스마트 카드 터미널를 위한 EMV Specification의 구현

박 창 현[†] · 두 명 택[††] · 이 백 순[†††] · 권 오 규[††††]

요 약

스마트 카드는 독립적인 프로세서와 메모리를 내장한 독립적인 컴퓨팅 장치로서 최근 금융권에서는 기존의 마그네틱 카드를 대체하려는 움직임을 보이고 있다. 이러한 스마트 카드에 대한 표준은 세계 3대 국제 결제 기관인 Europay, Mastercard, Visa가 공통으로 제안한 EMV 표준을 따르고 있다. 본 논문에서는 스마트 카드용 터미널 상에서 다양한 응용 프로그램을 지원할 수 있도록 하는 EMV 표준 프로토콜의 구현에 대해 기술한다. 또한, 본 논문에서 구현한 EMV 터미널이 국제 규격 인증 과정을 통과한 내용을 기술한다.

An Implementation of EMV Specifications for Smartcard Terminals

Chang-Hyeon Park[†] · Myung-Taek Doo[††] · Baek-Soon Lee[†††] · Oh-kyu Kwon[††††]

ABSTRACT

Smartcard is a kind of computing device with it's own processor and memory. Thus, the international money market is going to accomodate the use of smartcards instead of traditional magnetic cards for the future money market. EMV is a standard protocol for smardcards, proposed by Europay, Mastercard, and Visa, which are three famous international settlement organizations. This paper presents the implementation of EMV specifications for the smartcard terminals for supporting various application programs. This paper shows that the implemented EMV terminal passes the international approval tests.

키워드 : EMV Spec., 스마트 카드(Smartcard), 카드 터미널(Card Terminal)

1. 서 론

EMV[1-4]란 세계 3대의 국제 결제 기관인, Europay, Mastercard와 VISA의 약어로써 스마트카드를 세계적으로 확산시키기 위한 공통의 표준안을 말한다. 이 표준안은 ISO 7816[5-8]과 VISA, MASTER 카드프로그램 등 여러 응용 프로그램을 위한 공통의 플랫폼을 제공한다. 기존의 마그네틱 카드는 결제 방법에 따라 다소 차이가 있으나, 간단한 데이터만 카드로부터 수신하고, 연산 장치로서의 카드는 없었다. 즉, 데이터의 송수신은 MSR(Magnetic Stripe Reader)로 처리했으며, 카드 자체에 데이터를 저장하는 장치로 이용하지는 않는다. 카드의 이용은 단순히 머천트(merchant)에서 고객 관리용이나, 소액 지불, 신용결제 등의 기능만 지원할 뿐이다. 또한 응용프로그램이 고유의 카드에 내장되어, 사용자들이 2가지 이상의 서비스를 받기 위해서는 2가지 이상의 카드를 지니고 다녀야 하는 번거러움을 가지고 있다. 최근 국제 동향은 이러한 문제점을 하나의 카드로 통합하자는 움직임을 보이고 있다. VISA International은 다가오는 2005년부터 마그네틱 카드를 더 이상 생산하지 않겠다고 발표하였으며, MASTER Card에서도 2004년부터는 마그네틱 카드를 더 이상 생산하지 않겠다고 공식 발표한 것이 바로 그것이다.

스마트 카드는 기존의 마그네틱 카드와 다르게 자체 프로세서와 메모리를 내장한 독립적인 컴퓨팅장치로서, 카드 자체에 부착된 8개의 접속단자를 통하여 카드 터미널과 송수신 한다. 이것은 전자 상거래나, 전자 지갑, 향후 구현될 전자 주민등록증, 개인 의료카드, 신용 결제 지불 시스템과 같이 각별한 보안을 요구하는 여러 분야에서 활용될 수 있다. 스마트 카드는 저장장치에 자체 파일 시스템을 가지고 있으면서, 하나의 카드가 여러 개의 응용프로그램을 지원하기 위한 멀티세션을 지원한다. 여러 개의 응용프로그램을 하나의 카드에서 지원한다는 것은 모든 프로그램이 표준화된 프로토콜을 가지고 데이터 송수신을 해야 하며, 보안을 위한 키의 암호화, 복호화 프로그램 수정 등도 표준 프로토콜을 이용하도록 해야 한다는 것을 의미한다. 현재 국내에서는 금융과 관련된 몇몇 기업에서 스마트카드를 위한 EMV 인증과 핀패드(PIN pad) 모듈을 위한 EMV 인증이 승인된 경우가 있으며, 카드조회 터미널 상에 EMV 표준을 구현한

† 종신회원 : 영남대학교 컴퓨터공학과 교수
†† 준 회 원 : 영남대학교 컴퓨터공학과
††† 정 회 원 : (주)J-Technology 경영관리 본부 상무이사
†††† 정 회 원 : (주)J-Technology 연구개발본부/기술연구소 연구원

논문접수 : 2002년 9월 27일, 심사완료 : 2002년 12월 5일

예는 없다[14].

본 논문에서는 EMV 표준을 만족하는 스마트 카드를 위해서 EMV 표준 Spec.을 만족하는 EMV 터미널의 구현에 대해서 기술한다. 본 논문에서 기술하는 EMV Spec. 구현 범위는 다양한 응용프로그램을 지원하기 위한 표준 프로토콜(Level 1)을 설계, 구현하는 것이다.

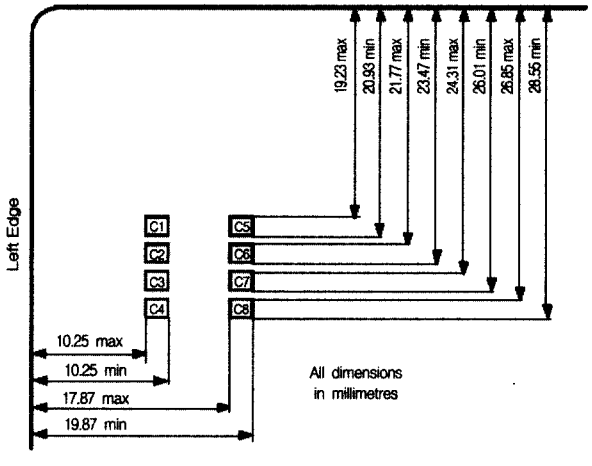
본 논문 2장에서는 물리적인 전송부분부터 송수신을 위한 명령어에 대한내용을 소개하고, 3장에서는 EMV 터미널의 하드웨어 구조와 소프트웨어 구현 방법을 기술한다. 4장에서 본 논문의 EMV 터미널이 국제적 인증과정을 통과한 절차를 제시하고, 5장에서 결론을 기술한다.

2. EMV Spec. 개요

스마트 카드 표준은 ISO 7816에 정하고 있으며, EMV는 ISO 7816 기반 위에 작성된 표준안이다. EMV Book 1에서는 스마트카드와 터미널간의 인터페이스 요구사항을 정하고 있으며[1], Book 2에서는 보안을 위한 키의 관리에 대해[2], Book 3에서는 응용프로그램[3], Book 4에서는 카드홀더 및 공급자, 발급자의 인터페이스 요구사항을 정한다[4]. 이장에서는 EMV 터미널 개발을 위한 EMV Spec.의 내용중에서 본 논문의 구현 목표인 표준 프로토콜(Level 1)과 관련된 내용을 기술한다.

2.1 전기적 인터페이스

스마트 카드와 터미널이 정확한 데이터를 송수신 하기 위해서는, 물리적인 인터페이스를 우선 구현하여야 한다[9].

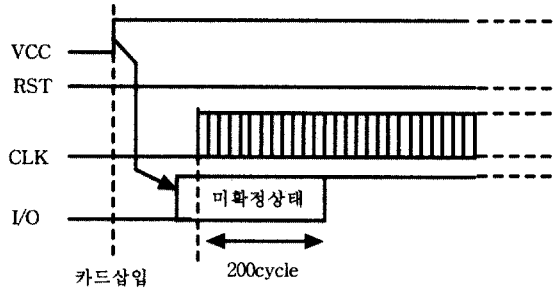


(그림 1) IC 카드의 접점

스마트 카드 (그림 1)는 8개의 접점을 가지고 있으며, 이중 사용하는 접점은 6개이고 2개는 나중을 위해 미리 확보된 접점(Resered Future Use ; RFU)이다[5, 6]. 각각을 살펴보면 C1은 카드로 전원을 공급하기 위한 전압단자(VCC) 이며, C5는 접지(GND)이며, C2는 카드로부터 초기 신호를 보내기 위한 리셋(RST), C6은 프로그래밍 전압(VPP)이며, C3는 클럭(CLK), C7은 데이터 송수신을 위한 I/O 단자이다.

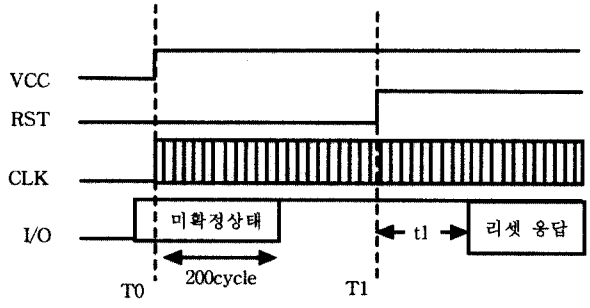
2.2 문자의 물리적 전송

스마트 카드와 터미널이 실제 데이터를 송수신 하기 전에 카드와 터미널간의 처리는 다음과 같다.

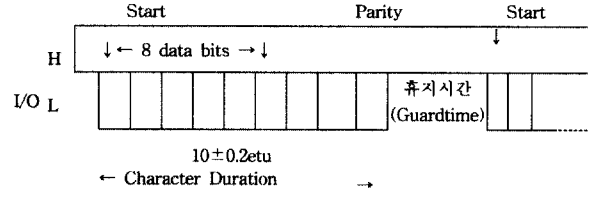


(그림 2) IC 카드의 초기신호

(그림 2)처럼 터미널에서 스마트 카드에 전기적인 VCC을 공급해주면 일정시간 이후에 CLK이 활성화되면서, I/O 라인상의 데이터는 미확정상태(indeterminate)상태로 된다. 나중에 터미널에서 리셋 신호를 카드쪽으로 전송하면, (그림 3)에서처럼 미결정상태이던 I/O 라인 상에 카드로부터 받는 초기 문자들의 집합인 리셋응답(Answer To Reset ; ATR)을 수신 받는다. 수신 받는 문자열들은 ISO 7816에 규정된 비트 간격(bit duration)에 따라서 한 개의 문자를 전송하기 위한 절차를 준수해야 한다.



(그림 3) 리셋 순서



(그림 4) 문자 프레임

(그림 4)는 각 문자를 전송하는 시간을 정의한 것으로, 한 개의 비트를 전송하는데 걸리는 기본 시간 단위(elementry

time unit, 이하 etu로 표기)로 정의한다.

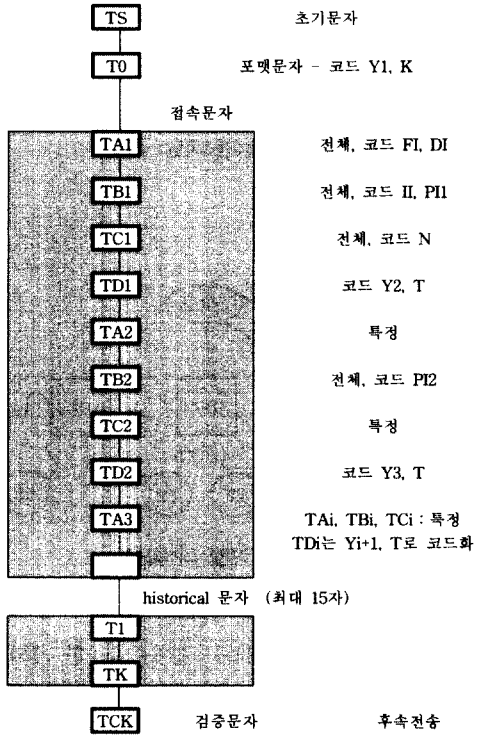
한 개의 문자를 전송하기 위해서는 카드와 터미널 양간에 규약이 필요한데, 먼저 데이터의 전송이 없는 I/O는 High를 유지하고 있다가 전송이 일어나는 쪽에서 I/O를 1etu동안 Low상태를 유지한다. 이때 수신 쪽은 데이터를 읽을 준비를 하고, 송신측은 한 개의 비트씩 I/O를 1etu에 준하여 변경하게 된다. 한 개의 비트는 1etu±0.2etu를 초과할 수 없으며, 만약 초과할 경우에는 잘못된 데이터의 전달이 일어나면서, 수신측에서 세션종료(deactivation sequence)가 일어난다. 비트간격은 EMV Spec.에 정의된 계산식을 준수해야 하는데 그 계산식은 다음과 같다.

$$\text{초기 } etu = \frac{372}{f} \text{초}, \ f = \text{입력클럭}(Hz)$$

$$\text{현재의 } etu = \frac{F}{Df} \text{초}, \ f = \text{입력클럭}(Hz)$$

초기 etu는 카드와 터미널간에 처음 송수신을 위한 etu이고, ATR 값을 받은 이후에 현재 etu값으로 변할 수 있다. f는 터미널에서 카드로 공급하는 주파수이며, 전송인자 D와 F는 etu 결정 인자로서 ATR값을 통하여 얻을 수 있으며 디폴트 값은 D = 1, F = 372이다. 한 개의 문자는 1개의 시작비트(start bit)와 8개의 데이터비트(data bit), 1개의 패리티 비트(parity bit)를 전송후에 대기시간(guard time)으로 I/O를 High 상태로 유지하게 된다.

2.3 리셋 응답(ATR)



(그림 5) ATR의 구조

터미널이 카드에 리셋 신호를 보낸 일정시간(42000CLK 이하)이 지나면 ATR값이 카드에서 송신된다. 이 값은 후에 카드 세션을 처리하기 위한 특별한 요소들을 보유하고 있다.

(그림 5)는 ATR의 구조로서, 첫 번째 값은 초기문자(TS)이며 이것은 '3B' 혹은 '3F' 값을 가지게 된다. '3B'는 순참조(direct convention)를 뜻하는 문자로서 I/O 라인상의 High 상태가 논리적인 1의 값이고 데이터 바이트의 least significant bit(LSB)가 첫 번째 비트임을 나타낸다. TS문자가 '3F' 이면 I/O 라인상의 Low 상태가 논리적인 1이며, 데이터 바이트의 most significant bit(MSB) 첫 번째가 비트 8 임을 나타낸다[1].

두 번째 수신하는 문자는 T0(format character) 문자이며, 첫 번째 상위 nibble(b5~b8)은 후속 인터페이스 문자(TA1~TD1)의 존재 여부를 1로 표시한다. 두 번째 하위 nibble(b1~b4)는 ATR 수신후에 전송되는 일련의 문자열(historical characters)의 수를 나타낸다.

〈표 1〉 T0 문자의 예

b8	b7	b6	b5	b4	b3	b2	b1
0	1	1	0	X	X	X	X

<표 1>와 같은 T0 문자일 경우 상위 nibble(b8~b5) 값이 0110이므로 후속 문자는 TB1, TC1 값이 수신 받게 된다. TA1 값은 etu계산의 인자값인 F와 D값을 가지며, TB1는 프로그래밍 전압값을 정하며, TC1은 한 개의 문자에서 대기시간(guard time)을 정하는데 추가적인 대기시간(extra guard time)을 정하며, TD1값은 T0와 마찬가지로 후속문자를 상위 nibble(b8~b5)에서 나타내고 자신은 프로토콜 T = 0와 T = 1을 결정하게 된다. 프로토콜 T = 0은 문자전송 프로토콜로서 송수신 데이터의 양이 적을 경우에 사용되는 프로토콜이고, T = 1은 블록 전송 프로토콜로서 송수신 데이터를 블록단위로 전송하는 프로토콜이다.

TD1 값을 수신하면, 후속문자 TA2~TD2값이 결정 된다. TA2는 추가모드(negotiable) 에서 작동여부를 판단하고, TB2는 VPP값을 결정하는데 사용되며, TC2는 T = 0 프로토콜에서 작업 대기 시간(work waiting time)을 전송한다. 작업 대기 시간은 카드에서 송신되는 임의의 문자의 시작비트의 에지(start leading edge)에서 카드 혹은 터미널에서 응답하는 이전문자의 시작비트의 에지간의 최대 간격이다. TD2의 경우 하위 nibble은 프로토콜 T = 1이 사용될 경우 1로 세트하며, 상위 nibble은 후속 문자 TA3~TD3 값을 결정한다.

TA3는 프로토콜 T = 1이 사용될 경우의 정보필드 크기수(information field size)를 결정한다. 정보필드 사이즈란 블록이 전송될 때, 데이터 바이트의 크기를 의미한다. TB3는 프로토콜 T = 1상의 연속되는 문자간의 최대 대기 시간

(Character Waiting Time ; CWT)과 블록과 블록간의 최대 대기 시간(Block Waiting Time ; BWT)을 계산하는데 필요한 인자를 전송한다. TC3는 블록 에러검출 형태를 전송하며, 모든 후속문자의 전송이 완료되면 마지막 데이터인 TCK 문자가 ATR의 마지막으로 전송된다. TCK 문자는 에러검출 문자로서 ATR로 보내진 데이터의 무결성을 검사하는 데 사용되며, 모든 데이터 바이트를 XOR(exclusive-OR)한 값을 전송한다.

2.4 Protocol Type T

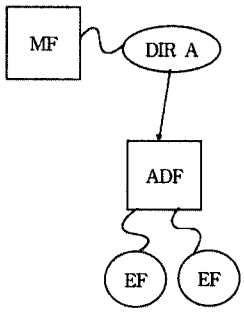
2.3절에서 인터페이스 바이트 TDi(i = 1, 2, 3, …)의 하위 nibble은 프로토콜 타입 T를 나타내는데 전송 프로토콜을 처리하기 위해 사용되는 규칙은 T = 0는 비동기 반이중 문자 전송 프로토콜을 나타내며, T = 1은 비동기 반이중 블록 전송 프로토콜을 나타내며, T = 2, T = 3은 향후 전이중 운용을 위한 프로토콜이며, T = 4은 향상된 비동기 반이중 문자 전송 프로토콜, T = 5~T = 13은 향후사용을 위해 유보중인 프로토콜, T = 14는 ISO에 의해 표준화되지 않은 프로토콜을 위해 유보, T = 15는 향후 확장을 위해 유보된 프로토콜이다. 이중 프로토콜 T = 0, T = 1이 현재 스마트 카드에서 사용되는 주요 프로토콜인데, 물리적 계층, 데이터 링크 계층, 전송계층을 갖는다[7].

물리적 계층은 본 논문의 2.2절에서 소개한 문자의 물리적 전송부분을 말하며, 데이터 링크 계층은 타이밍과 특정 옵션 그리고 T = 0, T = 1상의 프로토콜의 에러 처리부분을 말한다. T = 0상의 데이터링크 계층의 EMV 규약은 타이밍 부분으로 축약할수 있는데, 터미널에 의해 스마트카드로 보내진 두 개의 연속 문자 시작비트 상한에지 사이의 최소구간은 ATR에서 전달된 TC1의 값에 의해 표시된 대로 12~266etu가 되어야 한다. 또한 스마트 카드에 의해 터미널로 보내진 두 개의 연속문자 시작비트의 상한에지 사이의 최소구간은 12etu가 되어야 한다. 스마트 카드에 의해 보내진 임의문자 시작비트의 상한에지와 터미널에 의해 보내진 이전문자(previous character)의 시작문자의 상한에지 사이의 작업대기 시간은 최대 960×D×WI = 9600etu를 초과해서는 안된다(전송 인자인 D와 WI값의 디폴트는 각각 1, 10임). 반대 방향으로 보내진 두 개의 연속문자 시작 비트의 상한에지 사이의 최소구간은 16etu가 되어야한다.

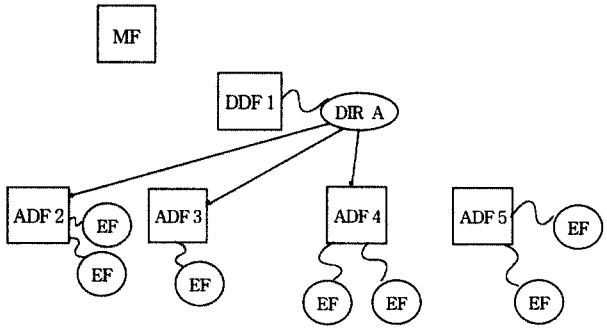
2.5 파일 구조

스마트 카드의 파일 구조는 전용파일(Dedicated File ; DF) 및 요소파일(Elementary File ; EF)로 구성되며, 트리 형태의 디렉터리 구조를 갖는다. 전용파일은 특정 응용프로그램이나 서비스와 관련된 요소파일의 기능적인 그룹으로 나누어지는 파일로써 디렉터리값을 가진다. 루트 레벨밑에는 디렉터리 파일(Directory File ; DIR)인 DF가 필수적으로 존재하는데, 신용, 직불, 금융공동망, 전자화폐, 인증서, 자행 및 제휴업무 등 여러 개의 응용프로그램을 추가적으로 DIR DF에 삽입할수 있다. 특히 루트레벨의 DF를 MF(Master File)라고 부른다. 요소파일은 반드시 전용파일아래에 존재하며, 내부요소파일(internal EF)과 작업 요소파일(working EF), 전자화폐 파일로 나눌 수 있다. 내부 요소파일이란 운영체제에 의해서만 이용되는 내용접근불가능 파일이며, 키(KEY) 파일과 사용자입력(PIN) 파일이 있다[7]. 작업 요소 파일은 표준형태의 요소파일로서 터미널에 의하여 읽기/수정 이 가능한 응용프로그램 데이터 저장용 파일이며, 전자화폐 파일은 전자화폐만을 위한 고유데이터를 관리하는 요소파일이다.

스마트 카드에 한 개의 응용프로그램을 가질 경우의 디렉터리 구조는 (그림 6)과 같다. MF(Master File)는 전용파일이며 그의 요소파일로 하위 디렉터리 구조를 나타내고 있는 DIR A를 갖는다. DIR A값을 스마트 카드로부터 읽어오면 하위 디렉터리 구조를 파악할 수 있는데, 한 개의 응용프로그램을 가지므로 ADF(Application DF)를 하나만 가지면서 하위에 요소파일인 EF를 두 개를 가진다. 두 개의 EF는 응용프로그램 데이터를 위한 파일이다.



(그림 6) 단일 응용 파일 구조

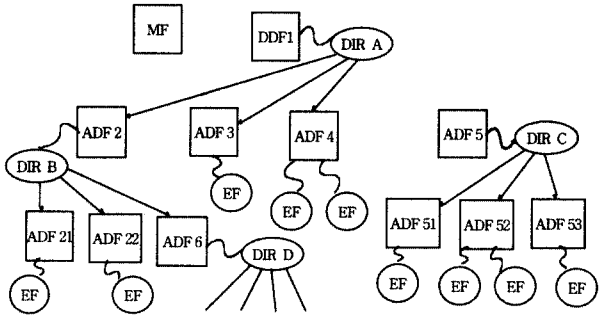


(그림 7) 단일 레벨 디렉터리

(그림 7)에서 보인 단일 레벨 디렉터리는 스마트 카드에 한 개의 디렉토리 파일이 존재하는 경우이다. 한 개의 디렉터리 파일로서 스마트 카드의 파일 구조를 파악할 수 있으며, 그림에서 MF에서 DF로 DIR A를 읽어오면 하위에 3개의 응용프로그램이 존재하는 것을 알 수 있다. ADF5는 스

마트 카드의 EMV Spec.의 범위를 벗어난 부분으로써, 이것은 DIR A의 파일로 참조하는 것이 아니라, 특정 주소로 참조한다. 이 파일은 예외로 처리한다.

(그림 8)은 하위 디렉토리 파일이 그 파일하위의 디렉토리 파일을 보유하는 형태를 나타내고 있다. 한 개의 디렉터리 정의파일(directory definition file)인 DDF1는 MF를 나타내며, 이 파일에 EF로서 DIR A 파일을 가진다. DIR A 파일을 스마트 카드로부터 읽어오면 하위에 응용 프로그램 2개와 DDF2의 파일을 읽어 올 수 있는데, DDF2 파일은 EF로서 DIR B를 EF로 갖고, 나머지 두 개의 응용프로그램은 각각 왼쪽부터 1개, 2개의 데이터 EF를 갖는다. DDF2 파일의 EF값을 읽어오면 하위에 응용 프로그램 2개와 DDF6값을 알 수 있는데 이것 또한 하위에 디렉터리를 DIR D로서 표시된다.



(그림 8) 3레벨 디렉터리

현재 선택된 응용프로그램은 가장 최근에 선택된 DF이며, 다른 DF가 선택될 경우는 현재의 DF에서 새로 선택된 DF로 이동하게 된다. 처음 터미널에서 스마트카드로 VCC가 인가되면 MF가 자동적으로 선택되고 현재의 DF가 된다. DF에 대한 선택방법은 DF 이름으로 바로 찾는 방법과 FID (File IDentification)를 통하여 찾는방법이 있다. DF 이름에 의한 파일을 참조할 경우 임의의 DF는 5~16bytes의 중복되지 않은 AID(Application IDentification)에 의해 선택·참조될 수 있는데, 각 AID는 스마트 카드내에서 유일하여야 하며, EMV Spec.에 정의된 DF이름 '1PAY.SYS.DDF01'으로 참조한다. 참조 방법은 5byte의 RID(Registered Application Provider Identifier)와 0~11byte의 PIX(Proprietary Application Identifier Extension)를 합쳐서 AID를 만들고, 이렇게 만든 AID를 SELECT, READ 명령에 인자로 전송하게 된다.

RID는 국제거래일 경우(<표 2>) 등록범위(4 bits : 1010)과 등록된 응용프로그램의 제공자 번호(36bits(9BCD))으로 구성되며, 국내거래일 경우(<표 3>) 등록형태(4bit : 1101), 국가코드(12bit(3BCD) : ISO3166 참조), 국가 표준화 기구(국가기술 품질원) 지정필드(24bit(6BCD))로 구성된다.

〈표 2〉 국제 거래 참조

구 분		RID	PIX
VISA	신 용 DF	A000000003	1010
	직 불 DF	A000000003	3010
MASTER	신 용 DF	A000000004	1010
	직 불 DF	A000000004	3060

〈표 3〉 국내 거래 참조

구 분	RID	PIX
신 용 DF	D410650990	1010
직 불 DF	D410650990	3010
금융공동망 DF	D410650990	0010
전자화폐 DF	D410650990	0020
인 증 서 DF	D410650990	0030

2.6 명령어

스마트 카드와 터미널간의 데이터 송수신을 위해서는 특정 명령어와 데이터 구조가 필요하다. 메시지는 전송 프로토콜(T = 0, T = 1)에 따라 터미널과 카드사이에서 전송되며, 터미널 및 카드는 물리적 계층, 데이터링크 계층, 전송계층을 수행한다. 응용프로그램의 실행을 위해서는 추가 계층인 어플리케이션 계층이 터미널에 존재하여야 하는데, 어플리케이션 계층의 수행단계는 명령어 전송, 명령어 처리 및 명령어에 대한 카드 응답으로 구성된다. 따라서 특정 응답은 특정 명령어와 일치하여야 하며, 명령어-응답의 쌍으로 표시된다.

APDU(Application Protocol Data Unit)는 카드에서 터미널 또는 터미널에서 카드로의 전송되는 명령어 메시지나 응답 메시지를 표시하는 단위로써, 명령어 - 응답 쌍내에 존재하는 명령어 메시지 및 응답 메시지에 포함된 데이터를 송수신한다. APDU는 4바이트의 필수헤더와 가변길이 추가 데이터(conditional body)로 구성된다[10].

CLA	INS	P1	P2	Lc	Data	Le
← Mandatory Header →				← Conditional Body →		

(그림 9) 명령어 APDU의 구조

(그림 9)는 명령어 APDU의 구조를 나타내고 있다. 여기에서 Lc는 APDU에서 전송되는 데이터 바이트 수를 말하여, Le는 응답 APDU에서 예상되는 최대 바이트로서, '00' 값을 가지면 256바이트를 의미한다.

<표 4>에서는 클래스 바이트(class byte)를 보이고 있으며, EMV에서는 '00' 값을 쓰고, <표 5>에서 보이는 INS(instruction) 바이트는 명령어 코드로써 길이는 1이며 명령어마다 다른 값을 사용한다[11].

〈표 4〉 클래스 바이트

Hex	의 미
'0'	Inter-Industry 명령어
'9'	전용 명령어
'A'	Data Confidentiality 사용
X	기타 RFU

〈표 5〉 INS 바이트

INS	명 령 어 명
'B0'	READ BINARY
'D6'	UPDATE BINARY
'0E'	ERASE BINARY
'B2'	READ RECORD
'E2'	APPEND RECORD
'DC'	UPDATE RECORD
'CA'	GET DATA
'DA'	PUT DATA
'A4'	SELECT FILE
'20'	VERIFY
'88'	INTERNAL AUTHENTICATE
'82'	EXTERNAL AUTHENTICATE
'84'	GET CHALLENGE
'C0'	GET RESPONSE
'1A'	BLOCK
'14'	UNBLOCK
'E0'	CREATE FILE
'E8'	SET LIFE
'24'	PUT KEY
'8A'	CREATE SESSION
'8C'	PUT KEY SET
'1C'	FORBIDDEN AC
'E4'	GET ENCIPHER

P1, P2는 파라미터 바이트로써 각각 '00'~'FF' 사이의 값을 가질 수 있으며, 특별한 의미를 갖지 않는 경우는 '00' 값을 사용한다. 데이터 바이트(Data byte)는 특별히 TLV(Tag, Length, Value) 형식을 따르는데, 1바이트의 템플릿과 값의 길이와 실제 데이터로 구성되는 구조체이다.

〈표 6〉 APDU 명령어 - 응답

Case	명령어 데이터	응답 데이터
1	없 음	없 음
2	없 음	있 음
3	있 음	없 음
4	있 음	있 음

명령 APDU와 응답 APDU는 모두 데이터를 포함할 수 있고 <표 6>과 같이 4가지 경우를 가진다. 명령어는 SELECT, READ 등이 있는데, SELECT 명령은 DF이름에 의한 응용프로그램을 선택할 경우와 파일 식별자에 의한 파일선택의 경우, DF 선택에서 파일 제어 정보(File Control Information ; FCI) 파일이 존재하면 FCI 파일의 첫 번째 레코드를 얻을 경우에 사용된다. 명령 APDU는 <표 7>과 같으며, 응답 APDU는 <표 8>과 같다.

〈표 7〉 SELECT 파일 명령 APDU

코 드	내 용
CLA	00
INS	A4
P1	Selection 제어
P2	00
Lc	Empty 또는 후속 데이터 필드 길이
Data	P1 P2에 따름 • 파일 식별자 • DF Name
Le	최대 예상 응답 데이터 길이

P1값은 '00000000'일 경우는 파일 식별자에 의한 참조이며 MF, DF 혹은 EF을 선택할 때 사용하며, '00000001'는 Child DF를 선택할경우에 사용한다. '00000100'은 DF 이름에 의하여 직접 참조할 때 사용되며 이 경우 데이터는 '1PAY.SYS.DDF01'가 사용된다.

〈표 8〉 SELECT 파일 응답 APDU

이 름	내 용	길 이
DATA	파일에 대한 정보	Var
SW1, SW2	COMPLETION CODE	2

<표 8>에서 보이는 응답 APDU에서는 데이터 값을 TLV 형식으로 가지는데, Tag가 '84'일 경우는 길이가 '01'~'10'이며 DF 이름을 나타낸다. SW1, SW2 값은 2byte의 길이를 갖는 에러코드를 나타낸다. SW1, SW2가 '9000'일 경우는 정상적인 처리를 의미하고, '6283'일 경우는 잘못된 파일을 선택했거나 선택된 파일이 블럭(block)되었을 경우의 에러코드이다.

〈표 9〉 READ 파일 명령 APDU

코 드	내 용
CLA	00, 04, A4
INS	B0
P1~P2	P1의 b8 = 1이면 P1의 b5~b1은 Short 파일식별자 P2는 Read할 Offset을 표시 P1의 b8 = 0이면 P1~P2는 Read할 Offset
Lc	Empty
Data	Empty
Le	Read될 바이트의 수

READ 명령은 선택된 DF의 디렉터리 파일을 읽어올 때 사용한다. 이 명령어는 비확인 데이터구조를 확인할 수 있는 Non-Transparent 구조(Record 구조)의 요소파일을 지원한다. <표 9>는 READ 명령어의 명령 APDU를 의미하는 것으로 CLA바이트에 '00', '04', 'A4'를 사용하고 있다.

<표 10>은 READ명령어의 응답 APDU를 보이고 있으며, 가변길이 데이터와 에러코드 SW1, SW2가 응답으로 수신되는데, 에러코드는 SELECT 명령과 동일하다.

〈표 10〉 READ 파일 응답 APDU

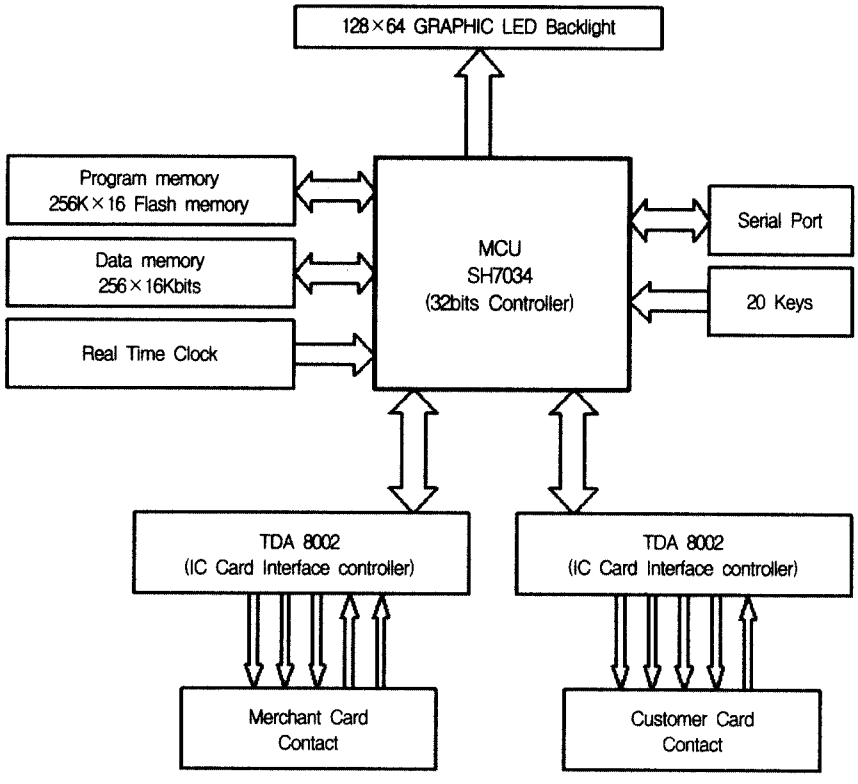
이 름	내 용	길 이
DATA	Data Read(Le bytes)	Var
SW1, SW2	COMPLETION CODE	2

3. EMV 터미널 구현

본 논문의 EMV 터미널은 EMV Spec.의 모든 응용프로그램이 동작할 수 있도록 프로토콜을 정의하는 Level 1을 구현하는데 그 목적을 두고 있다. 구현한 터미널의 원할한 테스트를 위해서 EMV L1 Pre-Certification이라는 INTEGRI 데이터 테스트 장비를 이용하였으며, 카드의 ATR값을 읽어오기 위한 디버그 문자를 삽입시켰다. 구현은 ANSI C를 이용하여 모니터 프로그램(monitor program)을 만들고, 실제 동작을 위한 데이터를 모니터 프로그램을 통하여 시리얼로 전송하여 기록한다. 본논문의 모니터 프로그램은 실제 터미널 동작과는 상관없이, ROM 영역 0x000000에 위치하여 실제 ROM 데이터를 읽어올 때, 모니터 프로그램부터 동작하여 실제 메인(main)이 존재하는 곳으로 분기 하도록 자체 개발한 프로그램이다. 스마트 카드 인터페이스는 스마트 카드의 8개의 입출력 포트와, EMV Spec에 맞게 VCC와 CLK를 인가하였으며, 테스트를 통하여 국제 인증과정을 통과하였다.

3.1 하드웨어

(그림 10)은 본 논문의 EMV 터미널 하드웨어 구조를 블록 다이어그램으로 나타낸 그림이다. 구현을 위해서 MCU SH7034 프로세서, 256K×16의 플래시메모리(flash memory)와, 256×16Kbits 데이터 메모리(data memory)를 연결하고, 출력을 위한 128×64 도트 출력용 LCD(GRAPHIC LED Backlight) 화면과, 데이터 전송을 위한 시리얼 포트(serial port) 2개, 사용자의 입력(PIN)을 입력받을 수 있는 PIN패드, 실시간 클럭(real time clock), 그리고 두 개의 카드 인터페이스 컨트롤러를 사용한다. TDA 8002(IC card interface controller)는 스마트카드 삽입부분이며, 실제 데이터의 송수신을 담당하는 부분이다. 프로그램 데이터 영역은 플래시메모리(flash memory) 부분에 기록하였으며, 스마트 카드에 인가하는 클럭과 CPU에 인가하는 클럭을 계산하여 스마트카드와 터미널간의 데이터 송수신에서 사용되는 etu를



(그림 10) 하드웨어 다이어그램

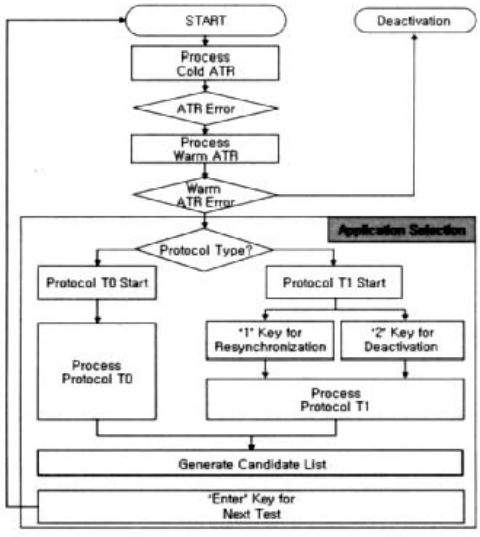
정확히 계산하여 허용 오차범위인 ±0.02를 만족하였다.

3.2 소프트웨어

본 논문의 EMV 터미널을 위한 소프트웨어 구현 범위는 스마트 카드에 리셋 신호를 보내고 응답을 받는 ATR처리부터 어플리케이션 계층인 프로토콜 T=0와 프로토콜 T=1 처리를 통한 응용프로그램 선택(application selection)까지이다. 이것은 EMV Spec.의 Level 1 범위와 일치한다[12].

3.2.1 소프트웨어 전체 구성

(그림 11)은 본 논문에서 작성한 EMV Spec. Level 1을 위한 프로그램의 전반적 흐름 개요도이다. 프로그램의 흐름을 설명하기 위해서 (그림 11)의 순서도에서 처리부분(사각형)을 위주로 순서대로 각각의 작업을 설명한다.

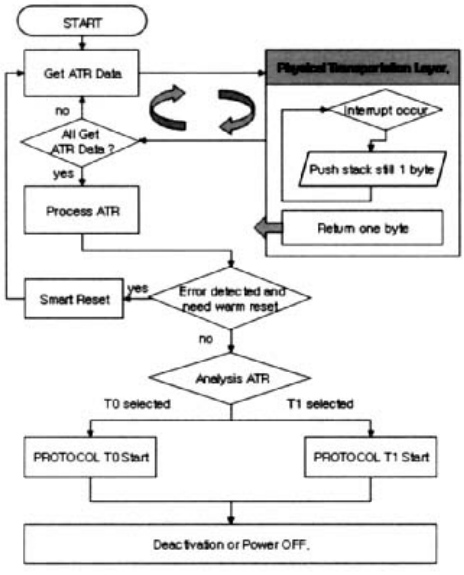


(그림 11) 소프트웨어 전체 개념도

먼저 그림의 초기리셋처리(Process Cold ATR)부분에서 스마트 카드에게 리셋 신호를 보낸다. 후에 스마트 카드로부터 ATR를 수신 받으면 받은 데이터의 무결성을 보증하는 에러 검출을 실시한다. 받은 데이터는 후에 etu계산이나, 실제 작업대기시간등 데이터 송수신에 필요한 값을 저장한다. 초기리셋처리에서 실패를 하게 되면 일정시간이 지난 후 웜 리셋(warm reset)을 시도를 하는데 웜 리셋후에 ATR을 받는 부분(Process Warm ATR)을 말한다. 콜드 리셋과 웜 리셋에서 수신받는 ATR값은 동일하다.

다음으로, ATR에서 프로토콜 타입이 분석되면 프로토콜 T=0와 T=1을 선택하게 된다. 선택된 프로토콜은 활성화되고, 선택되지 않은 프로토콜은 비활성화 되면서 후에 카드 세션은 선택된 프로토콜로 계속 이루어진다. Generate Candidate List 부분에서는 본 논문 2.5절에서 소개한 디렉터리 파일을 읽어와서 현재 디렉터리에 존재하는 AID를 트리구조로 저장하여, 지원하는 응용프로그램의 리스트를 이후 카드세션(EMV Level 2)에 이용한다.

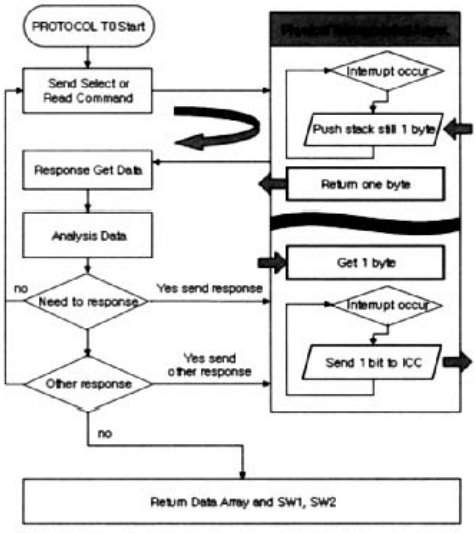
3.2.2 리셋응답(ATR) 처리



(그림 12) ATR 순서도

(그림 12)는 ATR을 수신할경우의 처리순서도이다. Get ATR Data는 타이머 인터럽트를 통하여 수신받은 일련의 데이터를 ATR 포맷(2.2절 그림 5)에 맞추어 저장한 다음, 본 논문 2.3절에 소개한 ATR의 에러 처리과정을 수행하게 된다. 콜드 리셋후 ATR과 웜 리셋후 ATR에서 에러가 없으면, 현재 ATR값을 이용하여, 카드세션에 필요한 데이터를 계산하여 저장하며, 프로토콜 T=0와 T=1을 판단하여 이후 카드세션에 반영한다. ATR에서 실패하면 터미널은 더 이상 카드 세션을 계속하지 않고 세션을 종료한다.

3.2.3 프로토콜 T = 0 처리



(그림 13) 프로토콜 T = 0 순서도

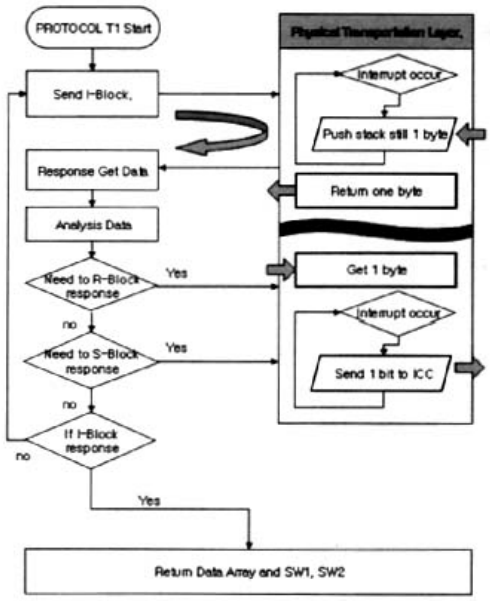
(그림 13)는 선택되어진 프로토콜이 T=0일 경우의 처리 순서도이며, 처음에 명령 APDU(Send Select or Read Command)를 생성하여 스마트 카드로 전송할 때, 타이머 인터럽트가 발생하면서 문자단위의 데이터를 스마트 카드로 전송한다.

명령 APDU를 모두 전송하면 스마트 카드로부터 응답 APDU를 통해 데이터가 전송되는데(Response Get Data), 이 과정에서 발생할 수 있는 예외처리(패리티 에러, 잘못된 바이트 순서 등)를 모두 수행한다.

다음으로, Analysis Data 부분은 더 이상 받을 데이터가 없을 경우에 데이터 및 SW1, SW2를 합쳐서 어플리케이션 계층(application layer)으로 전송한다. 리턴받은 SW1, SW2는 본 논문 2.6절에서 소개한 에러 검출 코드이다.

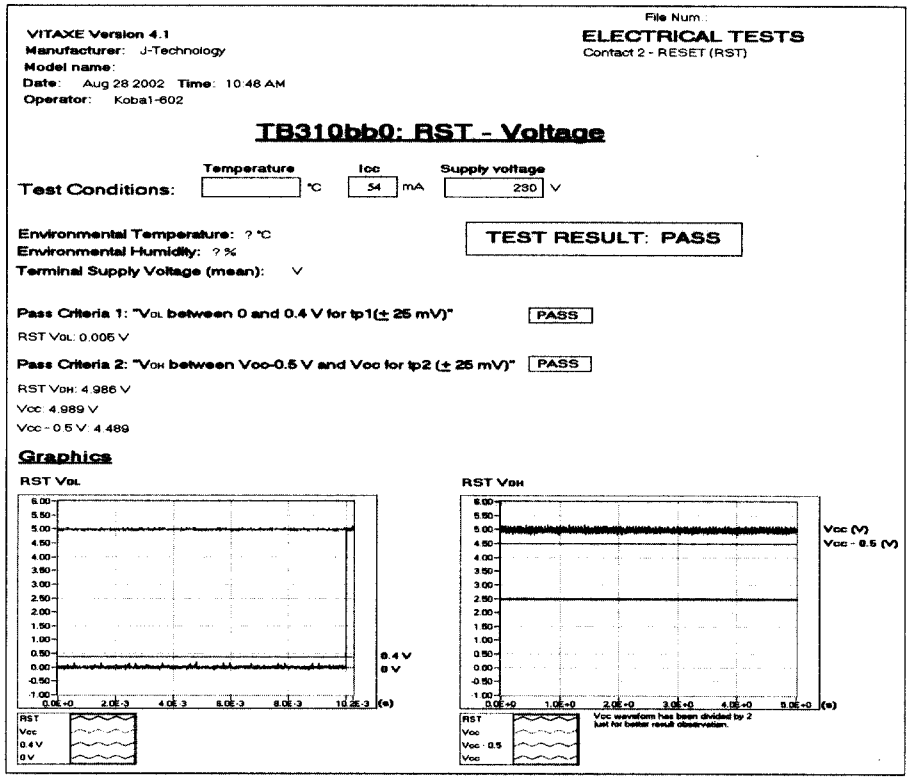
3.2.4 프로토콜 T = 1 처리

(그림 14)는 ATR에서 프로토콜 T=1이 선택되었을 경우 이후 카드 세션을 처리할 때 쓰이는 프로토콜 T=1의 처리 순서도이다. 기본적인 구성은 프로토콜 T=0와 동일하나, 전송되는 데이터의 무결성 보장을 위한 부분이 T=0와는 다르게 블록 전체의 무결성 보장을 해주어야 하는 등 블록전송에 따른 기타 에러처리가 부가적으로 반영시켰다. 기본적은 프로토콜 T=1은 명령어 블록인 I-Block과 응답 블럭인 R-Block과 동기나 기타 예외상황을 처리하기 위한 S-Block으로 구분되어진다. I-Block은 순차적인 번호(sequence number)가 있어서, 전송된 I-Block마다 '0'과 보수 형태인 '1'의 값을 오가며 순서를 정하며, 받은 블럭에 패리티 에러나 잘못된 I-Block의 번호 등 블록오류에 대해서 재전송 요구를 할 수 있는데 그때 R-Block에 I-Block 번호를 포함하여 전송한다. 스마트 카드와 터미널은 양간의 데이터 송수신 동안 이러한 R-Block을 만나게 되면 다음 데이터의 전송을 미루고 이전 데이터를 재 전송하게 된다.



(그림 14) 프로토콜 T=1 순서도

만약에 3번이상의 연속적으로 R-Block이 전송되거나, 에



(그림 15) 하드웨어 : 리셋 신호 전압

러가 연속적으로 3번이상 나게 되면, S-Block을 전송하여 동기를 맞추는데 이때, 터미널은 S-Block을 전송하여 동기를 맞추거나, 세션 종료 처리를 시작할 수 있다. 본 논문의 EMV 터미널에서는 세션 종료를 선택했다. 모든 전송은 타이머 인터럽트를 통해서 처리했으며, 하나의 I-Block의 응답 APDU로 최종 I-Block을 수신하면 명령 APDU 이후 받은 모든 I-Block과, SW1, SW2를 합쳐서 상위계층인 어플리케이션 영역으로 전송하게 된다.

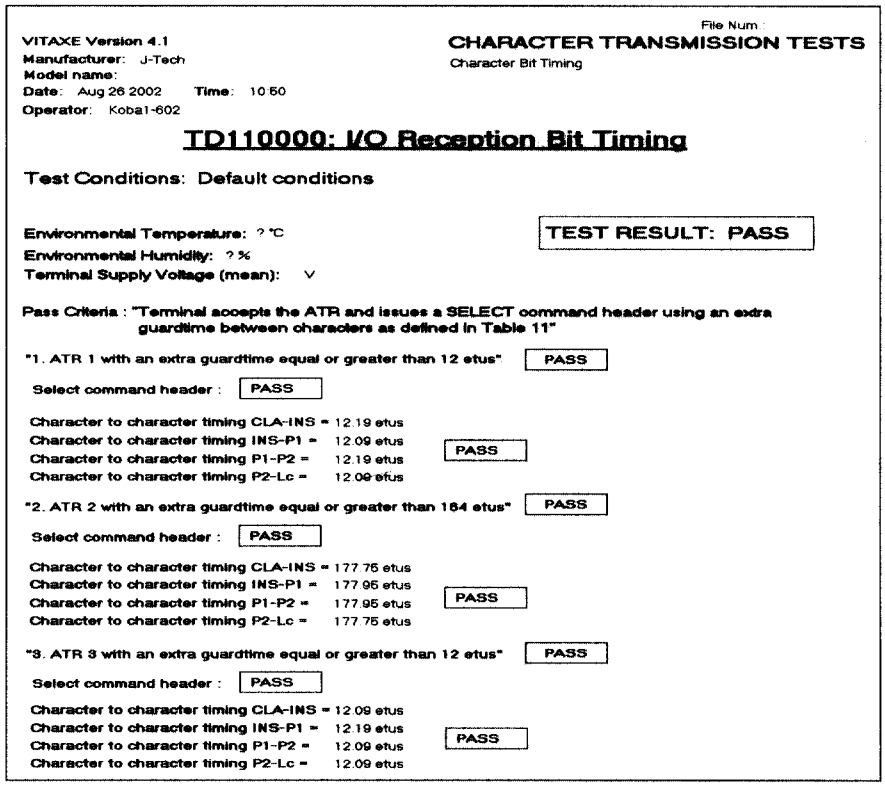
터미널에서 처리하는 내용과 스마트 카드에서 전송하는 데이터의 흐름을 INTEGRI 데이터 테스트 장비로 테스트해본 결과 정상적인 송수신을 확인할 수 있었으며, EMVCo [13]에서 정하는 타입 승인(type approval)시나리오를 모두 지원한다.

4. EMV 터미널 인증

EMV 국제 인증이란 EMVCo에서 정하는 스마트 카드부분, 터미널 부분을 규약에 정해서 테스트를 통하여 터미널이 모든 예외 처리에 대한 능력을 검증 받는 절차이다. 현재 미국, 독일, 일본 등 세계 몇몇 나라에서 스마트카드의 인증을 위한 테스트를 진행하고 있는데, 이들 중에서 본 논문에서 개발한 터미널은 일본의 품질인증기관(TUV RHEINLAND JAPAN)에서 인증절차를 밟았다. TUV JAPAN은 아시아 스마트 카드부분을 테스트, 인증하고 있다. EMV 인증 테스트 시나리오에는 각각의 EMV 테스트 케이스가 있는데, 터미널이 EMV를 지원하는 모든 스마트 카드를 지원하기 위해서는 이 절차가 반드시 필요하다[12].

(그림 15)은 실제 TUV에서 테스트 항목중에 하드웨어의 리셋 신호 전압을 테스트하는 항목으로 리셋 동안의 VCC의 안정성을 테스트하는 항목이다. 리셋을 위해서 터미널에서 카드로 I/O라인을 Low 상태를 유지하는데 필요한 전압의 범위를 정하는데, 이부분은 00.4v일 때의 오차범위와 0.5v에서 VCC까지의 오차범위가 ±0.25mv를 만족해야 한다. 하드웨어 테스트는 스마트카드의 8개의 단자와 터미널의 접점을 통하여 통신하는 물리적인 범위를 정하는 것이 대부분이다. 그림에서 그래픽으로 표시된 항목은 테스트 데이터를 바탕으로 작성된 것이다.

(그림 16)는 프로토콜 T=0의 테스트결과를 나타낸다. TD110000 테스트 항목은 ATR 수신 이후에 첫 번째 명령 APDU를 송신할 경우의 비트 타이밍을 테스트하는 경우로서, 첫 번째 ATR에서 추가 대기 시간을 수신 받는다면 명령 APDU를 전송할 때, 이렇게 반영된 추가대기 시간을 적용하여 전송하게 된다. 이 경우, 각각의 바이트를 테스트항목에 맞게 오차범위를 계산하여 성공 혹은 실패 여부를 가리게 된다. 그림에서 CLA-INS 혹은 INS-P1이라고 표시된 각각의 테스트 세부 항목은 본 논문의 2.6절에서 소개한 클래스 바이트와 인스트럭션 바이트를 그리고 파라미터를 의미한다. 각각의 바이트 역시 비트 간격을 준수해야 하므로


VITAXE Version 4.1
Manufacturer: J-Tech
Model name:
Date: Aug 26 2002 Time: 10:50
Operator: Kobal-602

File Num :
CHARACTER TRANSMISSION TESTS
Character Bit Timing

TD110000: I/O Reception Bit Timing

Test Conditions: Default conditions

Environmental Temperature: ? °C
Environmental Humidity: ? %
Terminal Supply Voltage (mean): V

TEST RESULT: PASS

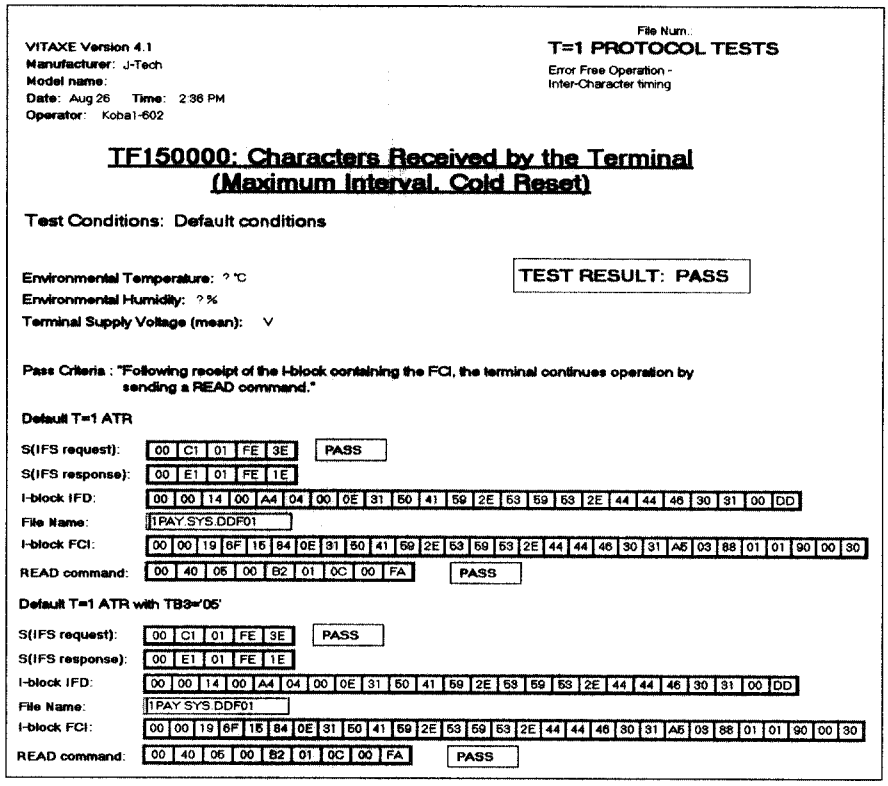
Pass Criteria : "Terminal accepts the ATR and issues a SELECT command header using an extra guardtime between characters as defined in Table 11"

"1. ATR 1 with an extra guardtime equal or greater than 12 etus" PASS
Select command header : PASS
Character to character timing CLA-INS = 12.19 etus
Character to character timing INS-P1 = 12.09 etus
Character to character timing P1-P2 = 12.19 etus
Character to character timing P2-Lc = 12.09 etus
PASS

"2. ATR 2 with an extra guardtime equal or greater than 184 etus" PASS
Select command header : PASS
Character to character timing CLA-INS = 177.75 etus
Character to character timing INS-P1 = 177.95 etus
Character to character timing P1-P2 = 177.95 etus
Character to character timing P2-Lc = 177.75 etus
PASS

"3. ATR 3 with an extra guardtime equal or greater than 12 etus" PASS
Select command header : PASS
Character to character timing CLA-INS = 12.09 etus
Character to character timing INS-P1 = 12.19 etus
Character to character timing P1-P2 = 12.09 etus
Character to character timing P2-Lc = 12.09 etus
PASS

(그림 16) 소프트웨어 : 프로토콜 T = 0



(그림 17) 소프트웨어 : 프로토콜 T = 1

테스트 항목은 명령 APDU를 대표로 테스트한 경우이다.

(그림 17)은 프로토콜 T = 1의 테스트 항목 중 타이밍에 관한 항목이며, 카드에서 터미널로 ATR을 전송한 후에, 터미널은 S-Block을 전송하는데, 이때 전송하는 S-Block은 후에 카드세션 처리를 위한 데이터의 크기를 정하게 된다. 이후, 명령어 APDU인 첫 번째 I-Block을 전송하고, 응답 APDU로 FCI를 수신 받는다. 이렇게 블록 전송의 경우에 디폴트 ATR을 수신받았을 경우와 ATR값에서 TB3 값을 '05'을 선택했을 경우를 테스트하는 경우로서, TB3값이 '05'이란 BWT는 0이고 CWT값은 43etu를 의미한다. 즉, 그림에서는 확인되지 않는 문자와 문자 사이의 간격이 43etu를 만족하는지를 테스트하여, 디폴트 ATR값과 임의의 ATR값에 따른 최대 문자간격(maximum interval)을 반영하고 있는지를 테스트하는 부분이다.

본 논문에서 개발한 EMV터미널은 이러한 하드웨어, ATR, 프로토콜 T = 0, 프로토콜 T = 1등 EMV 국제 인증을 위한 모든 EMV 테스트 항목들을 통과하였다.

5. 결 론

본 논문은 국제 표준은 ISO 7816, EMV Book 1, 2, 3, 4를 준수하고 다양한 응용프로그램을 위한 터미널을 설계하고, 구현하였다. 또한 데이터의 전송 중에 나타나는 모든 예외상황을 처리, 확인하기 위해서 국제 인증기관의 인증을 받았다. 하드웨어를 자체 개발하고, 모니터 프로그램을 이용하여 필요한 프로토콜을 실제 터미널에 구현함으로써 실제 카드의 데이터를 액세스 할 수 있었다. 사용하는 응용프로그램의 범위를 넓히고자 프로토콜 T = 0와 T = 1을 구현하여 실제 문자 전송 프로토콜뿐만 아니라 블록전송 프로토콜을 구현하여 향후 스마트 카드로 멀티미디어 정보도 송수신할 수 있는 준비상황을 마쳤다. 본 논문의 결과로 구현된 EMV 스마트 카드 터미널의 활용 및 기대 성과는 다음과 같다. 첫째, 국내외 여러 응용프로그램이 모두 EMV을 준수한다면 하나의 터미널상에 여러 응용 프로그램을 간단히 추가만 함으로써, 모든 응용 프로그램의 서비스를 지원할 수 있다. 이것은 하나의 카드로 여러 개의 서비스를 받을 수 있는 것을 가능하게 한다. 둘째, 향후에 지원할 응용프로그램을 터미널 재개발 없이 소프트웨어 매니저(software manager)를 통하여 지원 가능하여 개발시간의 단축을 기대할 수 있다. 개발시간의 단축은 제품의 단가에 막대한 영향을 가져와서 국가 스마트 응용분야에 경쟁력의 상승을 기대할 수 있다.

참 고 문 헌

[1] EMV2000 Version 4.0, Book1 - Application independent ICC to Terminal Interface Requirements, December, 2000.
[2] EMV2000 Version 4.0, Book2 - Security and Key Management, December, 2000.
[3] EMV2000 Version 4.0, Book3 - Application Specification, December, 2000.
[4] EMV2000 Version 4.0, Book4 - Cardholder, Attendant and Acquirer Interface Requirements, December, 2000.
[5] ISO/IEC 7816-1, Identification cards - Integrated circuit(s)

cards with contacts., 1988.
[6] ISO/IEC 7816-2, Dimensions and location of the contacts, 1988.
[7] ISO/IEC 7816-3, Electronic signals and transmission protocols, 1988.
[8] ISO/IEC 7816-5, Numbering system and registration procedure for application identifiers, 1988.
[9] ISO/IEC 7810, Identification cards - Physical characteristics, 1995.
[10] ISO 9992-2, Financial transaction cards Part 2 : Functions, messages(commands and responses), data elements and structures, 1989.
[11] ISO 9992-1, Financial transaction cards Part 1 : Concepts and structures, 1990.
[12] ISO/IEC 10373, Identification cards - Test methods, 1993.
[13] http://www.emvco.com.
[14] http://www.emvco.com/cgi_bin/detailapp.pl?id=1.

박 창 현

e-mail : park@cse.yu.ac.kr
1986년 경북대학교 전자공학과 졸업(공학사)
1988년 서울대학교 계산통계학과 전산학 전공(이학석사)
1992년 서울대학교 계산통계학과 전산학 전공(이학박사)
1992년~1993년 서울대학교 컴퓨터신기술공동연구소 특별연구원
1998년~1999년 University of Maryland, Institute of Advanced Computer Systems, Visiting Researcher
1993년~현재 영남대학교 컴퓨터공학과 부교수
관심분야 : 인공지능, 지식기반 시스템, 데이타 마인닝 및 지식 발견, 에이전트 시스템

두 명 택

e-mail : perseusx@orgio.net
2001년 (주)Willbes 스마트카드 연구원
2002년~현재 영남대학교 컴퓨터공학과 재학 중
관심 분야 : 임베디드 시스템, 무선인터넷, XML, 스마트 카드

이 백 순

e-mail : baeksoon@j-tech.co.kr
1983년 경대 전자공학과 졸업 (공학사)
1987년 (주)KISC 연구원
1989년 (주)C&I 연구원
1995년 (주)KIS 통신기술 개발 담당 부서장
2002년~현재 (주)J-Technology 경영관리 본부 상무이사
관심분야 : 전자화폐, 스마트 카드

권 오 규

e-mail : okkwon@j-tech.co.kr
2001년 (주)Willbes 스마트카드 연구원
2002년 영남대학교 컴퓨터공학과 졸업 (공학사)
2002년~현재 (주)J-Technology 연구개발 본부/기술연구소 연구원
관심분야 : 임베디드 시스템, 스마트카드, 리눅스 보안