

논문 2013-08-29

# AUTOSAR 플랫폼 기반 CDD를 활용한 비휘발성 메모리 수명 연장 기법

(A Non-volatile Memory Lifetime Extension Scheme Based on  
the AUTOSAR Platform using Complex Device Driver)

신 주 석, 손 정 호, 이 은 령, 오 세 진, 안 광 선\*

(Ju-Seok Shin, Jeong-Ho Son, Eun-Ryung Lee, Se-Jin Oh, Kwang-Seon Ahn)

**Abstract** : Recently, the number of automotive electrical and electronic system has been increased because the requirements for the convenience and safety of the drivers and passengers are raised. In most cases, the data for controlling the various sensors and automotive electrical and electronic system used in runtime should be stored on the internal or external non-volatile memory of the ECU(Electronic Control Units). However, the non-volatile memory has a constraint with write limitation due to the hardware characteristics. The limitation causes fatal accidents or unexpected results if the non-volatile memory is not managed. In this paper, we propose a management scheme for using non-volatile memory to prolong the writing times based on AUTOSAR(AUTOMotive Open System Architecture) platform. Our proposal is implemented on the CDD(Complex Device Driver) and uses an algorithm which swaps a frequently modified block for a least modified block. Through the development of the prototype, the proposed scheme extends the lifetime of non-volatile memory about 1.08 to 2.48 times than simply using the AUTOSAR standard.

**Keywords** : Automotive, ECU, Non-volatile memory, AUTOSAR

## 1. 서 론

오늘날 자동차 ECU(Electric Control Unit)는 차량과 컴퓨터 성능 발전과 함께 자동변속기 제어를 비롯하여 구동, 제동, 조향 계통 등 차량의 많은 부분을 제어하는 역할을 한다. 또한 차량에 탑재되는 ECU의 개수가 증가하고, 소프트웨어의 복잡성이 증가함에 따라 하드웨어 또는 ECU에 비 의존적인 표준화된 통합 소프트웨어 플랫폼의 요구가 늘어나 2003년에 유럽 자동차 업체인 BMW, Bosch, Continental, Daimler-Chrysler, Volkswagen을 주축으로 AUTOSAR(AUTOMotive Open System

ARchitecture)가 소개 되었다[1, 2]. 쓰기 횟수가 제한되어 있는 비휘발성 메모리의 경우[3, 4], 특정 영역에 데이터가 반복적으로 쓰여 지게 되면 해당 영역이 손상되고 저장된 데이터가 손상될 수 있다. 이 손상된 데이터를 기반으로 ECU들을 제어하게 되면 치명적인 사고로 이어질 수 있다. 기존 임베디드 시스템에서는 비휘발성 메모리의 수명 연장 등을 고려하고 메모리를 효율적으로 이용하기 위해 플래시 메모리 전용 파일 시스템을 이용하고 있다 [5]. 하지만 AUTOSAR 표준에서는 파일 시스템이 존재하지 않으며, 비휘발성 메모리의 수명 관리에 대해서도 고려하지 않고 있다.

본 논문에서는 위와 같은 문제점을 해결하기 위하여 AUTOSAR 표준 메모리 스택 구현 이외에 비휘발성 메모리의 수명 연장을 고려한 기법을 CDD(Complex Device Driver)의 형태로 구현하였으며, 비휘발성 메모리의 수명이 약 1.08~2.48배 연장되는 것을 실험 결과를 통하여 확인하였다.

\* Corresponding Author (gsahn@knu.ac.kr)

Received: 20 May 2013, Revised: 10 July 2013,  
5 Aug. 2013, Accepted: 5 Aug. 2013.

J.S. Shin, J.H. Son, E.R. Lee : ETRI

S.J. Oh, K.S. Ahn : Kyungpook National  
University

본 논문의 구성은 다음과 같다. 2장 관련 연구에서는 플래시 메모리 전용 파일 시스템에서 메모리의 수명을 연장하기 위한 기법으로 마모도 평준화(Wear leveling) 기법들에 대하여 기술하고, AUTOSAR 플랫폼 및 플랫폼에서 메모리 스택의 동작에 대해 기술한다. 3장에서는 본 논문에서 제안하는 AUTOSAR 플랫폼 기반의 비휘발성 메모리 수명 연장 기법에 대해 설명한다. 4장에서는 제안한 기법에 대한 실험결과를 분석하고, 마지막으로 5장에서는 결론을 맺는다.

## II. 관련 연구

본 장에서는 플래시 메모리 전용 파일 시스템에서 사용되는 마모도 평준화 기법들에 대해 기술하고, AUTOSAR 플랫폼에 대해 살펴본다. 또한 플랫폼 전체에서 메모리 스택 부분의 기본적인 동작 방식에 대해 기술한다.

### 1. 마모도 평준화 기법

최근 내장형 기기에서 필요한 저장 장치에 대용량의 메모리가 요구되고 있어 가전기기, 통신기기, 휴대기기 등의 다양한 기기의 저장매체로 NAND 플래시 메모리의 사용이 증가되고 있다. NAND 플래시 메모리를 효율적으로 사용하기 위해서는 플래시 메모리 전용 파일 시스템을 이용하여야 한다[5]. 대표적인 파일 시스템으로 YAFFS2(Yet Another Flash File System 2), JFFS2(Journaling Flash File System 2) 등이 있으며, 마모도 평준화, 가비지 컬렉션(Garbage Collection) 등의 기능을 제공하고 있다. 플래시 메모리에서 특정 블록에 기록 및 소거 작업이 집중되면, 일부 블록의 수명이 다하게 되어 저장 장치로서의 내구성에 문제가 발생한다. 이를 방지하기 위하여 전체 블록의 기록 및 소거주기가 고루 분산되도록 관리되어야 하는데 이러한 과정을 마모도 평준화라고 한다.

#### 1.1 삭제색인을 이용한 블록삭제

삭제색인(Cleaning index)을 블록 삭제 비용과 삭제 빈도의 곱으로 표시하고, 이 색인을 통하여 블록을 삭제하는 작업을 수행하는 기법이다. 삭제색인이 큰 블록을 우선적으로 선택하는 방식이기 때문에 삭제 빈도가 높은 블록에 기록 및 소거의 동작이 집중되는 문제점이 있다[6]. 이를 해결하기 위하여 소거횟수가 큰 블록과 소거횟수가 적은 블록에서 유효

한 데이터를 교환하는 방법으로 문제를 해결하였으나 소거 작업 수행시마다 삭제색인을 다시 계산하여 전체 블록과 비교해야 하므로 계산 비용이 많이 드는 문제점이 있다[7].

#### 1.2 Greedy & Cost-Benefit 정책

Greedy 정책은 소거과정에서 발생하는 유효 블록의 복사로 인한 작업을 최소화하기 위해 유용한 데이터를 포함하고 있지 않은 무효 블록이 가장 많은 세그먼트를 선택하여 지우는 방법이다[8]. Cost-Benefit 정책은 다음 식에 의해 세그먼트를 선택한다.

$$\frac{Benefit}{Cost} = \frac{age*(1-u)}{2u} \quad (1)$$

$u$ 는 세그먼트의 이용률로서 전체 세그먼트 크기에 대한 유효한 블록의 크기이다.  $2u$ 는 세그먼트를 지우기 위해 유효한 블록을 읽어서 다른 세그먼트에 저장하는 비용을 의미하며,  $(1-u)$ 는 새롭게 생성되는 빈 공간의 양이다.  $age$ 는 가장 마지막에 블록을 무효로 변경시킨 후의 시간이며,  $2u$ 는 유효 블록을 읽은 후 다른 빈 공간에 기록하는데 사용되는 시간을 의미한다. Cost에 비해 Benefit이 큰 세그먼트를 선택하도록 하는 기법으로 비용이 가장 적고 이익이 가장 많은 세그먼트를 선택하는 기법이다[9].

#### 1.3 순차적 저장방식

로그 구조 파일시스템에 기반한 JFFS2, YFFS2에서 데이터의 기록과 갱신을 플래시 메모리에 순차적으로 수행하는 방식이다[10-12]. 모든 블록을 다한 번씩 사용하면 다시 처음 블록부터 다시 사용하는 방법을 반복하는 기법으로 가장 균등하게 데이터를 저장할 수 있는 방법이다. 하지만 파일 시스템에서는 파일의 접근 형태가 균일하지 못할 경우, 파일의 갱신빈도에 따라 해당 블록의 소거 횟수의 편차가 커지는 문제점이 있다.

## 2. AUTOSAR 플랫폼

최근 차량용 전자장비의 개수가 늘어나고 차량 부품에 소프트웨어가 탑재되면서 하드웨어나 ECU에 의존적인 소프트웨어의 복잡성 및 모듈의 재사용성, 호환성 등의 문제가 발생하게 되었다[2, 13]. 이를 해결하기 위해 표준화된 통합 소프트웨어 플랫폼 중에 하나인 AUTOSAR가 소개 되었으며, AUTOSAR 플랫폼은 자동차 전자장치의 소프트웨어 재사용성 및 확장성 향상과 자동차의 생산 비용 절감을 목표로 하고 있다[2, 13, 14]. AUTOSAR는

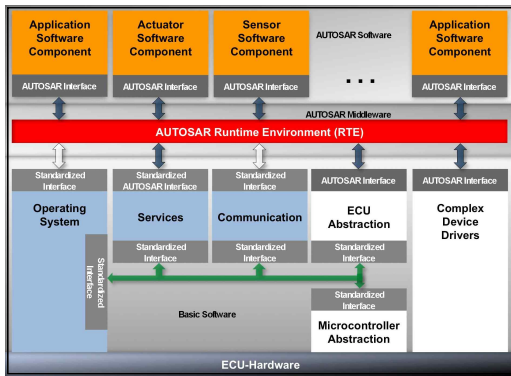


그림 1. AUTOSAR 플랫폼  
Fig. 1 The AUTOSAR Platform

크게 3개의 계층으로 다음과 같이 구성되어 있다. 그림 1은 AUTOSAR 플랫폼을 나타내고 있다.

- Autosar SW-C(Component): 응용 소프트웨어 부분으로 자동차의 실제 기능을 담당하는 부분이다.
- RTE(Run-Time Environment): SW-C와 BSW(Basic Software) 사이의 데이터 교환 및 연결시키는 역할을 하며 상위에서 제작된 SW-C들을 수정 없이 사용할 수 있도록 만들어 주는 일종의 미들웨어이다.
- BSW(Basic Software): 하드웨어와 소프트웨어를 분리시켜 다양한 ECU에 적용 가능하게 하는 역할을 한다. ECU와 하드웨어와 연결하는 Layer로 OS, EAL(ECU Abstraction Layer), MCAL(Microcontroller Abstraction Layer), CDD(Complex Device Driver)로 구성된다. MCAL에서는 하드웨어 의존성이 있는 부분으로 ADC(Analogue Digital Conversion), DIO(Digital Input Output) 디바이스 드라이버 등으로 구성이 되며, 이 외에 표준화 되지 못한 영역은 CDD영역에 포함된다.

### 3. AUTOSAR 플랫폼에서 비휘발성 메모리 접근 방식

BSW 영역에서 NVRAM(Non-Volatile Random Access Memory)를 통하여 Flash 메모리, EEPROM(Electrically Erasable Programmable Read-Only Memory) 등과 같은 비휘발성 메모리에 접근한다. 그림 2는 AUTOSAR 플랫폼에서 비휘발성 메모리에 접근하는 단계를 나타낸 것이다.

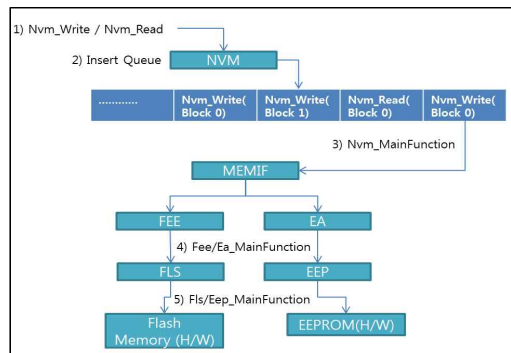


그림 2. 비휘발성 메모리 접근 방법  
Fig. 2 The Access Method of the Non-Volatile Memory on the AUTOSAR Platform

AUTOSAR 표준에서는 공통 플랫폼과 소프트웨어를 모델 기반으로 개발할 수 있는 도구 기반의 개발 방법론도 다루고 있으며[1, 15-17], 응용 개발자가 도구를 통하여 사용 용도에 맞추어 메모리 영역을 블록단위로 나누고 다음과 같은 세부 과정을 통해 메모리의 특정 블록을 읽거나 쓰는 동작을 수행한다[18, 19]. 응용 개발자가 설정한 블록들 중 특정 블록에 대해 AUTOSAR 표준 함수 Nvm\_WriteBlock 또는 Nvm\_ReadBlock 함수를 통하여 비휘발성 메모리에 데이터를 쓰거나 읽는 작업이 가능하며, 함수가 호출될 때마다 NVM(Non-Volatile Memory) 계층에서는 해당되는 작업들을 큐에 삽입한다.

이후, BSW 스케줄러가 동작할 때 스케줄러에 등록된 함수 중 Nvm\_MainFunction 함수가 실행되며, 큐에서 작업을 가져와 MEMIF(Memory Interface)를 거쳐 해당 블록의 정보에 따라 Flash 메모리에 접근할 경우 Fee\_Read 또는 Fee\_Write 함수를 수행하고 EEPROM에 접근할 경우 Ea\_Read 또는 Ea\_Write 함수를 수행한다. 또한 FEE(Flash EEPROM Emulation)/ EA(EEPROM Abstraction) 계층에서도 스케줄러에 의해 Fee\_MainFunction 함수 또는 Ea\_MainFunction 함수가 호출되면 Fls\_Read/Write 함수나 Eep\_Read/Write 함수가 실행된다.

마지막으로 실제 비휘발성 메모리에 접근하기 위해 스케줄러에 의해 Fls/Eep\_MainFunction 함수가 실행되면, 마이크로 컨트롤러의 내부 또는 외부 비휘발성 메모리인 Flash 메모리 또는 EEPROM에 데이터를 읽거나 쓰게 된다.

### III. 제안 기법

본 장에서는 본 논문에서 제안한 AUTOSAR 플랫폼 기반의 비휘발성 메모리 수명 연장 기법에 대해 설명한다. AUTOSAR 표준에 따라 도구로 설정된 특정 블록에 쓰기 요청이 많아지면 비휘발성 메모리인 Flash메모리 또는 EEPROM의 특정 영역에 지속적으로 업데이트가 된다. 쓰기 제한이 있는 비휘발성 메모리의 경우, 특정 영역에 데이터를 계속 쓰게 되면 해당 영역이 손상되고, 저장된 데이터가 손상될 수 있다. 또한 손상된 데이터를 기반으로 ECU를 제어할 경우 예기치 못한 오류가 발생할 수 있으며 치명적인 사고로 이어질 수도 있기 때문에 비휘발성 메모리의 수명을 고려하여 구현하여야 한다.

2장의 관련 연구에서 살펴본바와 같이 기존 임베디드 시스템에서 파일 시스템을 이용한 마모도 평준화 기법들이 존재함에도 불구하고, AUTOSAR 플랫폼에서는 파일 시스템을 사용할 수 없기 때문에 본 논문에서는 비휘발성 메모리의 수명을 연장시키기 위한 기법을 CDD(Complex Device Driver)의 형태로 구현하여 비휘발성 메모리의 수명을 관리하는 방법을 제안한다. 제안하는 기법에서의 가정사항은 다음과 같다.

먼저 도구를 이용하여 메모리를 그림 3와 같이 4개의 블록으로 나누며, 메모리의 모든 영역을 사용한다고 가정한다. 예를 들어 128Kbyte의 비휘발성 메모리를 사용할 경우, 32Kbyte 4개 또는 16Kbyte 8개로 블록을 구성할 수 있다. 또한 블록 B의 경우, 빈번히 데이터가 수정되는 반면에 블록A와C는 상대적으로 수정 빈도 횟수가 적다고 가정한다.

각 블록의 크기는 동일하게 설정하며, 9바이트를 추가적으로 설정하는데, 하위 9바이트는 메모리 관리를 위해 그림 4과 같이 사용되어 진다.

- MC(Modified Count): 비휘발성 메모리의 실제 수정된 횟수를 의미한다.
- SF(Swap Flag): 스왑 가능한 블록인지 아닌지를 체크하기 위해 사용되어지는 플래그이다. 스왑을 수행할 때 임계 값에 도달한 블록의 SF를 셋팅한다.
- PF(Progress Flag): 연속적으로 블록들이 스왑을 요청할 경우에 체크되어 지는 플래그이다. MC를 기반으로 스왑을 수행하기 때문에 먼저 스왑을 수행한 블록이 해당 블록의 PF를 셋팅하여 다른 블

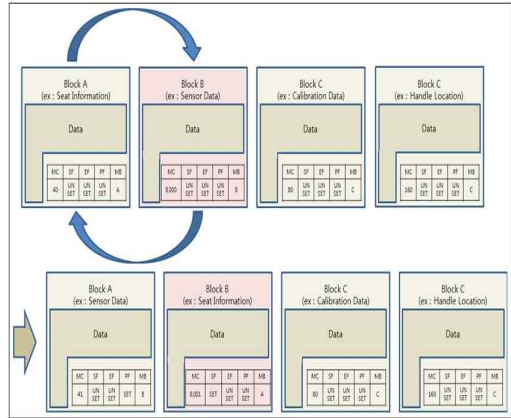


그림 3. CDD의 형태로 구현된 비휘발성 메모리 관리 기법

Fig. 3 A Non-Volatile Memory Management Scheme on the CDD(Complex Device Driver)

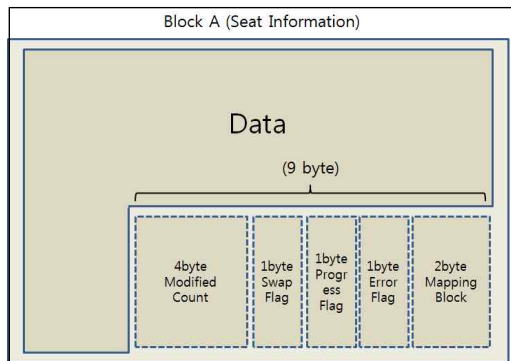


그림 4. 각 블록의 구성도

Fig. 4 The Block Diagram for the each Block

록이 해당 블록과 스왑하지 못하도록 방지하는 역할을 한다.

- EF(Error Flag): 에러의 발생 여부를 나타내기 위해 사용되어지는 플래그이다.
- MB(Mapping Block): 스왑된 블록을 찾아가기 위한 인덱스이다.

그림 4는 CDD의 형태로 구현된 비휘발성 메모리 관리 기법을 도식화 한 것이고 그림 5는 비휘발성 메모리 관리 기법의 순서도이다. 메모리 관리 기법의 동작 방법은 다음과 같다.

- 1) 해당 블록 이동: 응용 프로그램에서 비휘발성 메모리에 쓰기 요청이 들어오면 요청된 블록의 Mapping Block을 참조하여 해당 블록으로 이동한다.

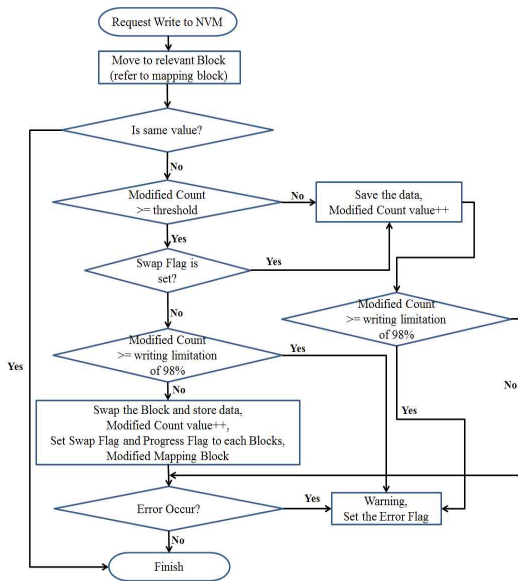


그림 5. 비휘발성 메모리 관리 기법 순서도  
Fig. 5. The Flowchart of the Non-Volatile Memory Management Scheme on the CDD(Complex Device Driver)

2) 임계 값 체크: 응용 프로그램에서 파라미터로 전달된 길이만큼 Fls/Ea\_Compare 함수를 통하여 현재 쓸려는 데이터와 기존의 데이터가 동일한지를 판단한다. 만약 동일하다면 데이터를 업데이트 하지 않고 종료한다.

그렇지 않다면 해당 블록의 MC(Modified Count)를 읽어 MC가 임계 값에 도달하였는지를 체크한다. 임계 값은 비휘발성 메모리의 최대 쓰기 가능 횟수의 80%로 휴리스틱하게 결정되었다.

3) 스왑 수행: 블록의 MC가 임계 값에 도달하면 나누어진 블록 중 PF가 셋팅 되어 있지 않고 MC가 가장 작은 블록과 스왑을 수행한다. 그리고 스왑이 요청된 블록들의 MB를 찾은 후, 블록들의 MB를 각각 수정하고 SF와 PF를 셋팅 한다.

4) 데이터 저장: 응용 프로그램으로부터 요청된 데이터를 저장할 경우 다음과 같은 조건에 따라 데이터를 비휘발성 메모리에 저장한다.

4-1) 2번 과정에서 MC가 임계 값보다 작을 경우, MC를 증가시키고 해당 블록에 데이터를 저장하고 프로세스를 종료한다.

4-2) 3번 과정에서 MC가 임계 값에 도달하였지만 해당 블록의 SF가 셋팅된 경우, MC를 증가시키고 데이터를 저장한다. 이때 해당 블록의 최대 쓰기

횟수를 초과하여 데이터를 저장하면 정상적인 데이터로 업데이트 되지 않기 때문에 MC가 비휘발성 메모리의 최대 쓰기 가능횟수의 98%에 도달하면 상위 어플리케이션으로 알려줄 수 있도록 EF를 셋팅한다.

4-3) 3번 과정에서 스왑을 수행하면 스왑되는 각 블록들의 MC를 증가시키고 데이터를 저장한다.

## IV. 실험

본 장에서는 AUTOSAR 메모리 서비스 부분과 CDD의 형태로 비휘발성 메모리 관리 기법을 구현한 내용을 기반으로 실험한 결과를 분석한다.

### 1. 가정사항

실험을 위한 실험 환경 및 가정사항은 다음과 같다. Tasking 툴 셋으로 모듈을 구현하였고, XC2365A Easy-Kit에 구현된 모듈을 TRACE32 디버깅 툴을 이용하여 포팅 하였다. 외부 EEPROM(AT25128N-128K)에 데이터를 주기적으로 업데이트하는 것으로 실험하였고, SPI(Serial Peripheral Interface)를 사용하여 외부 EEPROM에 접근하였다.

가정사항으로는 스왑을 하기 위한 조건인 임계 값을 비휘발성 메모리의 최대 쓰기 가능 횟수의 80%로 휴리스틱하게 설정하였다.

또한 비휘발성 메모리의 최대 쓰기 횟수가 넘어서도 계속 데이터를 쓰게 되면 정상적으로 업데이트 되지 않기 때문에 이를 방지하기 위한 EF를 셋팅하는 시점을 최대 쓰기 가능 횟수의 98%로 휴리스틱한 방법으로 결정하였다. 그리고 EEPROM 하드웨어 명세서에는 100,000번으로 쓰기 제한 횟수를 정의하고 있으나, 여러 번의 실험을 수행하기 위하여 최대 임계 값을 10,000번으로 가정하고 실험하였다.

### 2. 실험 결과

EEPROM에 데이터를 쓰기 위하여 2장 관련연구에서 살펴본바와 같이 Nvm\_WriteBlock함수 요청 이후, Nvm\_MainFunction, Ea\_MainFunction, Eep\_MainFunction함수를 통하여 EEPROM에 접근한다. 그리고 실제 Eep\_MainFunction함수에서 EEPROM에 쓰기 동작을 할 경우, CDD에 구현된 스왑 정책에 따라 해당 블록에 데이터를 저장하게 된다.

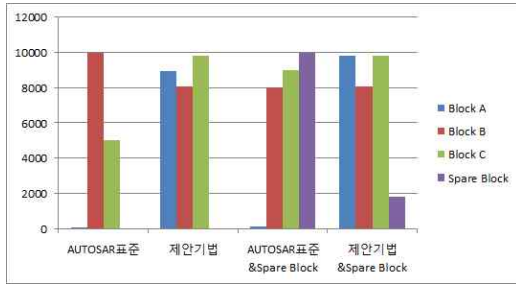


그림 6. 블록을 3개로 설정한 실험 결과  
Fig. 6 The Result of the Experimentation using the three Blocks and spare Block

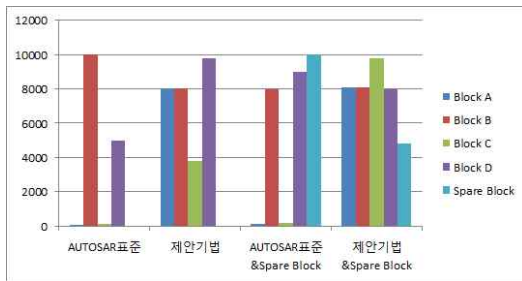


그림 7. 블록을 4개로 설정한 실험 결과  
Fig. 7 The Result of the Experimentation using the four Blocks and spare Block

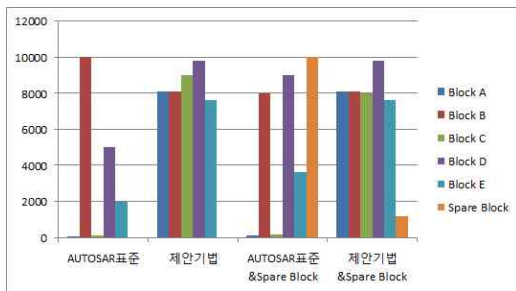


그림 8. 블록을 5개로 설정한 실험 결과  
Fig. 8 The Result of the Experimentation using the five Blocks and spare Block

비휘발성 메모리 블록은 도구를 통하여 3~5개의 블록으로 설정을 변경하면서 실험 하였다. 하나의 블록이라도 임계 값의 98%에 도달하는 조건으로 중단점을 두었고, 동작하다가 중단점에 도달하면 디버깅 창에서 블록들의 수정 횟수를 측정하는 것으로 실험하였으며, Spare블록을 사용하기 때문에 각각 2회씩 반복 실험하였다.

표 1. 실험 결과

Table 1. The Result of the Experimentation

	(1)	(2)	(3)	(4)
3개 블록 사용	15,050	26,760	27,090	29,435
4개 블록 사용	15,150	29,669	27,270	38,781
5개 블록 사용	17,150	42,574	30,870	42,773
메모리 사용 가능 공간	All Area	All Area - (Number of Block * 9Byte)	All Area	All Area - (Number of Block * 9Byte)

- (1) AUTOSAR 표준 EEPROM Write
- (2) AUTOSAR 표준과 CDD의 형태로 메모리 관리 기법을 적용하여 EEPROM에 Write
- (3) AUTOSAR 표준과 Spare Block을 이용한 EEPROM에 Write
- (4) AUTOSAR 표준과 CDD의 형태로 메모리 관리 기법을 적용하고 Spare Block을 이용하여 EEPROM에 Write

그림 6~8은 블록을 각각 3개, 4개, 5개로 설정하고 실험했을 때의 실험 결과를 보여 주고 있으며, 표 1은 실험결과를 토대로 제안한 기법과 단순히 AUTOSAR 표준에 근거하여 비휘발성 메모리에 기록할 경우를 비교 분석한 결과이다. CDD의 형태로 메모리 관리 기법을 구현하여 비휘발성 메모리에 반복적으로 쓰기 동작을 수행한 결과, 단순히 AUTOSAR 표준의 비휘발성 메모리 접근 방법을 통하여 메모리에 쓰기 동작한 것 보다 약 1.77~2.48배 정도 메모리의 수명을 연장하는 것을 확인하였다.

또한 Spare 블록을 이용하여 하나의 블록이라도 임계 값에 도달하거나 최대 쓰기 횟수와 동일하게 되면 Spare 블록에 쓰는 것으로 실험하였다. 이 경우에는, 본 논문에서 제안한 기법으로 비휘발성 메모리에 쓰기 동작을 할 경우 약 1.08~1.38배 정도 메모리의 수명을 연장 시키는 것을 확인하였다.

그러나 표 1과 같이 제안한 기법을 사용하지 않을 경우, 설정한 메모리의 크기 모두 사용가능하지만 제안한 기법을 사용할 경우에는 설정한 블록의 개수마다 약 9바이트 용량을 사용하지 못하는 단점이 있다.

## V. 결 론

본 논문에서는 CDD의 형태로 AUTOSAR 플랫폼 기반의 비휘발성 메모리 수명 연장 기법을 제안

하였다. 제안한 기법으로 비휘발성 메모리에 데이터를 저장할 경우 비휘발성 메모리의 수명을 약 1.08~2.48배 정도 연장 시키는 것을 실험을 통하여 확인하였다.

차량 전장 부품이 증가하고 소프트웨어가 복잡해짐에 따라 앞으로 AUTOSAR 플랫폼 기반의 차량 전장 부품이 꾸준히 증가할 것으로 기대된다. 또한 사용자의 편의성을 기반으로 하는 전장 부품이 늘어남에 따라 각종 센서들이 늘어나고 제어 데이터들이 저장되어야하기 때문에 비휘발성 메모리의 사용빈도가 늘어날 것으로 기대된다. 따라서 응용 개발자가 하드웨어의 제약사항을 모두 알지 못하고 AUTOSAR 플랫폼 기반으로 비휘발성 메모리에 각종 데이터를 저장하고자 할 경우, 비휘발성 메모리의 제약 사항을 고려하여 본 논문에서 제안한 기법과 같이 메모리의 수명을 관리하는 것이 필요하다.

## References

- [1] <http://www.autosar.org>
- [2] H. Heinecke, J. Bielefeld, K.P. Schnelle, N. Maldener, H. Fennel, O. Weis, T. Weber, J. Ruh, L. Lundh, T. Sanden, P. Heitkamper, R. Rimkus, J. Leflour, A. Gilberg, U. Vernich, S. Voget, K. Nishikawa, K. Kajio, T. Scharnhorst, B. Kunkel, "AUTOSAR-Current results and preparations for exploitation," Proceedings on Euroforum conference, 2006.
- [3] J. Brewer, M. Gill, Nonvolatile Memory Technologies with Emphasis on Flash: A Comprehensive Guide to Understanding and Using Flash Memory Devices, Wiley-IEEE press, 2008.
- [4] G. Tao, "Reliability of advanced embedded non-volatile memories: The 2T-FNFN device," Proceedings on ICICDT, pp.79-82, 2008.
- [5] S.O. Park, S.J. Kim, "An efficient multimedia file system for NAND Flash Memory Storage," IEEE Transactions on Consumer Electronics, Vol. 55, No. 2, pp.139-145, 2009.
- [6] M. Wu, W. Zwaenepoel, "eNVy: A Non-Volatile, Main Memory Storage System," Proceedings on International Conference of ASPLOS-6, pp.86-97, 1994.
- [7] D.Y. Kim, H.S. Yoo, S.H. Park, S.W. Park, "k-leveling: An Efficient Wear-leveling scheme for Flash Memory," Proceedings on KCC, pp.787-789, 2005 (in Korean).
- [8] A. Kawaguchi, S. Nishioka, H. Motoda, "A Flash-Memory Based File System," Proceedings on USENIX Technical Conference, pp.155-164, 1995.
- [9] J.G. Kim, S.M. Park, C.G. Kim, "A Ranking Cleaning Policy for Embedded Flash File Systems," Vol. 9-A, No. 4, pp.399-404, 2002 (in Korean).
- [10] <http://www.yaffs.net>
- [11] <http://sourceware.org/jffs2/>
- [12] S.H. Park, T.H. Lee, K.D. Chung, "A Flash File System Support Fast Mounting for NAND Flash Memory Based Embedded Systems," Proceedings on International Conference of SAMOS, pp.415-424, 2006.
- [13] D.H. Kum, G.M. Park, S.H. Lee, W.Y. Jung, "AUTOSAR migration from existing automotive software," Proceedings on ICCAS, pp.558-562, 2008.
- [14] O. Scheickl, M. Rudorfer, "Automotive Real Time Development Using a Timing-augmented AUTOSAR Specification," Proceedings on ERTS, 2008.
- [15] J.Y. Kim, J.W. Lee, K.K. Kwon, K.S. Kim, "Implementation and Design System Generation Tool model supporting the AUTOSAR Operating System," Journal of KIIT, Vol. 9, No. 4, pp.29-39, 2011 (in Korean).
- [16] E.R. Lee, J.H. Son, J.Y. Kim, "Qplus-Auto: The Realtime Operating System based on AUTOSAR," Proceedings on Annual Conference and Exhibition of KSAE, pp.2159-2162, 2010 (in Korean).
- [17] K.S. Lee, I.S. Park, M.H. Sun, W.T. Lee, "AUTOSAR-ready Light Software Architecture for Automotive Embedded Control Systems," Journal of KSAE, Vol. 21, No. 1, pp.68-77, 2013 (in Korean).
- [18] W.J. Won, J.K. Son, G.M. Park, D.H. Kum, S.H. Lee, "Design and Implementation Procedure of the AUTOSAR I/O Driver Cluster," Proceedings on ICCAS-SICE,

pp.5618-5623, 2009.

[19] H.K. Ryu, S.R. Cho, W.Y. Jung, "NVRAM Manager and System Service Architecture Design for AUTOSAR-based Automotive Software," Journal of KISS, Vol. 35, No. 1(B), pp.437-440, 2008 (in Korean).

**저 자 소개**

**신 주 석**



2006년 경일대학교 컴퓨터공학과 학사.  
2010년 경북대학교 전자전기컴퓨터학부 석사.  
현재, 경북대학교 컴퓨터학부 박사과정, 한국전자통신연구원 연구원.

관심분야 : RFID, 정보보호, 임베디드 시스템, 차량 전장 시스템.  
Email: jsshin@etri.re.kr

**손 정 호**



2002년 경북대학교 정보통신학 석사  
2007년 경북대학교 컴퓨터공학 박사  
현재, 한국전자통신연구원 선임연구원

관심분야: 컴퓨터 유/무선 통신 프로토콜, 임베디드 시스템, 영상처리.  
Email: phdson@etri.re.kr

**이 은 령**



1998년 경북대학교 전자공학과 학사  
2001년 경북대학교 컴퓨터공학과 석사  
2003년 경북대학교 전자전기컴퓨터학부 박사 수료

현재, 한국전자통신연구원 대경권연구센터 선임연구원  
관심분야: 컴퓨터 유/무선 통신 프로토콜, 임베디드 시스템, 영상처리.  
Email: blue4dia@etri.re.kr

**오 세 진**



2009년 경운대학교 컴퓨터공학과 학사  
2011년 경북대학교 전자전기컴퓨터학부 석사  
현재, 경북대학교 컴퓨터학부 박사과정

관심분야: RFID, 정보보호, 충돌방지, 임베디드 리눅스 시스템.  
Email: 170m3@knu.ac.kr

**안 광 선**



1972년 연세대학교 전기공학과 학사  
1975년 연세대학교 전기공학과 석사  
1980년 연세대학교 전기공학과 박사

현재, 경북대학교 컴퓨터학부 교수  
관심분야: 임베디드 시스템 설계, RFID, 정보보안.  
Email: gsahn@knu.ac.kr