# 목차

**Device API** ..... 1
*구조체* ..... 1
BS2SimpleDeviceInfo ..... 1
BS2SimpleDeviceInfoEx ..... 4
BS2ResourceElement ..... 5
BS2IPv6DeviceInfo ..... 6
BS2AuthOperatorLevel ..... 7
BS2DeviceCapabilities ..... 7

# Device API

장치 정보를 제어하거나 펌웨어를 업그레이드할 수 있습니다.

- BS2_GetDeviceInfo: 장치 정보를 가져옵니다.
- BS2_GetDeviceInfoEx: [+ 2.6.0] 장치 추가정보를 가져옵니다.
- BS2_GetDeviceTime: 현재 시간을 가져옵니다.
- BS2_SetDeviceTime: 현재 시간을 설정합니다.
- BS2_ClearDatabase: 사용자 정보와 Blacklist를 초기화합니다.
- BS2_FactoryReset: 모든 설정 정보와 데이터 베이스를 초기화합니다.
- BS2_RebootDevice: 장치를 다시 시작합니다.
- BS2_LockDevice: 장치 잠금을 설정하여 사용자 인증을 거부합니다.
- BS2_UnlockDevice: 장치 잠금을 해제하여 사용자 인증을 허용합니다.
- BS2_SetKeepAliveTimeout: 장치의 keep-alive 시간을 설정합니다.
- BS2_UpgradeFirmware: 펌웨어를 업그레이드합니다.
- BS2_UpdateResource: 리소스를 업데이트합니다.
- BS2_GetSpecifiedDeviceInfo: [+ 2.6.3] 장치 정보를 선택적으로 가져옵니다.
- BS2_GetAuthOperatorLevelEx: [+ 2.6.3] 주어진 장치 관리자를 가져옵니다. (1000명 관리자 지원)
- BS2_GetAllAuthOperatorLevelEx: [+ 2.6.3] 모든 장치 관리자를 가져옵니다. (1000명 관리자 지원)
- BS2_SetAuthOperatorLevelEx: [+ 2.6.3] 장치 관리자를 추가합니다. (1000명 관리자 지원)
- BS2_RemoveAuthOperatorLevelEx: [+ 2.6.3] 주어진 장치 관리자를 삭제합니다. (1000명 관리자 지원)
- BS2_RemoveAllAuthOperatorLevelEx: [+ 2.6.3] 모든 장치 관리자를 삭제합니다. (1000명 관리자 지원)
- BS2_GetDeviceCapabilities: [+ 2.8] 장치가 지원하는 정보를 일괄 취득합니다.

## 구조체

### BS2SimpleDeviceInfo

```
typedef struct
{
    uint32_t id;
    uint16_t type;
    uint8_t connectionMode;
    uint32_t ipv4Address;
    uint16_t port;
    uint32_t maxNumOfUser;
    uint8_t userNameSupported;
    uint8_t userPhotoSupported;
    uint8_t pinSupported;
    uint8_t cardSupported;
    uint8_t fingerSupported;
    uint8_t faceSupported;
    uint8_t wlanSupported;
    uint8_t tnaSupported;
```

```
    uint8_t triggerActionSupported;
    uint8_t wiegandSupported;
    uint8_t imageLogSupported;
    uint8_t dnsSupported;
    uint8_t jobCodeSupported;
    uint8_t wiegandMultiSupported;
    uint8_t rs485Mode;
    uint8_t sslSupported;
    uint8_t rootCertExist;
    uint8_t dualIDSupported;
    uint8_t useAlphanumericID;
    uint32_t connectedIP;
    uint8_t phraseCodeSupported;
    uint8_t card1xSupported;
    uint8_t systemExtSupported;
    uint8_t voipSupported;
}BS2SimpleDeviceInfo;
```

1. *id*
장치 식별자이며 항상 1 이상입니다.

2. *type*
장치 유형의 코드 값입니다.

| 값 | 설명 |
|---|---|
| 0x01 | BioEntry Plus |
| 0x02 | BioEntry W |
| 0x03 | BioLite Net |
| 0x04 | Xpass |
| 0x05 | Xpass S2 |
| 0x06 | Secure IO 2 |
| 0x07 | DM-20 |
| 0x08 | BioStation 2 |
| 0x09 | BioStation A2 |
| 0x0A | FaceStation 2 |
| 0x0B | IO Device |
| 0x0C | BioStation L2 |
| 0x0D | BioEntry W2 |
| 0x0E | CoreStation 40 |
| 0x0F | Output Module |
| 0x10 | Input Module |
| 0x11 | BioEntry P2 |
| 0x12 | BioLite N2 |
| 0x13 | XPass2 |
| 0x14 | XPass S3 |
| 0x15 | BioEntry R2 |
| 0x16 | XPass D2 |
| 0x17 | Door Module 21 |
| 0xFF | Unknown Type |

3. *connectionMode*
BioStar 애플리케이션과 장치간의 접속 모드를 뜻하며, 접속 주체에 따라 direct mode(0x0)와 server mode(0x1)로 구분됩니다. direct mode는 BioStar 애플리케이션이 장치로 접속하며 server mode는 장치가 BioStar 애플리케이션으로 접속합니다. 장치에 설정된 기본 접속 모드는 direct mode이며 접속 모드를 수정하려면 IP Config를 참고하십시오.

4. *ipv4Address*
현재 설정된 장치의 IP 주소입니다.

5. *port*
현재 설정된 장치의 TCP 포트 번호입니다.

6. *maxNumOfUser*
장치에서 저장할 수 있는 최대 사용자 개수입니다.

7. *userNameSupported*
장치에서 사용자 이름 사용을 지원하는지 알 수 있는 flag입니다.

8. *userPhotoSupported*
장치에서 사용자 프로필 사진 사용을 지원하는지 알 수 있는 flag입니다.

9. *pinSupported*
장치에서 PIN 사용을 지원하는지 알 수 있는 flag입니다.

10. *cardSupported*
장치에서 스마트 카드 인증을 지원하는지 알 수 있는 flag입니다.

11. *fingerSupported*
장치에서 지문 인증을 지원하는지 알 수 있는 flag입니다.

12. *faceSupported*
장치에서 얼굴 인식을 지원하는지 알 수 있는 flag입니다.

13. *wlanSupported*
장치에서 무선랜을 지원하는지 알 수 있는 flag입니다.

14. *tnaSupported*
장치에서 근태관리를 지원하는지 알 수 있는 flag입니다.

15. *triggerActionSupported*
장치에서 trigger action을 지원하는지 알 수 있는 flag입니다.

16. *wiegandSupported*
장치에서 Wiegand를 지원하는지 알 수 있는 flag입니다.

17. *imageLogSupported*
장치에서 이미지 로그를 지원하는지 알 수 있는 flag입니다.

18. *dnsSupported*
장치에서 DNS를 지원하는지 알 수 있는 flag입니다.

19. *jobCodeSupported*
장치에서 Job Code를 지원하는지 알 수 있는 flag입니다.

20. *wiegandMultiSupported*
장치에서 wiegandMulti를 지원하는지 알 수 있는 flag입니다.

21. *rs485Mode*
장치에서 RS485 모드입니다.

22. *sslSupported*
장치에서 ssl를 지원하는지 알 수 있는 flag입니다.

23. *rootCertExist*
장치에서 root인증서가 존재하는지를 알 수 있는 flag입니다.

24. *dualIDSupported*
장치에서 dualID(숫자,문자)를 지원하는지 알 수 있는 flag입니다.

25. *useAlphanumericID*
장치에서 AlphanumericID를 사용중인지 여부를 말하는 flag입니다.

26. *connectedIP*
장치가 접속하고 있는 ip주소입니다. (0xFFFFFFFF if disconnected)

27. *phraseCodeSupported*
장치에서 사용자별 개인 메세지를 지원하는지 알 수 있는 flag입니다.

28. *card1xSupported*
장치에서 1.x ToC 카드 읽기를 지원하는지 알 수 있는 flag입니다.

29. *systemExtSupported*
장치에서 RS-485 키 설정을 지원하는지 알 수 있는 flag입니다.

30. *voipSupported*
장치에서 VoIP를 지원하는지 알 수 있는 flag입니다.

## BS2SimpleDeviceInfoEx

BS2SimpleDeviceInfo 외 지원정보를 가져옵니다.

```
typedef struct
{
    enum
    {
        BS2_SUPPORT_RS485EX     = 0x00000001,
        BS2_SUPPORT_CARDEX      = 0x00000002,
        BS2_SUPPORT_DST         = 0x00000004,
        BS2_SUPPORT_DESFIREEX   = 0x00000008,
        BS2_SUPPORT_FACE_EX = 0x00000010,

        BS2_SUPPORT_FINGER_SCAN = 0x00010000,
        BS2_SUPPORT_FACE_SCAN   = 0x00020000,
        BS2_SUPPORT_FACE_EX_SCAN = 0x00040000,
```

```
        BS2_SUPPORT_ALL         = BS2_SUPPORT_RS485EX | BS2_SUPPORT_CARDEX |
BS2_SUPPORT_DST | BS2_SUPPORT_DESFIREEX | BS2_SUPPORT_FACE_EX |
BS2_SUPPORT_FINGER_SCAN | BS2_SUPPORT_FACE_SCAN | BS2_SUPPORT_FACE_EX_SCAN,
    };

    uint32_t supported;
    uint8_t  reserved[4];
}BS2SimpleDeviceInfoEx;
```

1. *supported*
현재 장치가 BS2SimpleDeviceInfo 에서 제공하는 기능 외의 정보를 추가적으로 가져옵니다.
아래 정의된 값으로 bit masking을 하여 지원 여부를 확인할 수 있습니다.

| 정의 | 값 | 설명 |
|---|---|---|
| BS2_SUPPORT_RS485EX | 0x00000001 | RS485 확장 지원 여부 (CoreStation 40에 해당) |
| BS2_SUPPORT_CARDEX | 0x00000002 | iClass SEOS 카드 사용 여부 |
| BS2_SUPPORT_DST | 0x00000004 | 일광절약 시간 사용 여부 |
| BS2_SUPPORT_DESFIREEX | 0x00000008 | DesFire 카드 고급설정 지원 여부 [+ V2.6.4] |
| BS2_SUPPORT_FACE_EX | 0x00000010 | 얼굴인증 지원 여부 FaceStation F2 [+ V2.7.1] |
| BS2_SUPPORT_FINGER_SCAN | 0x00010000 | 지문 검출이 가능한 장치 인지 여부 [+ V2.7.1] |
| BS2_SUPPORT_FACE_SCAN | 0x00020000 | 얼굴 검출이 가능한 장치 인지 여부 FaceStation2, FaceLite [+ V2.7.1] |
| BS2_SUPPORT_FACE_EX_SCAN | 0x00040000 | 얼굴 검출이 가능한 장치 인지 여부 FaceStation F2 [+ V2.7.1] |
| BS2_SUPPORT_ALL | 0x0000000F | 추가 정보 전체 제공 여부 |

2. *reserved*
예약된 공간입니다.

## BS2ResourceElement

```
typedef struct
{
    uint8_t type;
    uint32_t numResData;
    struct {
        uint8_t index;
        uint32_t dataLen;
        uint8_t* data;
    }resData[128];
}BS2ResourceElement;
```

1. *type*
리소스 데이터 타입입니다.

| 값 | 설명 | 지원하는 데이터 형식 |
|---|---|---|
| 0 | UI(Langauge pack) | 슈프리마 언어팩 |
| 1 | Notice message | UTF-8 문자열 |

| 값 | 설명 | 지원하는 데이터 형식 |
|---|---|---|
| 2 | Image(Background) | PNG |
| 3 | Slide image | PNG |
| 4 | Sound | WAVE |

2. *numResData*
리소스 데이터 개수입니다.

3. *index*
리소스 인덱스 번호입니다.

4. *dataLen*
리소스 데이터 크기입니다.

5. *data*
바이너리 리소스 데이터입니다.

## BS2IPv6DeviceInfo

```
enum {
    BS2_MAX_IPV6_ALLOCATED_ADDR = 8,
};

typedef struct
{
    BS2_DEVICE_ID id;
    uint8_t reserved[1];
    uint8_t bIPv6Mode;
    char ipv6Address[BS2_IPV6_ADDR_SIZE];
    uint16_t portV6;
    char connectedIPV6[BS2_IPV6_ADDR_SIZE];
    uint8_t numOfAllocatedAddressV6;
    char
allocatedIpAddressV6[BS2_IPV6_ADDR_SIZE][BS2_MAX_IPV6_ALLOCATED_ADDR];
}BS2IPv6DeviceInfo;
```

1. *id*
장치 식별자입니다.

2. *reserved*
예약된 공간입니다.

3. *bIPv6Mode*
현 장치가 IP V6로 동작하는지 여부를 나타내는 flag입니다.

4. *ipv6Address*
현재 장치의 IP V6 주소입니다.

5. *portV6*
현재 장치의 IP V6 포트 번호입니다.

6. *connectedIPV6*
장치가 연결된 서버의 IP V6 주소입니다.

7. *numOfAllocatedAddressV6*
현재 장치에 할당되어진 IP V6 주소의 갯수를 나타냅니다.

8. *allocatedIpAddressV6*
현재 장치에 할당되어진 IP V6 주소입니다. numOfAllocatedAddressV6는 할당되어진 주소의 갯수를 나타냅니다.

## BS2AuthOperatorLevel

```
typedef struct {
    char userID[BS2_USER_ID_SIZE];
    uint8_t level;
    uint8_t reserved[3];
} BS2Operator;

typedef BS2Operator BS2AuthOperatorLevel;
```

1. *userID*
사용자 식별자입니다.

2. *level*
사용자가 인증되었을 때 식별자에 해당하는 권한을 지정합니다.

| 값 | 설명 |
|---|---|
| 0 | 권한 없음 |
| 1 | 관리자 권한 |
| 2 | 시스템 구성을 변경할 수 있는 권한 |
| 3 | 사용자 정보를 변경할 수 있는 권한 |

3. *reserved*
예약된 공간입니다.

## BS2DeviceCapabilities

[+ 2.8]

```
typedef struct {
    uint32_t maxUsers;                  ///< 4 bytes
    uint32_t maxEventLogs;              ///< 4 bytes
    uint32_t maxImageLogs;              ///< 4 bytes
    uint32_t maxBlacklists;             ///< 4 bytes
    uint32_t maxOperators;              ///< 4 bytes
    uint32_t maxCards;                  ///< 4 bytes
    uint32_t maxFaces;                  ///< 4 bytes
    uint32_t maxFingerprints;           ///< 4 bytes
```

```
    uint32_t maxUserNames;            ///< 4 bytes
    uint32_t maxUserImages;           ///< 4 bytes
    uint32_t maxUserJobs;             ///< 4 bytes
    uint32_t maxUserPhrases;          ///< 4 bytes
    uint8_t maxOutputPorts;           ///< 1 byte
    uint8_t maxRelays;                ///< 1 byte
    uint8_t maxRS485Channels;         ///< 1 byte

    uint8_t cameraSupported: 1;
    uint8_t tamperSupported: 1;
    uint8_t wlanSupported: 1;
    uint8_t displaySupported: 1;
    uint8_t thermalSupported: 1;
    uint8_t maskSupported: 1;
    uint8_t faceExSupported: 1;
    uint8_t unused: 1;

    union {
        uint32_t mask;                ///< 4 bytes
        struct {
            uint32_t EM: 1;
            uint32_t HIDProx: 1;
            uint32_t MifareFelica: 1;
            uint32_t iClass: 1;
            uint32_t ClassicPlus: 1;
            uint32_t DesFireEV1: 1;
            uint32_t SRSE: 1;
            uint32_t SEOS: 1;
            uint32_t NFC: 1;
            uint32_t BLE: 1;
            uint32_t reserved: 21;
            uint32_t useCardOperation: 1;
        };
    } cardSupported;

    struct {
        BS2_BOOL extendedMode;        ///< 1 byte
        union {
            uint8_t mask;             ///< 1 byte
            struct {
                uint8_t card: 1;
                uint8_t fingerprint: 1;
                uint8_t face: 1;
                uint8_t id: 1;
                uint8_t pin: 1;
                uint8_t reserved: 3;
            };
        } credentials;
        uint8_t reserved[2];          ///< 2 bytes
        union {
            struct {
```

```
                union {
                    uint8_t mask;     ///< 1 byte
                    struct {
                        uint8_t biometricOnly: 1;
                        uint8_t biometricPIN: 1;
                        uint8_t unused: 6;
                    };
                } biometricAuth;

                union {
                    uint8_t mask;     ///< 1 byte
                    struct {
                        uint8_t cardOnly: 1;
                        uint8_t cardBiometric: 1;
                        uint8_t cardPIN: 1;
                        uint8_t cardBiometricOrPIN: 1;
                        uint8_t cardBiometricPIN: 1;
                        uint8_t unused: 3;
                    };
                } cardAuth;

                union {
                    uint8_t mask;     ///< 1 byte
                    struct {
                        uint8_t idBiometric: 1;
                        uint8_t idPIN: 1;
                        uint8_t idBiometricOrPIN: 1;
                        uint8_t idBiometricPIN: 1;
                        uint8_t unused: 4;
                    };
                } idAuth;
            } legacy;

            struct {
                union {
                    uint32_t mask;    ///< 4 bytes
                    struct {
                        uint32_t faceOnly: 1;
                        uint32_t faceFingerprint: 1;
                        uint32_t facePIN: 1;
                        uint32_t faceFingerprintOrPIN: 1;
                        uint32_t faceFingerprintPIN: 1;
                        uint32_t unused: 27;
                    };
                } faceAuth;

                union {
                    uint32_t mask;    ///< 4 bytes
                    struct {
                        uint32_t fingerprintOnly: 1;
                        uint32_t fingerprintFace: 1;
```

```
                    uint32_t fingerprintPIN: 1;
                    uint32_t fingerprintFaceOrPIN: 1;
                    uint32_t fingerprintFacePIN: 1;
                    uint32_t unused: 27;
                };
            } fingerprintAuth;

            union {
                uint32_t mask;    ///< 4 bytes
                struct {
                    uint32_t cardOnly: 1;
                    uint32_t cardFace: 1;
                    uint32_t cardFingerprint: 1;
                    uint32_t cardPIN: 1;
                    uint32_t cardFaceOrFingerprint: 1;
                    uint32_t cardFaceOrPIN: 1;
                    uint32_t cardFingerprintOrPIN: 1;
                    uint32_t cardFaceOrFingerprintOrPIN: 1;
                    uint32_t cardFaceFingerprint: 1;
                    uint32_t cardFacePIN: 1;
                    uint32_t cardFingerprintFace: 1;
                    uint32_t cardFingerprintPIN: 1;
                    uint32_t cardFaceOrFingerprintPIN: 1;
                    uint32_t cardFaceFingerprintOrPIN: 1;
                    uint32_t cardFingerprintFaceOrPIN: 1;
                    uint32_t unused: 17;
                };
            } cardAuth;

            union {
                uint32_t mask;    ///< 4 bytes
                struct {
                    uint32_t idFace: 1;
                    uint32_t idFingerprint: 1;
                    uint32_t idPIN: 1;
                    uint32_t idFaceOrFingerprint: 1;
                    uint32_t idFaceOrPIN: 1;
                    uint32_t idFingerprintOrPIN: 1;
                    uint32_t idFaceOrFingerprintOrPIN: 1;
                    uint32_t idFaceFingerprint: 1;
                    uint32_t idFacePIN: 1;
                    uint32_t idFingerprintFace: 1;
                    uint32_t idFingerprintPIN: 1;
                    uint32_t idFaceOrFingerprintPIN: 1;
                    uint32_t idFaceFingerprintOrPIN: 1;
                    uint32_t idFingerprintFaceOrPIN: 1;
                    uint32_t unused: 18;
                };
            } idAuth;
        } extended;
    };
```

```
    } authSupported;

    uint8_t intelligentPDSupported: 1;
    uint8_t unused2: 7;

    uint8_t reserved[431];
} BS2DeviceCapabilities;
```

1. *maxUsers*
장치에 저장 가능한 정보의 최대 갯수를 나타냅니다. (사용자)

2. *maxEventLogs*
장치에 저장 가능한 정보의 최대 갯수를 나타냅니다. (이벤트로그)

3. *maxImageLogs*
장치에 저장 가능한 정보의 최대 갯수를 나타냅니다. (이미지로그)

4. *maxBlacklists*
장치에 저장 가능한 정보의 최대 갯수를 나타냅니다. (블랙리스트)

5. *maxOperators*
장치에 저장 가능한 정보의 최대 갯수를 나타냅니다. (관리자)

6. *maxCards*
장치에 저장 가능한 정보의 최대 갯수를 나타냅니다. (카드)

7. *maxFaces*
장치에 저장 가능한 정보의 최대 갯수를 나타냅니다. (얼굴)

8. *maxFingerprints*
장치에 저장 가능한 정보의 최대 갯수를 나타냅니다. (지문)

9. *maxUserNames*
장치에 저장 가능한 정보의 최대 갯수를 나타냅니다. (사용자 명)

10. *maxUserImages*
장치에 저장 가능한 정보의 최대 갯수를 나타냅니다. (사용자 이미지)

11. *maxUserJobs*
장치에 저장 가능한 정보의 최대 갯수를 나타냅니다. (Job code)

12. *maxUserPhrases*
장치에 저장 가능한 정보의 최대 갯수를 나타냅니다. (사용자 구문)

13. *maxCardsPerUser*
장치에 저장 가능한 정보의 최대 갯수를 나타냅니다. (사용자별 카드)

14. *maxFacesPerUser*
장치에 저장 가능한 정보의 최대 갯수를 나타냅니다. (사용자별 얼굴)

15. *maxFingerprintsPerUser*
장치에 저장 가능한 정보의 최대 갯수를 나타냅니다. (사용자별 지문)

16. *maxInputPorts*
장치에 저장 가능한 정보의 최대 갯수를 나타냅니다. (장치의 입력포트)

17. *maxOutputPorts*
장치에 저장 가능한 정보의 최대 갯수를 나타냅니다. (장치의 출력포트)

18. *maxRelays*
장치에 저장 가능한 정보의 최대 갯수를 나타냅니다. (장치의 릴레이)

19. *maxRS485Channels*
장치에 저장 가능한 정보의 최대 갯수를 나타냅니다. (RS485 채널)

20. *시스템 지원 정보*
장치가 지원 가능한 시스템 정보를 bit 단위로 아래와 같이 나타냅니다.

| 비트위치 | 비트 수 | 멤버명 | 설명 |
|---|---|---|---|
| 0 | 1 | cameraSupported | 카메라 지원 여부. |
| 1 | 1 | tamperSupported | 탬퍼 지원 여부. |
| 2 | 1 | wlanSupported | 무선랜 지원 여부. |
| 3 | 1 | displaySupported | 화면의 지원 여부. |
| 4 | 1 | thermalSupported | 열화상 카메라 지원 여부. |
| 5 | 1 | maskSupported | 마스크 검출 지원 여부. |
| 6 | 1 | faceExSupported | Visual camera 인증 지원 여부. |
| 7 | 1 | unused | 미할당 필드. |

21. *cardSupported*
카드관련 지원 정보를 나타냅니다. mask값으로 전체를 또는 bit 단위로 각각의 지원항목에 접근할 수 있습니다.

| 비트위치 | 비트 수 | 멤버명 | 설명 |
|---|---|---|---|
| - | 전체 | mask | 전체정보 |
| 0 | 1 | EM | EM 카드 |
| 1 | 1 | HIDProx | HID Proximity 카드 |
| 2 | 1 | MifareFelica | MIFARE / FeliCa |
| 3 | 1 | iClass | iClass 카드 |
| 4 | 1 | ClassicPlus | Classic plus 카드 |
| 5 | 1 | DesFireEV1 | DESFire EV1 |
| 6 | 1 | SRSE | iClass SR, iClass SE |
| 7 | 1 | SEOS | iClass SEOS |
| 8 | 1 | NFC | NFC 카드 |
| 9 | 1 | BLE | BLE |
| 10 | 21 | reserved | 미할당 필드. |
| 31 | 1 | useCardOperation | 카드 사용 여부 |

22. *authSupported*
인증관련 지원 정보를 나타냅니다.

23. *extendedMode*
true인 경우, 확장인증모드를 지원하며, authSupported.extended를 참조합니다.
false인 경우 비확장인증 모드를 지원하며, authSupported.lagacy를 참조합니다.

24. *credentials*
지원되는 인증 수단을 나타냅니다. mask값으로 전체를 또는 bit 단위로 각각의 지원항목에 접근할 수 있

습니다.

| 비트위치 | 비트 수 | 멤버명 | 설명 |
|---|---|---|---|
| - | 전체 | mask | 전체정보 |
| 0 | 1 | card | 카드 |
| 1 | 1 | fingerprint | 지문 |
| 2 | 1 | face | 얼굴 |
| 3 | 1 | id | ID |
| 4 | 1 | pin | PIN |
| 5 | 3 | reserved | 미할당 필드. |

25. *reserved*
예약된 공간입니다.

26. *legacy*
비확장 인증모드 지원 시, 참조되는 정보입니다.

27. *biometricAuth*
(비확장 인증모드)Biometric 인증 조합을 나타냅니다.

| 비트위치 | 비트 수 | 멤버명 | 설명 |
|---|---|---|---|
| - | 전체 | mask | 전체정보 |
| 0 | 1 | biometricOnly | Biometric only |
| 1 | 1 | biometricPIN | Biometric + PIN |
| 2 | 6 | unused | 미할당 필드. |

28. *cardAuth*
(비확장 인증모드)Card 인증 조합을 나타냅니다.

| 비트위치 | 비트 수 | 멤버명 | 설명 |
|---|---|---|---|
| - | 전체 | mask | 전체정보 |
| 0 | 1 | cardOnly | Card only |
| 1 | 1 | cardBiometric | Card + Biometric |
| 2 | 1 | cardPIN | Card + PIN |
| 3 | 1 | cardBiometricOrPIN | Card + Biometric/PIN |
| 4 | 1 | cardBiometricPIN | Card + Biometric + PIN |
| 5 | 3 | unused | 미할당 필드. |

29. *idAuth*
(비확장 인증모드)ID 인증 조합을 나타냅니다.

| 비트위치 | 비트 수 | 멤버명 | 설명 |
|---|---|---|---|
| - | 전체 | mask | 전체정보 |
| 0 | 1 | idBiometric | ID + Biometric |
| 1 | 1 | idPIN | ID + PIN |
| 2 | 1 | idBiometricOrPIN | ID + Biometric/PIN |
| 3 | 1 | idBiometricPIN | ID + Biometric + PIN |
| 4 | 4 | unused | 미할당 필드. |

30. *extended*

확장 인증모드 지원 시, 참조되는 정보입니다.

31. *faceAuth*
(확장 인증모드)얼굴 인증 조합을 나타냅니다.

| 비트위치 | 비트 수 | 멤버명 | 설명 |
|---|---|---|---|
| - | 전체 | mask | 전체정보 |
| 0 | 1 | faceOnly | Face only |
| 1 | 1 | faceFingerprint | Face + Fingerprint |
| 2 | 1 | facePIN | Face + PIN |
| 3 | 1 | faceFingerprintOrPIN | Face + Fingerprint/PIN |
| 4 | 1 | faceFingerprintPIN | Face + Fingerprint + PIN |
| 5 | 27 | unused | 미할당 필드. |

32. *fingerprintAuth*
(확장 인증모드)지문 인증 조합을 나타냅니다.

| 비트위치 | 비트 수 | 멤버명 | 설명 |
|---|---|---|---|
| - | 전체 | mask | 전체정보 |
| 0 | 1 | fingerprintOnly | Fingerprint only |
| 1 | 1 | fingerprintFace | Fingerprint + Face |
| 2 | 1 | fingerprintPIN | Fingerprint + PIN |
| 3 | 1 | fingerprintFaceOrPIN | Fingerprint + Face/PIN |
| 4 | 1 | fingerprintFacePIN | Fingerprint + Face + PIN |
| 5 | 27 | unused | 미할당 필드. |

33. *cardAuth*
(확장 인증모드)카드 인증 조합을 나타냅니다.

| 비트위치 | 비트 수 | 멤버명 | 설명 |
|---|---|---|---|
| - | 전체 | mask | 전체정보 |
| 0 | 1 | cardOnly | Card only |
| 1 | 1 | cardFace | Card + Face |
| 2 | 1 | cardFingerprint | Card + Fingerprint |
| 3 | 1 | cardPIN | Card + PIN |
| 4 | 1 | cardFaceOrFingerprint | Card + Face/Fingerprint |
| 5 | 1 | cardFaceOrPIN | Card + Face/PIN |
| 6 | 1 | cardFingerprintOrPIN | Card + Fingerprint/PIN |
| 7 | 1 | cardFaceOrFingerprintOrPIN | Card + Face/Fingerprint/PIN |
| 8 | 1 | cardFaceFingerprint | Card + Face + Fingerprint |
| 9 | 1 | cardFacePIN | Card + Face + PIN |
| 10 | 1 | cardFingerprintFace | Card + Fingerprint + Face |
| 11 | 1 | cardFingerprintPIN | Card + Fingerprint + PIN |
| 12 | 1 | cardFaceOrFingerprintPIN | Card + Face/Fingerprint + PIN |
| 13 | 1 | cardFaceFingerprintOrPIN | Card + Face + Fingerprint/PIN |
| 14 | 1 | cardFingerprintFaceOrPIN | Card + Fingerprint + Face/PIN |
| 15 | 17 | unused | 미할당 필드. |

34. *idAuth*

(확장 인증모드)ID 인증 조합을 나타냅니다.

| 비트위치 | 비트 수 | 멤버명 | 설명 |
|---|---|---|---|
| - | 전체 | mask | 전체정보 |
| 1 | 1 | idFace | ID + Face |
| 2 | 1 | idFingerprint | ID + Fingerprint |
| 3 | 1 | idPIN | ID + PIN |
| 4 | 1 | idFaceOrFingerprint | ID + Face/Fingerprint |
| 5 | 1 | idFaceOrPIN | ID + Face/PIN |
| 6 | 1 | idFingerprintOrPIN | ID + Fingerprint/PIN |
| 7 | 1 | idFaceOrFingerprintOrPIN | ID + Face/Fingerprint/PIN |
| 8 | 1 | idFaceFingerprint | ID + Face + Fingerprint |
| 9 | 1 | idFacePIN | ID + Face + PIN |
| 10 | 1 | idFingerprintFace | ID + Fingerprint + Face |
| 11 | 1 | idFingerprintPIN | ID + Fingerprint + PIN |
| 12 | 1 | idFaceOrFingerprintPIN | ID + Face/Fingerprint + PIN |
| 13 | 1 | idFaceFingerprintOrPIN | ID + Face + Fingerprint/PIN |
| 14 | 1 | idFingerprintFaceOrPIN | ID + Fingerprint + Face/PIN |
| 15 | 18 | unused | 미할당 필드. |

35. *시스템 지원 정보*
장치가 지원 가능한 시스템 정보를 bit 단위로 아래와 같이 나타냅니다.

| 비트위치 | 비트 수 | 멤버명 | 설명 |
|---|---|---|---|
| 0 | 1 | intelligentPDSupported | Intelligent PD 지원 여부. |
| 1 | 7 | unused2 | 미할당 필드. |

36. *reserved*
예약된 공간입니다.

From:
https://kb.supremainc.com/kbtest/ - **BioStar 2 Device SDK**

Permanent link:
**https://kb.supremainc.com/kbtest/doku.php?id=ko:device_api&rev=1622011840**

Last update: **2021/05/26 15:50**