Table of Contents

Device API

API that controls the device information or upgrades the firmware.

- BS2_GetDeviceInfo: Gets the device information.
- BS2_GetDeviceInfoEx: [+ 2.6.0] Gets additional device information.
- BS2_GetDeviceTime: Gets the device time.
- BS2_SetDeviceTime: Sets the device time.
- BS2_ClearDatabase: Initializes the user information and blacklist.
- BS2_FactoryReset: Initializes all configurations and the database.
- BS2_RebootDevice: Restarts the device.
- BS2_LockDevice: Doesn't allow user authentication by locking the device.
- BS2_UnlockDevice: Allows user authentication by unlocking the device.
- BS2_SetKeepAliveTimeout: Configures the keep-alive time of the device.
- BS2_UpgradeFirmware: Upgrades the firmware.
- BS2_UpdateResource: Updates the resource.
- BS2_GetSpecifiedDeviceInfo: [+ 2.6.3] Gets specified device information.
- BS2_GetAuthOperatorLevelEx: [+ 2.6.3] Gets specified device operator. (Support operator up to 1000)
- BS2_GetAllAuthOperatorLevelEx: [+ 2.6.3] Gets all device operators. (Support operator up to 1000)
- BS2_SetAuthOperatorLevelEx: [+ 2.6.3] Sets device operator. (Support operator up to 1000)
- BS2_RemoveAuthOperatorLevelEx: [+ 2.6.3] Removes specified device operator. (Support operator up to 1000)
- BS2_RemoveAllAuthOperatorLevelEx: [+ 2.6.3] Removes all device operators. (Support operator up to 1000)
- BS2_GetDeviceCapabilities: [+ 2.8] Gets available function information of the device.
- BS2_RunAction: [+ 2.8.1] Commands the device to take certain actions.
- BS2_GetMasterAdmin: [+ 2.9.12] Gets the master admin from the device.
- BS2_SetMasterAdmin: [+ 2.9.12] Sets the master admin on the device.

Structure

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*
The device identifier which is always above 1.

2. *type*
Code value of device type.

Value	Description
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

Value	Description
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
0x23	BioStation 3
0x24	3rd party OSDP device
0x25	3rd party OSDP IO device
0x26	BioStation 2a

3. *connectionMode*
It indicates the connection mode between the BioStar application and device which is separated by the subject of the connection as direct mode(0x0) and server mode(0x1). The BioStar application connects to the device in direct mode, and the device connects to the BioStar application in server mode. The default settings for the devices are direct mode, and to change the connection mode refer to IP Config.

4. *ipv4Address*
IP address of the selected device.

5. *port*
TCP port number of the selected device.

6. *maxNumOfUser*
Maximum capacity of users that can be stored in the device.

7. *userNameSupported*
Flag that notifies whether the device supports user name.

8. *userPhotoSupported*
Flag that notifies whether the device supports user profile picture.

9. *pinSupported*
Flag that notifies whether the device supports PIN.

10. *cardSupported*
Flag that notifies whether the device supports Smart card authentication.

11. *fingerSupported*

Flag that notifies whether the device supports finger authentication.

12. *faceSupported*
Flag that notifies whether the device supports face recognition.

13. *wlanSupported*
Flag that notifies whether the device supports wireless LAN.

14. *tnaSupported*
Flag that notifies whether the device supports time and attendance.

15. *triggerActionSupported*
Flag that notifies whether the device supports trigger action.

16. *wiegandSupported*
Flag that notifies whether the device supports wiegand.

17. *imageLogSupported*
Flag that notifies whether the device supports image logs.

18. *dnsSupported*
Flag that notifies whether the device supports DNS.

19. *jobCodeSupported*
Flag that notifies whether the device supports job codes.

20. *wiegandMultiSupported*
Flag that notifies whether the device supports Multi-Wiegand.

21. *rs485Mode*
RS-485 mode of the device.

22. *sslSupported*
Flag that notifies whether the device supports SSL communication.

23. *rootCertExist*
Flag that notifies whether the device has a root certificate.

24. *dualIDSupported*
Flag that notifies whether the device supports alphanumeric ID.

25. *useAlphanumericID*
Flag that notifies whether the device is currently using Alphanumeric ID.

26. *connectedIP*
IP address where the device is connected to. (0xFFFFFFFF if disconnected)

27. *phraseCodeSupported*
Flag that notifies whether the device supports personal messages.

28. *card1xSupported*
Flag that notifies whether the device supports reading 1.x ToC cards.

29. *systemExtSupported*

Flag that notifies whether the device supports configuring RS-485 keys.

30. *voipSupported*
Flag that notifies whether the device supports VoIP.

BS2SimpleDeviceInfoEx

Retrieves BS2SimpleDeviceInfo and supported information.

```
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*
The current device additionally obtains information beyond the functionality provided by BS2SimpleDeviceInfo.
By bit masking with the values defined below, you can check if it is supported.

Definition	Value	Description
BS2_SUPPORT_RS485EX	0x00000001	Whether RS485 extensions are supported (In case of CoreStation 40)
BS2_SUPPORT_CARDEX	0x00000002	Whether iClass SEOS card is used

Definition	Value	Description
BS2_SUPPORT_DST	0x00000004	Whether daylight savings time is used
BS2_SUPPORT_DESFIREEX	0x00000008	Whether DesFire advanced setting is supported [+2.6.4]
BS2_SUPPORT_FACE_EX	0x00000010	Whether support face matching for FSF2 [+ V2.7.1]
BS2_SUPPORT_QR	0x00000020	Whether support QR matching XStation 2 QR [+ V2.8.0]
BS2_SUPPORT_FINGER_SCAN	0x00010000	Whether support fingerprint scan [+ V2.7.1]
BS2_SUPPORT_FACE_SCAN	0x00020000	Whether support face scan for FS2 and FL [+ V2.7.1]
BS2_SUPPORT_FACE_EX_SCAN	0x00040000	Whether support face scan for FSF2 [+ V2.7.1]
BS2_SUPPORT_QR_SCAN	0x00080000	Whether support QR scan XStation 2 [+ V2.8.0]
BS2_SUPPORT_ALL	0x000FFFFF	Whether to provide additional full information

2. *reserved*
Reserved space.

BS2ResourceElement

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

1. *type*
Resource data type.

Value	Description	Supported data format
0	UI(Language pack)	Suprema language pack
1	Notice message	UTF-8 string
2	Image(Background)	PNG
3	Slide image	PNG
4	Sound	WAVE

2. *numResData*
Number of resource data.

3. *index*
Resource index number.

4. *dataLen*
Resource data length.

5. *data*
Binary resource 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*
Device ID

2. *reserved*
Reserved space

3. *bIPv6Mode*
Flag to determine whether to work IPv6 mode or not.

4. *ipv6Address*
IPv6 address of device

5. *portV6*
IPv6 port of device

6. *connectedIPV6*
IPv6 address of server which device is connected.

7. *numOfAllocatedAddressV6*
Number of IPv6 addresses currently allocated to device. 8. *allocatedIpAddressV6*

IPv6 addresses currently allocated to device.

BS2AuthOperatorLevel

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

typedef BS2Operator BS2AuthOperatorLevel;
```

1. *userID*
User ID

2. *level*
Sets operator level when user authenticates.

Value	Description
0	No auth
1	Administrator level
2	System configuration level
3	User information lelve

3. *reserved*
Reserved space

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 CustomClassicPlus: 1;
            uint32_t CustomDesFireEV1: 1;
            uint32_t TOM_NFC: 1;
            uint32_t TOM_BLE: 1;
            uint32_t CustomFelica: 1;
            uint32_t reserved: 16;
            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 updateUserSupported: 1;
    uint8_t simulatedUnlockSupported: 1;
    uint8_t smartCardByteOrderSupported: 1;
    uint8_t treatAsCSNSupported: 1;
    uint8_t rtspSupported: 1;
    uint8_t lfdSupported: 1;
    uint8_t visualQRSupported: 1;

    uint8_t maxVoipExtensionNumbers;      ///< 1 byte

    uint8_t osdpStandardCentralSupported : 1;       ///< 1 byte
    uint8_t enableLicenseFuncSupported : 1;   ///< 1 byte
    uint8_t keypadBacklightSupported: 1;
    uint8_t uzWirelessLockDoorSupported: 1;
    uint8_t customSmartCardSupported: 1;
    uint8_t tomSupported: 1;
    uint8_t tomEnrollSupported: 1;
    uint8_t showOsdpResultbyLED: 1;

    uint8_t customSmartCardFelicaSupported: 1;
    uint8_t ignoreInputAfterWiegandOut: 1;
    uint8_t setSlaveBaudrateSupported: 1;
    uint8_t rtspResolutionChangeSupported: 1;
    uint8_t voipResolutionChangeSupported: 1;
    uint8_t voipTransportChangeSupported: 1;
    uint8_t authMsgUserInfoSupported: 1;
    uint8_t scrambleKeyboardModeSupported: 1;


    uint16_t visualFaceTemplateVersion;

    //------------------------------------------------------------------------
---------- 2 byte

    uint8_t authDenyMaskSupported: 1;
    uint8_t mifareExSupported: 1;
    uint8_t lockOverrideSupported: 1;
    uint8_t doorModeOverrideSupported: 1;
    uint8_t alternateAccessTimerSupported: 1;
    uint8_t realtimeIOStatusReportSupported:1;
    uint8_t dynamicSlaveDeviceNumSupported: 1;
    uint8_t secureTamperSupported: 1;

    //------------------------------------------------------------------------
---------- 1 byte
    uint8_t customSmartcardSlaveSupported: 1;
    uint8_t serverPrivateMsgSupported: 1;
    uint8_t facilityCodeSupported: 1;
    uint8_t masterAdminSupported: 1;
    uint8_t adminTwoStepAuthSupported: 1;
```

```
    uint8_t qrDetectGuideLedSupported: 1;
    uint8_t unused: 2;

    uint8_t reserved[424];      ///< 424 bytes

} BS2DeviceCapabilities;
```

1. *maxUsers*
Indicates the maximum number of information that can be stored on the device. (User)

2. *maxEventLogs*
Indicates the maximum number of information that can be stored on the device. (Event log)

3. *maxImageLogs*
Indicates the maximum number of information that can be stored on the device. (Image log)

4. *maxBlacklists*
Indicates the maximum number of information that can be stored on the device. (Blacklist)

5. *maxOperators*
Indicates the maximum number of information that can be stored on the device. (Operator)

6. *maxCards*
Indicates the maximum number of information that can be stored on the device. (Card)

7. *maxFaces*
Indicates the maximum number of information that can be stored on the device. (Face)

8. *maxFingerprints*
Indicates the maximum number of information that can be stored on the device. (Fingerprint)

9. *maxUserNames*
Indicates the maximum number of information that can be stored on the device. (Username)

10. *maxUserImages*
Indicates the maximum number of information that can be stored on the device. (user image)

11. *maxUserJobs*
Indicates the maximum number of information that can be stored on the device. (Job code)

12. *maxUserPhrases*
Indicates the maximum number of information that can be stored on the device. (User phrase)

13. *maxCardsPerUser*
Indicates the maximum number of information that can be stored on the device. (Card per user)

14. *maxFacesPerUser*
Indicates the maximum number of information that can be stored on the device. (Face per user)

15. *maxFingerprintsPerUser*
Indicates the maximum number of information that can be stored on the device. (Fingerprint per user)

16. *maxInputPorts*
Indicates the maximum number of information that can be stored on the device. (input port of device)

17. *maxOutputPorts*
Indicates the maximum number of information that can be stored on the device. (output port of device)

18. *maxRelays*
Indicates the maximum number of information that can be stored on the device. (relay on device)

19. *maxRS485Channels*
Indicates the maximum number of information that can be stored on the device. (RS485 channel)

20. *System support information*
It indicates the system information supported by the device in bit units as follows.

Bit position	Number of bit	Member	Description
0	1	cameraSupported	Camera Support or not
1	1	tamperSupported	Tamper Support or not
2	1	wlanSupported	WLAN Support or not
3	1	displaySupported	Available LCD or not
4	1	thermalSupported	TCM 10 (Thermal Detection) Support or not
5	1	maskSupported	Mask Detection Support or not
6	1	faceExSupported	Visual Face device such as FaceStation F2 or not
7	1	unused	Unassigned

21. *cardSupported*
This indicates the card support relevant. Referring to mask value, you can access each item in its entirety or in bit units.

Bit position	Number of bit	Member	Description
-	Total	mask	total Information
0	1	EM	EM Card
1	1	HIDProx	HID Proximity Card
2	1	MifareFelica	MIFARE / FeliCa
3	1	iClass	iClass Card
4	1	ClassicPlus	Classic plus Card
5	1	DesFireEV1	DESFire EV1
6	1	SRSE	iClass SR, iClass SE
7	1	SEOS	iClass SEOS
8	1	NFC	NFC Card
9	1	BLE	BLE
10	1	ClassicPlus(Custom card)	ClassicPlus(Custom Card)
11	1	DesFireEV1(Custom card)	DesFireEV1(Custom Card)
12	1	TOM NFC	TOM NFC
13	1	TOM BLE	TOM BLE
14	1	FeliCa(Custom card)	FeliCa (Custom Card)
15	16	reserved	Unassigned

Bit position	Number of bit	Member	Description
31	1	useCardOperation	Card operation enabled or not

22. *authSupported*
This indicates support information related to authentication.

23. *extendedMode*
If true, extended authentication mode is supported, refer to authSupported.extended.
If false, non-extended authentication mode is supported, refer to authSupported.lagacy.

24. *credentials*
This indicates supported authentication methods. You can access each item in its entirety or in bit units as a mask value.

Bit position	Number of bit	Member	Description
-	Total	mask	Total Information
0	1	card	Card
1	1	fingerprint	Fingerprint
2	1	face	Face
3	1	id	ID
4	1	pin	PIN
5	3	reserved	Unassgined

25. *reserved*
Reserved space.

26. *legacy*
Information referenced when non-extended authentication mode is supported.

27. *biometricAuth*
(Non-extended authentication mode) Indicates the biometric authentication combination.

Bit position	Number of bit	Member	Description
-	Total	mask	Total Information
0	1	biometricOnly	Biometric only
1	1	biometricPIN	Biometric + PIN
2	6	unused	Unassigned

28. *cardAuth*
(Non-extended authentication mode) Indicates the card authentication combination.

Bit position	Number of bit	Member	Description
-	Total	mask	Total Information
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	Unassigned

29. *idAuth*
(Non-extended authentication mode) Indicates the ID authentication combination.

Bit position	Number of bit	Member	Description
-	Total	mask	Total Information
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	Unassigned

30. *extended*
This information is referenced when supporting extended authentication mode.

31. *faceAuth*
(Extended authentication mode) Indicates the face authentication combination.

Bit position	Number of bit	Member	Description
-	Total	mask	Total Information
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	Unassigned

32. *fingerprintAuth*
(Extended authentication mode) Indicates a combination of fingerprint authentication.

Bit position	Number of bit	Member	Description
-	Total	mask	Total Information
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	Unassigned

33. *cardAuth*
(Extended authentication mode) Indicates the card authentication combination.

Bit position	Number of bit	Member	Description
-	Total	mask	Total Information
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

Bit position	Number of bit	Member	Description
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	Unassigned

34. *idAuth*
(Extended authentication mode) Indicates the ID authentication combination.

Bit position	Number of bit	Member	Description
-	Total	mask	Total Information
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	Unassigned

35. *System support information*
It indicates the system information supported by the device in bit units as follows.

Bit position	Number of bit	Member	Description
0	1	intelligentPDSupported	Whether Intelligent PD is supported. (BS2Rs485Config)
1	1	updateUserSupported	Whether User information update is supported.
2	1	simulatedUnlockSupported	Whether simulated button unlock is supported.
3	1	smartCardByteOrderSupported	Whether smartCardByteOrder is supported. (BS2CardConfig)
4	1	treatAsCSNSupported	Whether treatAsCSN is supported. (BS2BarcodeConfig)

Bit position	Number of bit	Member	Description
5	1	rtspSupported	Whether RTSP is supported. (BS2RtspConfig)
6	1	lfdSupported	Whether LFD is supported.
7	1	visualQRSupported	Whether Visual QR is supported.

36. *maxVoipExtensionNumbers*
It is the number of registered internal numbers in the extension phone book at the extended VoIP setting information.

37. *System support information 2*
It indicates the system information supported by the device in bit units as follows.

Bit position	Number of bit	Member	Description
0	1	osdpStandardCentralSupported	Whether OSDP Standard is supported. (BS2OsdpStandardConfig)
1	1	enableLicenseFuncSupported	Whether Device license is supported.(BS2LicenseConfig)
2	1	keypadBacklightSupported	Whether Keypad backlight is supported.
3	1	uzWirelessLockDoorSupported	Whether U&Z wireless lock is supported.
4	1	customSmartCardSupported	Whether Custom card is supported. (BS2CustomCardConfig)
5	1	tomSupported	Whether ToM is supported.
6	1	tomEnrollSupported	Whether ToM enrollment is supported.
7	1	showOsdpResultbyLED	Whether OSDP standard device supports the ability to display authentication results. (BS2DisplayConfig)

38. *System support information 3*
It indicates the system information supported by the device in bit units as follows.

Bit position	Number of bit	Member	Description
0	1	customSmartCardFelicaSupported	Whether FeliCa custom smart card is supported.
1	1	ignoreInputAfterWiegandOut	Whether a feature to filter out repetitive re-entry of Wiegand output results is included.
2	1	setSlaveBaudrateSupported	Whether to support baudrate setting of RS485 slave device.
3	1	rtspResolutionChangeSupported	[+2.9.8] Whether to support changing RTSP video resolution.
4	1	voipResolutionChangeSupported	[+2.9.8] Whether to support changing intercom video resolution.
5	1	voipTransportChangeSupported	[+2.9.8] Whether to support changing intercom data transmission method.
6	1	authMsgUserInfoSupported	[+2.9.8] Whether to support changing the method of displaying user information on the authentication screen.
7	1	scrambleKeyboardModeSupported	[+2.9.8] Whether to support changing the use of scramble keypad.

39. *visualFaceTemplateVersion*
[+2.9.8] For visual face devices, indicates template version information supported by the device.

40. *System Support Information 4*
[+2.9.12] Indicates the system information supported by the device in bit units as follows.

Bit Position	Number of Bits	Member Name	Description
0	1	authDenyMaskSupported	Whether to support mask wearing prohibition in facial authentication
1	1	mifareExSupported	Whether Mifare CardEx is supported
2	1	lockOverrideSupported	Whether lock override is supported
3	1	doorModeOverrideSupported	Whether door mode override is supported
4	1	alternateAccessTimerSupported	Whether user override is supported
5	1	realtimeIOStatusReportSupported	Whether real-time device I/O port status information is provided
6	1	dynamicSlaveDeviceNumSupported	Whether dynamic slave device connection (up to 128 devices) is supported
7	1	secureTamperSupported	Whether secure tamper is supported

41. *System Support Information 5*
[+2.9.12] Indicates the system information supported by the device in bit units as follows.

Bit Position	Number of Bits	Member Name	Description
0	1	customSmartcardSlaveSupported	Whether custom smart card slave is supported
1	1	serverPrivateMsgSupported	Whether server private message is supported
2	1	facilityCodeSupported	Whether facility code is supported
3	1	masterAdminSupported	Whether master admin is supported
4	1	adminTwoStepAuthSupported	Whether two-step authentication for master admin is supported
5	1	qrDetectGuideLedSupported	Whether QR detection LED can be displayed (XPQ2)
6	2	unused	Unused

42. *reserved*
Reserved space.