

Device API 1

..... 1

BS2SimpleDeviceInfo 1

BS2SimpleDeviceInfoEx 4

BS2ResourceElement 5

BS2IPv6DeviceInfo 6

BS2AuthOperatorLevel 6

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

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*

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 mode(0x1) . direct mode BioStar
 가 BioStar
 IP Config

direct mode(0x0) server
 server mode
 direct mode

4. *ipv4Address*

IP

5. <i>port</i>	TCP	.
6. <i>maxNumOfUser</i>		.
7. <i>userNameSupported</i>		flag .
8. <i>userPhotoSupported</i>		flag .
9. <i>pinSupported</i>	PIN	flag .
10. <i>cardSupported</i>		flag .
11. <i>fingerSupported</i>		flag .
12. <i>faceSupported</i>		flag .
13. <i>wlanSupported</i>		flag .
14. <i>tnaSupported</i>		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. *rootCertExist*

root 가 flag .

24. *dualIDSsupported*

dualID(,) flag .

25. *useAlphanumericID*

AlphanumericID flag .

26. *connectedIP*

가 ip . (0xFFFFFFFF if disconnected)

27. *phraseCodeSupported*

flag .

28. *card1xSupported*

1.x ToC flag .

29. *systemExtSupported*

RS-485 flag .

30. *voipSupported*

VoIP flag .

BS2SimpleDeviceInfoEx

BS2SimpleDeviceInfo 가 .

```
typedef struct
{
    enum
    {
        BS2_SUPPORT_RS485EX = 0x00000001,
        BS2_SUPPORT_CARDEX = 0x00000002,
        BS2_SUPPORT_DST = 0x00000004,
        BS2_SUPPORT_ALL = BS2_SUPPORT_RS485EX | BS2_SUPPORT_CARDEX |
        BS2_SUPPORT_DST,
    };

    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_ALL	0x00000007	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(Langauge 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];  
} BS20operator;  
  
typedef BS20operator BS2Auth0operatorLevel;
```

1. *userID*

2. *level*

가

0	
1	
2	
3	

3. *reserved*

From:

<https://kb.supremainc.com/bs2sdk/> - **BioStar 2 Device SDK**

Permanent link:

https://kb.supremainc.com/bs2sdk/doku.php?id=ko:device_api&rev=1557821911

Last update: **2019/05/14 17:18**