는 신뢰성 있는 실시간 데이터의 전송을 위해 통신 데이터를 다음과 같이 분류한다.

- 실시간 데이터 – 시스템의 각종 이벤트와 경고신호 같은 데이

터 길이가 짧고, 시간 제약을 받는다.
- 주기적 데이터 - 피드백 제어를 위한 센서 데이터와 같은 주기적으로 발생되며, 일정 시간 이내에 전송되어야 한다.
- 다량의 비 실시간 데이터 - 프로그램 및 데이터 파일과 같은 데이터의 길이가 길고, 시간 제약을 받지 않는다.

로봇 제어 네트웍에서 실시간 데이터의 전송은 매우 중요시되는 특성중 하나이다. IEEE1394 표준에서는 실시간 데이터에 대한 직접적인 지원을 하지 않는다. IEEE1394에서는 비동기 전송을 최소 20%의 대역폭을 할당하며, MAC Layer의 데이터 전송 방법은 아비트레이션을 기본으로 한다. 실시간 데이터의 전송은 고정 대역폭을 할당한 등시성 전송을 사용하거나, 디바이스 드라이버상에서 실시간 데이터 처리를 위한 이벤트 구조를 사용한 비동기 전송을 사용할 수 있다.

IEEE1394의 등시성 전송은 Cycle Master 에 의해 125us 주기로 발생되는 Cycle Start 패킷에 의해 시작된다. 이러한 특징은 주기적인 데이터의 전송에 좋은 특성을 나타낸다. 등시성 전송은 IEEE1394 표준에서 80%를 넘지 않도록 규정하고 있으며, 이러한 특징은 일정한 대역폭을 할당 받은 노드의 데이터 전송을 보장하게 된다. 등시성 전송은 Point-to-Point 뿐만 아니라 Broadcast 가 가능하므로 전 시스템의 데이터 동기화에 아주 적합한 특성을 나타낸다.

IEEE1394는 실행시간 재구성 가능한 특성을 갖고 있다. 이러한 기능은 로봇 구조에 유연한 특성을 제공할 수 있다. 필요에 따라 필요로 하는 기능을 추가하여 사용할 수 있음을 나타내고 IEEE1394 표준에 따르는 기능성 모듈이라면 제작한 벤더에 관계없이 잘 동작할 수 있음을 나타낸다.

3. IEEE1394를 이용한 로봇 제어용 소프트웨어 구조

IEEE1394는 홈 네트웍에 많이 사용되고 있으며, 각종 AV 기기나 디지털 TV등을 연결하는 인터페이스로 연구되어 왔다. 최근에 실시간성이 요구되는 제어 시스템에 적용을 연구하고 있으나, 대부분이 IEEE1394 위에 TCP/IP 등의 프로토콜 스택[7]을 올린 형태로 연구가 되고 있다.

이러한 방법을 로봇 제어에 그대로 적용하는 것은 몇 가지 문제점이 있다. 첫 번째 문제점으로 실시간 데이터를 처리하지 못한다는 문제점이 있다. 이를 개선하기 위한 노력들이 있으나[7] 각 데이터의 처리에 우선순위에 대한 효과적인 반응에 문제점이 있다. 두 번째로 신뢰성 있는 데이터의 전달을 보장하는가에 있다. 등시성 전송은 재전송이 없으므로 사용자 어플리케이션 레벨에서 재전송 기능을 추가해야 하며, 이때에는 실시간성을 만족하는지 확인을 해야 된다. 비동기 전송을 사용 할 때는 데이터의 신뢰성 있는 전달을 보장하지만 우선순위가 낮은 데이터에 의해 우선순위가 높은 데이터의 전송이 지연되는 우선순위 역전 현상을 제거해야 된다.

본 논문에서는 위의 문제점을 해결하고 IEEE1394가 갖고 있는 성능을 최대한 유지하기 위해 3가지의 데이터 종류에 대해 2계층의 독립된 데이터 전달방법을 제안한다.

로봇은 많은 모듈들의 집합체이다. 이러한 모듈들을 분류해보면 전 시스템의 제어를 담당하는 컨트롤러(Controller)와 데이터를 수집하거나 원하는 위치로 이동을 담당하는 기능성 모듈(Actuator)들로 분류할 수 있다. 이러한 모듈들이 서로 정보를 공유하기 위해서는 서로의 기능과 특성을 인지할 수 있는 정보가 필요하다. 이를 위해 Actuator Description table 을 만들고 이를 통해 각 모듈들 간에 데이터를 주고 받도록 구성한다.

그림 1은 각 Actuator 을 구분할 수 있는 ID 와 기능을 나타내고 있다. 디바이스가 초기화 될 때 32bit 의 고유한 ID를 생성하게 되고, Controller 와 Actuator 사이의 데이터 전송이나 제어신호 전송에 이 ID가 사용되게 된다. 이 테이블은 Actuator 에서 주기적 데이터가 발생되었을 때 데이터의 전송 목적지에 대한 참조 테이블 역할을 한다.

통신 데이터 중 실시간 데이터와 비 실시간성이며 비 주기적인 데이터는 IEEE 1394의 비동기 전송을 이용하고 Transaction Layer에서 동작하며, IEEE1394 표준으로 제공하는 Read, Write, Lock 프로토콜을 이용한다.

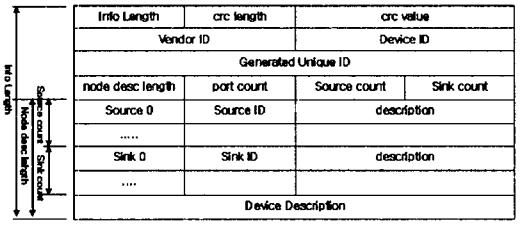

그림 1. Actuator description format

IEEE1394 표준은 실시간 데이터에 대한 직접적인 지원을 하지 않으므로 이를 구현하기 위해서는 소프트웨어적으로 구현을 해야 된다. 이를 위해서 그림 2와 같이 디바이스 드라이버레벨에서 우선순위 큐를 이용하여 실시간 데이터와 비 실시간 데이터를 구분한다. 모터의 제어 데이터나 위치제어 데이터는 실시간 데이터로 구분되어 우선순위가 높은 큐에 삽입되고 빠르게 처리된다. 기타 데이터는 비 실시간 데이터로 사용되고 우선순위가 낮은 큐에 삽입되어 처리된다. 이때 발생될 수 있는 우선순위 역전 현상은 네트웍 어댑터가 낮은 우선순위 데이터를 처리하는 동안 높은 우선순위 데이터가 큐에 들어왔을 때부터 낮은 우선순위 데이터의 처리가 완료 될 때까지 걸리는 시간이 된다.

실시간 데이터와 비 실시간 데이터는 해당 목적지 어드레스에 의해 구분을 함으로서 트랜잭션 계층의 Read, Write, Lock 프로토콜을 이용할 수 있다.

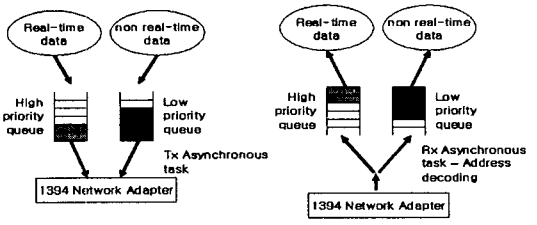

그림 2. Real-time data and non real-time data by priority queue mechanism on Asynchronous transfer

그림 3은 등시성 전송을 통해 데이터의 전달을 나타낸 그림이다. 등시성 전송은 주기적인 데이터를 처리하는데 사용된다. Robot Agent는 각 Actuator의 하부 드라이버로 동기화 데이터의 관리 역할을 담당한다. Agent는 하부 디바이스 드라이버로부터 전달되는 데이터를 각 Actuator Description table을 참조하여 전 노드에 전달하는 역할을 담당하고 있으며, 등시성 전송을 통해 데이터를 전달한다.

이렇게 함으로서 Controller(제어 모듈)에서는 빠른 정보 수집이 가능하며, 각 Actuator(제어 데이터에 의해 동작하는 슬레이

브)에서는 모든 데이터에 대한 사용자 레벨의 처리가 필요 없으므로 프로세스의 이용률이 좋아지고, 커널 레벨에서 데이터 전송을 지원함으로 불필요한 오버헤드를 줄일 수 있다

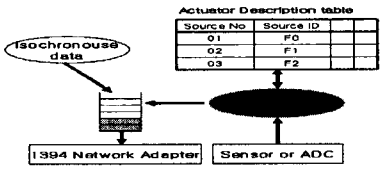

그림 3. Robot Transmit Agent

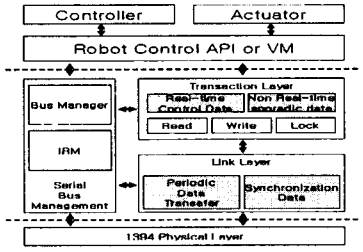

그림 4. 로봇 제어용 IEEE1394 프로토콜 계층

그림 4는 본 논문에서 제안하는 소프트웨어에 대한 전체적인 구조를 나타내고 있다. 계층간 데이터의 전달은 CSR Register의 로봇 제어 용 어드레스를 사용하며, Controller 기능을 갖는 모듈은 각 Actuator에 대한 데이터베이스 역할을 하는 테이블(Actuator lookup table)을 관리해야 된다.

4. 구현 및 성능평가

제안한 IEEE1394를 이용한 로봇 제어용 소프트웨어 구조는 다음과 같은 환경에서 구현 하였다. 시스템의 구성은 두 개의 Actuator와 하나의 Controller로 구성하였다.

OS : Linux2.4.18
CPU : Main Controller – Intel Pentinum 4
IEEE1394 네트웍 카드 : TI 의 OHCI1394표준을 따르는 TSB12LV26을 이용한 Sarotec 의 FHA-431

그림 5는 구현에 사용된 소프트웨어의 구조를 나타낸다. IEEE1394의 Link Layer 와 Transaction Layer를 Linux의 디바이스 드라이버 상에서 지원을 하도록 구현을 했다. 우선순위 큐를 이용하여 실시간 데이터의 처리가 가능하도록 하였고. 등시성 전송으로 주기적 데이터의 전송을 담당하도록 구성하였다.

<표 1>은 제안된 소프트웨어 구조를 이용하여 데이터를 전송할 때 일반적인 디바이스 드라이버를 이용한 TCP/IP over IEEE1394와의 데이터 전달 특성에 대해 비교한 결과이다.

결과에서 알 수 있듯이 우선순위 큐를 사용한 제안한 소프트웨어 구조에서 실시간 데이터가 일반 소프트웨어 구조를 사용했을 때와 비교하여 RTT(Round trip time)에 대한 변위가 작게 나옴을 알 수 있었다.

	실시간 데이터	비 실시간 데이터	TCP/IP over IEEE1394
평균	0.224 ms	0.634 ms	0.569 ms
표준편차	0.036	1.786051	1.445

<표 1> 제안된 소프트웨어 구조의 데이터 전송 특성 (Round trip time)

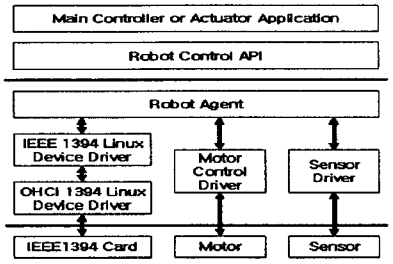

그림 5. The Software structure for implemented system

5. 결 론

본 논문에서는 IEEE1394가 로봇 제어를 위한 네트웍에 적합한 특성을 보임을 알아보았고, IEEE1394의 성능을 잘 살릴 수 있는 제어용 소프트웨어 구조를 제안하였다.

제어 데이터의 실시간성을 보장하기 위해 우선순위 큐를 이용하였고, 신뢰성을 보장하기 위해 비동기 전송을 이용하였다. 비실시간성이며 주기적인 데이터를 위해서는 사이클 스타트 패킷에 의해 동기되는 등시성 전송을 이용하였다.

로봇을 구성하는 각 모듈의 정보를 표현하기 위해 Actuator Descriptor format을 사용하였고, 이 데이터에 기초하여 Agent는 동기화 데이터를 구분하여 전송하도록 하였다.

구현의 예에서 알 수 있듯이 IEEE1394는 홈 네트웍이나 멀티미디어 전송매체로서의 기능뿐 아니라 로봇 제어용 네트웍으로도 좋은 성능을 나타냄을 알 수 있었다. 앞으로 산업용 네트웍으로의 사용도 기대해 볼만 하겠다.

참 고 문 헌

[1] IEEE1394, Std for High Performance Serial Bus, 1995-2000
[2] IEEE Standard 1212,' Information technology - Microprocessor systems - Control and Status Registers (CSR) Architecture for Microcomputer buses'
[3] S.H.Hong, ' Bandwidth Allocation Scheme in the Cyclic-Service FieldBus Networks' , IEEE/ASME Trans., 1999
[4] 박동환, 오봉진, 강순주, ' IEEE1394 기반의 홈네트워크에서 이벤트 등시성 서비스의 실시간성 보장을 위한 소프트웨어 구조' , 정보과학회지, 2002
[5] P.Pleinevaux and J.D. Decotignie, ' Time Critical Communication Networks: Field Busses' , IEEE Network Magazine, vol.2, pp.55-63, 1988
[6] Tai-Yeon Ku, Dong-Hwan Park and Kyeong-Deok Moon, ' A Java-based home network middleware architecture supporting ieee1394 and TCP/IP' , IEEE Transactoin on Consumer Electronics, Vol.48, No.3, August 2002
[7] Bok-Jin Youm, Jaehyun Park, ' A modified TCP/IP over IEEE1394 for real-time control communication' , ICCAS, 2002