

FOTA 기반 지능형 자동차를 위한 범용 ECU 시뮬레이터 설계 및 구현

The Design and Implementation of ECU Simulator for the Smart Vehicle based on FOTA

박인혜¹ · 고재진¹ · 곽재민^{2*}

¹전자부품연구원 소프트웨어디바이스연구센터

²목포해양대학교 해양전자통신공학부

In-Hye Park¹ · Jae-Jin Ko¹ · Jae-Min Kwak^{2*}

¹Software Device Research Center, KETI, Seongnam, Gyeonggi-do, 463-490, Korea

²*Division of Marine Electronics & Communication Engineering, Mokpo National Maritime University, Mokpo, Jeollanam-do, 530-729, Korea

[요 약]

본 논문은 FOTA 기반의 지능형 자동차를 위한 범용 ECU 시뮬레이터의 설계 및 개발에 대해 서술한다. 특정 목적용이 아닌 범용적인 ECU 시뮬레이터를 하드웨어와 소프트웨어 플랫폼 두 부분으로 나눠 설계 및 개발 한다. 하드웨어는 32bit Atmel AVR의 코어를 장착하고 CAN/LIN 인터페이스 그리고 사용자를 위한 LCD화면과 터치 버튼을 설계하여 범용적인 지능형 자동차용 ECU의 조건을 만족시켰다. 소프트웨어는 설계/개발된 하드웨어의 모든 기능들을 지원하면서도 데이터를 효율적으로 처리하기 위하여 FSM의 구조로 개발했다. 설명된 방법으로 개발된 ECU 시뮬레이터의 유효성 검증을 위해 자동 업데이트 시스템의 데이터를 전달해주는 ECU 관리자 추가 개발하여 테스트를 진행했다. 실험을 위하여 ECU와 DM간 상호 통신이 가능한 총 4가지의 시나리오를 지정해 업데이트 과정을 포함한 테스트를 진행했고, 결과적으로 모든 시나리오에서 ECU 시뮬레이터로서 동작의 유효성을 확인할 수 있었다.

[Abstract]

This paper presents the design and development of ECU simulator in order to verify automatic firmware update system of ECU SW on the vehicle. We designed and developed general-proposed ECU simulator as divide into HW and SW parts. HW adopt 32bit MPU, CAN, LIN, LCD, and touch-pad button to satisfy general-proposed ECU level. And SW constructed as finite-state-machine structure in order to support HW spec. We developed ECU manager which deliveries update data to verify validity of operation. And we tested ECU simulator with ECU manager. For test of validity of ECU simulator, we set 4 scenarios between ECU simulator and manager. As the result, we confirmed validity of operation in ECU simulator for automatic SW update system.

Key word : ECU, ECU simulator, Firmware update, FOTA, Automatic update system

<http://dx.doi.org/10.12673/jkoni.2014.18.1.22>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 17 January 2014; Revised 20 February 2014

Accepted 16 February 2014

*Corresponding Author; Jae-Min Kwak

Tel: +82-61-240-7268

E-mail: kjm@mmu.ac.kr

I. 서론

자동차 산업과 IT산업의 융합이 확대됨에 따라 스마트 폰과 스마트 TV의 시장과 더불어 지능형 자동차 시장의 규모가 점차 커지고 있다. 지능형 자동차는 차량 내 운전자 편의를 고려한 능동 제어 장치를 도입하여 필요시 운전자에게 경고를 해주거나 직접 위험 상황을 방지하는 등의 기능을 탑재한다 [1]. 이러한 지능형 자동차에 가장 핵심적인 요소 중 한 가지가 ECU(electronic control unit; 전자제어장치)이다.

최근의 자동차는 25개에서 71개의 ECU를 탑재할 정도로 ECU가 지능형 자동차의 핵심 부분으로 자리하고 있다. 특히 지능형 자동차의 경우 70개에서 80개의 ECU가 적용되고 있어, 현재 자동차의 전장품 비율이 전체 자동차 부품의 상당 부분을 차지하는 상황임이 분명하다[2]. 또한 앞으로 자동차의 지능화에 따라 자동차의 주요부분을 다루는 ECU 사용의 확대는 더욱 가속화될 것으로 보인다.

ECU의 사용 확대와 함께 지능형 자동차에 대한 사용자들의 인식과 요구 또한 날로 증가되고 있다. 사용자의 다양하고 복잡한 요구사항을 충족하기 위해서는 소프트웨어(software)의 중요성이 매우 대두되고 있다. 특히나 플랫폼에 상관없이 손쉬운 동작과 빠른 업데이트가 가능한 소프트웨어에 대한 요구사항들에 초점이 맞춰지고 있다. 이러한 요구사항의 만족을 위한 소프트웨어의 중심에는 웹(web)이 있다. 이러한 변화는 네트워크의 안정적인 관리에 맞춰져 있던 서비스 제공 사업자들의 전통적인 관심을 DM(device management; 장치관리)으로 변화시키는 계기가 되었다. 이에 따라 DM의 기능은 펌웨어 관리, 자동개통, 원격진단 및 제어 뿐 아니라 사용자의 편의를 위한 서비스 관리, CRM(customer relationship management; 고객 관계 관리)으로 확대됐다[3],[4].

DM의 확대된 기능 중 차량내 ECU를 사용자 편의에 맞춰 빠르고 안정적으로 관리하는 것이 중요하게 여겨지고 있다. 따라서 DM을 이용해 표준 플러그인 확장을 통하여 상용화 및 멀티 플랫폼 지원에 유리한 인포테인먼트(infotainment) 시스템(즉, AVN; audio, visual and navigation) 및 ECU의 소프트웨어가 FOTA(firmware over the air) 업데이트 중심으로 변화하고 있다.

FOTA는 ECU 소프트웨어와 펌웨어를 원격으로 업데이트 할 수 있는 기술이다. FOTA 업데이트 기술은 무선으로 새로운 소프트웨어를 제공할 수 있기 때문에 운전자가 판매 대리점으로 직접 움직일 필요가 없다. 또한 펌웨어 오류를 수정하기 위한 소프트웨어를 내려받을 수 있을 뿐 아니라 자동차의 품질과 소프트웨어 안정성 개선, AS 비용 절감에 보다 적극적으로 대응할 수 있는 장점이 있다[5].

이러한 추세에 맞춰 FOTA를 이용한 업데이트 시스템 동작의 완전성을 검증해줄 범용적인 ECU 시뮬레이터들의 개발을 통해 업데이트를 검증해볼 필요가 있다. 따라서 본 논문에서는 FOTA를 이용하는 지능형 자동차용 ECU를 타겟으로 한

ECU 시뮬레이터를 설계 및 개발 방법에 대해 설명하고 동작의 유효성을 검증한다.

본 논문의 주요 구성은 다음과 같다. 2장에서 FOTA를 이용한 업데이트를 진행했던 기존 연구에 대해 살펴볼 것이다. 이어 3장에서는 업데이트를 위한 ECU 플랫폼을 하드웨어와 소프트웨어 부분으로 구분하여 설계 및 개발에 대해 서술한다. 다음 4장에서 전용 ECU 관리자를 구현하여 ECU 플랫폼의 성능을 검증하고 마지막 5장에서 결론을 맺는다.

II. 관련 연구

ECU는 과거에 적용된 내장형 시스템보다 더 많은 소프트웨어 모듈을 탑재하고 있으며, 이런 상황은 벤더(vendor)들에게 개발 기간의 증가와 비용의 부담으로 작용한다. 개발 부담을 줄여주고자 논문 [6]에서는 차량용 ECU 소프트웨어 플랫폼을 개발했다. 보통 하드웨어의 성능이 주를 이루기에 논문 [3]과 같이 특정 ECU를 타겟으로 하여 소프트웨어를 개발할 수는 있지만, 이 경우에는 타겟이 된 하드웨어가 아닌 다른 플랫폼에서의 재사용이 힘들고 소프트웨어 자체의 검증이 어렵기 때문에 소프트웨어의 신뢰성도 떨어지게 된다. 따라서 단순 제어가 아닌 소프트웨어 적으로 보다 고급의 조절(예를 들어 네비게이션)이 필요한 경우에는 소프트웨어 플랫폼을 개발하여 신뢰성있는 소프트웨어의 개발이 상용제품 생산의 비용 및 시간적인 측면에서 매우 효율적일 것이다. 따라서 논문 [5]에서 개발된 소프트웨어 플랫폼은 표준화된 개발환경 구축 및 제어기에 적용 가능한 것으로, 논문의 저자들은 차후 자동차에 실제 탑재되기 위함을 목적으로 설계했다고 언급한다. 따라서 설계된 플랫폼의 동작은 실제 ECU 기기에 적용 가능하기 때문에 모듈의 재사용성을 통한 및 소프트웨어의 신뢰성 증가 그리고 개발기간의 단축이라는 장점을 얻을 수 있었다. 이렇듯 범용적인 목적을 가진 ECU 시뮬레이터를 통한 소프트웨어의 검증 과정은 실질 개발품으로의 직접적인 개발의 경우에 비하여 많은 비용과 시간을 줄일 수 있다.

또한 ECU 시뮬레이터 및 플랫폼의 개발과는 별개로 펌웨어를 무선으로 업데이트하는 기술도 많은 연구가 이뤄져있다. 논문 [7]에서는 ECU와 개념이 유사한 단말기기(지능형 전자 계량기)의 원격 펌웨어 업데이트 기술을 하드웨어 메모리 구조를 기반으로 소개하고 있다. 총 세 가지 방법을 소개하고 있으며, 각 방법의 업데이트 시퀀스에 대해 설명하고 있다. 논문 [7]을 통해서 업데이트 데이터의 특성에 따라서 저장되는 메모리 위치를 다르게 지정하는 것이 효율적임을 알 수 있다. 하지만 설명하는 방법은 스마트 미터라는 특수 목적용 기기 전용의 업데이트 데이터이기 때문에 해당 펌웨어 및 업데이트 방법과 그에 따른 메모리 사용은 제한적으로 이용될 수밖에 없다.

논문 [8]에서는 무선 센서 네트워크에서 원격으로 센서 노드들의 펌웨어를 업데이트 하는 방법을 고안 및 직접 개발했

다. 무선 센서 네트워크의 모든 노드들이 주요 업데이트 데이터를 스스로 전달토록 하였으며, 전달받은 데이터를 스스로 업데이트 하도록 개발됐다. 본 논문에서 타겟이 되는 자동 업데이트 시스템과 유사하게 업데이트 데이터 전달 과정 중 무선의 구간이 존재하고 또한 센서 노드(우리는 센서노드 대신 ECU)가 업데이트 데이터를 스스로 전달 및 업데이트 한다는 점이 유사하다. 논문 [8]에서는 전체적으로 사용자 데이터 송수신 구간과 업데이트 데이터 구간 그리고 업데이트를 위한 재부팅 시점을 조정한 업데이트 메커니즘에 대해 설명하고 있다.

논문 [9]에서는 스마트 그리드 구현을 위해 AMI기반으로 원격 펌웨어 업그레이드 시스템과 관련한 연구를 진행했다. 스마트 그리드에 핵심인 지능형 계량기에 요구되는 기능, 프로토콜 및 구조 등을 단계적으로 구분한 FUMS(firmware upgrade management system)를 제안했다. 이를 위해 펌웨어 업그레이드의 전체 과정을 분석하고 외부 서버에서 전달되는 업그레이드 데이터 들을 효과적으로 적용하는 방법에 대해 연구하고 설명하고 있다.

이러한 펌웨어의 원격 업그레이드 자체에 대한 연구 뿐 아니라, 지능형 기기의 원격 검침 및 원격 제어를 목적으로 서버 및 시스템에 대한 연구도 함께 연구됐다. 논문 [10]에서 스마트 워터그리드 네트워크와 같이 사람이 직접 수정할 수 없는 환경에서, 사용되는 단말기가 원격으로 검침 될 수 있도록 암호화 모듈이 구현된 펌웨어의 사용에 관한 연구를 진행했다. 또한 [11]에서는 지능형 임베디드 장치의 원격 제어를 위해 임베디드 웹 서버 구현에 관한 연구를 진행했다. 이러한 연구는 본 논문에서 사용한 외부 서버를 통한 자동 업데이트 시스템의 전체 흐름과 유사한 구조를 갖는다.

III. ECU 시뮬레이터 설계 및 구현

본 장에서는 FOTA 기반의 지능형 자동차를 위한 범용 ECU 시뮬레이터의 설계 및 개발 방법을 하드웨어와 소프트웨어로 나누어 설명한다.

ECU 시뮬레이터의 설명에 앞서, 자동 업데이트 시스템에 대해 먼저 간략히 설명한다. 아래 그림 1에는 본 논문에서 설계 및 개발한 ECU 시뮬레이터의 범위와 타겟이 되는 FOTA를 장착한 지능형 자동차의 대략적인 구성도가 나타나있다. 오른쪽의 서버로부터 무선(WiFi 혹은 3G)을 통해 특정 ECU의 업데이트 데이터를 VDM(virtual data management)으로 전달한다. ECU의 업데이트를 통합적으로 관리하는 VDM는 전달받은 데이터를 자동차 내부의 특정 ECU에게 전달한다. VDM과 ECU는 자동차 안에 탑재되었으며, 상호 CAN(controller area network)으로 연결됐다고 설정했다. 타겟이되는 VDM의 ECU 자동 업데이트 확인을 위해서는 반드시 ECU를 통해야 하고, 보다 다양하고 확장성있는 검증을 위하여 본 논문에서는 실제 ECU가 아닌 ECU 시뮬레이터를 개발했다.

3-1 하드웨어 플랫폼 설계 및 구현

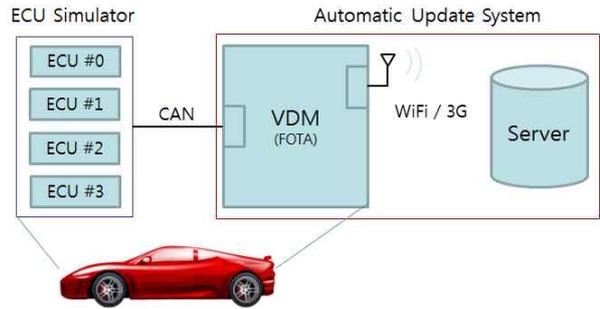


그림 1. 구현된 ECU 시뮬레이터와 자동 업데이트 시스템 관계 개요도

Fig. 1. The concept diagram of ECU simulator and automatic update system for vehicle

설계하려는 ECU는 특정 데이터에 집중하지 않고 범용적인 목적을 갖는다. 따라서 기존에 존재하는 ECU의 기본 요구조건들과 동일한 수준에서 하드웨어의 구조를 갖춰야 한다. 따라서 차량용 ECU로서 인포테인먼트를 조정 가능한 수준으로 설계하기 위하여 다음과 같은 조건으로 하드웨어를 설계했다.

Main processing unit은 32-bit Atmel AVR AT32UC CPU를 사용했고 내부 플래쉬와 램은 각각 512, 256 KB를 채택했다. CAN 통신이 가능하도록 CAN bus를 지원하도록 설계하고 2개의 인터페이스를 생성하여 타 기기(혹은 타 ECU)와의 이중 연결을 가능하게 설계했다. 메모리 영역의 인터페이스로는 EEPROM, DataFlash, SD/MMC, 그리고 SDRAM을 지원하도록 하였다. 마지막으로 사용자에게 지원하는 인터페이스로는, 340*240의 LCD화면, 상하좌우실행 5버튼의 터치버튼, LED(light emitting diode), 마이크, 스피커 단자로 이뤄져 있다. 그 밖에 ECU 시뮬레이터 하드웨어의 상세에 대해 다음 표 1에 나타냈다.

지금까지 설명된 하드웨어 플랫폼의 전체 블록도를 다음 그림 2에 나타냈다. 표 1에 간략히 나타냈던 하드웨어의 구조를 모두 가진 모습이다. 또한 그림 2의 설계대로 제작된 하드

표 1. 하드웨어 플랫폼의 주요 사양

Table 1. The specification of HW platform for ECU simulator

Parameter	Value
Main Processor Unit	32-bit Atmel AT32UC3C
	512KB internal Flash, 256KB internal RAM
	Up to 91 DMIPS Running at 66MHz from Flash
Networking Interface	CAN(2x), LIN(2x)
Debugging Tool	JTAG & NEXUS interface
USER I/O	TWI I/F(6-touch button), EBI I/F(QVGA LCD display)
Support	SD/MMC slot, etc.

웨어 플랫폼이 그림 3에 나타나있다. 그림 3의 보트 상단에 위치한 인터페이스는 CAN이고 가운데 위치한 것이 main processing unit이다. 또한 왼쪽에 위치한 화면이 LCD이고 창 아래에 5개의 터치버튼이 존재함을 확인할 수 있다.

3-2 소프트웨어 플랫폼 설계 및 구현

ECU로서 기능 제공을 위한 하드웨어 플랫폼을 필수적인 부분의 구동을 위한 소프트웨어 플랫폼(즉, firmware)을 설계해야한다. 따라서 우리는 ECU 시뮬레이터 플랫폼으로서의 기능을 만족하게하기 위하여 사용자 인터페이스(LCD; liquid crystal display, 터치버튼)와 CAN 인터페이스의 손쉬운 조정에 기반을 두어 소프트웨어 플랫폼(이하 펌 웨어)을 개발했다.

그림 4에는 설계된 소프트웨어 플랫폼의 프로그램 블록도가 나타나있다. 사용자 인터페이스를 위한 LCD 출력 모듈과 터치 버튼 모듈이 있고, VDM으로부터 데이터를 주고 받기 위한 CAN 모듈이 존재한다. 가장 왼쪽에 위치한 사용자에게 현재 상태를 보여주는(output) LCD모듈은 터치 버튼 모듈(input)의 입력과 VDM으로부터의 데이터 입력에 따라 출력내용을 조금씩 변화 시킨다. 또한 터치 버튼 모듈은 LCD 뿐 아니라 CAN 모듈로 명령을 전달하여 현재 ECU 시뮬레이터가 가진 데이터 정보를 VDM으로 전송토록 한다.

소프트웨어 플랫폼은 시뮬레이터로서 다양한 상황의 동작을 위해 총 4개의 ECU를 가상으로 설계했다. 하드웨어 플랫폼에 구비된 상하좌우 터치버튼을 이용해 가상의 ECU 4개 중 한 가지를 선택할 수 있도록 개발하였으며, 실행 버튼을 누르면

특정 번호의 ECU의 업데이트 버전 및 패킷 수신 상태등의 상세 정보를 볼 수 있도록 내부 동작을 설계했다. 4개의 ECU는 각자 독립적인 개체로 이뤄져있으며 오직 FOTA를 이용한 업데이트 상황을 검증하기 위한 목적으로 구현됐기 때문에 하드웨어에 구애받지 않는 범용적인 특성을 갖는다.

앞서 설명된 세 가지 주요 모듈인 CAN, LCD, 터치버튼은 4개의 ECU를 수월하게 표현하기 위해 특정 상태(state) 단위로 움직이는 FSM(finite state machine)으로 설계했다. 각 모듈에서 상태 이동은 터치 버튼을 통한 사용자 입력이 들어왔을 경우, VDM으로부터 새로운 데이터가 도착했을 경우에 인터럽트 형식으로 발생한다. 그림 5에 LCD와 CAN 모듈의 FSM이 간략히 나타나있다. 그림 5 왼쪽의 LCD 출력 모듈 그림 중 ECU select 상태는 사용자에게 모든 ECU의 상태를 한 화면에 간략히 보여준다. 이 때 사용자의 상하좌우 입력에 따라 해당 ECU를 선택할 수 있도록 설계하였고, 실행 버튼을 통해 상세 내용을 확인할 수 있도록 설계 했다(ECU# Zoon 상태, #는 0부터 3까지). 그림 5 오른쪽의 CAN 모듈의 그림은 사용자의 ECU 선택 및 새로운 데이터 유입에 따라서 IDLE 상태와 RX/TX/Set or get ECU #의 상태를 이동하도록 설계 및 개발했다.



그림 3. 하드웨어 플랫폼
Fig. 3. The produced HW platform of ECU simulator

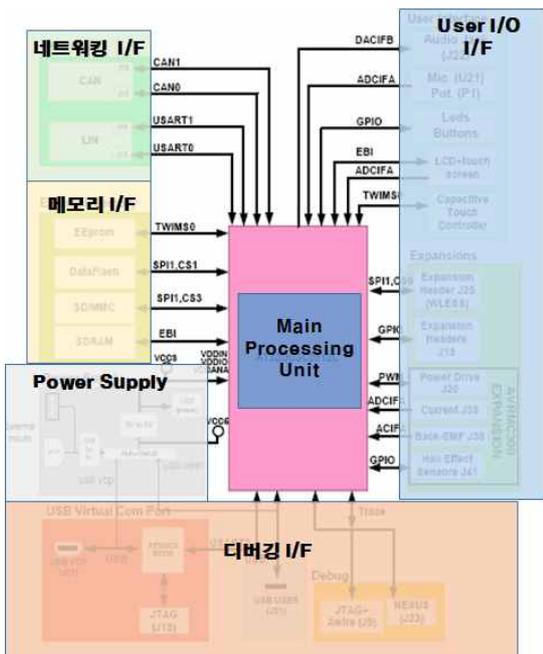


그림 2. ECU 시뮬레이터의 하드웨어 플랫폼 구조도
Fig. 2. Architecture of HW platform for ECU simulator

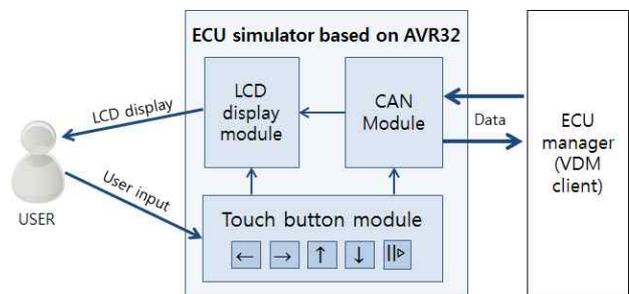


그림 4. ECU 시뮬레이터의 소프트웨어 플랫폼 블록 다이어그램
Fig. 4. The block diagram of SW platform for ECU simulator

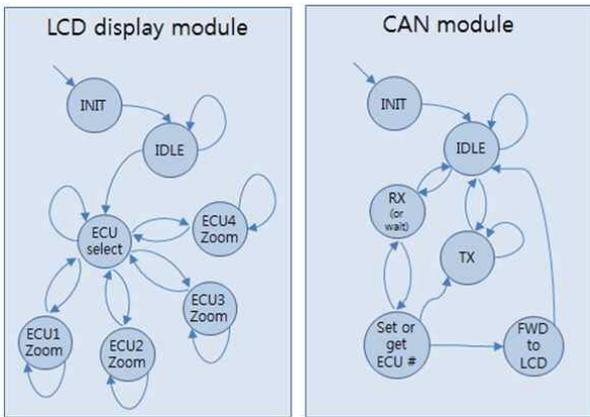


그림 5. 주요 모듈(LCD, CAN)의 FSM
 Fig. 5. The finite-state-machines of main module(left and right are LCD and CAN module, respectively)

IV. ECU 시뮬레이터 성능 검증

4-1 검증을 위한 ECU 관리자 개발

FOTA 업데이트 기술이 적용된 ECU의 성능을 확인하기 위해서는, 시스템에서 ECU로 직접적으로 데이터를 전달하여 데이터 전달의 유효성을 검증해야 한다. 따라서 우리는 업데이트 데이터를 시나리오대로 송신하는 가상의 ECU 관리자를 설계하여 개발된 ECU 플랫폼과의 성능을 확인했다.

ECU 관리자는 현재 AVN의 플랫폼으로 가장 널리 사용되는 FreeScale사의 i.MX6 보드를 채택하였다. 채택된 플랫폼 보드는 ARM사의 Cortex-A9 코어를 장착한 Linux 기반의 AI(automotive infotainment) 보드이다. ECU 플랫폼과의 데이터 통신에는 CAN을 통해 통신하도록 했다. iMX6 보드의 간단한 스펙에 관해서 다음 표2에 나타냈다.

4-2 ECU 시뮬레이터 동작 확인

표 2. ECU 관리자 하드웨어 플랫폼의 주요 사양
 Table 2. The specification of HW platform for ECU manager

Parameter	Value
Processor	ARM Cortex-A9 4xCPU
Memory	2GB DDR3
CAN	2x
Support	WLAN, Bluetooth, GPS, AM/FM tuner, SD card slot, etc.
Support	SD/MMC slot, etc.

다음 그림 6에 개발된 ECU 시뮬레이터와 가상의 ECU 관리자가 개발된 플랫폼 간 연결 모습이 나타나있다. 그림 6의 왼

쪽에 위치한 보드가 ECU 시뮬레이터 플랫폼이고 오른쪽의 보드가 가상의 ECU 관리자가 개발된 iMX 플랫폼 보드이다. 두 보드는 CAN으로 연결돼 통신을 진행하며, iMX 보드에 올려진 ECU 관리자 프로그램은 RGB 케이블을 통해 모니터와 시리얼 통신을 진행하며 터미널 창에 프롬프트 형식을 취하여 데이터 송수신 명령을 내리도록 했다. 즉, VDM으로의 역할을 ECU 관리자가 모의로 수행하도록 사용자(개발자)가 임의의 명령을 주어 테스트를 진행했다. CAN 통신은 가장 보편적으로 사용되는 속도인 115200 bps로, 주소 송수신 모드는 CAN 2.0A로 설정하여 진행했다.

다음 그림 7에 ECU 시뮬레이터를 검증하기 위한 ECU 관리자와의 통신 시나리오를 나타내고 있다. ECU 시뮬레이터 내부에 ECU 4개가 존재하기 때문에, 각각의 ECU가 ECU 관리자와 데이터를 주고받도록 하였다. 그림 7에 나타나있듯이 ECU 관리자에서 ECU#0~#3로 통신 요청 시나리오가 3개, 반대로 ECU#0~#3에서 ECU 관리자로의 통신 요청 시나리오를 1개 지정했다.

ECU 관리자로부터 통신 요청되는 시나리오 세 가지는 각각 다음과 같다. ECU#0~#3 동작의 유효성 및 고장 유무를 확인하기 위한 연결 확인 시나리오, ECU#0~#3에 설치된 현재 소프트웨어의 버전을 확인하는 시나리오, ECU#0~#3의 버전을 업데이트하는 시나리오로 구성돼있다. ECU#0~#3로부터 요청되는 시나리오는, 현재 가상의 업데이트 서버에 저장된 소프트웨어의 최신 버전 확인 요청 이다.

설정된 각 시나리오들은 자동 업데이트 요청 시스템에서 가장 주요/빈번하게 발생하는 통신 시나리오이다. 따라서 ECU#0~#3가 각각의 시나리오에 따라 정확한 동작을 하면, 개발된 하드웨어/소프트웨어 플랫폼이 ECU 시뮬레이터로서 유효함을 확인할 수 있다.

개발된 ECU 시뮬레이터는 각 시나리오에서 마다 업데이트 결과 및 상태를 LCD 화면을 통해 알 수 있도록 패킷 송수신 상황 및 업데이트 버전을 알려준다. 기본적으로 4개의 ECU 업데이트 버전을 최초 화면에 모두 나타냈으며(그림 7의 왼쪽 상단의 LCD 그림), 버튼 입력에 따라 각 ECU의 상태를 세부적으로 볼 수 있도록 했다. 앞선 설명과 같이 ECU 관리자에서의 명령은 프롬프트 형식으로 사용자가 임의로 설정하였다(그림 7의 오른쪽 상단).

다음 그림 8은 실험 결과에 따른 LCD 동작을 ECU 시뮬레이터 내 ECU #0의 상태를 통해 보여준다. 동작 순서는 그림 7에 명시된 시나리오와 동일하다. 즉, 그림 8의 1번 화면은 ECU #0가 시나리오 1의 상황에서 패킷을 수신한 경우를 나타낸다. 화살표는 각 시나리오마다 첫 패킷이 전달되는 방향을 나타낸다. 그림 8의 각 화면 하단의 숫자들은 실행 버튼을 누르는 지속 시간을 나타낸다. 즉, 5초 동안 지속적으로 누른 상태를 유지하면 ECU에서 ECU 관리자 쪽으로 최신 업데이트 버전 확인을 요청(시나리오 4번) 패킷을 송신하도록 했다.



그림 6. ECU 시뮬레이터와 ECU 관리자의 연결 모습
 Fig. 6. The scene in which ECU simulator and ECU manager are being connected with each other(left and right are ECU simulator and manager, respectively)

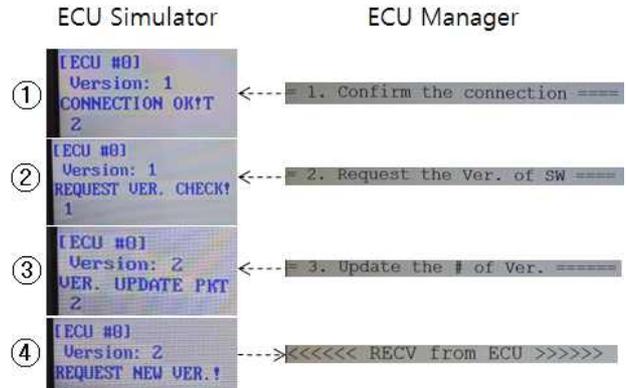


그림 8. 실험 결과
 Fig. 8. Simulation results

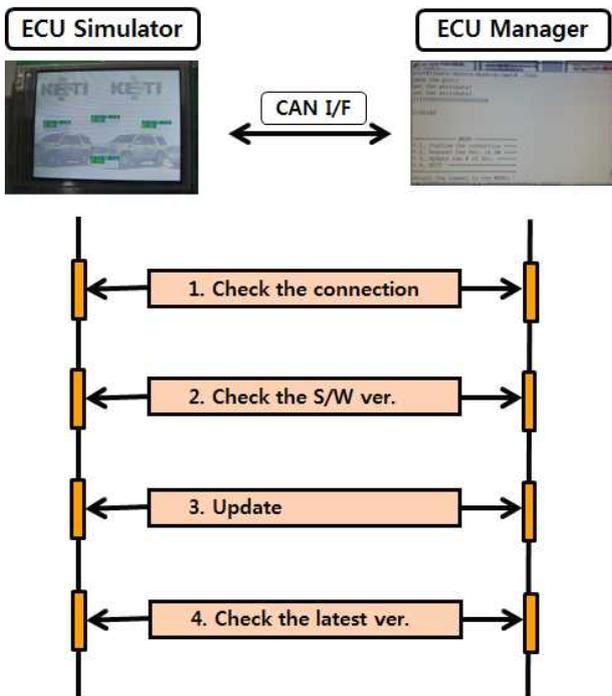


그림 7. 통신 시나리오
 Fig. 7. Communication scenario(from #1 to #4)

그림 8번의 3번 시나리오 실행 결과를 확인해보면, ECU 시뮬레이터 내의 ECU #0의 소프트웨어 버전이 1에서 2로 상승된 것을 볼 수 있다. 즉, 소프트웨어 업데이트가 진행된 것이다.

실험을 통해 ECU 시뮬레이터가 패킷의 전달 순간에 업데이트 정보를 포함한 정보를 LCD로 정확히 표현하며, 반대로 ECU 관리자로의 데이터 전달 역시 시나리오대로 동작함을 확인했다. 따라서 개발된 ECU 시뮬레이터는 하드웨어/소프트웨어 수준 모두에서 지능형 자동차 내부의 업데이트 시스템 검증에 사용되기 위한 사용이 유효함을 알 수 있다.

V. 결 론

본 논문은 FOTA 기반의 지능형 자동차를 위한 범용 ECU 시뮬레이터의 설계 및 구현에 대해 서술했다. 특정 ECU가 아닌 범용적인 데이터 송수신을 위한 ECU 시뮬레이터를 하드웨어와 소프트웨어 플랫폼 두 부분으로 나눠서 개발을 진행했다. ECU 시뮬레이터 동작의 검증을 위해 외부로부터 업데이트 데이터를 전달해주는 ECU 관리자를 개발하여 테스트를 진행했다. 총 4가지의 시나리오를 적용했고, ECU 시뮬레이터 내부에 4개의 가상 ECU를 설정하여 테스트를 진행했다. 그 결과 ECU 시뮬레이터로서 동작됨을 확인할 수 있었다.

ECU 시뮬레이터의 신뢰성을 높이고 전체 시스템의 완성도를 높이기 위하여 ECU 시뮬레이터로 실제 장치를 제어하는 기능을 추가할 예정이다. 오디오 혹은 네비게이션 등의 인포테인먼트를 CAN 입출력 포트에 연결하여 제어할 계획이다.

감사의 글

본 논문은 지식경제부의 IT 산업 핵심 기술 개발 프로그램(10041705, 과제명: 차량용 AVN/ECU 소프트웨어 통합 업데이트 시스템 기술 개발)의 지원을 받아 수행하였음을 밝힙니다.

참고문헌

[1] K. Lee, J. Choi, "Integrated vehicle safety control system for smart vehicles," *Journal of Auto Journal Publication* 2012, Vol. 34, No. 6, pp. 20-25, June 2012
 [2] J Oh, "Necessity and Methodology for the Integration of ECU," *Journal of Auto Journal Publication* 2012, Vol. 34, No. 9, pp.

- 37-41, Arg, 2012
- [3] S. H. Park, S. Y. Min, H. T. Ju, "CAN-based automatic seatbelt control system," in *Proceeding of IEEK Atom conference 2010*, Korea, pp. 295-296, Nov, 2010
- [4] S. K. Moon, "A survey on the works of designing an SoC platform for smart motor vehicle info-tainment," in *Proceeding of KIICE Spring conference 2011*, Korea, pp. 699-701, May, 2011
- [5] Y. Berholtz, "Complex software control of connected car," Arg. 2012, pp. 76-77. [Internet]. Available: <http://www.autoelectronics.co.kr>
- [6] J. W. Lee, J. M. Yeon, D. H. Won, "Development of the software platform for the ECU of the Vehicles," in *Proceeding of KASE Spring conference 2011*, Korea, pp. 1208-1211, May, 2011
- [7] H. K. Kim, J. G. Choi, "A study of remote firmware upgrading schemes," in *Proceeding of KIEE Summer conference 2011*, Korea, pp. 726-727, July, 2011
- [8] J. H. Moon, D. I. Kim, R. J. Park, H. B. Lee, T. Y. Jeong, "A remote firmware update mechanism for a TDMA-based bidirectional linear wireless sensor networks", *Journal of Institute of Control, Robotics and System 2009*, Vol. 15, No. 8, pp 867-875, Aug. 2009
- [9] N. J. Jeong, I. K. Yang, S. H. Choi, Y. T. Jin, "A Study on the Requirements Analysis and Design for System Development of Remote Firmware Upgrade Management System Based-on AMI," *Journal of The Transactions of the Korean Institute of Electrical Engineers*, Vol. 61, No. 1, pp 155-161, Jan. 2012
- [10] S. H. Park, H. M. Park, C. B. Kim, "A Study on Encryption Module for Remote Terminal Security of Smart Water-Grid Network," *Journal of advanced national technology* 2013, Vol. 17, No. 6, pp 719-726, Dec. 2013
- [11] M. E. Lee, "The Implementation of Embedded Web Server System for a Remote Control and Measurement," *Journal of advanced national technology* 2013, Vol. 16, No. 5, pp 839-845, Oct. 2012



박인혜 (In-Hye Park)

2008년 : 광운대학교 컴퓨터공학과 공학사
 2010년 : 광운대학교 컴퓨터공학과 공학석사
 2012년 ~ 현재 : 전자부품연구원 소프트웨어디바이스연구센터 연구원
 ※ 관심분야 : 임베디드 시스템, 지능형 자동차 시스템, 무선 센서 네트워크



고재진 (Jae-Jin Ko)

1997년 : 광운대학교 컴퓨터공학과 공학사
 2000년 : 광운대학교 컴퓨터공학과 공학석사
 2012년 : 광운대학교 컴퓨터공학과 공학박사
 2000년 ~ 현재 : 전자부품연구원 소프트웨어디바이스연구센터 책임연구원/센터장
 ※ 관심분야 : 임베디드 시스템, 지능형 자동차 상황인식



곽재민 (Jae-Min Kwak)

1998년 : 한국항공대학교 통신정보공학과 공학사
 1999년 : 한국항공대학교 통신정보공학과 공학석사
 2002년 : 한국항공대학교 통신정보공학과 공학박사
 2002년 ~ 2003년 : 한국전자통신연구원 네트워크 연구소 Post-doc
 2003년 ~ 2008년 : 전자부품연구원 SoC 연구센터 책임연구원
 2008년 ~ 현재 : 목포해양대학교 해양정보통신공학과 조교수
 ※ 관심분야 : 디지털 통신 시스템, 유무선 통신신호처리, 임베디드 시스템