

멀티코어 상의 AUTOSAR 플랫폼을 활용한 차량용 LDW 응용 서비스 개발

Development of Vehicle LDW Application Service using AUTOSAR Platform on Multi-Core MCU

박미룡*, 김동원**

Mi-ryong Park*, Dongwon Kim**

요약 본 논문은 최근 각광을 받고 있는 차선 이탈 알람 서비스인 LDW(Lane Departure Warning)와 같은 안정성 서비스를 제공하기 위하여, 비대칭 멀티코어 플랫폼을 구성한다. 멀티코어 플랫폼은 고속 영상처리를 담당하는 고속 영상 MCU(Micro Controller Unit) 코어와 안정적인 제어를 요하는 곳에 저속 제어 MCU코어를 사용하는 멀티코어 H/W 플랫폼상에 AUTOSAR S/W 플랫폼을 포팅하고, AUTOSAR 개발방법론에 따른 MBD(Model Based Development) 기반 모델을 활용하여 LDW 소프트웨어 컴포넌트(SW-C)를 설계하고 동작을 검증한다. 또한 고속 영상 MCU와 저속 제어 MCU간에는 가상화 기법을 사용하지 않고 타이머 기반 공유 메모리를 이용한 폴링 기법의 IPC(Inter Processor Communication) 기능을 개발하고, 외부 타 ECU(Electronic Control Unit)와의 CAN 통신기능을 개발하여 알람 신호, 차량 시뮬레이션 신호와 같은 제어 신호 송수신을 처리할 수 있도록 AUTOSAR S/W 플랫폼을 적용한다. 본 연구를 통하여 고속 및 저속 비대칭 멀티코어상에 AUTOSAR가 탑재된 ECU 기능 개발이 가능함을 확인함으로써, ADAS(Advanced Driver Assistance System)와 같은 다양한 응용 서비스들을 제공할 수 있게 되며, ISO 26262로 대변되는 차량 기능안전성 확보가 가능하게 된다.

Abstract In this paper, we examine Asymmetric Multi-Processing Environment to provide LDW service. Asymmetric Multi-Processing Environment consists of high-speed MCU to support rapid image processing and low-speed MCU for controlling with other ECU at the control domain. Also we designed rapid image process application and LDW application Software Component(SW-C) according to the development process rule of AUTOSAR. To communicate between two MCUs, timer based polling based IPC was designed. Also to communicate with other ECUs(Electronic Control Units), we designed CAN messages to provide alarm information and receiving CAN message to catch the Turn signal. We confirm the possibility of the various ADAS development using an Asymmetric Multi-Processing Environment and AUTOSAR platform. We also expect providing ISO 26262 functional safety.

Key Words : AUTOSAR, MBD, Asymmetric Multi-Core, LDWS, Functional Safety

*정희원, 한국전자통신연구원 자동차IT플랫폼연구실

**정희원, 충북도립대학 전자통신전공

접수일: 2014년 6월 2일, 수정완료일: 2014년 7월 2일

게재확정일: 2014년 8월 8일

Received: 2 June, 2014 / Revised: 2 July, 2014

Accepted: 8 August, 2014

**Corresponding Author: won@cpu.ac.kr

Dept. of Electronics Communication, Chungbuk Prov. Univ., Korea

I. 서 론

최근 컴퓨터 기술의 급속한 발전으로 인해 기존의 텍스트 위주의 사용자 환경에서 벗어나 이미지, 그래픽, 오디오 및 비디오 데이터 등을 제공하는 멀티미디어 사용자 환경으로 변화하고 있다. 최근 들어 자동차 응용 서비스 개발 과정에서 응용 서비스 복잡도 증가에 따라 발생하는 문제점을 최소화하기 위해 차량에서 운전자나 탑승자의 미칠 위험 정도에 따라 서비스의 각 기능별 안정성 검사를 제공할 수 있도록 ISO-26262 국제표준 규격을 마련하였다[4]. 기능 안정성 표준규격에 따라 개발 프로세스를 가져가는 것은 향후 응용 서비스에서 발생할 문제에 대한 대비책 마련에서 시작될 뿐만 아니라 현존하는 최신 기술을 적용하였다는 이력을 바탕으로 리콜 등의 사태에 대해 합리적으로 대응할 수 있도록 하고 있다.

이에 따라 업체별 응용 서비스 개발과정에서 ISO-26262의 기능 안정성 규격에 준수하여 개발할 때 안정적인 모델을 재사용할 수 있도록 하기 위하여 MBD(Model based Development) 기법을 적용하면서 전자장치에 적용할 SW 플랫폼을 적용하여 재사용 가능한 응용 소프트웨어 컴포넌트를 개발하는 과정을 준수하고 있다. 자동차 분야의 전자장치용 소프트웨어 플랫폼은 10여년 이상 산업계 표준과정을 거쳐 AUTOSAR 소프트웨어 플랫폼이 형성되었으며, 기능 안정성의 소프트웨어 개발에 필수 불가결한 요소가 AUTOSAR 플랫폼이다. AUTOSAR 플랫폼을 적용함에 있어서 기능 안정성을 만족하기 위하여 최신 멀티코어 지원과 AUTOSAR 표준 규격 4.0 이상을 제시하고 있다^{[1-4][13-16]}.

AUTOSAR 4.0 이상의 규격에서는 기능안정성을 제공할 수 있도록 기술 리포트를 마련하고 있는데 주요한 표준의 항목은 멀티코어 MCU 지원, 메모리 보호, API 흐름 모니터링, 태스크 간의 타이밍 보호, 태스크 간 동작 영역 보호를 위한 메모리 파티션, 종단간 통신 보호 기법, 사용자 모드와 수퍼바이저 모드 보호모드 동작 등을 제시하고 있으며, 특히 멀티코어 상에서 응용 서비스간의 보호와 통신 기능에 대한 규격을 강화하고 있다^{[5-7][13-16]}.

차량의 응용 서비스들은 편의와 안전 응용들이 기하급수적으로 증가됨에 따라 최근 들어 최고급 차량에는 100여개의 응용 서비스를 지원하는 ECU(Electronic Control Unit) 들이 장착되고 있다. 또한 중형급 차량에서 조차도 수십 개의 ECU가 분산 배치되어 있어 이들의 모

듈화가 진행되고 있을 뿐만 아니라 멀티코어를 이용하여 다량의 분산된 응용 서비스들을 한 개의 다중 코어 MCU에 동작될 수 있도록 DCU(Domain Control Unit)이라는 형태의 집중형 ECU의 연구 개발, 안전과 관련된 분야의 응용 서비스를 위한 기능안전성 연구가 진행되고 있다. 다중 응용 서비스들을 제공할 수 있는 멀티코어 MCU의 적용은 차량 모델 개발에서 ECU 개수를 줄여 배선의 감소, ECU의 스페이스 감소와 함께 완성차의 조립라인을 단순화 시킬 수 있는 장점을 제공하고 있어 유럽의 완성차 업체들과 Europa FP7등의 연구가 진행되고 있다^{[3][8-10]}.

이러한 기능안전성 규격과 멀티코어 적용 및 안전/편의 등의 응용 서비스의 개발에 대한 시도들이 완성차 업체 및 연구자들을 중심으로 조심스럽게 진행되고 있으며, 기능 안전성 규격에 따라 사고가 발생되지 않도록 하는 시도들과 사고가 발생하더라도 안전 목표치를 벗어나지 않도록 하는 시도들이 이루어지고 있다.

이에 본 연구에서는 영상 처리용 고속 MCU나 DSP 보드와 제어 처리를 위한 제어용 MCU로 분리하여 사용하던 환경을 고려하여 단일 MCU에 다중 코어를 적용하여 고속 영상처리와 전자제어 장치를 동시에 처리할 수 있도록 가상화 기술을 사용하지 않고 두 코어 간 내부 IPC 체계를 수립하고, 상호 교환 메시지를 정의하여 전자제어 장치에 알람 이벤트를 전송할 수 있도록 AUTOSAR 기반 응용 서비스를 개발하였다. 이를 통하여 단일 MCU상에서 영상처리와 제어처리를 동시에 처리할 수 있도록 하였으며, 향후 ISO-26262 차량 기능 안전성 규격을 만족할 수 있도록 AUTOSAR 기반 응용 소프트웨어를 표준 규격에 따라 프로세서와 응용 서비스의 모델을 설계 및 검증하였다.

2장 본론에서는 다중코어 기반 기능 안전성을 제공할 기초 플랫폼으로 사용되는 AUTOSAR 기반 응용 서비스 설계 과정과 개발 과정에서 발생하는 문제점 들을 살펴보고, 3장에서는 개발된 서비스의 동작시험을 위한 참조 보드와 개발된 응용 서비스의 실험보드 상에서 시험을 위한 SIL(Software In-the-Loop) 환경과 SIL 시험 및 그 결과들을 살펴보고, 4장에서 결론을 맺는다.

II. 차량용 LDW 응용 서비스 개발

1. 멀티코어 참조 플랫폼 환경 구성

차량에 사용하는 멀티코어 MCU 개발과정에 필수적으로 FPGA 기반 설계 IP들을 검증 과정을 거친다. 현재 사용한 검증 플랫폼으로 업계에서 지속적으로 검증에서 사용하는 FPGA를 기반으로 참조용 보드를 구성하며, FPGA 내부에는 다양한 IP코어들이 제작되어 동작된다. 개발용 참조 플랫폼에서는 클락의 설정에 따라 동작환경을 달리 설정할 수 있게 되며, 영상용 코어에는 빠른 영상처리가 요구되며, 제어용 코어에서는 안정성을 기본적으로 제공하도록 설계 된다. 코어들이 내부 버스를 통해 상호 연결되어 있고, 다양한 IP 들은 설정에 따라 어느 코어에나 연동되어 동작할 수 있다. 또한 외부 연동장치(Peripheral) 들도 설정에 따라 코어와 연결된 IP들과 연결하여 사용될 수 있다^[10].

영상코어에는 빠른 영상처리를 처리하기 위한 주변장치가 연결되며, 제어코어에서는 저속 제어 처리를 위한 주변장치가 연동된다. 그림1의 개념모델에서는 비디오 입력과 같은 다양한 기능을 위한 USB나 이더넷 외부 연동 인터페이스와 I2X, SPI 등의 통신 인터페이스를 제공한다. 또한 디버깅이나 저속 통신을 위한 UART, CAN 인터페이스도 제공한다. 코어들간의 연동이나 주변장치의 연동을 위한 내부 버스가 연결되며, 버스 사용권을 제어하기 위한 제어가 동작된다. 다양한 외부 장치들이나 내부의 인터럽터를 처리하기 위한 인터럽터 컨트롤러와 내외부 메모리 제어기, 플래시 메모리 처리, 실시간 클럭, 빠른 메모리 접근과 이동을 위한 DMA 기능들을 포함하고 있다.

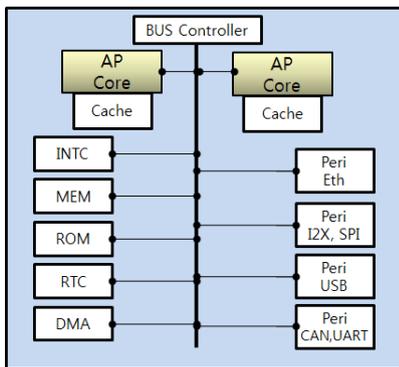


그림 1. 멀티코어 개념 모델
 Fig. 1. A Conceptual model of Multi-core

2. 제어기 AUTOSAR SW 플랫폼 적용

차량의 제어기에 적용할 수 있는 산업계 표준 플랫폼은 AUTOSAR SW 플랫폼으로 AUTOSAR 표준의 경우 차량 응용 서비스 개발 프로세스로부터, HW와 연관성이 높은 소프트웨어 플랫폼, 응용 서비스용 API 등 다양한 영역에 표준을 구축했다.

그림 2에서 보는 것처럼 AUTOSAR는 MCU에 의존적인 HW 장치 관련 MCAL(MCU Abstraction Layer)과 ECU 보드별 특성을 표준화한 EAL(ECU Abstraction Layer) 계층, 그리고 서비스를 제공하는 계층(OS, COM, MEM 등) 및 이들을 상위 응용 소프트웨어 컴포넌트(SW-C)과 디커플링 시켜주기 위한 런타임 환경(AUTOSAR RTE)으로 구성되어 있다. 특히 MCU 의존적인 MCAL 계층은 MCU의 변경이 전체 시스템에 미치는 영향을 최소화 할 수 있도록 한 계층이며, EAL 계층은 ECU의 변경에도 응용 서비스 SW-C의 수정이 없이 동작될 수 있도록 한 계층이다. AUTOSAR 응용 서비스의 개발을 위하여 모델 기반 개발도구를 활용하게 되며, ETRI에서 개발한 AutoWorks 도구^[11]를 활용하여 응용 서비스를 개발한다.

AutoWorks의 경우 개발도구와 플랫폼을 모두 포함되어 있으며, 기존 다른 프로세서에 개발된 계층구조 플랫폼 가운데 MCAL을 멀티코어 환경의 새로운 MCU에 맞추어 수정 포팅 한다. 또한 AutoWorks의 멀티코어 적용을 위해 GPT(General Purpose Timer)를 포함한 AUTOSAR OS 부분 및 차량 통신을 위한 CAN 통신 드라이버의 수정 포팅이 되었고, 기존 MCU에서 CAN 통신을 위해 사용하던 SPI/PORT 등이 수정되었으며, 코어간 공유메모리를 통한 IPC 통신을 위한 LaneDepartCDD 기능이 추가 개발 되었다.

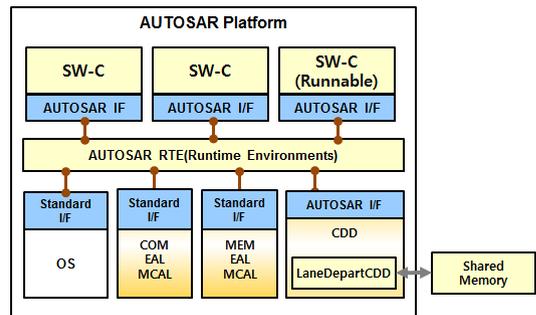


그림 2. AUTOSAR 적용 계층 및 모듈
 Fig. 2. AUTOSAR apaptation layer and modules

3. LDW 응용 서비스 설계

AUTOSAR 응용 서비스 설계는 시스템 구조설립 단계, 내부동작 설정단계, 하드웨어 토폴로지 설정단계, 네트워크 설정단계, 시스템 매핑 단계를 거쳐 이루어진다.

첫째, 시스템 구조설계 단계에서는 시스템의 요구사항에 맞게 소프트웨어 컴포넌트로 분리시키고, 이들 간 연관관계를 설계하는 단계이다. 그림 3은 AUTOSAR 응용 서비스 개발방법론에 따른 시스템 구조설계 그림이다. 시스템 구조설계에서는 소프트웨어 컴포넌트들을 정의하고 그들 간 연결을 위한 포트-인터페이스-신호메타데이터의 세부사항들을 설계한다. TurnSigSrc는 방향전환 CAN 신호 메시지를 수신하기 위한 소프트웨어 컴포넌트(SW-C)이고, InterDomain은 차량별 다양한 CAN신호를 알람매니저에서 사용할 수 있도록 처리하는 SW-C이며, LaneDeSrc에서는 영상코어의 신호를 수신하는 SW-C이다. AlarmManager에서는 방향지시 신호(TurnDirection)와 차선이탈 신호(LaneDepart)를 바탕으로 알람상태(AlarmStatus)를 처리하는 SW-C이며, AlarmAct는 차량별 다양한 알람장치 연동에 따라 이를 처리하기 위한 SW-C이다.

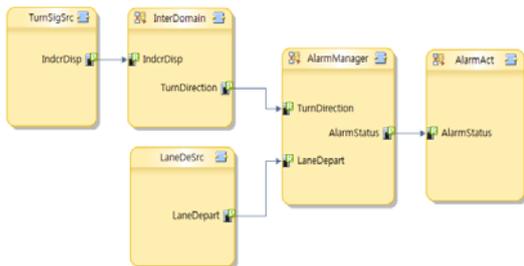


그림 3. LDW 응용 서비스 구조 설계
Fig. 3. A design of LDW application service structure

둘째, 내부 동작 설정단계는 각 소프트웨어 컴포넌트들의 RTE(Runtime Environment) 동작을 표현하기 위한 과정으로 데이터 수신에서 발생하는 이벤트 설정, 데이터 수신/송신 설정, 주기적인 타이머 설정, 내부 변수값 설정 등을 처리한다. 내부 동작 설정단계에서 설계된 값들은 실제 RTE의 코드 생성에 영향을 미치게 된다. 일반적인 설계 도구에서는 내부동작 설정을 처리하고 모델 도구로 연동하여 내부 동작 설계를 하며, 이를 통하여 MIL(Model in-the-Loop) 시험을 진행하게 된다.

셋째, 하드웨어 토폴로지 설정단계에서는 설계된

SW-C들을 어떠한 ECU에 CAN 클러스터로 연동할지 설계하는 단계이다. 아래 그림 4에서 보여주는 LDW 응용을 위한 하드웨어 토폴로지에서는 방향지시 ECU(TSS)와 차선이탈 ECU(LDS) 및 차선이탈 애플레이터 외부 가상 ECU(LES)들이 CAN 클러스터를 통하여 상호 연동되도록 설계한다.

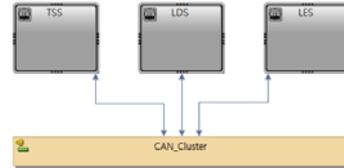


그림 4. LDW 하드웨어 토폴로지 설계
Fig. 4. Hardware topology design of LDW application

넷째, 네트워크 설정단계는 시스템 구조설계 단계에서 설계된 인터페이스 및 내부 신호들의 시그널, PDU 및 프레임으로 매핑하는 과정이다. 최종적으로 시스템 매핑단계에서는 개발된 SW-C들을 하드웨어 토폴로지서 설정한 ECU로 매핑 한다.

4. SW-C 내부 동작 설계

대부분의 AUTOSAR 설계도구들은 도구간 상호 연동성을 제공하고 있다. 내부 동작 설계단계에 설계된 ARXML(AUTOSAR XML) 파일은 내부 동작을 개발하기 위한 원형만 제공한다. 내부 동작을 개발하기 위하여 MBD 툴을 활용한다.

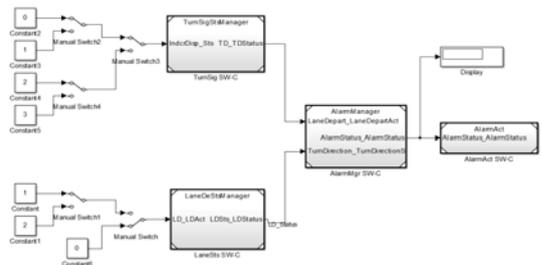
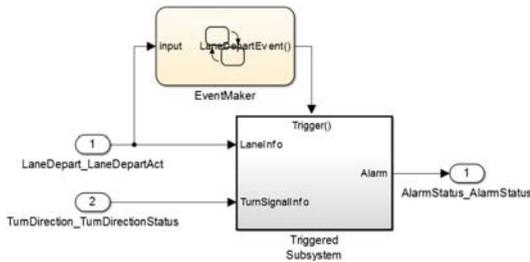


그림 5. SW-C IB(Internal Behaviour) 설계
Fig. 5. A design of SW-C internal behaviour

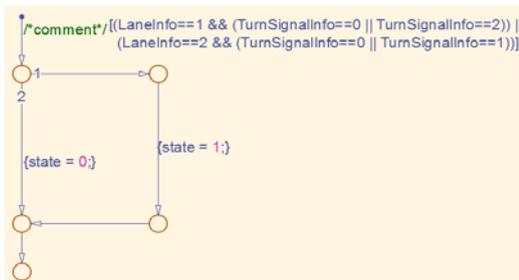
모델기반 설계 툴 가운데 내부 동작에 대하여서는 이미 잘 개발된 Matlab/Simulink 등의 툴을 인터페이스 하여 사용하게 되며, AutoWorks에서도 모델의 내부 동작 설계를 Simulink와 연동하기 위해 Matlab의 AUTOSAR

모델 호환을 위한 arxml export/import 과정을 거치며, Simulink로 arxml을 import 하는 기능이 포함되어 있다. 그림 5는 그림 3에서 설계된 응용 서비스 구조의 내부 동작 설계 결과 arxml을 MBD 툴에서 불러와 설계된 모델 구조를 보여주고 있다.

모델의 내부동작은 상태 머신을 통하여 개발할 수도 있고, 기존 모델들의 조합으로 개발할 수 있으며, 진리 테이블을 통하여서도 개발할 수 있다. 다양한 방법을 적용하여 제작된 내부 모델에 대한 동작 검증하게 되며, 검증된 모델을 코드 생성기를 거쳐 사용자가 사용할 수 있는 코드로 만들게 된다.



(a) AlarmManager SW-C 모델



(b) Triggered Subsystem 내부 상태 흐름도

그림 6. AlarmManager SW-C 모델 설계 개략도
 Fig. 6. A brief designed model of AlarmManager SW-C

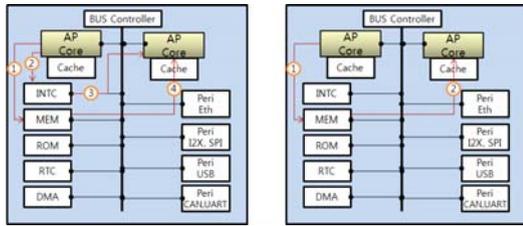
그림 6에서는 LDW 서비스의 핵심 기능인 AlarmManager SW-C의 내부 시뮬링크 모델 동작 설계 예를 보여주고 있다. AlarmManager는 EventMaker와 Triggered Subsystem으로 구성된다. 그림6-(a)는 EventMaker는 LaneDeSrc SW-C로부터 차선이탈 정보(LaneDepart)에 따라 차선이탈 이벤트(LaneDepartEvent) 신호를 만들고, 차선이탈 이벤트(LaneDepartEvent) 신호가 발생한 경우에 Triggered Subsystem은 차선이탈 정

보(LaneInfo)와 방향지시등 정보(TurnSignalInfo)를 가지고 알람(Alarm) 상태를 판단하게 된다. 그림6-(b)는 트리거 발생 시 차선이탈 정보(LaneInfo)와 방향지시등 정보(TurnSignalInfo)의 상태조합으로 알람 상태(State)를 계산하고 이를 외부 AlarmStatus 신호로 출력하는 Triggered Subsystem 내부 상태흐름도를 보여주고 있다.

5. 코어 간 IPC 설계

대부분의 AUTOSAR 설계도구들은 도구간 상호 연동성을 제공하고 있다. 내부 동작 설계단계에 설계된 ARXML(AUTOSAR XML) 파일은 내부 동작을 개발하기 위한 원형만 제공한다. 내부 동작을 개발하기 위하여 MBD 툴을 활용한다. 고속 코어와 제어 코어 간에 영상 처리 후 감지된 차선 이탈정보와 같은 상호 제어메시지를 전송하기 위하여 IPC 메커니즘을 채택한다.

그림 7에서는 인터럽트 기반 IPC 및 타이머기반 폴링 IPC 방식을 멀티코어 개념 모델을 통해 보여주고 있다. 그림 7-(a)에서 인터럽트 기반 IPC는 어떤 한 코어에서 ①공유메모리에 메시지를 써 놓고, ②인터럽트를 걸어주면, ③다른 코어에서 인터럽트를 감지하고, ④공유메모리에서 메시지를 읽어내어 처리하는 방식이다. 본 연구에서는 그림 7-(b)의 타이머 기반 공유 메모리를 이용한 폴링 IPC 기법을 사용한다. 이 기법은 기존 대부분 차량 응용 서비스에서 이용되는 CAN을 통한 메시지 송수신에서 사용되는 방식과 유사한 방식으로 현재의 상태정보가 최신의 정보라고 상호 판단하고 ①송신측에서는 주어진 타이머 주기 내에 현재의 상태 정보를 전송하게 되고, ②수신측에서는 마지막으로 수신된 메시지가 최신 메시지라 판단하고 읽어 처리하는 방식이다. 제어 코어의 AUTOSAR 하부에서는 공유메모리 IPC 기능이 기존 표준화된 계층이나 모듈을 통해 지원하지 않으므로 이를 AUTOSAR CDD로 개발하였으며(이를 LaneDepartCDD라고 함), LaneDepartCDD 내부에서는 상호 정의한 공유 메모리 정보를 읽어 AUTOSAR 인터페이스로 매핑하여 RTE에서 사용할 수 있도록 한다. AUTOSAR CDD는 주기적인 동작을 위하여 OS 스케줄러 테이블에 등록이 되며, OS는 타이머 동작에 따라 스케줄러 테이블에 등록된 CDD_Lane_Check()를 동작시켜 영상 코어에서 전송하는 차선이탈 정보를 수신한다.



(a) 인터럽트기반 IPC (b) 타이머기반 폴링 IPC

그림 7. 멀티코어간 메시지 교환 방식

Fig. 7. Message exchange methods between Multi-core

III. 개발 및 시험 환경

모든 설계가 완성되면 AutoWorks 코드 생성기를 통하여 코드를 자동으로 생성하고 설계과정에서 변하지 않는 코어코드와 설계를 통해 변경되는 설정정보 코드 파일(c, h)들을 모두 통합하여 컴파일하며 OS, MCAL, EAL, RTE 및 응용 SW-C들을 통합하게 된다. 이때 외부 MBD툴을 통해 개발된 코드들도 C 코드 생성기를 통하여 모델을 코드로 변환한다.

고속 영상처리를 위한 영상 코어에는 PoC(Proof of Concept) 확인을 위하여 FW(Firmware) 기반 영상처리 소프트웨어를 개발하여 탑재하며, 제어를 위한 제어코어에는 AUTOSAR 기반 개발방법론에 따라 설계 개발된 LDW SW-C 내부 모델들을 코드 생성기들을 통하여 생성하게 되고, 기 제작된 BSW(Basic SW)의 C 코드와 결합한다. 통합 과정에서 각 모듈별 컴파일 및 링크 시험을 거쳐 에러가 없는 환경을 만들고 이를 타겟 보드에 올려 검증한다. 전체 타겟 빌드 환경에는 임베디드용 크로스 컴파일러로 gnu의 gcc를 사용하였으며, gnu의 MAKE를 활용하여 통합 빌드 환경을 구축하고 컴파일을 하여 통합 이미지를 제작한다.

각 개발 과정 중 설계 변경에 따라 변경된 부분만 코드 생성기를 통하여 수정된 코드를 통합 빌드 환경에 적용하여 재 컴파일하여 새롭게 생성한 바이너리 이미지를 타겟 보드에 동작시켜 검증한다.

현재 참조 보드에서는 LCD 가 부착되어 있어 영상처리 결과를 디스플레이를 통하여 확인할 수 있으며, 외부 타 ECU들과의 연동을 위하여 CAN 인터페이스를 제공하고 있어 USB-CAN 소프트웨어나, CANoe 등의 장비를 활용하여 CAN 메시지의 송수신 상태를 디버깅할 수

있다.

시험을 위하여 노트북의 개발환경에서 AUTOSAR 개발도구를 통한 SW-C 설계, ECU 설정, C 코드 생성 등을 처리하고, Matlab/Simulink를 활용하여 SW-C의 내부 동작을 모델링하고 임베디드 코드를 이용한 내부동작을 C 코드로 생성하여 RTE코드 생성기에서 생성된 코드와 결합하여 통합 적용하며, 통합 빌드 환경에서 AUTOSAR의 코어코드, ECU 설정에 따른 생성코드, RTE 생성코드, 외부 MBD툴에서 생성된 SW-C 내부동작 코드 등의 전체 코드들을 컴파일 하여 통합된 바이너리 파일을 만든다.

컴파일 된 바이너리 이미지는 FPGA기반 타겟 참조 보드에 적재하기 위하여 USB를 통한 이미지 다운로드를 수행되며, USB 시리얼을 통한 이미지의 동작 상태를 모니터링하고, CANoe나 USB-CAN 장치를 통하여 CAN 메시지를 송수신 시험하여 동작을 검증한다.

그림 8에서는 전체 LDW 응용 서비스를 시험할 수 있는 통합 시험환경을 보여주고 있다. 통합 시험환경의 좌측 노트북에서는 AutoWorks를 통한 디자인, 코드 생성, 전체 컴파일, 이미지 생성, 이미지 다운로드, USB-UART를 통한 동작 모니터링이 이루어지고, 우측 노트북에서는 USB-CAN을 통한 CAN 제어 메시지 전송 및 수신 등을 처리하며, 중앙에 멀티코어 참조 플랫폼을 보여주고 있다. 특히 우측 노트북의 USB-CAN 알람 소프트웨어는 외부 알람 ECU를 대신하여 수신한 CAN 알람 메시지를 바탕으로 Windows용 비프 동작으로 알람을 시험하였다. USB-CAN 알람 소프트웨어는 USB-CAN SDK를 활용하며 주기적으로 전달되는 메시지의 포맷을 상호 정의하고 알람 온(1) 상태의 메시지를 수신한 경우 일정시간 비프 알람을 제공하도록 동작됨을 확인하였다.



그림 8. 응용 서비스 통합 시험환경

Fig. 8. Test environment of application service

IV. 결론

본 논문에서는 지능형 차량의 영상기반 응용서비스와 같이 고속 영상처리와 제어가 동시에 필요한 차량 제어 기에서 멀티코어를 활용하여 고속 영상처리와 제어기용 SW-C를 분리하여 개발함으로 저속 코어에 동작되는 AUTOSAR SW-C의 검증만으로도 향후 ISO-26262에서 요구되는 기능 안정성을 만족시킬 수 있도록 설계, 개발 방법을 제공한다.

기존 영상처리 DSP와 제어기용 LDW 응용 서비스를 개발하기 위해 두 개의 독립적인 ECU 보드를 만들어 접속하거나 두개의 MCU를 결합하여 하나의 ECU 보드를 만드는 방식을 사용하던 것에 비하여 멀티코어 MCU를 사용함으로 ECU 보드 스페이스를 감소시켜 차량 설계 시 부품들의 배치가 쉬워지고, 두 칩 혹은 두개의 보드를 결합하여 개발하는 것에 비하여 선로 라우팅 감소로 안정성을 확보할 수 있으며, 칩이나 보드의 결합의 배선 라우팅 시 발생하는 EMI/EMC 문제들을 감소시킬 수 있는 장점을 제공할 수 있다.

AUTOSAR 개발 프로세스에 준수하는 개발도구와 SW 플랫폼을 적용한 AUTOSAR 기반 LDW 응용 서비스의 개발과정을 통하여 새롭게 검증하는 다중코어 MCU위에서도 짧은 시간에 OS와 MCU의 하드웨어 관련 추상 계층의 드라이버만의 개발로도 기존 ECU 추상화 계층의 재사용, 서비스 계층 및 RTE(Runtime Environment)의 재사용을 통하여 응용 서비스의 재활용이 가능하게 되었다.

ISO 26262 기능안정성 규격을 만족시키기 위하여 멀티코어 기반 MCU들을 활용하는 단계에서 본 연구를 통하여 영상처리와 제어 처리를 분리시켜봄으로 제어처리를 단순화 시켜 기능 안정성 규격 만족이 간단하게 될 수 있음을 볼 수 있다.

향후 실제적인 응용서비스 개발에서 ISO 26262 프로세스의 적용과 멀티코어의 가상화 환경을 구축을 통하여 다양한 응용 서비스를 멀티코어 환경에 동시에 적용할 수 있도록 연구를 확장하여야 한다.

References

[1] MiRyong Park, et al., "A SW Convergence Trend between IT Technology and Automotive

Technology," KICS, Vol. 25. No. 6, Pp. 3-11, Jun. 2008.

- [2] MiRyong Park, Jaeyoung Kim, "Automotive Electronics SW Platform(AUTOSAR) Standardization," TTA Journal No. 117, Pp. 89-100, Jun. 2008.
- [3] Brewerton, S., Willey, N., Gandhi, S., Rosenthal, T. et al., "Demonstration of Automotive Steering Column Lock using Multicore AutoSAR® Operating System," SAE Technical Paper 2012-01-0031, Apr. 2012, from <http://papers.sae.org/2012-01-0031/>
- [4] ISO 26262, Road Vehicles- Functional Safety- Part 3: Concept phase, 2011.
- [5] AUTOSAR, AUTOSAR Technical Safety Concept Status Report, from http://www.autosar.org/download/R4.0/AUTOSAR_TR_SafetyConceptStatusReport.pdf
- [6] K. Sung, T. Han, "Development of Automotive Embedded System based on AUTOSAR Methodology," ACIT 2013, ASTL Vol.22, pp. 132-137, Xian, China, May. 2013.
- [7] J. Lee, et al., "A Study on AUTOSAR COM I-PDU Transmission Mode," SIA 2013, ASTL Vol. 22, pp. 86-91, Xian, China, May 2013.
- [8] RECOMP project (Reduced Certification of Multicore Processors), <http://atc.ugr.es/recomp/>
- [9] C. Aussagues, et al., "PharOS, A Multicore OS ready for safety-related automotive systems: results and future prospects." In Proc. of The Embedded Real-Time Software and Systems (ERTS2), Toulouse, France, May 2010.
- [10] Mi-Ryong Park, Sang-Ha Kim, Dongwon Kim, "Traffic Information Service System Using Gateway between VANET and Infrastructure Network," VOL. 12 No. 5, pp.61~66, October 2012
- [11] Automotive ECU SW Solution 'AUTOWORKS' http://m.etri.re.kr/html/not/not_01020100.jsp?no=11539&pageNum=8, Jan. 2010.
- [12] MiRyong Park, et al., "Traffic Information Service System Using Gateway between VANET and Infrastructure Network," The Journal of The Institute of Webcasting, Internet and

Telecommunication VOL. 12 No. 5

- [13] Stefana N., et al., "Combining Model-based Analysis and Testing," ERTS2014, Feb. 2014.
- [14] Bucaioni A., et al., "From Modeling to Deployment of Component-Based Vehicular Distributed Real-Time Systems," IEEE International Conference on Information Technology, Apr. 2014.
- [15] Longfei, Z., et al., "A Timing Verification Framework for AUTOSAR OS Component Development based on Real-Time Maude," IEEE International Symposium on Theoretical Aspects of Software Engineering, Dec. 2013.
- [16] Jongbok Lee., "Performance Study of Asymmetric Multicore Processor Architectures," The Journal of The Institute of Internet, Broadcasting and Communication(JIIBC), Vol. 14, No. 3, pp. 163-169, 2014.

저자 소개

박 미 룡(정회원)



- 1998년 2월 : 경북대학교 전자공학 석사 졸업
- 2013년 2월 : 충남대학교 컴퓨터공학과 박사 수료
- 1999년 4월 ~ : 한국전자통신연구원 책임연구원
- 관심분야 : 임베디드 Linux, AUTOSAR, GENIVI, VANET, 차량 IVN

• e-mail : mrpark@etri.re.kr

김 동 원(정회원)



- 1983년 2월 : 경북대학교 전자공학과 졸업
- 1990년 2월 : 경북대학교 전자공학과 석사 졸업
- 1998년 2월 : 충북대학교 전자공학과 박사
- 1988년 3월 ~ 1998년 2월 : 한국전자통신연구원

• 1998년 3월 ~ : 충북도립대학 전자통신전공 교수
 • 관심분야 : USN, VANET, 차량IT, 임베디드시스템
 • e-mail : won@cpu.ac.kr