
CF(CompactFlash)메모리 카드를 이용한 부트 시스템 구현에 관한 연구

2. 부트로더의 구현

이광철* · 김영길**

The Study of the Implementation of the Boot System Using CF(CompactFlash) memory card
2. Implementation of the Boot Loader

Kwang-cheol Lee* · Young-kil Kim**

요 약

CF 메모리카드를 위한 부트 시스템의 시스템을 초기화하고 CF 카드를 확인한 후 CF 메모리카드의 종류를 확인하여 적절한 CF 메모리카드입이 확인되면 파일 시스템을 초기화하고 원하는 시스템 프로그램을 읽어 들여 DRAM 영역에 재배치하는 부트로더를 구현하였다. 시스템 프로그램을 CF 메모리카드에서 읽어 들이는 과정에서 시스템 프로그램의 무결성을 보장해주는 알고리즘을 넣어 시스템이 좀더 안정적으로 동작할 수 있다. 또한 CF 메모리카드를 사용하고 파일 시스템 표준을 준수하기 때문에 이기종간의 호환성이 유지되며 이를 양산 시점에 그대로 이용할 수 있기 때문에 양산성이 향상되는 특징이 있다.

ABSTRACT

This paper describe the implementation of the boot loader for the boot system using CF memory card. This boot loader consist of the system initialization, CF card checking, CF memory card checking, file system and the program relocater. This boot loader increase the system stability with program consistency checking algorithm in the read phase from the CF memory card. And this system have the compatibility in CF memory card file system, so system manufacturing productivity increase.

키워드

CF 메모리카드, boot loader, system update

1. 서 론

본 연구실에서 CNS(Car Navigation System)용으로 구현한 하드웨어는 고비용의 NOR 형태의 Flash 메모리를 많이 사용하지 않으면서 시스템의 성능을 높이고 시스템 소프트웨어의 유지 보수를 쉽게 하기 위해서 CF 메모리카드와 DRAM을

도입하였다. 이런 시스템의 소프트웨어를 구동하기 위해서는 전용의 부트 시스템을 구현해야 한다. 따라서 CF 메모리카드를 이용한 부트 시스템을 연구하게 되었다.

CF 메모리카드를 이용한 부트 시스템은 적은 양의 NOR 형태의 Flash 메모리 또는 ROM

* 아주대학교 의용공학과
접수일자 : 2003. 9. 17

** 아주대학교 전자공학과

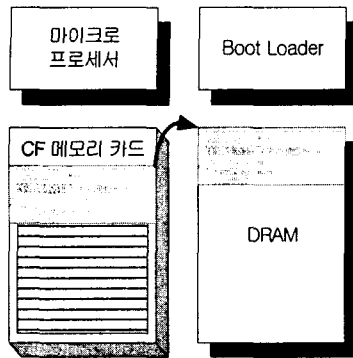


그림 4. 시스템의 구현 방법
Fig. 1 System Implementation method

(Read Only Memory)과 CF 메모리카드 및 DRAM(Dynamic Random Access Memory)을 이용하여 시스템을 구성한다. 이 시스템은 시스템 프로그램을 CF 메모리카드에 저장해두고 시스템 부팅 시에 이를 DRAM에 복사하여 마이크로프로세서가 이를 실행하는 시스템이다. 이런 동작을 수행하기 위해서는 먼저 위의 하드웨어를 구성한다. 여기에 CF 메모리카드의 표준을 따르는 CF 메모리카드 제어 소프트웨어를 구성한다. CF 메모리카드 제어 소프트웨어를 기반으로 CF 메모리카드의 파일 시스템 액세스 블록을 구성한다. 이 블록을 이용해 CF 메모리카드에 저장되어 있는 파일을 읽어 DRAM에 복사한다. DRAM에 복사된 시스템 프로그램을 마이크로프로세서가 바로 실행할 수 있는 형태로 재배치하고 시스템 동작을 초기화하는 부트로더(boot loader)를 구현한다. 그림 1은 이와같은 동작을 수행하는 시스템의 블록도를 나타낸 것이다.

이와 함께 시스템의 유지 보수를 위하여 시스템 프로그램을 업데이트하는 방안도 고려한다.

II. CF 메모리카드 제어

CF 메모리카드 제어를 위한 하드웨어 및 소프트웨어 인터페이스 구성은 그림 2와 같다.

CF 메모리카드의 파일을 액세스하기 위해서는 CF 메모리카드의 전기적 구조적 특성 표준에 맞

춰 하드웨어를 제어해야 한다.

CF 카드를 제어하기 위해서는 기본적으로 각 블록 액세스를 위한 시스템 버스와 CF 카드 버스 간 충돌을 방지하는 기능 및 CF 카드에서 발생하는 인터럽트를 수행하는 기능을 구현한다. 이를 기반으로 CF 카드의 블록 읽기/쓰기 기능을 구현한다. 이 기능은 CF 메모리카드의 주기능인 데이터 저장 장치로 사용하는 측면에서 가장 중요한 블록이므로 성능을 고려하여 최적화를 수행한다.

또한 CF 카드의 전원 제어와 인식 블록이 필요하다. CF 카드의 표준에는 CIS(Card Information Structure)의 내용에 따라서 CF 카드의 형태와 카드가 제공하는 기능을 알아낼 수 있는 방법을 제공한다. CIS는 tuple이라는 정보들이 linked list 형태로 구성되어 있는데 각 tuple은 카드의 특성

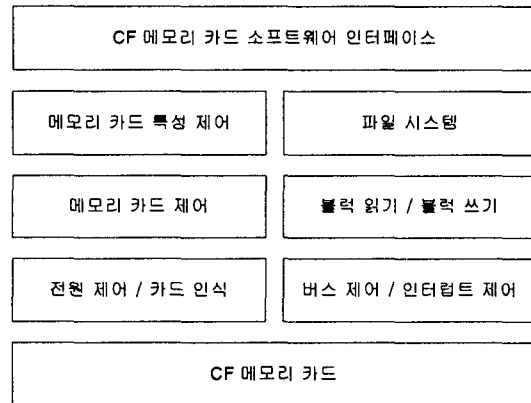


그림 5. CF 메모리카드 인터페이스 블록도
Fig. 2 CF memory card interface block diagram

을 정의하는 정보를 가지고 있다. 이 논문에서 사용하는 CF 메모리카드는 ATA(Advanced Technology Attachment) 형태로 액세스가 가능한 메모리카드이다. 그런데 CF 카드 중에는 네트워크 어댑터나 RAM과 같은 다른 기능을 수행하는 디바이스들이 많이 존재한다. 따라서 시스템이 CF 메모리카드를 인식하기 위해서는 CF 카드 CIS의 tuple을 읽어 분석하는 과정이 필요하다.

앞에서 기술한 기능을 이용하여 CF 메모리카드 CIS의 tuple을 분석하여 해당 정보에 맞게 CF 메모리카드 인터페이스를 재설정하고, CF 메모리카

드에 의한 인터럽트 제어 부분을 초기화 한 후 CF 메모리카드 내의 데이터 영역에 있는 실제 데이터를 읽어 낸다.

CF 메모리카드의 데이터 영역은 임의의 구조를 갖는 파일시스템을 구현할 수도 있지만 시스템간의 호환을 위해서 널리 사용되는 파일시스템을 구현한다. 이 논문에서는 Microsoft사의 MS-DOS 시스템 및 Windows 시스템을 비롯하여 여러 시스템의 표준으로 쓰이는 FAT12 / FAT16 / FAT32 (File Allocation Table 12 / 16 / 32)를 구현하였다. 이 기능은 그림 2의 파일시스템 블록에 해당한다. 이 블록에서는 다른 종류의 파일시스템에 대한 처리도 한다. 즉 지원하지 않는 파일시스템인 경우 상위 인터페이스에 해당 정보를 제공한다. 파일을 시스템에서 수정할 필요가 없는 경우에는 읽기 전용 파일 시스템을 구현한다. 일반적으로 쓰기 기능을 고려하는 경우 기능 구현이 더욱 어렵기 때문이다.

CF 메모리카드 제어를 위한 최상위 블록인 CF 메모리카드 소프트웨어 인터페이스는 파일 시스템 종류에 상관없이 파일을 읽고 쓰기 위한 사용자 인터페이스이다.

III. 부트로더 구현

본 논문에서는 그림 3의 블록도에 나타난 시스템을 바탕으로 부트로더를 구현한다. 부트로더는 그림 4와 같은 순서로 수행된다. 소프트웨어의 기능적으로는 시스템 초기화, 카드 확인, 메모리카드 확인, CF 메모리카드 초기화, 파일 시스템 초기화, 시스템 프로그램 파일 확인/로드/시작 등이 있다.

III-1. 시스템 초기화

시스템 초기화 단계에서는 시스템 각 부분의 전원을 초기 상태로 조정하고 각 시스템 블록의 초기화를 수행한다. DRAM을 사용하는 경우 DRAM의 동작 속도 및 각 신호의 동작 타이밍에 맞게 설정한다. 이 부분은 시스템 전체의 동작 성능에 중요하게 작용하므로 설정에 주의가 필요하다.

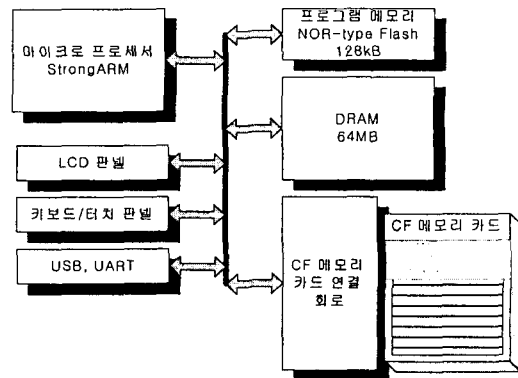


그림 6. 구현한 시스템 블록도
Fig. 3 Implemented system block diagram

다. 또한 LCD 시스템도 초기화하고 시스템 로고 및 사용자 정보를 표시하는 기능을 구현한다. 이외에 외부 시스템과 데이터 교환이 필요한 경우 USB와 UART 시스템을 초기화하여 데이터 송수신이 가능한 상태로 초기화를 한다.

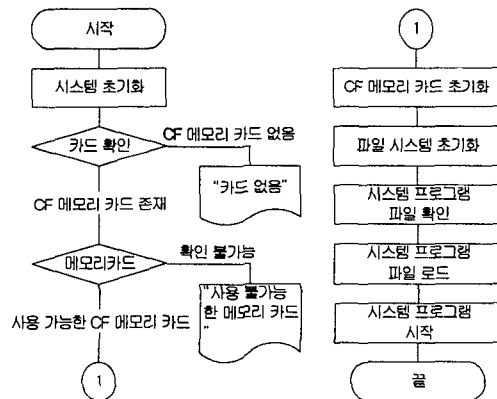


그림 7. 시스템 부트로더의 순서도
Fig. 4 Flowchart of the system boot loader

위와 같이 시스템이 기본적인 초기화를 수행한 후 CF 메모리카드를 초기화하고 CF 메모리카드 내의 시스템 파일을 확인한다. 시스템 파일을 열어 파일 포맷과 파일 데이터의 무결성을 확인한다. 이는 시스템 오동작 또는 사용자 실수에 의하여 시스템 설계 시에 예상 못했던 불안정한 시스템 상황을 배제하기 위한 방안이다.

III-2. CF 메모리카드 초기화

시스템 초기화에 이어서 카드 확인 동작을 수행한다. 카드의 유무를 먼저 판별한다. CF 카드가 CF 카드 소켓에 꽂혀있는 상태는 상태확인 전용 신호를 통하여 알 수 있다. 특정한 카드의 경우 자체 초기화 시간이 길기 때문에 일정 기간 동안 CF 카드 시스템이 안정될 때까지 대기해야 한다. 카드가 연결되어 있지 않은 경우 더 이상 시스템을 동작 시킬 수 없으므로 에러 메시지를 표시하고 사용자가 CF 메모리카드를 장착하기를 기다리는 수밖에 없다. 물론 일정 시간이 지나면 자동으로 시스템을 정지시키는 기능을 넣어야 한다.

카드가 들어 있음을 확인하고 나서는 원하는 CF 메모리카드인지를 확인하는 기능이 필요하다. 이는 그림 2의 CF 메모리카드 인터페이스 블록도의 메모리카드 특성 제어 블록에 해당하는 기능으로서 CF 카드 중에 Flash 메모리를 이용한 메모리카드임을 확인한다. RAM을 확장하는 CF 메모리카드는 비휘발성 특징이 필요한 시스템 부트 이미지 데이터를 저장할 수 없으므로 시스템 부트에 사용할 수 없고, 네트워크 어댑터 또는 모뎀 등과 같은 I/O 기능을 수행하는 CF 카드는 데이터를 저장할 수 없기 때문이다. 또한 CF 메모리카드 중에 ATA 인터페이스를 사용하지 않고 직접 메모리를 제어해야 하는 FFS (Flash Filing Systems) 또는 FTL (Flash Translation Layer)을 따로 구현해야 하는 카드도 배제해야 한다. 만약에 원하는 종류의 CF 메모리카드가 아닌 경우 사용자가 인지하여 조치를 취할 수 있도록 메시지를 표시해 준다.

위의 과정을 거친 후 마이크로프로세서가 CF 메모리카드를 사용하기 위한 초기화를 수행한다. 대량의 데이터를 읽어 저장하기 위한 메모리를 할당한다.

CF 메모리카드가 파일 구조를 지원하는 일반 저장 장치처럼 사용할 수 있도록 하기 위해 파일 시스템을 초기화한다. 그림 5와 같은 구조를 가지고 있는 FAT 시스템의 경우 MBR(Master Boot Record), PBR(Partition Boot Record)을 확인하여 파일 시스템 처리를 위한 각종 환경 변수들을 설정한다. 또한 FAT 부분을 확인하여 파일 시스템

에 문제가 있는 지를 확인한다. 이 과정에서 FAT12/FAT16/FAT32 시스템인지, 루트 디렉토리 엔트리의 위치 및 클러스터 크기, 섹터의 크기와 각 부분의 시작 위치 등을 확인하여 정확하게 파일 시스템을 액세스할 수 있도록 준비한다.

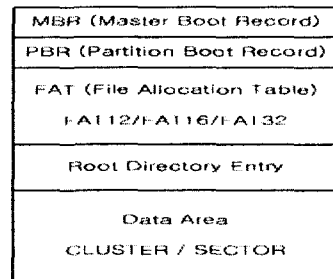


그림 8. FAT 파일시스템의 구조
Fig. 5 FAT file system structure

III-3. 시스템 프로그램 확인 및 로드

파일시스템 초기화가 완료되면 파일 시스템을 열어서 미리 정해둔 위치에 시스템 프로그램 파일이 존재하는 지 확인한다. 시스템 프로그램이 존재하지 않는 경우 에러 메시지를 표시하여 사용자에게 문제가 생겼음을 알린다. 시스템 프로그램 파일이 존재하면 파일 내용이 손상되었는지 확인한다. 파일 내용이 손상된 경우 초기 시스템 설계시에 고려하지 못했던 상황이 발생할 수 있기 때문에 이런 상황은 배제해야 한다. 파일 포맷을 확인하고 CRC(Cyclic Redundancy Check) 또는 블록 체크섬(Block Check-sum) 등의 방법을 사용하여 데이터의 무결성을 확인한다. 본 논문에서는 WindowsCE 이미지 파일용 부트로더를 작성하였다. WindowsCE 이미지 파일의 포맷은 표 1과 같다. 이 표를 살펴보면 이미지 파일을 확인할 수 있는 sync 바이트("B000FF\n")가 있고 데이터의 에러를 확인할 수 있는 Record Checksum이 있다.

표 1. WindowsCE 바이너리 이미지 포맷 (.bin)
Table. 1 WindowsCE Binary Image Format (.bin)

Field	Length (bytes)	Description
Sync	7	B, 0, 0, 0, F, F, '\n'
Header:		
Address	4	Physical starting address of the image.
Length	4	Physical length, in bytes, of the image.
Record:		
Address	4	Physical starting address of data record. If this value is zero, the record address is the end of the file, and record length contains the starting address of the image.
Length	4	Length of record data, in bytes.
Checksum	4	Signed 32-bit sum of record data bytes.
Data	Length	Record data.

파일이 안전함을 확인하고 파일을 열었으면 이를 DRAM의 특정 위치로 로드한다. 이때 속도의 향상을 위하여 DRAM의 버스트 모드를 이용하여 블록단위로 복사한다. 이때 사용자에게 시스템이 시작되는 상황을 알리기 위하여 상태표시를 한다.

III-4. 시스템 프로그램 시작

CF 메모리카드에 있던 시스템 프로그램이 DRAM으로 로드가 마무리되면 모든 시스템 동작을 로드된 프로그램으로 넘긴다. 시스템 프로그램 내에서는 DRAM 초기화를 수행하지 말아야 한다.

IV. 시스템 소프트웨어 업데이트

CF 메모리카드에 시스템 프로그램을 넣는 방법은 여러 가지로 구성할 수 있다. 본 논문에서 제안

하는 시스템은 CF 메모리카드의 파일시스템이 FAT 표준을 따르기 때문에 그림 6의 오른 쪽에 있는 Read/Write 장치를 이용하여 CF 메모리카드를 PC에 연결하여 바로 시스템 프로그램을 업데이트하는 것이다. 최근에는 이들 장치의 하드웨어 제조사가 제공하는 전용의 드라이버 파일을 PC에 설치하지 않아도 이들 장치를 PC에 연결하기만 하면 PC는 이 장치를 대용량 저장 장치로 인식하여 연결된 저장 매체에 데이터를 읽고 쓸 수 있다.



그림 9. SanDisk사의 CF 메모리카드와 CF 메모리카드 Read/Write 장치
Fig. 6 SanDisk's CF memory card and CF memory card Read/Write tool

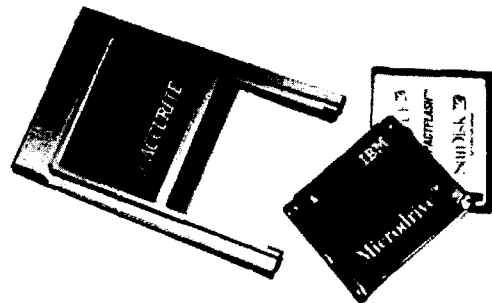


그림 10. PCMCIA-CF 어댑터
Fig. 7 PCMCIA-to-CF adaptor

요즘은 그림 7의 장치와 같은 노트북 컴퓨터 등의 PCMCIA 슬롯에 바로 연결해서 사용할 수 있는 PCMCIA-to-CF 어댑터도 시중에서 쉽게 구할 수 있다. 이와 같은 방법은 임베디드 시스템의 양산 시점에서도 동일하게 적용할 수 있어 안정성 및 양산성 증대에 도움이 된다.

또는 사용자가 USB 또는 UART 등을 이용하

여 외부의 시스템과 데이터 통신을 수행하여 업데이트하도록 구성할 수도 있다. 이러한 방법을 사용할 경우 개발자는 USB 또는 UART를 통한 외부 시스템과의 데이터 통신 프로토콜을 구현해야 하고 PC 등 외부 시스템의 프로그램도 동시에 개발이 되어야 한다. 만약의 경우 사용자의 잘못에 의하여 CF 메모리카드 내의 시스템 프로그램 파일이 손상이 되더라도 다시 업데이트를 수행할 수 있으므로 이 부분에 대해서는 안전성이 확보된 시스템이 된다. 또한 업데이트 후에 사용자의 추가 조작 없이 바로 다음 시스템 부팅 시점에 적용이 되기 때문에 좋은 방법이라고 할 수 있다.

본 논문에서 구현한 WindowsCE 운영체제를 기반으로 하는 장치의 경우에는 WindowsCE 운영체제가 동작하는 상태에서는 PC 기반 ActiveSync 소프트웨어를 이용하여 CF 메모리카드의 파일을 액세스할 수 있는 방법이 이미 구현되어 있다. PC에서 ActiveSync 소프트웨어를 실행하고 USB 또는 RS-232C 케이블을 이용하여 WindowsCE 운영체제 기반 장치와 연결하고 WindowsCE 운영체제 기반 장치의 repllog.exe 프로그램을 실행하면 PC의 Windows Explorer 창에서 WindowsCE 장치내의 데이터를 업데이트할 수 있다.

V. 실험 및 고찰

본 연구실에서 CNS용으로 구현한 CF 메모리카드를 이용한 부트 시스템과 본 논문에서 제안하는 방안이 기초하여 구현한 부트로더를 구현하여 실험을 해보았다. 구현된 시스템에 전원을 인가하면 CF 메모리카드에 미리 저장해놓은 WindowsCE 부트 이미지 파일인 NK.bin 파일을 이용하여 WindowsCE 운영체제가 부팅이 되고 정상 작동하고 있는 화면은 그림 8의 사진에서 확인할 수 있다.

WindowsCE가 떠있는 상황에서 PC와 연결하여 파일을 전송하는 실험을 하기 위하여 PC에 Microsoft사에서 제공하는 ActiveSync 소프트웨어를 설치한다. 이 실험에서 사용한 소프트웨어의

버전은 Microsoft ActiveSync Version 3.5 (Build1176)이다. 구현한 시스템을 WindowsCE 운영체제로 부팅하고 PC와 USB 케이블을 이용하여 연결하고 WindowsCE 시스템 내의 접속 프로그

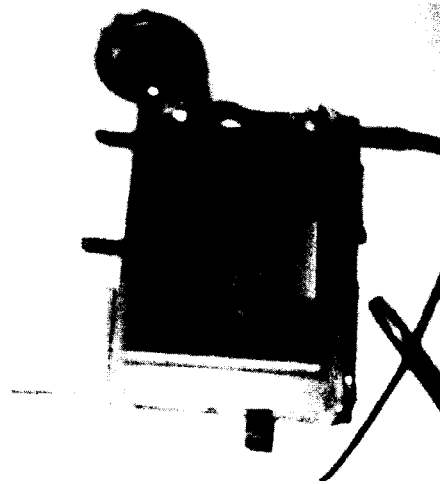


그림 8. WindowsCE를 실행시킨 화면
Fig. 8 WindowsCE Execution

램(repllog.exe)을 실행시킨다. 이때 PC에는 그림 9와 같은 접속 화면이 나타난다. Guest로 접속하여 파일 복사 동작을 수행한다.

위와 같은 방법으로 연결이 되면 Windows의 Windows Explorer 창에 Mobile Device라는 항목이 생성이 되며 이 항목을 선택하면 다시 “내 컴퓨터”라는 항목이 보인다. 이 부분은 WindowsCE 시스템의 “내 컴퓨터” 항목이다. 다시 이 항목을 선택하면 여러 개의 항목이 보일 것이다. 여기서 “기억 장치 카드” 항목은 WindowsCE 시스템에 연결된 대용량 메모리 장치를 나타낸다. 실험하고 있는 시스템의 경우 CF 메모리카드를 가리킨다. 따라서 “기억 장치 카드” 항목을 선택하여 파일을 쓰거나 읽으면 CF 메모리카드의 파일을 쓰거나 읽는 것이다.

선택된 기억 장치 카드의 디렉토리 및 파일 항목 중에 NK.bin 파일이 존재하는 것을 확인할 수 있다(그림10).

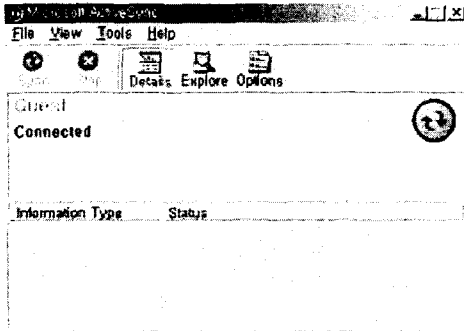


그림 12. ActiveSync 접속화면
Fig. 9 ActiveSync connection

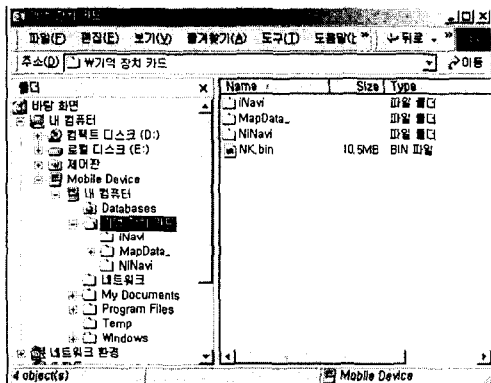


그림 13. 탐색기를 통한 WindowsCE 액세스
Fig. 10 File access in the WindowsCE system using Windows Explorer

이제 Windows Explorer를 통하여 파일 업로드 또는 다운로드를 해볼 수 있다. 그림 11은 NK.bin 파일을 다운로드하는 화면(왼쪽)과 업로드하는 화면(오른쪽)이다.

하지만 USB를 통한 ActiveSync를 이용해 파

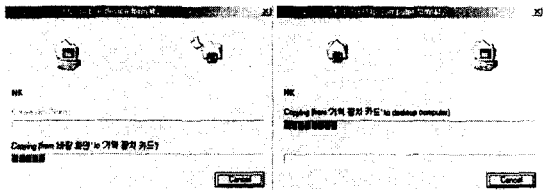


그림 11. 다운로드(좌) 및 업로드(우) 실행 화면
Fig. 11 Execution of the download and upload

일을 전송하는 경우도 그리 빠르지 않다. CF 메모리 카드로 파일 전송을 가장 빠르게 할 수 있는 방

법은 노트북 컴퓨터에 PCMCIA-to-CF 어댑터를 연결하여 파일을 전송하는 경우이다. 이는 물리적인 전송 매체의 속도차이가 가장 큰 원인이기 때문에 확실하게 가장 빠른 방법이라고 할 수 있다.

VI. 결론

본 구현은 CF 메모리카드를 사용한 부트 시스템의 부트로더를 구현한 것이다. CF 메모리 카드를 제어하여 CF 메모리카드의 파일시스템 구조에 맞게 파일을 액세스하여 시스템 프로그램을 읽어 DRAM에 재배치시키고 DRAM의 시스템 프로그램을 시작하는 부트로더를 구현하였다.

시스템 초기화 블록에 특히 많은 배려를 해야 하며 CF 메모리 카드의 표준 및 파일 시스템 표준을 따르는 부분에 많은 관심을 기울여 부트로더의 구조를 잡아야 한다.

동작 성능 면에서 초기 부팅 시에 프로그램이 안전한 지 확인하는 과정과 로드하는 과정에서 시간 소모가 있다는 것이 단점이다. 하지만 이 부분 역시 시스템 부팅 시에 시스템의 안전성을 확인한다는 측면에서 그 정도의 시간은 할당되어도 큰 문제가 없다고 판단된다.

양산 측면에서는 CF 메모리카드를 PC 등의 신뢰할 수 있는 시스템에서 프로그램 파일만 다운로드하기 때문에 안전하고 빠르게 진행된다. 또한 시스템 프로그램 메모리를 시스템과는 독립적으로 확인 가능하다는 점도 장점이라고 할 수 있다. 기존의 시스템은 양산 시 Flash 메모리 또는 ROM에 프로그램을 써넣는 시간이 너무 오래 걸렸고 쓰는 도중에 여러 가지 문제점이 발생할 수 있었다. 하지만 CF 메모리카드를 이용한 부트 시스템을 사용할 경우 양산 시점에서 대용량의 프로그램 메모리를 처리하는 과정에서 생길 수 있는 여러 가지 문제점이 발생하지 않으므로 양산 신뢰성을 높일 수 있다.

향후 연구방향은 CF 메모리 카드내의 여러 종류의 파일 시스템을 대응할 수 있도록 개선하는 방향과 WindowsCE 이외의 다른 운영체제를 위한 부트로더를 구현하는 것이다.

참고 문헌

- [1] Building Powerful Platforms with WindowsCE, James Y. Wilson, Aspi Havewala, 2001
- [2] CompactFlash™ Memory Card Product Manual, SanDisk, 2001
- [3] Microsoft Windows CE 3.0 Software Developer Documentation, Microsoft Corporation, 2000
- [4] PCMCIA PRIMER, Larry Levine, 1995
- [5] The PCMCIA Developer's Guide Second Edition, Michael T. Mori, W. Dean Welder, Sycard Technology, 1995

저자 소개



이광철(Kwang-cheol Lee)

2001년 8월 아주대학교 전자공학과 졸업(공학사)

2003년 8월 아주대학교 산업대학원 정보전자공학과 졸업(공학석사)

2003년 9월 ~ 현재 아주대학교 대학원 의용공학과 박사과정

※ 관심분야 : 리얼타임 시스템, 임베디드시스템 설계, 네트워크 시스템



김영길(Young-kil Kim)

1978년 고려대학교 전자공학과 졸업(공학사)

1980년 한국 과학원 산업전자공학과 졸업(공학석사)

1984년 ENST(France) 졸업(공학박사)

1978년 제 12회 기술고시 통신직: 체신부 정책국 전자 통신 전담반 통신기좌

1978년 9월 ~ 현재 아주대학교 전자공학과 정교수

※ 관심분야 : 초음파 의료시스템, 의료용 통신 시스템, 선박 전자