

목차

Smartcard API 1

구조체 1

 BS2CSNCard 1

 BS2SmartCardHeader 2

 BS2SmartCardCredentials 3

 BS2AccessOnCardData 3

 BS2SmartCardData 3

 BS2Card 4

Smartcard API

카드 데이터를 읽고 쓰는 함수를 제공하는 API입니다.

- [BS2_ScanCard](#): 장치에서 카드를 스캔하고 카드 데이터를 분석합니다.
- [BS2_WriteCard](#): SmartCard에 데이터를 저장합니다.
- [BS2_EraseCard](#): SmartCard를 포맷합니다.

구조체

BS2CSNCard

```
typedef struct {
    uint8_t type;
    uint8_t size;
    uint8_t data[BS2_CARD_DATA_SIZE];
} BS2CSNCard;
```

1. type

카드 유형의 코드 값입니다. 카드 유형은 카드 목적을 나타냅니다. 장치에서 서버로 사용자를 전송할 때, Access 카드는 카드에 기록된 정보만으로 동작이 가능하기 때문에 이슈 기록을 유지하는 데에만 사용됩니다.

값	설명	사용된 형식
0x00	알려지지 않은 카드	
0x01	CSN 카드	
0x02	Secure 카드	
0x03	Access 카드	
0x0A	Wiegand 카드	BS2WiegandConfig.format (이 형식은 BS2WiegandConfig.CSNIndex 혹은 BS2WiegandConfig.CardMask가 0으로 설정될 때 사용됩니다.)
0x1A	Wiegand 카드	BS2WiegandMultiConfig.formats[0]
0x2A	Wiegand 카드	BS2WiegandMultiConfig.formats[1]
0x3A	Wiegand 카드	BS2WiegandMultiConfig.formats[2]
0x4A	Wiegand 카드	BS2WiegandMultiConfig.formats[3]
0x5A	Wiegand 카드	BS2WiegandMultiConfig.formats[4]
0x6A	Wiegand 카드	BS2WiegandMultiConfig.formats[5]
0x7A	Wiegand 카드	BS2WiegandMultiConfig.formats[6]
0x8A	Wiegand 카드	BS2WiegandMultiConfig.formats[7]
0x9A	Wiegand 카드	BS2WiegandMultiConfig.formats[8]
0xAA	Wiegand 카드	BS2WiegandMultiConfig.formats[9]
0xBA	Wiegand 카드	BS2WiegandMultiConfig.formats[10]
0xCA	Wiegand 카드	BS2WiegandMultiConfig.formats[11]
0xDA	Wiegand 카드	BS2WiegandMultiConfig.formats[12]
0xEA	Wiegand 카드	BS2WiegandMultiConfig.formats[13]
0xFA	Wiegand 카드	BS2WiegandMultiConfig.formats[14]

2. *size*

카드 템플릿의 크기입니다.

3. *data*

카드 템플릿 데이터입니다.

BS2SmartCardHeader

```
typedef struct {
    uint16_t hdrCRC;
    uint16_t cardCRC;
    BS2_CARD_TYPE cardType;
    uint8_t numOfTemplate;
    uint16_t templateSize;
    uint16_t issueCount;
    uint8_t duressMask;
    uint8_t reserved[5];
} BS2SmartCardHeader;
```

1. *hdrCRC*

card header 체크섬 값입니다.

2. *cardCRC*

card data 체크섬 값입니다.

3. *cardType*

카드 유형의 코드 값입니다.

값	설명
0	알려지지 않은 카드
1	CSN 카드
2	Secure 카드
3	Access 카드
4	Wiegand 카드

4. *numOfTemplate*

지문 템플릿의 개수입니다.

5. *templateSize*

지문 템플릿의 크기입니다. 일반적인 지문 템플릿의 크기는 384 바이트로 고정되어 있습니다.

BioStar 2에서 스마트 카드를 사용하는 경우 기본값은 300 바이트이며 필요에 따라 변경할 수 있지만 템플릿의 크기가 너무 작게 설정되면 지문 매칭에 문제가 발생할 수 있으므로 300 바이트 이상으로 설정하는 것을 권장합니다.

6. *issueCount*

스마트 카드 발급 회차입니다.

7. *duressMask*

협박 지문 유무 마스크 값입니다.

BS2SmartCardCredentials

```
typedef struct {
    uint8_t pin[BS2_PIN_HASH_SIZE];
    uint8_t templateData[BS2_SMART_CARD_MAX_TEMPLATE_COUNT *
BS2_FINGER_TEMPLATE_SIZE];
} BS2SmartCardCredentials;
```

1. *pin*

pin code 값입니다.

2. *templateData*

지문 템플릿 데이터 영역으로 최대 4개의 지문 템플릿을 저장할 수 있습니다.

BS2AccessOnCardData

```
typedef struct {
    uint16_t accessGroupID[BS2_SMART_CARD_MAX_ACCESS_GROUP_COUNT];
    BS2_DATETIME startTime;
    BS2_DATETIME endTime;
} BS2AccessOnCardData;
```

1. ***accessGroupID***

출입 그룹 아이디 리스트입니다.

2. ***startTime***

사용자 인증이 가능한 시작 시간이며, 0일 경우 제한이 없습니다.

3. ***endTime***

사용자 인증이 가능한 마지막 시간이며, 0일 경우 제한이 없습니다.

BS2SmartCardData

```
typedef struct {
    BS2SmartCardHeader header;
    uint8_t cardID[BS2_CARD_DATA_SIZE];
    BS2SmartCardCredentials credentials;
    BS2AccessOnCardData accessOnData;
} BS2SmartCardData;
```

1. ***header***

SmartCard 헤더입니다.

2. ***cardID***

단말기에서 사용할 카드 식별자입니다. Access on Card의 경우 배열의 32 byte를 Card ID로 사용하고, Secure Credential card는 배열의 24 byte를 Card ID로 사용하게 됩니다.

3. ***credentials***

pin code나 지문 템플릿이 저장되어 있는 인증 데이터 영역입니다.

4. **accessOnData**

AOC 카드에서 사용하는 영역으로 출입 그룹 정보를 가지고 있습니다.

BS2Card

```
typedef struct {  
    uint8_t isSmartCard;  
    union {  
        BS2CSNCard card;  
        BS2SmartCardData smartCard;  
    };  
}BS2Card;
```

1. **isSmartCard**

SmartCard 여부를 나타내는 flag입니다.

2. **card**

일반 카드 데이터

3. **smartCard**

SmartCard 데이터

From:

<https://kb.supremainc.com/kbtest/> - **BioStar Device SDK**

Permanent link:

https://kb.supremainc.com/kbtest/doku.php?id=ko:smartcard_api&rev=1519172439

Last update: **2018/02/21 09:20**