자바 카드에서 접촉 및 비접촉 겸용 IC 카드 OS의 설계 및 구현

주 홍 일, 손 수 호, 전 용 성, 전 성 익
한국전자통신연구원 IC카드연구팀
전화 : 042-860-5988 / 핸드폰 : 016-537-7581

Implementation of the contact and contactless IC Card OS for Java Card

Hong-Il Ju, Yong-Sung Jeon, Soo-Ho Sohn, SungIk Jun
IC Card Research Team, ETRI
E-mail : juhong@etri.re..kr

Abstract

This paper describes the design and implementation of contact and contactless IC card OS(Operating System) for Java Card, namely JCOS(Java Card OS). The JCOS complies with ISO/IEC 7816 and ISO/IEC 14443 standards. The JCOS conforms to Java Card 2.1.2 specifications.

The JCOS is running on 32-bit ARM7TDMI with public key crypto-coprocessor. This paper describes only the dual-interface protocol of the JCOS which supports contact and contactless applications in a single chip. The JCOS has been completed with our sample banking service and access control service in ETRI up to now.

I. 서론

최근 인터넷과 전자상거래 기술의 발전으로 보다 안전한 사용자 인증 및 정보보호 수단으로 프로세서가 내장된 IC 카드의 사용이 점점 증가하는 추세이다. 또한, IC 카드는 통신, 금융, 교통, 신분 확인, 전자화폐, 출입 통제 시스템 등 여러 다양한 응용 서비스에 사용되고 있으며, 카드와 카드 단말기 사이의 통신 방식에 따라 접촉 또는 비접촉으로 사용되고 있다. 따라서, IC 카드도 기존의 단일 응용 서비스 지원에서 다중 응용 서비스 지원, 보다 안전한 개인 정보의 저장 및 사용, 사용자 인증 및 데이터 인증 등을 포함하는 보다 강화된 보안성 기능이 요구되고 있다.

본 논문에서는 이러한 추세에 맞춰 대두되고 있는 자바 카드 기술을 기반으로 하는 IC 카드를 개발함에 있어서, 접촉 및 비접촉 프로토콜을 모두 지원 가능한 IC 카드 OS의 구현에 대해 기술하고자 한다. 현재 개발하고 있는 차세대 IC 카드는 접촉과 비접촉 프로토콜을 지원하는 콤비 카드를 목표로 하고 있으며, 접촉과 비접촉 표준은 각각 ISO/IEC 7816과 ISO/IEC 14443을 준용한다. 또한, 차세대 IC 카드의 OS는 자바 카드 규격인 Java Card 2.1.2을 만족한다.

본 논문의 구성은 1절에서 서론을 간략히 기술하고, 2절에서는 자바 카드 기술 기반의 IC 카드, 3절에서는 접촉 및 비접촉 프로토콜 설계, 4절에서는 프로토콜의 구현 및 시험 결과에 대해 언급하고, 마지막 5절에서 결론을 기술한다.

II. 자바 카드 기술 기반의 IC 카드

2.1 자바 카드의 기본 구조

자바 카드 기술을 기반으로 하는 IC 카드의 특징은 하위의 운영체제(Operating System) 위에 존재하는 자바 카드 가상 기계(Java Card Virtual Machine)가 자

바 카드 애플릿(Java Card Applet)의 바이트 코드를 수행하고, 메모리나 I/O 같은 IC 카드 내의 모든 자원에 대한 접근을 제어한다는 것이다. 또한, 자바 카드는 발급 후에도 최종 사용자가 필요에 따라 다양한 응용 프로그램을 카드에 적재가 가능하다는 것이다. 이러한 특징을 가지는 자바 카드의 구조는 그림 1과 같다 [1-3].

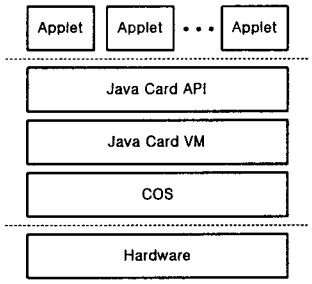

그림 1. 자바 카드의 구조

그림 1과 같이 자바 카드 구조는 맨 하위에 IC 카드를 구성하는 하드웨어가 존재하고, 그 위에 하드웨어에 의존적인 COS(Chip OS)가 위치한다. 그리고, COS 위에 플랫폼 독립성을 제공해주는 자바 카드 가상 기계가 탑재되며, 그 위에 자바 카드 API가 위치하게 된다. 여기서, 자바 카드 API는 애플릿 프로그램이 참조하는 패키지 형태의 클래스 파일로 애플릿은 프로토콜에 상관없이 같은 APDU(Application Protocol Data Unit) 메소드를 사용한다.

2.2 접촉 IC 카드의 동작 원리

접촉 IC 카드에서 카드의 동작 원리와 데이터 전송 절차를 간단히 살펴보면, 먼저 카드와 카드 리더의 접점이 서로 접촉하게 되면, 카드는 카드 리더로부터 전원과 클럭을 공급받고, 리셋 신호에 대해 응답을 하게 된다. 이때, 카드의 리셋에 대한 응답을 ATR(Answer To Reset)이라 한다. 또한 경우에 따라서는 PPS(Protocol and Parameters Selection)가 수행되기도 한다[4][6]. ATR과 PPS 가 성공적으로 수행되고 나면, 데이터 전송을 위해 통신 설정이 완료되고, 카드는 카드 리더로부터 첫번째 명령어를 기다리는 것으로 프로토콜이 시작된다.

카드와 카드 리더 사이의 데이터 전송 프로토콜은 카드 리더가 통신의 마스터(master)로, 카드는 슬레이브(slave)로 진행된다. 결국, 카드 리더가 카드로 명령어를 전송하면, 카드는 그 명령어에 대한 응답만 하게 된다. 또한, 카드와 카드 리더 사이의 데이터 전송은 카드 리더 쪽의 호스트에서 응용 프로그램을 실행시키기 위해 전달하는 것으로 명령어/응답 송수신을 반복하여 응용 서비스가 수행된다.

2.3 비접촉 IC 카드의 동작 원리

비접촉 IC 카드는 카드 리더와의 물리적인 접촉을 하지 않고 무선통신 방법을 기반으로 하고 있으며, 카드와 카드 리더 양쪽이 갖고 있는 안테나와 RF 모듈에 의해서 통신이 이루어진다[5].

비접촉 IC 카드는 카드 리더로부터 전원 공급과 데이터 송수신을 수행하고, 충분한 전원 공급이 되면 카드 리더로부터 송신되는 REQB(Request)명령어를 기다린다. 그러나, 비접촉 IC 카드는 접촉 IC 카드와 달리, 한번에 여러 개의 카드가 동시에 카드 리더가 보내는 REQB(Request) 명령어에 응답할 수 있다. 일반적으로 카드 리더와 카드는 각각 한번에 하나씩 통신을 해야 하는데, 이렇게 두개 이상의 카드가 동시에 응답하는 경우를 충돌이라 한다. 카드 리더는 이러한 충돌이 일어나면, 충돌 방지를 위해 몇 가지 명령어들을 사용해 한번에 하나씩 통신할 수 있도록 하는데, 이를 충돌 방지라 한다[7]. 따라서, 비접촉 IC 카드는 먼저 충돌방지 절차를 수행 후, 데이터 전송이 이루어진다. 비접촉 IC 카드의 동작 상태에 대한 자세한 사항은 ISO/IEC 14443 Part3 을 참조한다.

III. 접촉 및 비접촉 프로토콜의 설계

3.1 접촉 및 비접촉 모드 선택

접촉 및 비접촉 프로토콜을 모두 지원하는 겸용 IC 카드에서는 먼저 접촉 또는 비접촉에 대한 모드 선택이 우선적으로 이루어져야 한다. 차세대 IC 카드에서 접촉 및 비접촉 프로토콜을 위한 모드 선택은 카드를 구성하는 하드웨어의 I/O 인터페이스 부분에서 처리한다. I/O인터페이스 모듈에 접촉과 비접촉에 대한 각각의 상태 레지스터(status register)를 두고, 접촉 I/O 인터페이스에 의해 클럭과 리셋이 들어오면 접촉 상태 레지스터의 값이 '1'이 되고, 비접촉식 I/O 인터페이스에 의해 클럭과 리셋이 들어오면 비접촉 상태 레지스터의 값이 '1'이 된다. 따라서, 프로토콜을 수행하기에 앞서 각각의 상태 레지스터의 값을 확인 후, 프로토콜을 수행하게 된다.

3.2 접촉 프로토콜의 설계

접촉 프로토콜은 ISO/IEC 7816에 정의되어 있으며, 비동기 반이중 문자 전송방식(T=0)과 비동기 반이중 블록 전송 방식(T=1)을 사용하는데, 본 논문에서는 현재 많이 사용되고 있는 T=0 프로토콜에 대해서 언급한다. 문자 전송의 프레임은 하나의 시작 비트, 8개의 데이터 비트, 하나의 패러티(parity)비트로 구성되고, 다음 문자와의 사이는 감시시간(guard time)으로 구분된다.

카드와 카드 사이의 어플리케이션 층에서 데이터 전송을 위해 APDU라는 전송 규약을 사용하는데, 카드는 항상 카드 리더가 보내는 명령 APDU를 수신하면, 카드 내에서 명령 APDU에 해당하는 작업을 수행하고 응답 APDU를 송신한다. 그림 2는 명령 APDU 및 응답 APDU의 구조이다[4][6].

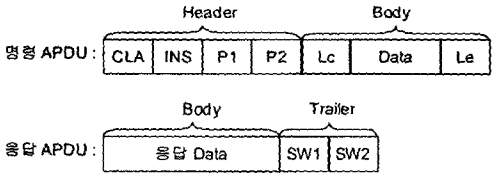

그림 2. 명령 APDU 및 응답 APDU의 구조

그림 2에서 명령 APDU의 Header 4 바이트와 응답 APDU의 Trailer 2바이트는 반드시 필요하며, 나머지 Body 부분은 전송되는 데이터 길이에 따라 가변적이다. 명령 APDU는 Body부분의 Lc, Le에 따라 4가지 경우로 나눌 수 있는데, 여기서 Lc는 전송하고자 하는 데이터의 길이를 나타내며, Le는 응답을 받고자 하는 데이터의 길이를 나타낸다. 그림 3은 명령 APDU의 4가지 경우에 대한 각각의 구조를 보여준다[4][6].

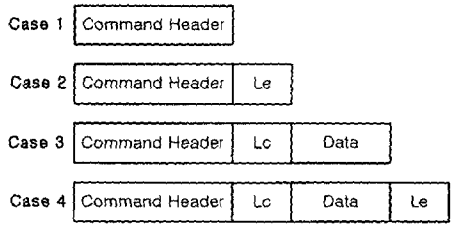

그림3. 명령 APDU의 4가지 구조

그림 3에서 Case 1은 Lc=0, Le=0 인 경우, Case 2는 Lc=0, Le≠0인 경우, Case 3은 Lc≠0, Le=0인 경우, 그리고 Case 4는 Lc≠0, Le≠0인 경우를 의미한다. 이러한 APDU는 어플리케이션 층에서 사용되며, 실제 카드와 카드 리더 사이에서는 각 APDU 형식에 해당하는 TPDU(Transport Protocol Data Unit)로 매핑(mapping)되어 전송된다.

그림 4는 접촉 IC 카드의 T=0 프로토콜에서 명령 APDU의 4가지 경우에 매핑되는 각각의 TPDU 수행 과정을 보여준다. 이러한 명령 APDU에 매핑되는 각각의 TPDU에 대하여, 어플리케이션 개발자는 TPDU에 무관하게 단지 APDU만 사용하여 어플리케이션을 개발하고, COS 개발자가 각 APDU에 해당하는 TPDU를 고려하여 설계해야 한다.

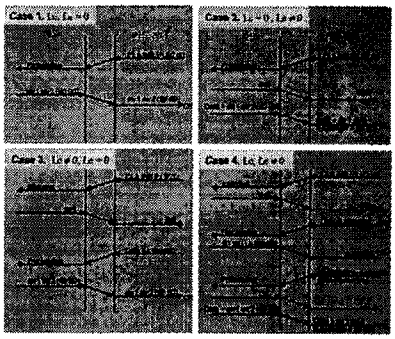

그림 4. 명령 APDU의 4가지 경우에 대한 각각의 TPDU 수행과정(T=0)

3.3 비접촉 프로토콜의 설계

비접촉 프로토콜은 반이중 블록 전송 프로토콜 방식을 사용하는데, 접촉 IC 카드의 T=1 프로토콜과 유사하지만, 블록을 구성하는 형식이 약간 다르다. 비접촉 프로토콜의 블럭 형식은 그림 5와 같다[7].

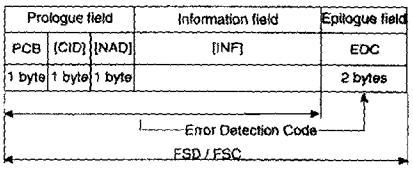

그림 5. 비접촉 프로토콜의 블록 형식

그림 5에서 프롤로그 부분은PCB(Protocol Control Byte), CID(Card Identification), NAD(Node ADdress) 바이트로 구성된다. PCB는 데이터 전송을 제어하기 위해 필요한 정보를 전달하는데 사용하며, 블록의 종류와 블록 번호에 대한 정보를 표시한다. 그리고, CID와 NAD는 카드 식별 인자로 사용된다. PCB에 따라 전송되는 블록은 S-블록, R-블록, I-블록으로 구분되며, I-블록이 어플리케이션 층에서 명령 APDU 및 응답 APDU를 전송하는데 사용된다. 또한, 그림 5에서 정보 영역(information field)은 선택 사항으로 I-블록에서 APDU를 전송하거나 S-블록에서 WTX명령어를 전달 할 때 사용하며, R-블록에서는 사용하지 않는다. 마지막으로 EDC(Error Detection Code) 부분은 에러 체킹 코드로 필요하며, 전송되는 전체 바이트에 대한 CRC_B 값을 가진다. 여기서 CRC_B에서 사용하는 다항식은 G(x)=x16+x12+x5+1 이다.

비접촉 프로토콜의 경우, 어플리케이션 층에서의 APDU는 I-블록의 정보 영역에 매핑되어 전달되는데, 그림 6은 이러한 블록 전송 방식에서의 명령 APDU의 각각의 경우에 해당하는 TPDU 수행 과정을 보여준다

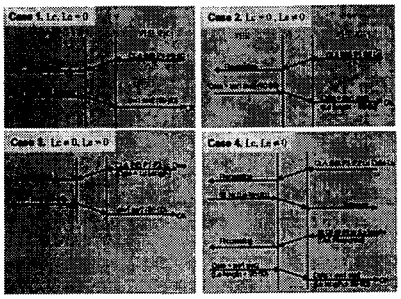

그림 6. 명령 APDU의 4가지 경우에 대한 각각의 TPDU 수행과정(비접촉)

Ⅳ. 프로토콜의 구현 및 시험 결과

4.1 프로토콜의 구현

접촉 및 비접촉 프로토콜을 구현함에 있어서 비접촉식 프로토콜을 위한 자바 카드 API를 추가하지 않고, 하위 COS 부분에서 접촉 및 비접촉을 위한 전송 프로토콜 계층을 두어 COS 상위에서는 동일한 전송 프로토콜에 수렴되도록 하여 구현했다. 즉, 접촉 및 비접촉에서 동일한 JCRE 환경을 사용한다. 또한, 접촉과 달리 비접촉 프로토콜에서는 여러 개의 카드가 동시에 응답을 할 경우, 먼저 충돌 방지 절차를 수행한 후 데이터 전송이 이루어진다. 그림 7은 접촉 및 비접촉 프로토콜의 수행 블록도이다.

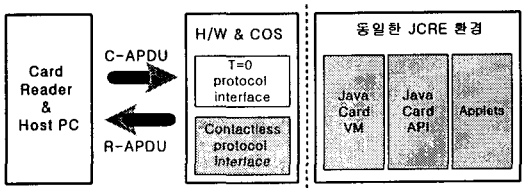

그림 7. 접촉 및 비접촉 프로토콜의 수행 블록도

4.2 프로토콜 시험 결과

<표 1>은 카드와 카드 리더 사이의 기본 APDU의 수행 결과이다. <표 1>에서 명령어는 카드 리더가 카드로 전송하는 명령어이며, 응답은 카드가 카드 리더로 전송하는 응답을 의미한다. 밑줄로 표현된 부분은 비접촉 프로토콜에서 I-블록의 정보 영역에 해당하는 부분으로 명령어/응답 APDU를 의미하는데, 이는 접촉 프로토콜에서와 같은 결과를 보여준다. 그리고, <표 1>에서 기울임꼴로 표현된 *0A 01* 과 *0B 01*은 비접촉식 블록 형식에서 프롤로그 부분의 PCB와 CID이고, 마지막 2바이트는 CRC_B에 해당한다. <표 1>에서 PCB의 값이 0A 또는 0B가 반복되는 것은 블록 번호 규칙과 블록 처리 규칙을 준수함을 보여준다.

<표 1> 카드와 카드 리더 사이의 APDU 수행 결과

```
//Select installer
명령어: 0A 01 00 A4 04 00 09 A0 00 00 00 62 03 01 08 01 35 A2
응 답 : 0A 01 90 00 F1 63
//Begin Installer
명령어: 0B 01 80 B0 00 00 00 10 15
응 답 : 0B 01 90 00 4A 7F
//Create Java Purse
명령어: 0A 01 80 B8 00 00 0C 0A A0 00 00 00 62 03 01 0c 02 01 00 43 20
응 답 : 0A 01 61 0A 7B A9
명령어: 0B 01 00 C0 00 00 0A 5A 6A
응 답 : 0B 01 A0 00 00 00 62 03 01 0c 02 01 90 00 06 3C
//Create Java Loyalty
명령어: 0A 01 80 B8 00 00 0C 0A A0 00 00 00 62 03 01 0c 05 01 00 46 AC
응 답 : 0A 01 61 0A 7B A9
명령어: 0B 01 00 C0 00 00 0A 5A 6A
응 답 : 0B 01 A0 00 00 00 62 03 01 0c 05 01 90 00 27 6B
//Create Wallet
명령어: 0A 01 80 B8 00 00 11 0A A0 00 00 00 62 03 01 0c 06 01 05 01 02 03 04 05 80 79
응 답 : 0A 01 61 0A 7B A9
명령어: 0B 01 00 C0 00 00 0A 5A 6A
응  답: 0B 01 A0 00 00 00 62 03 01 0c 06 01 90 00 EA 4E
//End Installer
명령어: 0A 01 80 B1 00 00 00 7E 96
응 답 : 0A 01 90 00 F1 63
```

Ⅴ. 결론

본 논문에서는 접촉 및 비접촉 기능을 모두 수행하는 콤비(combi) 카드를 개발함에 있어서, 자바 카드 가상 기계나 자바 카드 API는 접촉과 비접촉이 공유토록 하고, 하위 COS 부분에서 접촉과 비접촉 프로토콜 인터페이스 계층을 두어, 사용하고자 하는 애플릿을 겸용 카드에 동일하게 사용할 수 있도록 하는 장점이 있다. 본 논문에서 제시하고 구현한 프로토콜 스택은 계층적 접근 방법(layered approach)의 장점을 살려 서로 독립적으로 개발 유지 보수가 용이하게 하였다. 접촉 및 비접촉 프로토콜을 모두 지원 가능한 자바 카드 기반의 차세대 IC 카드는 자체 시험 결과 표준을 완벽히 준수하고 장시간 운용 시험 결과가 양호함을 확인하였다.

참고문헌

[1] Chen, Zhiqun, *Java Card Technology for Smart Cards*, Addison-wesley, 2000.

[2] http://java.sun.com/products/javacard.

[3] Sun Microsystems, Inc., *Java Card™ 2.1.2 Development Kit User's Guide*, Apr 11, 2001.

[4] W.Rankl & W.Effing, *Smart Card Handbook*, John Wiley & Sons, 2000.

[5] Klaus Finkenzeller, *RFID HANDBOOK*, Wiley, 1999.

[6] ISO/IEC 7816, International Standards.

[7] ISO/IEC 14443, International Standards.