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.

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
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
0x0F	Output Module
0x10	Input Module
0x11	BioEntry P2
0x0F	OM-120

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_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*
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
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_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_ALL	0x0000000F	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(Langauge 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