

Device API	1
.....	1
BS2SimpleDeviceInfo	1
BS2SimpleDeviceInfoEx	4
BS2ResourceElement	6
BS2IPv6DeviceInfo	6
BS2AuthOperatorLevel	7
BS2DeviceCapabilities	8

Device API

- [BS2_GetDeviceInfo](#): 가 .
- [BS2_GetDeviceInfoEx](#): [+ 2.6.0] 가 가 .
- [BS2_GetDeviceTime](#): 가 .
- [BS2_SetDeviceTime](#): .
- [BS2_ClearDatabase](#): Blacklist .
- [BS2_FactoryReset](#): .
- [BS2_RebootDevice](#): .
- [BS2_LockDevice](#): .
- [BS2_UnlockDevice](#): .
- [BS2_SetKeepAliveTimeout](#): keep-alive .
- [BS2_UpgradeFirmware](#): .
- [BS2_UpdateResource](#): .
- [BS2_GetSpecifiedDeviceInfo](#): [+ 2.6.3] 가 .
- [BS2_GetAuthOperatorLevelEx](#): [+ 2.6.3] 가 . (1000)
- [BS2_GetAllAuthOperatorLevelEx](#): [+ 2.6.3] 가 . (1000)
- [BS2_SetAuthOperatorLevelEx](#): [+ 2.6.3] 가 . (1000)
- [BS2_RemoveAuthOperatorLevelEx](#): [+ 2.6.3] . (1000)
- [BS2_RemoveAllAuthOperatorLevelEx](#): [+ 2.6.3] . (1000)
- [BS2_GetDeviceCapabilities](#): [+ 2.8] 가 .
- [BS2_RunAction](#): [+ 2.8.1] .

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*

1 .

2. *type*

.

0x00	Unknown Type
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
0x18	XPass D2 Keypad
0x19	FACELITE
0x1A	XPass2 Keypad
0x1B	XPass D2 Revision
0x1C	XPass D2 Keypad Revision
0x1D	FaceStation F2 Finger
0x1E	FaceStation F2
0x1F	XStation 2 QR
0x20	XStation 2
0x21	Input Module 120
0x22	XStation 2 Finger

3. connectionMode

BioStar

mode(0x1)

가 BioStar

direct mode

IP Config

BioStar

.

direct mode(0x0)

server

server mode

direct mode

4. ipv4Address

IP

.

5. port

TCP

.

6. maxNumOfUser

.

7. userNameSupported

flag

.

8. userPhotoSupported

flag

.

9. pinSupported

PIN

flag

.

10. cardSupported

flag

.

11. fingerSupported

flag

.

12. faceSupported

flag

.

13. wlanSupported

flag

.

14. tnaSupported

flag

.

15. <i>triggerActionSupported</i>	trigger action	flag	.
16. <i>wiegandSupported</i>	Wiegand	flag	.
17. <i>imageLogSupported</i>		flag	.
18. <i>dnsSupported</i>	DNS	flag	.
19. <i>jobCodeSupported</i>	Job Code	flag	.
20. <i>wiegandMultiSupported</i>	wiegandMulti	flag	.
21. <i>rs485Mode</i>	RS485		.
22. <i>sslSupported</i>	ssl	flag	.
23. <i>rootCertExist</i>	root 가	flag	.
24. <i>dualIDSsupported</i>	dualID(,)	flag	.
25. <i>useAlphanumericID</i>	AlphanumericID	flag	.
26. <i>connectedIP</i>	가 ip . (0xFFFFFFFF if disconnected)		
27. <i>phraseCodeSupported</i>		flag	.
28. <i>card1xSupported</i>	1.x ToC	flag	.
29. <i>systemExtSupported</i>	RS-485	flag	.
30. <i>voipSupported</i>	VoIP	flag	.

BS2SimpleDeviceInfoEx

BS2SimpleDeviceInfo 가 .

```
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_QR               = 0x00000020,

        BS2_SUPPORT_FINGER_SCAN      = 0x00010000,
        BS2_SUPPORT_FACE_SCAN        = 0x00020000,
        BS2_SUPPORT_FACE_EX_SCAN     = 0x00040000,
        BS2_SUPPORT_QR_SCAN          = 0x00080000,

        BS2_SUPPORT_ALL              = BS2_SUPPORT_RS485EX |
            BS2_SUPPORT_CARDEX |
            BS2_SUPPORT_DST |
            BS2_SUPPORT_DESFIREEX |
            BS2_SUPPORT_FACE_EX |
            BS2_SUPPORT_QR |
            BS2_SUPPORT_FINGER_SCAN |
            BS2_SUPPORT_FACE_SCAN |
            BS2_SUPPORT_FACE_EX_SCAN |
            BS2_SUPPORT_QR_SCAN,
    };

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

1. *supported*
가 BS2SimpleDeviceInfo 가 가 .
bit masking .

BS2_SUPPORT_RS485EX	0x00000001	RS485 (CoreStation 40)
BS2_SUPPORT_CARDEX	0x00000002	iClass SEOS
BS2_SUPPORT_DST	0x00000004	
BS2_SUPPORT_DESFIREEX	0x00000008	DesFire [+ V2.6.4]
BS2_SUPPORT_FACE_EX	0x00000010	FaceStation F2 [+ V2.7.1]
BS2_SUPPORT_QR	0x00000020	QR XStation 2 QR [+ V2.8.0]
BS2_SUPPORT_FINGER_SCAN	0x00010000	가 [+ V2.7.1]
BS2_SUPPORT_FACE_SCAN	0x00020000	가 FaceStation2, FaceLite [+ V2.7.1]
BS2_SUPPORT_FACE_EX_SCAN	0x00040000	가 FaceStation F2 [+ V2.7.1]

BS2_SUPPORT_QR_SCAN	0x00080000	QR 가 XStation 2 [+ V2.8.0]
BS2_SUPPORT_ALL	0x000FFFFFF	가

2. *reserved*

BS2ResourceElement

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

1. *type*

0	UI(Language pack)	
1	Notice message	UTF-8
2	Image(Background)	PNG
3	Slide image	PNG
4	Sound	WAVE

2. *numResData*

3. *index*

4. *dataLen*

5. data

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*

.

2. *reserved*

.

3. *bIPv6Mode*

가 IP V6

flag

.

4. *ipv6Address*

IP V6

.

5. *portV6*

IP V6

.

6. *connectedIPv6*

가

IP V6

.

7. *numOfAllocatedAddressV6*

IP V6

.

8. *allocatedIpAddressV6*

IP V6

. numOfAllocatedAddressV6

.

BS2AuthOperatorLevel

```
typedef struct {
    char userID[BS2_USER_ID_SIZE];
    uint8_t level;
    uint8_t reserved[3];
} BS2operator;

typedef BS2operator BS2AuthOperatorLevel;
```

1. *userID*

2. level
가

0	
1	
2	
3	

3. reserved

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_B00L 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*
가 . ()
2. *maxEventLogs*
가 . ()
3. *maxImageLogs*
가 . ()
4. *maxBlacklists*
가 . ()
5. *maxOperators*
가 . ()
6. *maxCards*

- 가. ()
7. *maxFaces*
가. ()
8. *maxFingerprints*
가. ()
9. *maxUserNames*
가. ()
10. *maxUserImages*
가. ()
11. *maxUserJobs*
가. (Job code)
12. *maxUserPhrases*
가. ()
13. *maxCardsPerUser*
가. ()
14. *maxFacesPerUser*
가. ()
15. *maxFingerprintsPerUser*
가. ()
16. *maxInputPorts*
가. ()
17. *maxOutputPorts*
가. ()
18. *maxRelays*
가. ()
19. *maxRS485Channels*
가. (RS485)
20.
가 가 bit .

0	1	cameraSupported	.
1	1	tamperSupported	.
2	1	wlanSupported	.
3	1	displaySupported	.
4	1	thermalSupported	.
5	1	maskSupported	.
6	1	faceExSupported	Visual camera .
7	1	unused	.

21. cardSupported

mask bit

-		mask	
0	1	EM	EM
1	1	HIDProx	HID Proximity
2	1	MifareFelica	MIFARE / FeliCa
3	1	iClass	iClass
4	1	ClassicPlus	Classic plus
5	1	DesFireEV1	DESFire EV1
6	1	SRSE	iClass SR, iClass SE
7	1	SEOS	iClass SEOS
8	1	NFC	NFC
9	1	BLE	BLE
10	21	reserved	
31	1	useCardOperation	

22. authSupported

23. extendedMode

true , authSupported.extended
false , authSupported.lagacy

24. credentials

mask bit

-		mask	
0	1	card	
1	1	fingerprint	
2	1	face	
3	1	id	ID
4	1	pin	PIN
5	3	reserved	

25. reserved

26. legacy

27. biometricAuth

()Biometric

-		mask	

0	1	biometricOnly	Biometric only
1	1	biometricPIN	Biometric + PIN
2	6	unused	.

28. *cardAuth*

()Card .

-		mask	
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	.

29. *idAuth*

()ID .

-		mask	
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	.

30. *extended*

, .

31. *faceAuth*

() .

-		mask	
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	.

32. *fingerprintAuth*

() .

-		mask	
0	1	fingerprintOnly	Fingerprint only

