

# A Design and Implementation for a Reliable Data Storage in a Digital Tachograph

Sung Hoon Baek<sup>†</sup> · Myunghee Son<sup>\*\*</sup>

## ABSTRACT

The digital tachograph is a device that automatically records speed and distance of a vehicle, together with the driver's activity and vehicle status at an accident. It records vehicle speed, break status, acceleration, engine RPM, longitude and latitude of GPS, accumulated distance, and so on. European Commission regulation made digital tachographs mandatory for all trucks from 2005. Republic of Korea made digital tachographs mandatory for all new business vehicles from 2011 and is widening the range of vehicles that must install digital tachographs year by year. This device is used to analyze driver's daily driving information and car accidents. Under a car accident that makes the device reliability unpredictable, it is very important to store driving information with maximum reliability for its original mission. We designed and implemented a practical digital tachograph. This paper presents a storage scheme that consists of a first storage device with small capacity at a high reliability and a second storage device with large capacity at a low cost in order to reliably records data with a hardware at a low cost. The first storage device records data in a SLC NAND flash memory in a log-structured style. We present a reverse partial scan that overcomes the slow scan time of log-structured storages at the boot stage. The scheme reduced the scan time of the first storage device by 1/50. In addition, our design includes a scheme that fast stores data at a moment of accident by 1/20 of data transfer time of a normal method.

**Keywords :** Tachograph, Storage

## 디지털 자동차운행기록계에서 안정적인 데이터 저장을 위한 설계 및 구현

백 승 훈<sup>†</sup> · 손 명 희<sup>\*\*</sup>

## 요 약

디지털 자동차운행기록계는 교통안전법에 따라 자동차의 운행상황과 교통사고 상황과 함께 자동차의 속도, 거리, 브레이크 상황, 가속도, GPS 위치 등을 자동적으로 저장장치에 기록하는 장치이다. 유럽에서는 디지털 자동차운행기록계 장착이 2005년부터 모든 트럭에게 의무화되어 있고, 대한민국은 2011년부터 신규로 등록되는 사업용 차량은 의무적으로 장착해야 하며, 해가 지날수록 의무적으로 장착해야하는 자동차의 범위가 확대되어가고 있다. 이 장치는 운전자의 일일 운행 현황 분석 및 사고 분석을 위하여 사용된다. 자동차 사고는 장치의 안정성을 예측불가능하게 한다. 그래서 불확실한 상황아래에서 최대한 안정적으로 데이터를 저장할 수 있는 기술은 매우 중요하다. 우리는 실제 디지털 자동차운행기록계를 설계하고 구현하였다. 본 논문은 이 장치의 설계와 구현에 있어서 저비용의 하드웨어 자원으로 안전하게 대용량 데이터를 저장하기 위해서 저용량이지만 안정적인 1차 저장장치와 대용량을 저비용으로 구현한 2차 저장장치로 구성된 계층적 저장 기법을 제안한다. 1차 저장장치는 용량이 SLC 낸드 플래시 메모리를 사용하여 로그 구조 형식으로 데이터를 저장한다. 로그 구조의 단점인 느린 부팅 문제를 해결하기 위해 역방향 부분 검색 기법을 제시한다. 이 방법은 1차 저장장치의 부팅 시간을 50분의 1로 감소시킨다. 추가적으로 사고 순간의 데이터를 신속하게 데이터를 저장하는 기법도 제시한다. 이 방법으로 저비용의 내장형 시스템에서 사고순간의 운행기록 시간을 일반적인 방법의 저장시간의 1/20만큼 단축하였다.

**키워드 :** 자동차운행기록계, 스토리지

<sup>†</sup> 종신회원 : 중원대학교 컴퓨터시스템공학과 조교수

<sup>\*\*</sup> 비 회 원 : 한국전자통신연구원 SW-SoC 융합기획연구팀 선임연구원

논문접수 : 2012년 6월 1일

수 정 일 : 1차 2012년 8월 23일

심사완료 : 2012년 8월 29일

\* Corresponding Author : Myunghee Son(mhson@etri.re.kr)

## 1. 서 론

자동차운행기록계는 운행상황과 교통사고 상황 등을 자동적으로 저장장치에 기록하는 장치이다. 교통안전법 제55조

에 따라 자동차 운행기록장치의 장착, 운행기록의 보관, 제출, 점검, 분석, 활용에 관한 업무를 효율적으로 실시하기 위하여 자동차 운행기록 및 장치에 관한 관리지침이 고시되었다[1]. 법률에서 정하는 자동차운행기록계는 차량 영상기록장치와 다른 장치로서, 자동차의 주행거리, 속도, 엔진회전수, 브레이크신호, 차량 위치, 방위각, 및 가속도 등을 기록하고 선택적으로 영상카메라와 인터페이스를 할 수 있는 모듈을 갖출 수 있다.

국도해양부는 사업용 차량의 디지털 운행기록계 장착 의무화와 관련된 교통안전법을 공포하였다. 사업용 차량은 자가용에 비하여 사고율이 5배 이상 높고, 교통법규 위반건수가 1.7배 높은 사업용 차량 운전자의 난폭운전습관을 개선하고자 디지털 운행기록계의 의무장착 범위를 넓혀나가고 있다. 유럽위원 규정은 2005년부터 모든 트럭에 운행기록계 장착을 의무화하고 있다[2]. 우리나라에서는 2011년 1월 1일 이후 신규로 등록하는 사업용 차량은 의무적으로 디지털 운행기록계를 장착해야 하며, 기존 등록차량의 경우 버스 및 일반택시는 2012년 12월 31일까지, 개인택시와 화물자동차는 2013년 12월 31일까지 디지털 운행기록계를 장착해야 한다[3]. 이렇게 운행기록계의 의무 장착 범위가 늘어가고 있어 디지털 운행기록계의 연구가 중요하게 되었다.

자동차운행기록계는 매 1초마다 자동차의 운행 상태를 기록해야 하고, 교통사고 순간의 전후 10초 동안 10밀리초 해상도로 운행 상태를 반드시 기록해야 한다. 문제는 심각한 교통사고와 같은 극단적인 사건이 발생하였을 때에도 데이터 손실 없이 안전하게 차량 운행 정보를 기록해야 한다는 것이다.

가볍지 않은 교통사고가 발생하여 자동차운행기록계에 인가되는 전력 공급이 중단되면, 교통사고 이후의 기록이 불가능할 뿐만 아니라, RAM에 버퍼링된 데이터를 저장하지 못해 잃어버리고, 파일 시스템의 손상으로 이전에 저장한 데이터조차도 한꺼번에 잃어버릴 수 있다. 배터리를 이용한 무정전원장치를 사용하여도, 배터리가 충분히 충전이 되어 있는지 않은 상황 또는 배터리의 수명이 다한 상황 또는 배터리 충전회로에 오류가 발생한 경우 또는 심한 충격으로 인해 배터리 연결 케이블이 끊어지는 상황 등의 예기치 않은 불안정한 상황들이 발생하여, 사고 순간의 데이터를 기록하지 못하거나 잃어버릴 수 있다.

최악의 모든 시나리오에서도 데이터를 완벽하게 저장할 수 있는 장치나 방법은 없다. 하지만 우리는 용납할 수 있는 조건하에서 가능한 한 매우 안정적으로 사고순간의 운행 정보를 잃어버리지 않고 저장할 수 있는 기법이 필요하다.

본 논문은 본문에서 실제 자동차운행기록계를 설계하고 구현한 기술 중에서 안정적이고 효율적으로 데이터를 저장하는 저장장치의 구조와 기법을 제시한다. 본 논문의 본문에 들어가기 전에 교통관리법에서 정한 관리지침에 따른 자동차운행기록계의 요구사항과 기존 데이터 저장방법들의 장단점들을 분석해보고자 한다.

## 2. 자동차운행기록계 요구사항

이 장은 전자식 운행기록장치의 저장방법의 이해를 돕기 위해서 자동차 운행기록 및 장치에 관한 관리지침[1]에서 정한 운행기록 요구사항을 소개한다.

전자식 운행기록장치는 1Hz 이상의 성능을 가지는 위치추적장치, 6개월 이상의 1초 단위 데이터를 기록/저장할 수 있는 기억장치, 장치에서 습득한 운행기록 정보를 분석 장비로 전송하기 위하여 범용적으로 유통되는 이동식 기억장치 등을 포함하고 있어야 한다.

운행기록은 문자형으로 구성된 범용자료저장방식인 텍스트 파일에 기록하여야 한다. 파일의 이름에는 연월일, 자동차번호, 운행횟수, 운전자코드, 초구분으로 구성되어 있다. 초구분은, 일반적인 상황에서 1초 단위로 저장되는 정보인지 가속도 충격이 발생한 순간의 10초 전후 동안 0.01초 단위로 저장되는 정보인지를 구분한다.

평상시에는 1초마다 운행정보를 저장해야 하며, 차량진행 방향으로 지구중력가속도의 1.2배 또는 차량좌우측방향으로 지구중력가속도의 0.5배 이상의 가속도가 감지되는 순간의 10초 전후 동안 0.01초 단위로 운행정보를 해당 텍스트 파일에 저장해야 한다. 이 요구사항은 0.01초마다 10초 이상의 운행정보를 메모리 버퍼링하고 있다가 사건이 발생하는 순간에 버퍼링된 과거 10초간의 0.01초 해상도의 데이터를 저장해야하고, 사건 발생 후 10초 동안 0.01초 해상도의 데이터를 저장해야한다. 그리고 6개월 이상의 1초 단위 데이터를 보관할 수 있어야 한다.

매 1초 마다 저장해야하는 최소 데이터양은 70바이트이다. 하지만 데이터 및 위조/변조를 방지하기 위해서 추가되는 암호화 코드 등을 포함하면 약 100바이트가 매 1초마다 저장되어야 한다. 이런 데이터를 6개월 이상 기록할 수 있어야 한다. 그래서 약 1.6GB의 저장공간이 필요하다.

운행정보를 담은 텍스트 파일의 첫줄은 운행기록장치 모델명, 차대번호, 자동차 유형, 자동차 등록번호, 운송사업자 등록번호, 운전자코드로 구성되어 있고, 매초 또는 매 0.01초마다 일일주행거리, 누적주행거리, 차량속도, 분당 엔진회전수, 브레이크 신호, 차량 GPS의 경도/위도/방위각, 가속도, 일일주행거리, 누적주행거리, 정보발생 일시, 차량속도, 기기 상태, 및 제조사가 정한 데이터로 구성된 한 줄의 텍스트형식의 정보가 1초 또는 0.01초마다 해당 텍스트 파일에 덧붙여진다.

운행기록 정보의 위변조를 막기 위하여 보안 관리가 필수적으로 요구된다. 모든 입력된 정보는 사용자 및 제3자가 임의로 위변조를 할 수 없도록 암호화하여 설계/제작되어야 한다. 또한 분해 조작을 막기 위하여 장치의 케이스를 한번 이상 열었는지 확인할 수 있도록 제작되어야 한다.

장치는 장치에서 습득한 운행기록 정보를 분석장비로 전송하기 위한 무선통신 모듈 및 이동식 기억장치를 포함하여야 하며, 영상기록장치에 운행기록 정보를 전송할 수 있는 인터페이스 모듈 또는 영상카메라와 인터페이스할 수 있는 모듈을 가질 수 있다.

### 3. 관련 기술

예기치 않은 여러 가지 상황에서도 최대한 안전하게 운행 정보를 기록해야 하며, 용납할 수 있는 비용으로 이것을 해결해야 한다. 우선, 우리는 기존의 저장 기술들을 비교 분석해 보고자 한다.

#### 3.1 저널링 파일시스템

일반 컴퓨터에서 주로 사용하고 있는 파일 시스템은 저널링 파일 시스템(Journaling File System)이다. EXT3[4], EXT4[5], XFS[6], NTFS[7], IBM JFS[8] 등이 대표적인 저널링 파일 시스템이다. 이것은 저널이라 불리는 원형 로그 영역에 변화된 데이터들을 기록하였다가, 저널에 commit을 하기 전에 저널에 기록한 데이터를 주 파일 시스템 영역으로 옮긴다. 그러므로 데이터를 두 번 기록해야 하기 때문에 쓰기 성능이 감소한다.

갑작스런 전력 중단이 발생하였을 때에 가장 최근에 commit한 순간의 파일 시스템 상태로 신속하게 복귀할 수 있다. 하지만, 가장 최근에 commit한 이후부터 저장된 데이터들은 복원될 수가 없다.

또한 저널링 파일 시스템은 일반 운영체제에서 사용하는 복잡한 파일 시스템이므로, 수십 KB의 메모리의 작은 하드웨어 자원을 가지고 있는 내장형 시스템에서는 사용할 수 없다.

#### 3.2 로그 구조 파일 시스템

로그 구조 파일 시스템(Log-Structured File System)은 파일 시스템의 전영역이 저널영역이다. 변화된 데이터를 항상 순차적으로만 기록한다. JFFS[9], YAFFS[10] 등이 내장형 시스템의 낸드 플래시 메모리를 사용하는 로그 구조의 파일 시스템에 해당한다.

이런 파일 시스템은 일단 저장한 어떤 데이터도 잃어버리지 않는 큰 장점이 있다. 하지만 로그 구조 파일 시스템은 저장장치의 용량에 비례해서 큰 램을 필요로 할 뿐만 아니라, 파일 시스템을 마운트 하는데 소요되는 시간이 매우 길다. 어떤 내장 시스템에서 64MB의 저장공간을 마운트하기 위해서, JFFS2 및 YAFFS는 각각 23초와 3초가 필요하였다 [11]. 만약 그 내장 시스템이 자동차운행기록계에서 필요한 최소 저장용량 1.6GB를 마운트하기 위해서는 1150초 또는 69초가 소요된다. 그러므로 이러한 파일 시스템을 운행기록계에 사용하기 곤란하다.

#### 3.3 FAT 파일 시스템

FAT(File Allocation Table) 파일 시스템에서 DOS에서부터 사용되어 왔다. 이것은 상대적으로 구현이 간단하고 작은 하드웨어 자원으로도 구현이 가능하여 많은 시스템에서 구현되어 있다. 그래서 시스템 간에 데이터를 교환하기 위해서 널리 사용된다. 플로피 디스크, 플래시 메모리 카드, 이동식 저장장치 및 내장형 장치에서 많이 사용되고 있다.

FAT 파일 시스템은 저널링 파일 시스템에 비해서 안정

성이 낮다. 데이터를 저장하고 있는 중간에 전력공급이 중단되면 상당부분의 파일들을 잃어버릴 수 있다.

#### 3.4 무정전전원장치

무정전전원장치는 주로 배터리와 그 배터리의 충전회로로 구성되어 있으며, 시스템에 외부 전력이 중단되어 있을 때에 충전된 배터리를 이용하여 시스템에 전력을 공급한다. 이 장치는 시스템의 안정성을 크게 향상시킬 수 있다.

하지만, 배터리가 충분히 충전이 되어 있는지 않은 상황 또는 배터리의 수명이 다한 상황 또는 심한 충격으로 인해 배터리 연결 케이블이 끊어지는 상황 등의 예기치 않은 불안정한 상황이 발생하여, 사고 순간의 데이터를 기록하지 못하거나 잃어버릴 수도 있다.

### 4. 안정적 저장 장치의 설계와 성능

무정전전원장치는 시스템의 안정성을 크게 높여주지만 그것이 문제를 완전히 제거하지 못한다. 즉, 사고로 인해서 무정전전원장치가 오동작 할 경우에도 저비용의 임베디드 환경에서 안정적으로 데이터를 저장할 수 있는 기술은 여전히 필요하다.

우리는 저비용의 임베디드 장치에서 사용할 수 있는 FAT 파일시스템을 사용하면서 데이터를 안정적으로 기록하기 위하여, 일차적으로 작은 NAND 플래시 메모리에 로그 구조 형식으로 데이터를 기록한 후에, 이차적으로 FAT 파일시스템의 대용량 저장장치로 다시 데이터를 기록하는 데이터 저장 기법을 제시한다. 1차 저장장치인 NAND 플래시 메모리는 로그 구조의 저장 방식을 사용하여 데이터를 가장 안정적으로 기록한다. 그러면서 작은 NAND 플래시 메모리를 사용하기 때문에 긴 마운트 시간을 나타내는 로그 구조 파일 시스템의 단점이 없다. 2차 저장장치는 FAT 파일시스템을 탑재하였기 때문에 작은 메모리와 적은 연산 자원으로도 구현이 가능하였다. 안전하지 않을 수 있는 FAT 파일 시스템에서 데이터 유실이 발생하였을 경우, 1차 저장장치에서 유실된 데이터를 얻을 수 있다.

본 논문은 이러한 이중적 데이터 저장기법으로 설계된 자동차운행기록계의 구체적인 설계를 제시한다. 또한 주어진 구조 내에서 효율적으로 쓰기 성능을 향상하는 기법을 제시하고 성능을 평가한다.

#### 4.1 하드웨어 구조

Fig. 1에는 본 연구에서 구현한 자동차운행기록계의 실물 사진이 나타나 있다. Fig. 2는 이 장치의 내부 구성도이다. 이 자동차운행기록계는 STM32F10을 포함하고 있다. STM32F10은 ARM기반 프로세서와 256KB 플래시 메모리, 64KB SRAM, CAN(Car Area Network, 자동차 영역 네트워크) 인터페이스, SD카드 인터페이스, USART와 I2C 인터페이스를 제공한다.

이 장치는 자동차의 CAN을 지원하는 OBD-II 포트에 연



Fig. 1. The picture of our digital tachograph

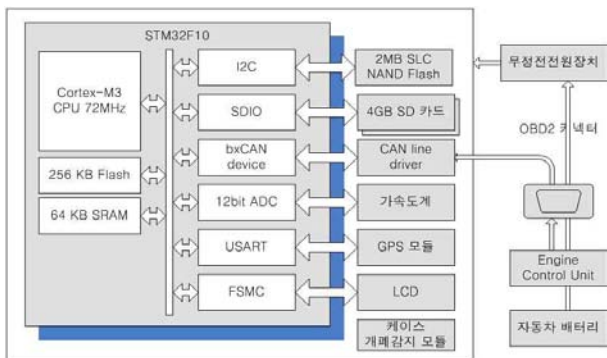


Fig. 2. The hardware block diagram of our digital tachograph

결되어 자동차의 개조 없이, ECU로부터 자동차속력, 분당 회전수, 연료분사량, 브레이크 정보, 및 기타 차량 진단 정보를 얻어 온다. 기존의 자동차운행기록계들은 자동차를 개조하여 속도센서, 연료분사량센서 등을 설치하였기 때문에 고장률이 높고 설치비용이 증가하였다. 하지만 OBD-II를 통한 방식은 자동차를 개조하지 않기 때문에 고장률이 매우 낮고 일반인들이 쉽게 장착할 수 있게 한다.

모든 운행기록 정보는 일차적으로 I2C인터페이스로 연결된 2MB SLC 낸드 플래시 메모리 즉 1차 저장 장치에 저장된다. 그리고 주기적으로 1차 저장 장치에 저장되어진 데이터를 4GB 이상의 두 개의 SD카드로 구성된 2차 저장 장치에 저장한다. SD카드는 내장형과 외장형 두 개가 설치되어 있어 외장형 SD카드는 사용자가 쉽게 데이터를 이동할 수 있게 한다.

1차 저장 장치로서 사용한 2MB SLC 낸드 플래시는 비트 오류율이 매우 낮고 10만 번의 쓰기와 지우기를 할 수 있는 특성을 지니고 있다. FTL 및 파일시스템을 1차 저장 장치에 설치하기보다 저널링 기법을 응용한 구조로 데이터를 저장한다.

2차 저장 장치로서 MLC 낸드 플래시를 직접 제어하는 대신 SD카드를 선택하였다. 최근 대량으로 생산되는 고용량의 MLC 낸드 플래시 메모리는 소프트웨어로는 처리가 불가능한 고도의 에러 정정 하드웨어(1KB당 24비트 오류정정 [12]) 사양을 요구한다. 미래에는 이런 요구조건이 점차 악화될 것으로 전망한다[13]. FTL 등의 소프트웨어를 추가로 필요로 한다. 그래서 개발 기간과 비용을 줄이기 위해서 ECC 및 FTL이 이미 들어 있는 SD카드를 선택하였다.

#### 4.2 안정적 데이터 저장 기법

낸드 플래시 메모리를 사용하는 1차 저장 장치에는 로그 형식으로 데이터를 순차적으로 기록하고, 메타 정보와 데이터를 한꺼번에 하나의 낸드 페이지에 저장하기 때문에 1차 저장 장치는 가장 우수한 안정성을 제공한다. 하지만 용량이 작기 때문에 6개월 이상의 운행기록을 보관해야하는 요구사항을 만족시키지 못한다. 그래서 2차 저장 장치를 두어 이 단점을 해결하였다.

본 운행기록계는 Fig. 3과 같이 3개의 저장장치로 구성되어 있다. 모든 운행 기록은 1차 저장장치로 저장되고, 주기적으로 1차 저장장치에 있는 데이터를 2차 저장장치로 복사한다. 2차 저장 장치는 안정성보다 작은 하드웨어 자원으로 구현이 가능한 FAT 파일 시스템과 SD카드를 선택하였다. 외부 이동식 저장장치는 탈착이 가능한 SD카드를 운행기록장치의 관리지침에 따라서 운행기록 정보를 분석장비로 전송하기 위하여 사용된다.



Fig. 3. The tiered structure of the storage device

##### 1) 낸드 플래시 메모리의 특징

1차 저장 장치로 사용한 낸드 플래시 메모리는 10만 번 쓰기와 읽기를 반복할 수 있고, 오류정정코드가 필요하지 않고, 32개의 블록으로 구성되어 있으며, 각 블록은 256바이트의 256개 페이지들로 구성되어 있는 SLC 낸드이다. 첫 번째와 두 번째 블록(메인 메타 블록)은 시스템의 고정 정보 즉, 장치의 고유식별번호, 자동차번호, 사용자등록번호 등이 저장된다. 나머지 30개의 블록(데이터 블록)은 차량운행정보를 순차적으로 저장하는데 사용된다.

##### 2) 1차 저장장치의 데이터 구조

낸드 플래시 메모리에서는 항상 페이지 단위로만 저장할 수 있으며, 블록 내에서는 페이지를 순차적으로 써내려가야 한다. 한번 기록한 페이지는 지우기 전에는 다시 기록할 수 없다. Fig. 4에는 데이터 블록에 저장되는 페이지 데이터의 구조가 나타나 있다. 모든 페이지 데이터는 항상 같은 구조의 메타 헤더를 포함하고 있다. 메타 헤더 뒤에 운행기록 데이터가 붙는다.

메타 헤더에 저장되는 *메타 버전*은 펌웨어 버전의 변경으로 메타 헤더의 구조가 변경되었을 경우를 인식하기 위하여 사용된다. *체크섬*은 페이지 데이터를 2바이트 단위로 더했을 때 0이 되도록 하는 값을 저장하여 정보의 무결성을 위하여 사용된다.

모든 페이지 데이터의 *페이지 버전*은 다르다. 모든 비트가 1인 값을 최초의 페이지 버전으로 사용하고, 신규 페이지 데이터가 저장될 때마다 1씩 감소된 페이지 버전을 사용한다. 그러므로 페이지 버전의 값이 가장 작은 것이 가장

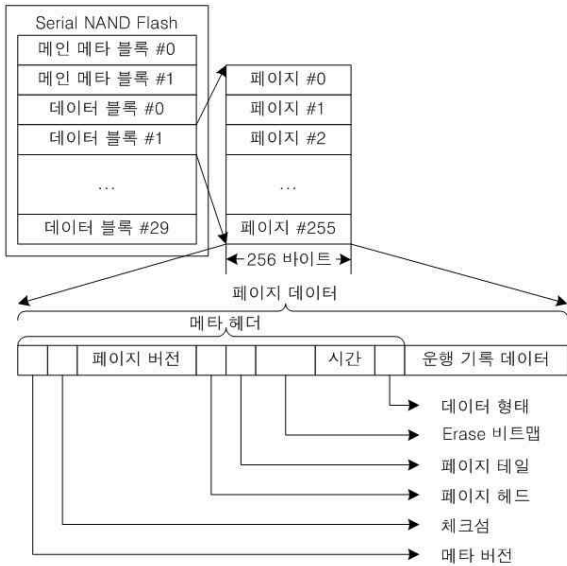


Fig. 4. The structure of the NAND flash and the structure of the page data

최신의 페이지 데이터가 된다. 지워진 페이지들은 모든 비트가 1이므로 무효한 페이지는 최신의 페이지로 인식되지 않는다. 페이지의 버전은 64비트이다. 0.01초단위로 페이지를 저장한다면, 페이지 버전이 0으로 도달하는데 60억년이 걸린다. 그러므로 버전 번호의 오버플로 문제가 없다.

Erase 비트맵 필드에 있는 각 비트는 각 블록이 지워졌는지 아닌지를 기록한다. 총 32개의 블록이 있으므로 Erase 비트맵은 4바이트로 구성되어 있다. 최신의 페이지 데이터에 있는 Erase 비트맵만이 유효하다.

시간 필드는 해당 운행 기록 데이터가 발생한 연월일시분초 및 센티 초를 저장한다. 데이터 형태 필드는 바로 뒤에 오는 데이터가 운행 기록 데이터인지 다른 의미의 데이터인지를 나타낸다.

운행 기록 데이터 필드에는 GPS의 위도, 경도, 방위각, X축 가속도 값, Y축 가속도 값, 엔진의 분당 회전수, 자동차속력, 브레이크 on/off 정보, 연비, 연료소모량, 주행거리, 기어 위치 등이 기록 된다.

### 3) 원형 큐 형식의 페이지 데이터 관리

데이터 블록은 Fig. 5에 나타난 원형 큐처럼 동작한다. 마지막 블록의 마지막 페이지에 데이터를 저장하였다면, 다음은 첫 번째 데이터 블록의 첫 번째 페이지에 데이터를 저장한다. 원형 큐에서 가장 오래된 데이터 및 가장 최신의 데이터를 나타내는 페이지 헤드와 페이지 테일은 Fig. 5에 나타난 것처럼 운행 기록 데이터와 함께 저장된다. 최신의 페이지 데이터에 있는 페이지 헤드 및 페이지 테일이 유효한 값이다.

페이지 테일은 가장 최근에 저장한 페이지의 다음의 빈 페이지를 가리키고, 페이지 헤드는 1차 저장장치에서 2차 저장장치로 복사하지 않은 가장 오래된 페이지를 가리킨다.

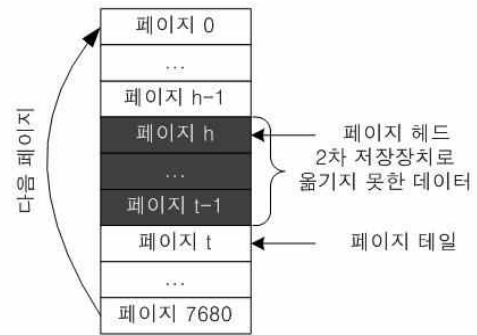


Fig. 5. The structure of the first-tier storage device that operates like a circular queue

페이지 테일과 페이지 헤드로 나타내는 원형 큐의 데이터들은 1차 저장장치에서 2차 저장장치로 옮기지 못한 데이터들이다.

주기적으로 1차 저장장치의 데이터의 운행기록 데이터를 1차 저장장치로 이동한다. 이동이 완료되면 해당 페이지 데이터는 이 원형 큐에서 dequeue된다. 즉 페이지 헤드는 다음 페이지 번호를 가리킨다. 새로운 운행 기록을 담은 페이지 데이터는 페이지 테일이 가리키는 곳에 저장되고, 페이지 테일은 다음 페이지를 가리킨다. 즉, 새로운 데이터는 이 원형 큐에 enqueue된다.

이 원형 큐의 페이지 헤드와 페이지 테일은 신규 데이터 페이지에 신규 운행 기록 데이터와 함께 저장된다. 즉, 가장 최신의 페이지 데이터가 이 원형 큐의 유효한 페이지 헤드와 페이지 테일을 가지고 있다. 시스템이 재시동할 때 가장 최신의 페이지 버전을 갖는 페이지를 검색해서 유효한 페이지 헤드와 페이지 테일을 얻는다.

### 4) 데이터의 안정성

페이지 헤드, 페이지 테일, 및 Erase 비트맵 등으로 구성된 메타 데이터가 운행기록정보와 동시에 한 개의 페이지로서 저장되기 때문에 메타 정보와 데이터의 불일치성이 발생하지 않는다. 갑작스럽게 전원이 중단되어도 재시동 후에 전체 페이지를 검색하여 유효한 체크섬을 가지는 최신의 페이지를 찾아 낼 수 있다. 최신의 페이지의 페이지 헤드와 페이지 테일로부터 아직 2차 저장장치로 옮기지 못한 데이터를 찾아 낼 수 있다.

1차 저장장치에 있는 데이터는 2차 저장장치로 옮겨졌다고 하더라도 지우지 않는다. 그러므로 어떤 예외사항으로 인하여 2차 저장장치의 파일 시스템에서 데이터 손실이 발생한 경우에, 1차 저장장치에 있는 운행기록을 이용할 수 있다. 2MB의 1차 저장장치는 2시간 가량의 운행기록 데이터를 기록할 수 있기 때문에 적어도 사고 순간의 데이터는 1차 저장장치에서 얻어 낼 수 있다.

NAND를 사용하는 1차 저장 장치에는 로그 형식으로 데이터를 순차적으로 기록하고, 메타 정보와 데이터를 한꺼번에 하나의 페이지에 저장하기 때문에 1차 저장 장치는 가장

우수한 안정성을 제공한다. 하지만 1차 저장 장치의 작은 용량 때문에 6개월 이상의 운행기록을 보관해야하는 요구사항을 만족시키지 못한다. 이 문제를 해결하기 위해서 2차 저장 장치를 사용하였다.

5) 재시동 후 최신 데이터 복원

시스템 재시동 후에 1차 저장 장치에서 가장 최신의 메타 정보를 찾아야하는데, 일반적인 로그 구조의 파일 시스템에서는 전체 영역을 검색함으로써 초기화 시간이 매우 많이 소요된다. 운행기록 장치의 초기화 시간이 길어질수록 자동차 시동직후에서 초기화를 완료할 때까지 운행기록을 하지 못하게 된다. 특히 급발진 사고와 같이 시동 직후에 발생하는 사고를 기록하려면 빠른 초기화 시간이 필요하다.

시스템이 재시동을 하게 되면 가장 마지막에 기록한 최신 페이지 데이터가 어느 것인지를 알아내기 위해서, 모든 데이터 블록들의 모든 페이지를 검색하는 방법을 택할 수 있다. 모든 페이지들 중에서 가장 작은 값의 페이지 버전 값을 가지는 페이지가 최신의 페이지 데이터이다.

모든 페이지를 읽어서 검색하는 방법은 시간이 많이 소모된다. 우리는 모든 페이지를 읽는 방법 대신에 일부 페이지만 검색하는 방법을 고안하여 구현하였다. 이 검색 방법은 전체 페이지 중에서 역방향으로 일부만 읽어서 검색하여도 가장 최신의 페이지를 찾는다. 항상 페이지가 순차적으로 기록된다는 특징을 사용하기 때문에 역방향 부분 검색이 가능하다.

Fig. 6은 역방향 부분검색의 간단한 검색 부위 예를 보여준다. 만약 네 개의 블록만 있고, 각 블록 당 페이지 개수는 8개이고, 각 페이지에 적힌 페이지 버전이 Fig. 6과 같을 때, 회색 페이지는 검색을 수행한 페이지이고 흰색 페이지는 검색을 하지 않은 페이지를 나타낸다.

Fig. 6의 예를 들어 검색과정을 설명하면 다음과 같다. 맨 처음 가장 마지막 블록의 첫 낸드 페이지부터 검색을 한다. 작은 값의 페이지 버전이 최신의 페이지이다. 현재 검색 페이지의 페이지 버전이 현재까지 발견한 페이지 버전보다 최신일 경우에 그 블록의 다음 페이지도 읽어서 페이지 버전을 비교한다. 그러므로 블록 3의 모든 페이지들을 읽어야 한다. 여기까지 최신 페이지 버전은 10001이 된다.

Block 0	Block 1	Block 2	Block 3
Page 0 (ver. 10000)	Page 0 (ver. 9992)	Page 0 (empty)	Page 0 (ver. 10008)
Page 1 (ver. 9999)	Page 1 (ver. 9991)	Page 1 (empty)	Page 1 (ver. 10007)
Page 2 (ver. 9998)	Page 2 (ver. 9990)	Page 2 (empty)	Page 2 (ver. 10006)
Page 3 (ver. 9997)	Page 3 (empty)	Page 3 (empty)	Page 3 (ver. 10005)
Page 4 (ver. 9996)	Page 4 (empty)	Page 4 (empty)	Page 4 (ver. 10004)
Page 5 (ver. 9995)	Page 5 (empty)	Page 5 (empty)	Page 5 (ver. 10003)
Page 6 (ver. 9994)	Page 6 (empty)	Page 6 (empty)	Page 6 (ver. 10002)
Page 7 (ver. 9993)	Page 7 (empty)	Page 7 (empty)	Page 7 (ver. 10001)

Fig. 6. An example for searching region in the reverse partial search

```

01: RecentVersion := 0xFFFFFFFFFFFFFFFF
02: LatestBlock := -1
03: LatestPageOffset := -1
04: FOR B := from 마지막 블록 번호 to 시작 블록 번호 DO
05:   B 번째 블록의 첫 번째 페이지를 읽는다.
06:   IF 읽은 페이지의 페이지 버전 >= RecentVersion
07:     Continue
08:   ENDIF
09:   FOR P := from 2 to 블록 당 페이지 개수 DO
10:     B 번째 블록 내에서 P 번째 페이지를 읽는다.
11:     If 읽은 페이지의 페이지 버전 < Recent Version
12:       LatestBlock := B
13:       LatestPageOffset := P
14:     ELSE
15:       BREAK
16:     ENDIF
17:   ENDFOR
18: ENDFOR
    
```

Fig. 7. The pseudo code of the reverse partial search

그 다음 블록2를 검색한다. 블록2에는 모두 지워진 페이지만 있다. 지워진 페이지의 데이터는 모든 비트가 1이므로 버전 값이 가장 높게(가장 오래된) 처리된다. 블록2의 첫 페이지는 현재까지 발견된 최신 페이지 버전, 10001 보다 크기 때문에 그 블록2의 다른 페이지는 검색하지 않고 다음 블록으로 검색을 이동한다.

블록1의 페이지0에서 페이지2는 최신 페이지 버전보다 작기 때문에 블록1의 다른 페이지들은 검색의 대상이 된다. 그리하여 페이지3까지 검색을 한다. 여기까지 최신 페이지 버전은 9990이 된다. 그 다음 블록0의 첫 페이지를 검색한다. 그 페이지 버전은 9990보다 최신이 아니므로, 블록0의 나머지 페이지들은 검색을 하지 않는다.

이 검색 방법은 Fig. 7의 의사코드로 표현될 수 있다. 마지막 블록부터 시작 블록까지 역순으로(4행) 각 블록의 첫 페이지를 검색한다(5행). 읽은 페이지의 페이지 버전이 현재까지 검색된 페이지 버전보다 작지 않다면(6행) 다음 블록을 검색하고(7행), 그렇지 않다면 그 블록의 나머지 페이지들에 대해서(9행) 순차적으로 읽고(10행) 그 페이지의 페이지 버전이 현재까지 검색된 페이지 버전보다 작다면, 그 블록의 페이지를 가장 최근의 페이지로 설정한다(12~13행).

항상 페이지는 순차적으로 저장되어 있으므로, 역방향 부분 검색 과정에 있어서, 어떤 블록의 전체 페이지를 검색해야 하는 블록은, 맨 처음 검색된 유효 블록 또는 가장 최신의 페이지가 있는 블록뿐이다. 그러므로 대부분의 블록에 대해서 첫 페이지만 읽어서 비교하면 되기 때문에 최신 데이터 검색 속도가 상당히 개선된다.

Fig. 8은 순방향 검색 방법과 역방향 부분 검색 방법의 성능차이를 보여준다. 순방향 검색은 첫 블록부터 마지막 블록까지 모든 블록마다 모든 페이지를 읽어서 최신 페이지를 검색하는 방법이다. 순차 검색은 1900ms가 소요되는 반면에 역방향 부분 검색 방법은 35ms가 소요되었다.

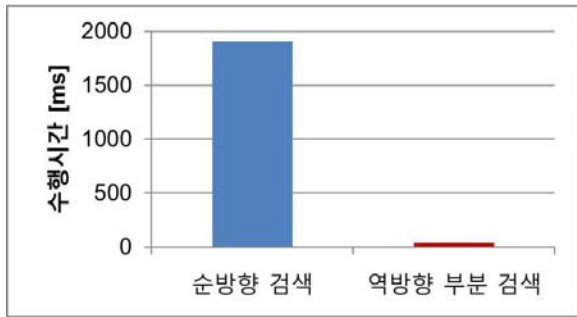


Fig. 8. Execution time for the forward search and the reverse partial search

### 4.3 충격 직후 운행정보 기록

자동차운행기록계는 가속도 값이 지정된 임계치를 넘는 순간 이후 10초 동안의 운행기록을 0.01초 간격으로 저장해야 한다. 이 요구 사항을 달성하기 위해서 본 시스템은 다음과 같이 구현하였다.

0.01초 간격의 운행기록을 램에 10초 동안 보관한다. 즉 10초 이상 지난 과거 운행기록은 램에서 제거한다. 충격이 발생하는 순간에 과거 0.01초간격의 10초 동안의 운행기록은 최대한 빨리 기록 한 후에, 앞으로 10초 후까지 0.01초 간격의 운행기록을 저장한다. 그런데 저사양의 내장형 시스템에서는 빠르게 기록하는 것이 문제가 된다. 본 장치에서는 기본적인 저장방식으로는 충격 전 10초 과거 운행 정보를 저장하기 위해서 40초의 시간이 소모되었다. 그러나 저장 효율을 높이는 기술을 사용하여 2초 내에 과거 10초 운행정보를 저장하였다.

기본적인 저장방법에서는 매순간(0.01초마다) 발생하는 운행정보를 1차 저장장치의 한 개의 낸드 페이지(256바이트)에 저장하고, 그것을 다시 2차 저장장치에 텍스트형식으로 저장하기 위해서, 한 개의 낸드 페이지 단위로 파일을 열어 운행기록을 저장하고 파일을 닫는 방식을 사용하였다.

저장 성능을 높이기 위해서 우리는 두 가지 방법을 결합하여 사용하였다. 첫 번째 방법은 매순간(0.01초 단위) 발생하는 운행정보를 1차 저장장치의 각각 다른 페이지에 저장하는 것이 아니라, 여러 순간의 운행정보를 모아서 하나의 낸드 페이지에 저장한다. 그리하여 1차 저장장치에서 소모되는 시간을 단축하였다.

SD카드에 작은 데이터를 자주 저장하는 것보다 큰 데이터를 한꺼번에 저장하는 것이 성능을 크게 높인다. 극소의 RAM자원을 가지는 SD카드는 블록 매핑 FTL 또는 블록 매핑과 유사한 FTL을 사용하기 때문에, 작은 데이터를 저장하여도 항상 큰 기록 단위(블록)로 저장하기 때문이다.

두 번째 성능 향상 기법은 앞 문단에서 언급한 SD카드의 특징을 이용하였다. 이 두 번째 기법에서는, 파일을 쓰기 위해서 파일을 열고 (file open) 닫는 (file close) 빈도를 낮춤으로써 쓰기 성능을 향상시킨다. 즉, 1차 저장장치의 각 낸드 페이지 단위(256바이트)로 파일을 열고 닫는 것이 아니라 여러 페이지 단위로 한번 파일을 열고 닫는다. 저사양의

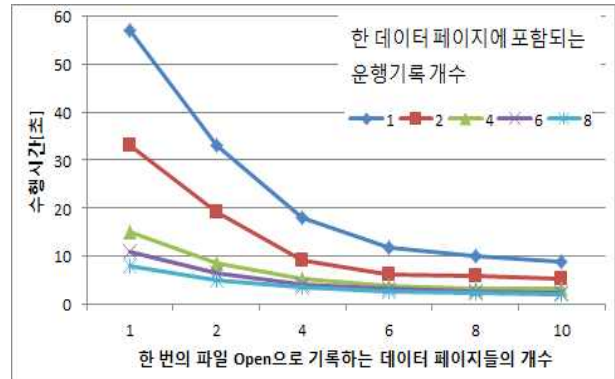


Fig. 9. Performance comparison for the number of driving records stored in a single NAND page of the first-tier storage and the frequency for opening and closing a file

내장형 시스템에서는 파일을 닫을 때마다 버퍼에 기록된 데이터가 저장장치로 기록된다. 즉 자주 파일을 닫을수록 작은 데이터들이 자주 SD카드에 전송되어 많은 블록 쓰기가 발생하므로 2차 저장장치의 성능이 느리게 된다.

파일을 너무 오랫동안 닫지 않으면 램에 버퍼된 데이터가 저장장치로의 전송이 지연되므로, 사고 등의 예외상황에 대한 데이터 신뢰성이 낮아 질 수 있다. 반면 파일을 닫는 빈도가 빠르면 데이터의 신뢰성은 높아지나 성능이 낮아지게 된다. 그런데 느린 성능은 오히려 신뢰성을 낮출 수도 있다.

Fig. 9에서 앞에서 소개한 두 가지 방법의 성능 향상 효과를 잘 보여준다. 범례에 나타난 각 라벨은 첫 번째 성능 향상 기법에 따라 1차 저장장치의 하나의 낸드 페이지에 저장되는 운행기록의 개수를 나타낸다. 여러 운행기록을 모아서 하나의 낸드 페이지에 저장함으로써 1차 저장장치에서 소모되는 시간을 줄이기 때문에 성능이 크게 향상됨을 볼 수 있다.

y축은 두 번째 성능 향상 기법에 따른 수행시간을 보여준다. x축에 나타난 개수의 낸드 페이지 단위로 2차 저장장치의 파일을 열고 닫는다.

하나의 낸드 페이지에 8개의 운행기록을 모아서 기록하고 10개 낸드 페이지 단위로 파일을 열어 10개의 낸드 페이지에 담긴 운행기록을 2차 저장장치로 저장하고 파일을 닫음으로써, 이런 방법을 사용하지 않을 때(Fig. 9에서 한 데이터페이지에 포함되는 운행기록개수는 1, x축은 1) 57초가 소모되는 성능을 2초로 낮출 수 있었다.

## 5. 결 론

우리는 안정성과 저비용이라는 두 가지 양립적인 요소를 얻기 위하여 두 가지 저장장치를 사용하였다. 1차 저장장치는 용량은 작지만 매우 안정적인 구조를 갖추고 있었다. 2차 저장장치는 안정성은 낮지만 저비용의 내장 장치로서 구현이 가능한 구조를 가지면서 6개월분 데이터를 저장할 수

있는 용량을 갖추고 있다. 본 연구는 1차 저장장치의 안정성과 2차 저장장치의 저비용 장점을 융합함으로써 저비용으로 최대한 안정적으로 자동차 운행정보를 기록할 수 있는 방법을 제시하였다.

1차 저장장치는 메타 정보와 데이터를 한꺼번에 하나의 낸드 페이지에 저장하여 안정성을 높였으며, 부팅 속도를 높이기 위하여 제시한 역방향 부분 검색은 순방향 검색 시간을 50분의 1로 단축하였다.

사고발생 직후 1차 및 2차 저장장치로 데이터를 기록하는 속도는 사고순간의 데이터의 안정성과 관련 있다. 빠르게 저장할수록 예기치 않은 상황으로 인한 데이터 손실을 방지할 수 있다. 데이터 저장 속도를 높이기 위하여 한 개의 낸드 페이지에 여러 기록을 모아서 기록하고, 파일의 열기와 닫기 빈도를 조절함으로써 기본적인 방식에 비해서 쓰기 시간을 20분의 1수준으로 감소하였다.

본 논문에서 제안하는 기록 방식은 디지털 운행기록장치 뿐만 아니라 최대한 안정적이면서 저비용의 하드웨어로 데이터를 저장해야하는 응용분야에도 적용하여 사용할 수 있을 것이다.

### 참 고 문 헌

[1] Ministry of Land, Transport and Maritime Affairs, "Management guidelines on driving record and devices", Notification 2010-667 of Ministry of Land, Transport and Maritime Affairs, September, 2010.

[2] K. Lemke, C. Paar, and M. Wolf, "Embedded Security in Cars", Springer, 2006.

[3] Serim Park, "Mandatory of installing digital tachographs for commercial cars", ITToday, July 22, 2010.

[4] M. Cao, T. Y. Ts'o, B. Pulavarty, S. Bhattacharya, A. Dilger, and A. Tomas, "State of the Art: Where we are with the Ext3 filesystem", Proc. of the Linux Symposium, pp.69-96, July, 2005.

[5] A. Mathur, M. Cao, S. Bhattacharya, A. Dilger, and A. Tomas, "The new ext4 filesystem: current status and future plans", Proc. of the Linux Symposium, pp.21-33, June, 2007.

[6] Daniel Robbins, "Common threads: Advanced filesystem implementor's guide, Part 9, Introducing XFS". Developer Works. IBM. Jan., 2002.

[7] R. Russon and Y. Fledel, "NTFS Documentation", <http://www.sourceforge.net>, June, 2004.

[8] S. Best, "JFS Overview", IBM, Jan., 2000.

[9] D. Woodhouse, "JFFS: The journaling flash file system", Proc. of the Linux Symposium, 2001.

[10] C. Manning, "YAFFS: Yet Another Flash File System", <http://www.aleph1.co.uk/yaffs>, 2004.

[11] S.H. Lim, K.H. Kim, "An Efficient NAND Flash File System for Flash Memory Storage", IEEE Trans. on Computers, pp.906-912, Vol.55, No.7, July, 2006.

[12] C.C. Wu, "Quality Comparison of SLC, MLC and eMLC", Flash Memory Summit, Aug., 2011.

[13] Michael Abraham, "NAND Flash Trends for SSD/Enterprise", Flash Memory Summit 2010 proceedings, August, 2010.



### 백 승 훈

e-mail : shbaek@jwu.ac.kr

1997년 경북대학교 전자공학과(학사)

1999년 한국과학기술원 전기및전자공학과 (공학석사)

1999년~2005년 한국전자통신연구원 컴퓨터시스템연구부 연구원

2008년 한국과학기술원 전기및전자공학전공(공학박사)

2008년~2011년 삼성전자 메모리사업부 책임연구원

2011년~현 재 중원대학교 컴퓨터시스템공학과 조교수

관심분야: 컴퓨터저장장치, 운영체제, 친환경자동차



### 손 명 희

e-mail : mhson@etri.re.kr

1998년 충남대학교 항공우주공학과(학사)

2000년 충남대학교 컴퓨터공학과(공학석사)

2005년 충남대학교 정보통신공학과 (공학박사)

2011~2012년 (주)스마트에코텍 대표

2000년~현 재 한국전자통신연구원 SW-SoC 융합기획연구팀 선임연구원

관심분야: 유무선통신프로토콜, 친환경자동차, 계산지능알고리즘