실시간 CAN 통신 디바이스 드라이버를 위한 미들웨어에 관한 연구

A Study on Middleware for Real-Time CAN Communication Device Drivers

김 석, 이철훈
충남대학교 컴퓨터공학과

Kim Seok, Lee Cheol-Hoon
ChungNam Univ.

요약

윈도우즈 시스템 상에서 실행되는 CAN 통신 소프트웨어는 CAN 통신 데이터의 정확한 측정 및 성능 검증을 위해 실시간성을 요구한다. 본 연구에서는 윈도우즈 시스템에 실시간성을 제공하는 RTiK-MP(Real-Time implant Kernel-Multi Processor)을 이용하여, CAN 통신 디바이스 드라이버의 실시간성 지원을 위한 미들웨어를 구현하였다. 미들웨어 사용의 확장성, 편의성을 위해 API(Application Program Interface) 형태로 제공하고, 이를 활용한 프로그램을 통해 미들웨어를 검증한 결과 10ms 주기에서 오차범위 내에서 정상 동작함을 확인하였다.

Ⅰ. 서론

CAN 통신(Controller Area Network)은 차량 내에서 마이크로 컨트롤러나 장치들이 서로 통신하기 위해 설계된 표준 통신 규격으로, 차량용 통신에 널리 사용되고 있다.[1] 최근 들어 자동차 부품의 전자화가 확산되면서 ECU(Electronic Control Unit) 간의 CAN 통신 성능 검증을 위한 CAN 통신 소프트웨어의 중요성이 강조되고 있다. 이러한 소프트웨어는 정확한 데이터를 측정하고, 검증하기 위해 실시간성을 지원해야 한다.

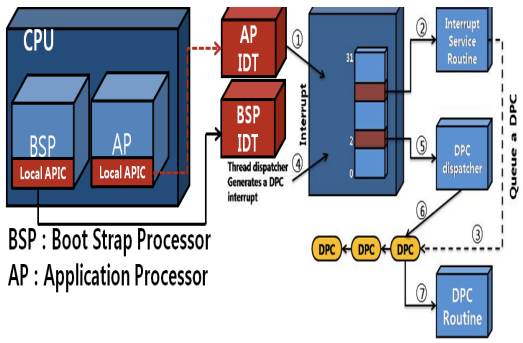
본 연구에서는 실시간성을 지원하지 않는 윈도우즈 기반 CAN 통신 디바이스 드라이버에 실시간성을 제공하기 위한 방법으로, RTiK-MP을 이용한 API 형태의 미들웨어에 대해 연구하였다.

본 논문은 2장에서 관련 연구로 RTiK-MP의 구조와 Local APIC에 대해서 설명하고, 3장에서는 실시간 CAN 통신 디바이스 드라이버를 위한 API 형태의 미들웨어에 대해 기술하고, 4장에서는 실험 환경 및 결과를 기술하였으며, 마지막 5장에서는 결론을 맺는다.

Ⅱ. 관련 연구

1. RTiK-MP

RTiK-MP는 x86 기반의 멀티프로세서 윈도우즈에 실시간성을 제공하기 위해 디바이스 드라이버 형태로 이식되며 윈도우즈와는 별개의 HAL(Hardware Abstraction Layer)을 가짐으로써 각각의 프로세서가 가지고 있는 Local APIC(Advanced Programmable Interrupt Controller)접근을 통해 실시간성을 지원한다. [그림1]에서 보듯이 멀티프로세서 시스템의 경우, 각각의 프로세서는 고유의 Local APIC를 가진다. RTiK-MP는 Application Processor의 Local APIC를 이용해 윈도우즈와는 다른 독립적인 타이머 인터럽트를 발생시켜 주기적인 동작을 보장하고, 이를 통해 윈도우즈 시스템에 실시간성을 제공한다.[2][3]



▶▶ 그림 1. RTiK-MP 구조

2. Local APIC

Local APIC Intel Architecture에서 제공하는 인터럽트 컨트롤러로써, 하드웨어 인터럽트를 인터럽트 핸들러의 주소를 가지고 있는 IDT(Interrupt Descriptor Table)로 전달해주는 역할을 한다.[4]

Ⅲ. 본론

RTiK-MP를 이용한 미들웨어 사용의 확장성, 편의성을 위해서 실시간 CAN 통신 드라이버의 API를 정의하였다. API는 헤더파일과 DLL(Dynamic Linking Library) 형태로 제공되고, 사용자는 DLL 링크를 이용해 편리하게 실시간 CAN 통신 드라이버를 사용할 수 있다.

[표 1]은 미들웨어가 제공하는 API를 정의한 표이다.

표 1. 미들웨어가 제공하는 API

API	Meaning
RTiK_Initialization	RTiK-MP 초기화
RTiK_CreateThread	실시간 통신 쓰레드 생성
RTiK_Enable	RTiK-MP 활성화
RTiK-Disable	RTiK-MP 비활성화
RTiK_Event_clear	RTiK-MP 이벤트 해제 (커널 영역)
RTiK_End	초기화 시에 생성된 이벤트핸들 해제 (유저 영역)
CAN_Driver_Init	CAN통신 드라이버 초기화
CAN_Driver_DeInit	CAN통신 드라이버 해제
CAN_Driver_Get_Config	CAN통신 Hardware 설정 얻기
CAN_Channel_Enable	CAN통신 채널 활성화
CAN_Channel_Disable	CAN통신 채널 비활성화
CAN_Tx_Enable	CAN통신 데이터 송신 활성화
CAN_Tx_Disable	CAN통신 데이터 송신 비활성화
CAN_Rx_Enable	CAN통신 데이서 수신 활성화
CAN_Rx_Disable	CAN통신 데이터 수신 비활성화

Ⅳ. 실험 환경 및 결과

1. 실험 환경

미들웨어의 실시간 CAN 통신 성능을 검증하기 위해 제공하는 DLL 링크를 이용한 프로그램을 만들어 Host-PC에서 실행하였다. Host-PC와 Target Board 사이에 CAN 통신 데이터 모니터링용 PC를 두고 Host-PC에서 송신한 데이터 로그를 저장 후, 송신 주기와 데이터의 정확성을 확인하였다. 워크로드는 while문을 무한 반복하는 윈도우즈의 프로세스를 이용하였고, 워크로드 개수에 따른 미들웨어의 성능을 검증하였다.

Host PC, Target Board의 실험 환경은 [표 2]와 같다.

표 2. 실험 환경

	Host PC	Target Board
CPU	Intel Core2 Duo E8400@ 3.00GHz	Renesas V850
OS	Windows 7 Ultimate K	OSEK OS
CAN Comm. Device	Vector CANCaseXL	Infineon TLE6250G

2. 실험 결과

[표 3]은 워크로드의 개수에 따른 미들웨어의 CAN 통신 10ms 송신 주기 측정 결과의 최대, 최솟값을 나타내는 표이다. 워크로드가 없을 경우 약 0.3%의 오차를 가지며 동작하고, 5개, 10개의 워크로드가 적용 될 시 약 0.7%의 오차로 동작하는 것을 확인할 수 있다.

전송된 데이터의 순서와 정확성을 확인하기 위해 실험에 사용한 8바이트의 CAN 데이터 중, 1번째 바이트는 순서 번호로 하여 송신시마다 하나씩 증가시켰고, 나머지 7바이트는 'R', 'T', 'i', 'K' ,'-', 'M', 'P' 문자로 구성하여 전송하는 방법을 사용하여 이상 없이 데이터 수신이 이루어지는 것을 확인할 수 있었다.

표 3. 워크로드 적용 미들웨어 CAN 통신 주기 측정

전송 주기	로드 개수	최 대	최 소	오차율(%)
10ms	로드 없음	10.027ms	9.978ms	0.3
	로드 5개	10.072ms	9.923ms	0.7
	로드 10개	10.068ms	9.927ms	0.7

Ⅴ. 결론

본 연구에서는 윈도우즈 시스템에 실시간성을 제공하는 RTiK-MP을 이용하여 실시간 CAN 통신을 위한 미들웨어를 설계 및 구현하였고, 10ms 주기에서 안정적인 실시간 CAN 통신 성능을 확인 할 수 있었다.

향후 CAN 통신 데이터의 정밀한 측정방법을 이용하여, 1ms 및 5ms 전송 주기에서의 성능을 검증하고, 안정적인 미들웨어의 성능을 위해 오차율을 줄이는 방안을 연구할 필요가 있다.

■ 참 고 문 헌 ■

[1] "CAN_bus", Wikipedia, The Free Encyclopedia. Wikimedia Foundation, Inc. 15 Mar. 2016. Web. 29 Mar. 2016.

[2] 이승훈, 조아라, 김효중, 조한무, 박영수, 이철훈, "윈도우즈 시스템 상에서의 군용 점검장비를 위한 실시간 통신", 한국차세대컴퓨팅학회논문지, 제8권 제4호, pp.47-57, 2012. 8.

[3] 송창인, 이승훈, 주민규, 이철훈, "멀티프로세서 윈도우즈 상에서 실시간성 지원", 한국콘텐츠학회논문지, 제12권, 제6호, pp.68-77, 2012.

[4] Intel, "Intel 64 and IA-32 Architecture Software Developer's Manual Volume 3B : Systems Programming Guide", 2012.