



CARD & READER TECHNOLOGIES

ACR122S-SDK 사용자 가이드

◎ ACR122S 리더기의 특징 및 스펙

- ACR122S 리더기는 13.56MHz 비접촉식 리더기로서 Mifare 시리즈 ISO14443 호환 태그뿐만 아니라 근거리 무선통신을 위한 ISO/IEC 18092(NFC) 태그까지 read/write 할 수 있는 리더기입니다.

<제품 스펙 및 지원되는가능 RF카드 종류>

인터페이스	RS-232 Serial 인터페이스
주파수 대역	13.56 MHz
동작범위	70mm 이내 (태그의 안테나 사이즈 및 성능에 비례)
동작확인	두개의 LED(녹색,적색)
통신속도 / 읽기,쓰기 속도	최대 115200bps, 8-N-1 / 212kpbs, 424Kbps(태그 역시 424Kbps 지원해야 함)
개발환경	PC/SC표준 API사용, Microsoft CCID 표준드라이버 사용
공급전압 및 전류	5V DC(USB로 전원 공급), 200mA (최대); 50mA (대기)
운영온도	0℃~50℃
제품사이즈 및 중량	72mm(가로)×120.5mm(세로)×20.4mm(두께), 148g
지원 Protocol	T=CL 프로토콜, Felica 프로토콜 *APDU커멘드셋은 PC/SC 표준을 따르며 단지 Felica태그, 비프음, 듀얼LED를 컨트롤 할 때만 ACR122 전용 APDU 커멘드 사용.
호환규격/Certification	ISO14443 A and B, ISO/IEC18092, Mifare®, Felica, CE, FCC, RoHS, MIC
지원 O/S	Win 98, Win ME, Win 2000, Win XP, Win Vista, Win 7, Win Server 2003, Win Server 2008 Win XP x64, Win Vista x64, Win 7 x64, Win Server 2003 x64, Win Server 2008 x64, Win Server 2008 R2 x64 Linux
SAM 슬롯	1개 SAM슬롯(ISO7816, T=0/T=1 프로토콜 지원)

* 지원가능카드

- Mifare® card (Classics, DESFire, UltraLight, Plus SL1)
- ISO14443 A&B 타입 카드
- ISO/IEC 18092(NFC)카드 ⇒ Topaz, Jewel카드 등
- Felica카드

◎ ACR122S 리더기의 프로그래밍 절차

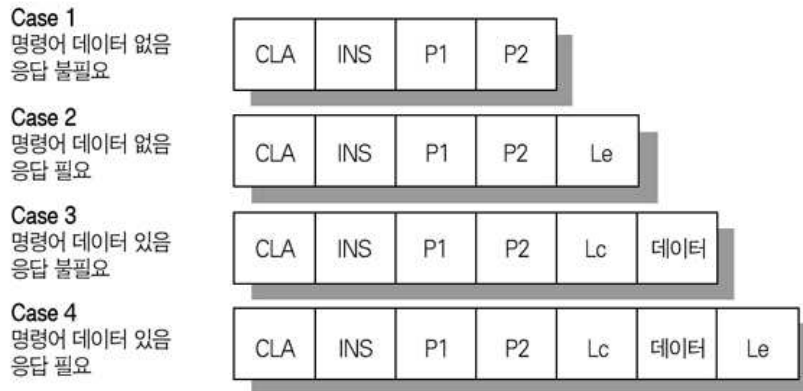
- ACR122S 리더기는 RS232(Serial) 방식이지만 호스트간의 통신 프로토콜은 CCID와 매우 유사합니다. 그렇기 때문에 스마트카드와 관련된 국제표준인 PC/SC API를 사용하여 프로그래밍 할 수 있습니다.

이때 스마트카드에 보내지게 될 데이터는 APDU(Application Protocol Data Unit)단위로 전송되는데 비접촉식 카드와의 통신은 유사 APDU(pseudo-APDU)로 통신하고 SAM 인터페이스는 표준 APDU로 통신 합니다.

사용할게 될 스마트카드(접촉식카드, RFID카드, SAM카드, USIM카드 등등)의 **고유 카드커맨드**를 알고 있어야 개발작업이 가능합니다.

※ PC/SC API중에 Direct Transmit()함수를 사용하여 전송되는 APDU의 형식은 아래와 같이 4가지 케이스로 구분됩니다.

<명령 APDU의 4가지 유형>



- CLA(1byte) : 명령클래스
- INS(1byte) : 명령클래스 내의 특정 명령
- P1(1byte) : 명령 매개 변수 1을 정의
- P2(1byte) : 명령 매개 변수 2를 정의
- Lc(1byte) : 이 선택적 필드는 명령의 데이터필드에 있는 byte의 길이를 나타냄.
- 데이터필드(Lc만큼의 byte길이) : 이 옵션 필드는 명령 데이터를 저장함.
- Le(1byte) : 이 선택적 필드는 응답 데이터필드의 예상 최대 byte길이를 나타냄.

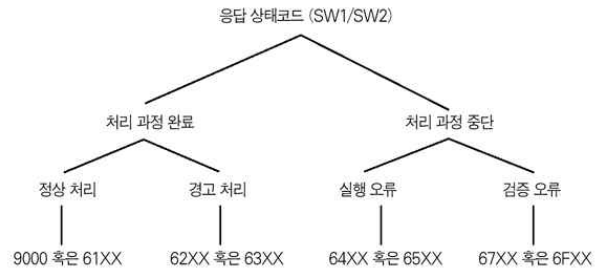
<ISO7816-4의 INS 값>

INS값	명령 설명	INS값	명령 설명
0E	바이너리 삭제	20	확인(Submit)
70	채널 관리	82	외부 인증
84	챌린지 수신(난수 요청)	88	내부 인증
A4	파일 선택(Select)	B0	바이너리 읽기
B2	레코드 읽기	C0	응답 수신
C2	엔벨로프	CA	데이터 읽기(UID 또는 CSN 읽기)
D0	바이너리 쓰기	D2	레코드 쓰기
D6	바이너리 업데이트	DA	데이터 저장
DC	레코드 업데이트	E2	레코드 추가

<응답 APDU 구조>

응답 APDU		
몸체(Body) (선택)	상태정보 (Status Word) (필수)	
데이터	SW1	SW2

<상태코드 구조도>

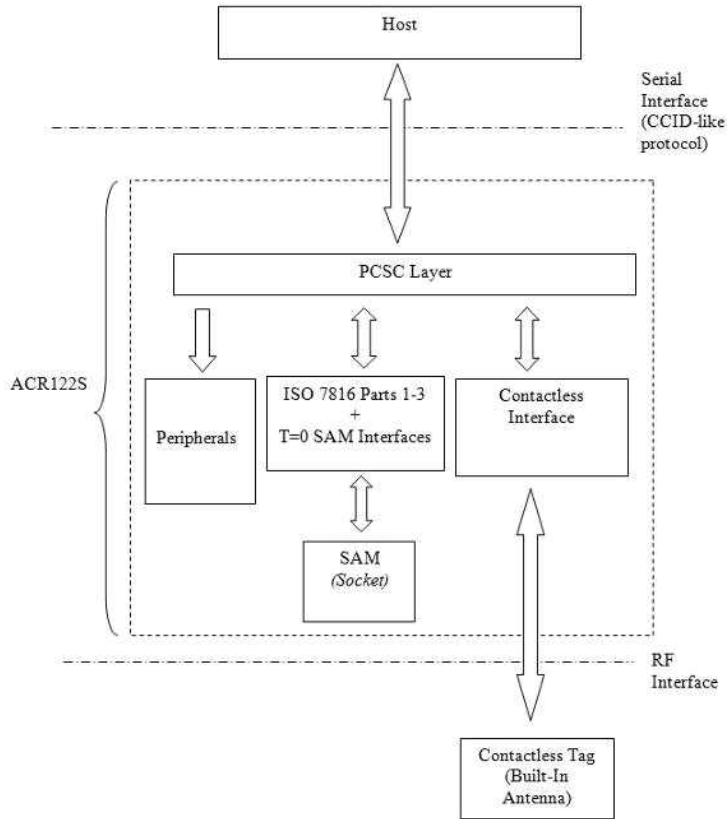


<상태코드(SW1, SW2) 정리>

SW1-SW2	의미
정상 처리(Normal processing)	
9000	정상 처리 완료
61XX	XX 크기만큼 처리할 응답 데이터가 존재함
경고 처리(Warning processings)	
62XX	메모리 영역이 변경되지 않았음
63XX	메모리 영역이 변경되었음
실행 오류(Execution errors)	
64XX	메모리 영역이 변경되지 않았음(SW2=00 경우 이외는 모두 RFU)
65XX	메모리 영역이 변경되었음
66XX	ISO 7816는 정의하지 않지만 차후 보안 관련 처리를 위해 예약한 영역
검증 오류(Checking errors)	
6700	잘못된 크기
68XX	지원하지 않는 CLA 명령
69XX	명령을 받아들일 수 없음
6AXX	P1 및 P2 값이 잘못되었음
6B00	P1 및 P2 값이 잘못되었음
6CXX	Le 값이 잘못되었음
6D00	INS 명령이 잘못되거나 지원하지 않음
6E00	해당 CLA를 지원하지 않음
6F00	어떠한 오류인지 판별 불가능함

* XX는 각각 경우에 따라 ISO 7816에 정의되어 있음

◎ ACR122S 리더기의 통신 방법



* RF카드 통신 :
유사 APDU로 통신
(pseudo-APDU)

* SAM인터페이스 :
표준 APDU로 통신

◎ Serial Interface (CMP_ACR122S 메뉴얼 6쪽)

- 통신 설정 : 9600 bps, 8-N-1
- 호스트와 ACR122S 간의 통신 프로토콜은 CCID와 매우 유사함
- RS232 통신 형식에 맞춰서 APDU 커맨드를 보낼 수 있습니다.

명령 포맷 (Command Frame Format)

STX (0x02)	Bulk-OUT Header	APDU Command Or Parameters	Checksum	ETX (0x03)
1 Byte	10 Bytes	M Bytes (If applicable)	1 Byte	1 Byte

상태 프레임 포맷 (Status Frame Format)

STX (0x02)	Status	Checksum	ETX (0x03)
1 Byte	1 Bytes	1 Byte	1 Byte

응답 포맷 (Response Frame Format)

STX (0x02)	Bulk-IN Header	APDU Command Or abData	Checksum	ETX (0x03)
1 Byte	10 Bytes	M Bytes (If applicable)	1 Byte	1 Byte

◎ RFID 카드 및 리더기 제어 (Pseudo-APDU)

1) RFID 카드 및 리더기 제어 : RFID에 명령을 보내거나 리더기내에 장착된 LED/비프음 등의 장치를 제어하기 위해서는 Direct Transmit 방식을 사용하여 Pseudo-APDU를 보낼 수 있습니다.

2) Direct Transmit command : 리더기의 칩셋(NXP PN532)에 직접 보내는 커맨드로써 API 문서에 기술되지 않는 태그를 사용하거나 특정 기능을 사용하고자 할때는 칩셋의 데이터시트(PN532 User Manual.pdf)를 참고하여 사용할 수 있습니다.

◎ Direct Transmit (CMP_ACR122S 메뉴얼 12쪽)

1) Direct Transmit 명령 포맷

Class	INS	P1	P2	Lc	Data In	
0 x FF	0 x 00	0 x 00	0 x 00	Number of Bytes to send	TAG Command	Data

* Lc : 최대 255byte

2) Data In : TAG Command (칩셋 데이터시트 참조)

- Input

Command	Data			의미
D4 40	Tg	DataOut[]		태그 데이터 교환(Tag Exchange Data)
D4 4A	MaxTg	BrTy	InitiatorData[]	태그 폴링(Tag Polling)

- Output

Data OUT				
D5 41	Status	DataIn[]		SW1
D5 4B	NbTg	TargetData1[]	TargetData2[]	SW2

3) 상태 코드값 : SW1 SW2

Results	SW1	SW2	의미
Success	90	00	성공
Error	63	00	실패
Time Out Error	63	01	TAG 응답 없음
Checksum Error	63	27	Checksum 응답값이 잘못됨
Parameter Error	63	7F	TAG 명령어가 잘못됨

[예제 1 : Mifare Classic] (CMP_ACR122S 메뉴얼 31~32쪽)

1) SELECT command (Polling)

STX	Bulk-OUT Header	Direct Transmit command				Checksum	ETX		
		Class	INS	P1	P2			Lc	Command
02	6F 09 00 00 00 00 01 00 00 00	FF	00	00	00	04	D4 4A 01 00	[03]	03

> 응답값

RDR -> 02 00 00 03 (Waiting the Tag)

RDR -> 02 80 0E 00 00 00 00 01 01 00 00

RDR -> D5 4B 01 01 00 02 18 04 F6 8E 2A 99 90 00 [Checksum] 03
① ② ③ ④ ⑤ ⑥ ⑦

- ① Number of Tag found = [01]; ⑤ Length of the UID = 4;
- ② Target number = 01 ⑥ UID = F6 8E 2A 99
- ③ SENS_RES = 00 02; ⑦ Operation Finished = 90 00
- ④ SEL_RES = 18,

※ Tip : 태그 유형은 SEL_RES을 인식하여 확인할 수 있습니다.

00 = Mifare Ultralight	18 = Mifare 4K	98 = Gemplus MPCOS
08 = Mifare 1K	20 = Mifare DESFire	
09 = Mifare MINI	28 = JCOP30	

2) AUTHENTICATE command (로그인)

적용 내용 : Block=04, KEY A=FF FF FF FF FF FF, UID=F6 8E 2A 99

STX	Bulk-OUT Header	Direct Transmit command					Checksum	ETX	
		Class	INS	P1	P2	Lc			Command
02	6F 14 00 00 00 00 01 00 00 00	FF	00	00	00	0F	D4 40 01 60 04 FF FF FF FF FF FF F6 8E 2A 99	[]	03

> 응답값

RDR -> 02 00 00 03 (Waiting the Tag)

RDR -> 02 80 05 00 00 00 00 01 01 00 00

RDR -> D5 41 [00] 90 00 [Checksum] 03
① ②

- ① Valid = 00
- ② Operation Finished = 90 00 [실패하면 오류코드를 반환 합니다. 에러코드 참조]
- ③ KEY A = 60, KEY B = 61

3) WRITE command (Update)

적용내용 : 04 블록 쓰기, write=01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10

STX	Bulk-OUT Header	Direct Transmit command					Checksum	ETX	
		Class	INS	P1	P2	Lc			Command
02	6F 14 00 00 00 00 01 00 00 00	FF	00	00	00	15	D4 40 01 A0 04 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10	[]	03

> 응답값

RDR -> 02 00 00 03 (Waiting the Tag)

RDR -> 02 80 05 00 00 00 00 01 01 00 00

RDR -> D5 41 [00] 90 00 [Checksum] 03
 ① ②

- ① Valid = 00
- ② Operation Finished = 90 00 [실패하면 오류코드를 반환 합니다. 에러코드 참조]

4) READ command

적용내용 : 04 블록 읽기

STX	Bulk-OUT Header	Direct Transmit command			Checksum	ETX
		Class INS P1 P2	Lc	Command		
02	6F 14 00 00 00 00 01 00 00 00	FF 00 00 00	0F	D4 40 01 30 04	[]	03

> 응답값

RDR -> 02 00 00 03 (Waiting the Tag)

RDR -> 02 80 05 00 00 00 00 01 01 00 00

RDR -> D5 41 [00] 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 90 00 [Checksum] 03
 ① ② ③

- ① Valid = 00
- ② 04번 블록 값
- ③ Operation Finished = 90 00[인증에 실패하면 오류코드를 반환 합니다. 에러코드 참조]

[예제 2 : T-Money 선불 A타입]

1) SELECT command (A타입 Polling)

STX	Bulk-OUT Header	Direct Transmit command			Checksum	ETX
		Class INS P1 P2	Lc	Command		
02	6F 09 00 00 00 00 01 00 00 00	FF 00 00 00	04	D4 4A 01 00	[03]	03

> 응답값

RDR -> 02 00 00 03 (Waiting the Tag)

RDR -> 02 80 16 00 00 00 00 01 01 00 00

RDR -> D5 4B 01 01 00 04 20 04 08 5C 88 0E 08 57 80 02 10 10 00 11 90 00 [Checksum] 03
 ① ② ③ ④ ⑤ ⑥ ⑦ ⑧

- | | |
|---------------------------------|------------------------------|
| ① Number of Tag found = [01]; | ⑤ Length of the UID = 4; |
| ② Target number = 01 | ⑥ UID = 08 5C 88 0E |
| ③ SENS_RES = 00 04; | ⑦ ATS 값(Answer To Select) |
| ④ SEL_RES = 04, | ⑧ Operation Finished = 90 00 |
| tip : B타입 Command = D4 4A 01 03 | |

2) READ command (T머니 16자리 카드번호 읽기)

STX	Bulk-OUT Header	Direct Transmit command			Checksum	ETX
		Class INS P1 P2	Lc	Command		
02	6F 10 00 00 00 00 01 00 00 00	FF 00 00 00	0B	D4 40 01 00 00 A4 00 00 02 42 00 00	[]	03

> 응답값

RDR -> 02 00 00 03 (Waiting the Tag)

RDR -> 02 80 3A 00 00 00 00 01 01 00 00

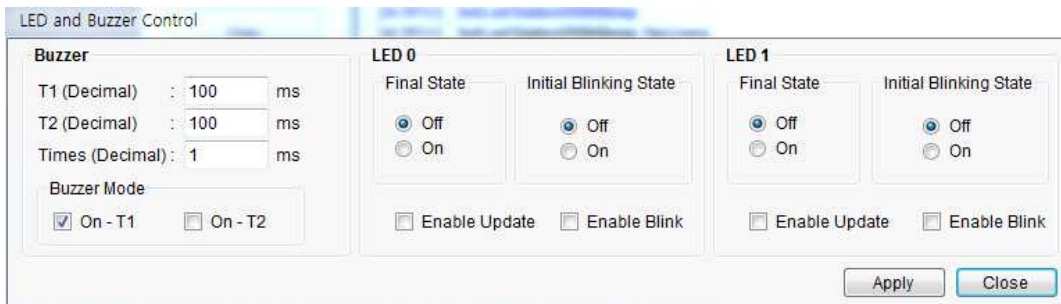
RDR -> D5 41 00 6F 31 B0 2F 00 10 01 08 10 10 00 00 64 19 96 80 00 96 93 84 18 20 07
 02 12 20 27 02 11 01 00 00 07 A1 20 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 90 00 90 00 [Checksum] 03 카드번호 16자리

◎ LED and Buzzer 제어 (CMP_ACR122S 메뉴얼 19~24쪽)

- LED and Buzzer 포맷(세부 옵션은 메뉴얼 참조)

Class	INS	P1	P2	Lc	Data In(4 Bytes)
0 x FF	0 x 00	0 x 40	LED State Control	0 x 04	Blinking Duration Control

* Tip : ACR122S Tool에서 간단하게 옵션 변경 가능(샘플코드의 Device Programming 참조)



※ 위의 설정값은 ACR122S 리더기의 메모리에 저장됩니다.

◎ SAM Interface (CMP_ACR122S 메뉴얼 9~11쪽)

- SAM 인터페이스를 사용하기 위해서는 활성화가 되어야 합니다.

1) SAM Interface 활성화 (IccPowerOn)

명령 포맷 (Command Frame Format)

STX (0x02)	Bulk-OUT Header (HOST to RDR IccPowerOn)	Parameters	Checksum	ETX (0x03)
1 Byte	10 Bytes	0 Byte	1 Byte	1 Byte
Offset	Field	Size	Value	Description
0	bMessageType	1	62h	Message-specific data length
1	dDwLength <LSB .. MSB>	4	00000000h	Identifies the slot number for this command. Default=00h
5	bSlot	1	00-FFh	Sequence number for command
6	bSeq	1	00-FFh	
7	bPowerSelect	1	00h, 01h, 02h, or 03h	Voltage that is applied to the ICC 00h-Automatic Voltage Selection 01h-5.0 volts 02h-3.0 volts 03h-1.8 volts
8	abRFU	2		Reserved for Future Use

예) To activate the slot 0 (default), sequence number = 1, 5V card.

HOST -> 02 62 00 00 00 00 00 01 01 00 00 [Checksum] 03
 RDR -> 02 00 00 03

RDR -> 02 80 0D 00 00 00 00 01 00 00 00 3B 2A 00 80 65 24 B0 00 02 00 82
90 00 [Checksum] 03

2) SAM Interface 비활성화 (IccPowerOff)

HOST -> 02 63 00 00 00 00 00 02 00 00 00 [Checksum] 03

RDR -> 02 00 00 03

RDR -> 02 81 00 00 00 00 00 02 00 00 00 [Checksum] 03

3) SAM Interface 데이터 전송 (Data Exchange)

HOST -> 02 6F 05 00 00 00 00 03 00 00 00 80 84 00 00 08 [Checksum] 03

RDR -> 02 00 00 03

RDR -> 02 80 0A 00 00 00 00 03 00 00 00 E3 51 B0 FC 88 AA 2D 18 90 00
[Checksum] 03

Response = E3 51 B0 FC 88 AA 2D 18; SW1 SW2 = 90 00

◎ ACR122-SDK의 각 샘플코드에 대한 기능 설명

< Contactless Cards 폴더 >

- Mifare Programming : Mifare Classic(1K, 4K, 320byte)카드를 사용하여 프로그램 개발시 참고 (Mifare Classic카드의 섹터에 특정키값으로 로그인하는 방법, 특정 Sector에 로그인 후 그 Sector에 있는 특정 Block에 데이터를 read/write하는 방법, 금액(Value)을 충전, 차감, 조회하는 기능을 구현)
- Other PICC Cards : Mifare Classic카드를 제외한 ISO14443 A또는 B타입카드중 part4 규격에 부합하여 ISO7816-4의 APDU커맨드를 보낼수 있는 스마트카드로 개발작업을 진행할 때 참고

< Device Programming 폴더 >

- Device Programming : ACR122S 리더기내의 부속장치들을 컨트롤할 수 있는 샘플 코드 (예 : LED 제어, 비프음 제어, 통신속도 등)
- Get ATR : 삽입된 스마트카드의 ATR값(Answer To Reset:리셋 명령을 보낸후 카드로부터 오는 응답값)을 얻는 샘플코드
- Peer to Peer : NFC리더와 ACR122S 리더기의 P2P 통신시 사용. 스마트폰과 P2P는 지원 안되고 일반적인 카드어플리케이션 개발시 참고할 필요 없음.
- Polling : 랜덤하게 스마트카드가 삽입(근접)되어졌을때 리더기가 카드를 인식하여 프로그램에 응답을 주는 절차를 구현 =>폴링이라고 표현함.

< SAM Programming 폴더 >

- Key Management : SAM Interface 사용을 위한 샘플코드로 구성 (예 : Power On ICC/활성화, 초기화, 키값 세팅 등)

◎ 적용할 RFID카드/태그 종류에 따른 참고 자료

- ISO14443 타입A 또는 B카드중 part4 규격에 맞는 카드와 Mifare Classic카드는 표준APDU 커맨드를 사용하여 명령을 전송하며 Felica카드, ISO18092카드(Topaz, Jewel등) 그리고 리더기의 부속장치(LED, 비프음, 안테나)를 제어할때는 Pseudo APDU(유사APDU)를 사용합니다.

1) Mifare Classic(1K, 4K, Mini)카드와 ISO14443 part4 규격에 부합되는 카드로 개발작업시
관련자료 : CMP_ACR122S 메뉴얼 31쪽 또는 샘플코드의 Contactless Cards 폴더 참고.

2) Mifare DESFire 카드로 개발시 관련자료 : 아래 링크된 사이트 참고.

카드커맨드가 나와있는 데이터시트의 경우 NXP와 NDA에 서명후에 NXP본사로부터 메일로
자료 받을수 있음.(1~2주 소요)

<http://ridrix.wordpress.com/2009/09/19/mifare-desfire-communication-example/>

(Mifare DESFire에 커맨드를 보내는 방식은 전용방식인 Native Command모드와 ISO7816 Wrapping모드, ISO7816 표준APDU모드가 있음)

3) Felica카드로 개발작업시 관련자료 : CMP_ACR122S 메뉴얼 40쪽 참고

4) ISO18092카드(Topaz, Jewel 등)로 개발작업시 관련자료 : CMP_ACR122S 메뉴얼 42~46쪽
참고, 또는 각 태그의 datasheet참고(예 : Topaz Card Specification.pdf)

※ Mifare 1K / Mifare UltraLight 카드의 구조에 대한 요약설명

Mifare 1K : 고유 씨리얼번호인 UID는 4byte(16진수 여덟자리)로 나타나며 총 16개(0x00~0x0F)의 섹터가 있으며 당연히 각 섹터는 64byte가 됩니다.(1024byte / 16 = 64byte)

그리고 각 섹터는 4의개 Block(0x00~0x03)이 있으며 각 Block의 사이즈는 16byte입니다. 이중 각 Sector별 0,1,2번 Block은 user영역으로 key값만 알면 마음대로 read/write가 가능합니다(각 Sector별로 로그인 키값을 다르게 세팅할 수 있습니다.)

그리고 각 Sector의 마지막 Block인 3번 Block(각 섹터의 4번째 블록)은 각 Sector의 key값(Key A : 6byte, access bit값: 4byte, Key B: 6byte)이 저장됩니다.

(KeyB값(3번 Block의 마지막 6byte)을 사용하지 않을 경우 user영역으로 사용할 수 있는데 일반적으로는 건드리지 않고 비워둡니다.)

이때 access bit값을 어떻게 주느냐에 따라 read only, read/write 가능, 둘다 불가 등등의 메모리 접근조건을 설정할 수 있습니다. (제조사 기본 출고값은FF FF FF FF FF FF(Key A) + FF 07 80 69(접근조건) + FF FF FF FF FF FF(Key B) 로 설정되어 있습니다.)

해서 키값을 바꾸고 싶으시면 원하시는 Sector에 기본 key값(FF FF FF FF FF FF)으로 로그인한 후에 3번 Block에 write하면 됩니다.

(예를 들어 4번 Sector의 키값을01 02 03 04 05 06으로 바꾸고 싶으시면 FF FF FF FF FF FF로 로그인한 후에 write명령을 날리신후 writing할 값을 01 02 03 04 05 06 FF 07 80 69 로 저장하면 됩니다. 이렇게 writing을 하면 다음번 로그인시 바뀐 key값으로 로그인하면 됩니다.)

그리고 아래 사이트는 Mifare 1K카드에 대해 잘 설명해놓은 블로그이니 참고하시기 바랍니다.
<http://blog.naver.com/toynbee12/130015487052>

Mifare UltraLight(Mifare UL)는 총 64byte로 구성되며 UID는 7byte(16진수 열네자리)로 나타납니다. 총 16개의 Page가 있고 한 Page당 4개의 바이트가 있습니다. Mifare UL은 key값을 넣어 로그인하는 절차가 없습니다. Default상태는 마음대로 read & write가 허용되는 상태입니다.

그리고 Mifare UL의 두가지 특수한 기능이 있는데 하나는 read only만 가능하게끔(write불가) Lock을 설정하는 기능과 OTP기능이라고해서 금액(Value)형태로 데이터를 저장(write)할 때 금액이 차감(Decrement)만 되고 충전(Increment)는 되지 않게 세팅할 때 사용됩니다.(이런 케이스는 보통 할인권이나 1회용 티켓용으로 사용되어질 때 더 이상 충전해서 사용 못하게 할 때 적용하는 기능입니다.) Mifare UL카드내의 세팅은 0x03 page의 0x00, 0x01, 0x02, 0x03 바이트 영역에있는 OTP0, OTP1, OTP2, OTP3의 비트값을 데이터시트의 6.5.3 OTP bytes에 따라 세팅해주시면 됩니다
 마찬가지로 Lock기능은 0x02 page의 0x02, 0x03 바이트 영역에 있는 Lock0, Lock1의 비트값을 데이터시트의 6.5.2 Lock bytes에 따라 세팅해주시면 됩니다.

간단히 예를 들면 만약 0x06 page를 다른 사람이 write못하게 read only상태로 만들고 싶다면 Lock0의 비트값을 01000010(0x42), Lock1의 비트값을 00000000(0x00)로 세팅하면되고 다른 예로 0x05 page와 0x11 page를 read only상태로 만들고 싶다면 Lock0의 비트값을 00100110(0x26), Lock1의 비트값을 00001000(0x08)로 세팅하면 됩니다.

◎ 스마트카드 / NFC관련 약어

스마트카드 약어	
ACK	Acknowledge 승인 응답문자 : 데이터 전송이 성공이면 ACK코드값을 되돌려 받음.
APDU	Application Protocol Data Unit 스마트카드에서 사용하는 응용 프로토콜 데이터 단위(스마트카드와 리더기간의 데이터 교환 형식)
ATQA	Answer To Request, type A 요청에 의한 응답, ISO14443 TypeA 카드로부터의 요청에 의한 응답
ATQB	Answer To Request, type B 요청에 의한 응답, ISO14443 Type B 카드로부터의 요청에 의한 응답
ATR	Answer To Reset 스마트카드에 보내진 리셋신호 대한 응답값으로 ISO 7816 표준으로 정의되어짐. 스마트 카드가 리더기에 삽입되면 외부 장치(PC등)로부터 보내오는 명령어에 따라서 리셋 작업을 수행하며, 리셋 작업의 결과로 카드의 종류, 모델(버전), 제작 회사 등의 정보를 담고 있는 응답신호(ATR)를 외부장치로 보냄. 외부장치<->리더기<->스마트카드간의 최초 정상통신 및 동작여부를 확인할 수 있는 응답값
CCID	Chip/Smart Card Interface Device 스마트카드 장치(리더기)에 관련된 표준규격으로 Microsoft에서는 Windows Vista이상버전의 O/S는 기본적으로 CCID드라이버를 탑재하고 있어서 제조사쪽에서 제공하는 별도의 장치(리더기)드라이버를 설치할 필요가 없음.
COS	Card Operating System(스마트카드 운영체제) 스마트카드 운영체제(O/S)는 크게 두가지로 나뉘는데 개방형 플랫폼과 폐쇄형 플랫폼이 있음. 개방형카드의 종류는 크게 MULTOS(Multi-Application Operating System)와 Java카드(Global platform)가 있으며 폐쇄형 카드는 Proprietary(전용) O/S를 사용하는 카드를 일컫음. MULTOS: 1996년 Mondex사가 개발하고 MASCO산업 컨소시엄이 지원하는 다기능 COS로 MULTOS의 COS위에 MULTOS의 인터프리트 언어인 MEL로 설계된 MEL가상머신을 탑재한 형태의 스마트카드 플랫폼 Java 카드: 자바카드는 COS위에 SUN사가 개발한 자바가상머신(JVM)을 탑재한 형태의 스마트카드 플랫폼으로 현재 Javacard 버전 2.2.1 사용되고 있음.
EMV	Europay / Master card / Visa card 세계 3대 신용 카드 회사인 벨기에의 유로페이, 미국의 마스터 카드, 비자 카드 등 3개사가 공동으로 결제하는 IC 카드의 표준 규격. 이름은 3개사의 머리글자를 따서 붙여짐. 미국의 아메리칸 익스

	프레스 등 다른 카드 회사도 이 규격을 지지하고 있으며, IC 카드형 전자 화폐를 대표하는 사실상의 표준 EMV인증은 H/W(스마트카드단말기)와 관련된 EMV2000 Level1 인증과 S/W(단말기펌웨어, 결재모듈)와 관련된 EMV2000 Level2 인증이 있음
NAK	Negative Acknowledge 부정 응답문자 : 데이터 수신에 에러가 있음을 확인하는 신호.
PC/SC	Personal Computer/Smart Card 종류가 다른 스마트카드와 PC간의 상호 호환을 위해 Microsoft, HP, Sun Microsystems등 의 7개가 Workgroup을 결성하여 PC/SC란 표준규격을 제정함. 이 표준규격은 제조회사가 다른 스마트카드와 리더기사이에 상호 호환을 준수하며 다중응용프로그램간에 자원공유를 가능하게함으로써 H/W구매 비용과 개발비용을 줄여주고 있음. 현재 PC/SC 1.0버전이 사용되고 있음(PC/SC part 1~8) PC/SC 구조는 Interface Device Handler, ICC Resource Manager, Service provider로 구성됨 - Interface Device Handler : 리더기를 위한 장치드라이버로써 리더기의 기능을 I/O인터페이스로 매핑시키는 역할을 수행함. - ICC Resource Manager : PC에서 사용하는 스마트카드와 리더기로의 모든 접근을 조정 및 제어하고 스마트카드와 관련된 모든 자원을 관리함. (Windows시리즈의 경우 ScardSvr.exe, ScardSvr.dll) - Service Provider : 특정 스마트카드에서 제공되는 기능을 캡슐화하고 API를 통하여 어플리케이션이 스마트카드로 접근할수 있도록 함.(Windows용 스마트카드 API는 winscard.dll에 구현 되어있음)
PCSC Lite	Personal Computer/Smart Card Lite Linux용에서 규정된 스마트카드 표준으로 Windows용 PC/SC 구조 및 PC/SC API와 유사
PICC	Proximity Integrated Circuit Card Contactless Card로써 RFID카드를 지칭함. ICC는 접촉식 카드
PIN	Personal Identification Number 스마트카드에 저장되어지는 개인별 식별(보안) 번호로 8자리 숫자로 구성됨.
R-APDU	Response APDU 응답 APDU
REQA	REQuest Answer, type A ISO14443 Type A카드에 대한 응답 요청
REQB	REQuest Answer, type B ISO14443 Type B카드에 대한 응답 요청
SAK	Select Acknowledge, type A Type A의 카드타입 응답값
SAM	Secure Access Module 카드 판독기(리더기) 내부에 장착되어 카드와 단말기의 유효성을 인증하고 통신 데이터를 암호화하여 정보의 노출 방지 및 통신 메시지의 인증 및 검증함. 또한 카드에서 이전된 전자적인 가치를 저장하기도 함. 인터넷 전자 상거래 시 또는 개인용 컴퓨터(PC) 사용 시 프로그램 안에 카드 인증용 SAM을 내장하기도 한다
SIM	Subscriber Identity Module(가입자 인증 모듈) 휴대폰에서 사용되는 가입자(사용자) 정도를 저장하는 커팅된 스마트카드의 일종
T=CL	ISO/IEC14443 part4를 지원하는 RFID카드에서 사용하는 프로토콜 접촉식카드의 ISO7816 part4 에서 사용하는 전송 프로토콜과 유사
WUPA	ISO14443 TypeA용 Wake-Up Command
WUPB	ISO14443 TypeB용 Wake-Up Command

NFC 약어																					
LLCP	Logical Link Control Protocol - NFC장치간 신뢰성있는 P2P(Peer To Peer)통신을 목적으로 설계된 프로토콜 - Peer Mode로 동작 시 디지털 프로토콜 상에서 링크제어계층 정의 - P2P 통신, 연결/비 연결 기반 전송계층 및 프로토콜 멀티플렉싱 제공 - NFC 칩 세트 또는 기기의 소프트웨어에 구현																				
NCI	NFC Controller Interface 호스트와 NFC 칩 간의 통신 인터페이스 정의																				
NDEF	NFC Data Exchange Format - NFC 응용데이터(애플리케이션)를 위한 표준 데이터형식 제공 - 메시지 형식 정의 - 하나 이상의 레코드로 구성 가능한 메시지 형식 - 분할 가능한 메시지 형태 정의 - NDEF 메시지는 최대 4GB 크기 지원 가능하지만, 실제적으로 태그의 메모리 크기에 따라 제한됨																				
NPP	NDEF Push Protocol 한 개의 NFC장치에서 다른 NFC장치로 NDEF 데이터를 보내기 위해 설계된 LLCP의 가장 윗단에 배치된 simple 프로토콜																				
SNEP	Simple NDEF Exchange Protocol - 두 개의 NFC 단말기간(P2P) 데이터 송수신을 구현하는 프로토콜 - SNEP 클라이언트와 서버 애플리케이션 간의 요청/응답 프로토콜 - 신뢰할 수 있는 데이터 교환을 위해 NFC LLCP의 Connection-oriented 전송 모드를 사용																				
	<table border="1"> <tr> <td>NFC 서비스 형태</td> <td>세부설명 (SNEP를 통해 구현 가능한 NFC 서비스 형태)</td> </tr> </table>	NFC 서비스 형태	세부설명 (SNEP를 통해 구현 가능한 NFC 서비스 형태)																		
	NFC 서비스 형태	세부설명 (SNEP를 통해 구현 가능한 NFC 서비스 형태)																			
	<table border="1"> <tr> <td>연락처 전송</td> <td>- NFC 기능이 탑재된 휴대폰을 이용해 명함 등 연락처 정보를 전송 --> 수신인은 해당 정보를 휴대폰 연락처에 입력 가능</td> </tr> </table>	연락처 전송	- NFC 기능이 탑재된 휴대폰을 이용해 명함 등 연락처 정보를 전송 --> 수신인은 해당 정보를 휴대폰 연락처에 입력 가능																		
	연락처 전송	- NFC 기능이 탑재된 휴대폰을 이용해 명함 등 연락처 정보를 전송 --> 수신인은 해당 정보를 휴대폰 연락처에 입력 가능																			
<table border="1"> <tr> <td>정보 수집</td> <td>- NFC 태그가 내장된 영화 포스터앞을 지나갈 때 휴대폰으로 영화 등에 관한 정보를 수집 --> 집에 도착했을 때 휴대폰을 NFC기능이 탑재된 리모콘에 가볍게 갖다 대기만 하면 TV에서 영화 광고를 시청 가능</td> </tr> </table>	정보 수집	- NFC 태그가 내장된 영화 포스터앞을 지나갈 때 휴대폰으로 영화 등에 관한 정보를 수집 --> 집에 도착했을 때 휴대폰을 NFC기능이 탑재된 리모콘에 가볍게 갖다 대기만 하면 TV에서 영화 광고를 시청 가능																			
정보 수집	- NFC 태그가 내장된 영화 포스터앞을 지나갈 때 휴대폰으로 영화 등에 관한 정보를 수집 --> 집에 도착했을 때 휴대폰을 NFC기능이 탑재된 리모콘에 가볍게 갖다 대기만 하면 TV에서 영화 광고를 시청 가능																				
<table border="1"> <tr> <td>정보 교환</td> <td>- 다른 사람의 휴대폰에 자신의 휴대폰을 가볍게 접촉시킴으로써 쿠폰, 특별할인 혜택, 광고, 게임, 마케팅 서비스 등 스마트 포스트에서 수집된 정보를 간단하게 교환 가능</td> </tr> </table>	정보 교환	- 다른 사람의 휴대폰에 자신의 휴대폰을 가볍게 접촉시킴으로써 쿠폰, 특별할인 혜택, 광고, 게임, 마케팅 서비스 등 스마트 포스트에서 수집된 정보를 간단하게 교환 가능																			
정보 교환	- 다른 사람의 휴대폰에 자신의 휴대폰을 가볍게 접촉시킴으로써 쿠폰, 특별할인 혜택, 광고, 게임, 마케팅 서비스 등 스마트 포스트에서 수집된 정보를 간단하게 교환 가능																				
RTD	NFC Record Type Definition - NDEF 메시지의 데이터 유형에 대해 식별하기 위한 확장 구조를 제공 - 레코드 유형은 데이터의 의미를 식별																				
	<table border="1"> <thead> <tr> <th>Definition</th> <th>NDEF Record Type</th> <th>Definition</th> <th>NDEF Record Type</th> </tr> </thead> <tbody> <tr> <td>Smart Poster</td> <td>Sp</td> <td>Handover Request</td> <td>Hr</td> </tr> <tr> <td>Text</td> <td>T</td> <td>Handover Select</td> <td>Hs</td> </tr> <tr> <td>URI</td> <td>U</td> <td>Handover Carrier</td> <td>Hc</td> </tr> <tr> <td>Generic Control</td> <td>Gc</td> <td>Signature</td> <td>Sig</td> </tr> </tbody> </table>	Definition	NDEF Record Type	Definition	NDEF Record Type	Smart Poster	Sp	Handover Request	Hr	Text	T	Handover Select	Hs	URI	U	Handover Carrier	Hc	Generic Control	Gc	Signature	Sig
	Definition	NDEF Record Type	Definition	NDEF Record Type																	
	Smart Poster	Sp	Handover Request	Hr																	
	Text	T	Handover Select	Hs																	
	URI	U	Handover Carrier	Hc																	
Generic Control	Gc	Signature	Sig																		

◎ **Error code 정리자료** (CMP_ACR122S 메뉴얼 47쪽)

Error	Error Code
No Error	0x00
Time Out, the target has not answered	0x01
A CRC error has been detected by the contactless UART	0x02
A Parity error has been detected by the contactless UART	0x03
During a Mifare anti-collision/select operation, an erroneous Bit Count has been detected	0x04
Framing error during Mifare operation	0x05
An abnormal bit-collision has been detected during bit wise anti-collision at 106 kbps	0x06
Communication buffer size insufficient	0x07
RF Buffer overflow has been detected by the contactless UART (bit BufferOvfl of the register CL_ERROR)	0x08
In active communication mode, the RF field has not been switched on in time by the counterpart (as defined in NFCIP-1 standard)	0x0A
RF Protocol error (cf. reference [4], description of the CL_ERROR register)	0x0B
Temperature error: the internal temperature sensor has detected overheating, and therefore has automatically switched off the antenna drivers	0x0D
Internal buffer overflow	0x0E
Invalid parameter (range, format, ...)	0x10
DEP Protocol: The chip configured in target mode does not support the command received from the initiator (the command received is not one of the following: ATR_REQ, WUP_REQ, PSL_REQ, DEP_REQ, DSL_REQ, RLS_REQ, ref. [1]).	0x12
DEP Protocol / Mifare / ISO/IEC 14443-4: The data format does not match to the specification. Depending on the RF protocol used, it can be: • Bad length of RF received frame, • Incorrect value of PCB or PFB, • Invalid or unexpected RF received frame, • NAD or DID incoherence.	0x13
Mifare: Authentication error	0x14
ISO/IEC 14443-3: UID Check byte is wrong	0x23
DEP Protocol: Invalid device state, the system is in a state which does not allow the operation	0x25
Operation not allowed in this configuration (host controller interface)	0x26
This command is not acceptable due to the current context of the chip (Initiator vs. Target, unknown target number, Target not in the good state, ...)	0x27
The chip configured as target has been released by its initiator	0x29
ISO/IEC 14443-3B only: the ID of the card does not match, meaning that the expected card has been exchanged with another one.	0x2A
ISO/IEC 14443-3B only: the card previously activated has disappeared.	0x2B
Mismatch between the NFCID3 initiator and the NFCID3 target in DEP 212/424 kbps passive.	0x2C
An over-current event has been detected	0x2D
NAD missing in DEP frame	0x2E