

Slave Control API 1

..... 1

BS2Rs485SlaveDevice 1

BS2Rs485SlaveDeviceEX 2

BS2OsdpStandardDevice 2

BS2OsdpStandardNotify 4

BS2OsdpStandardDeviceAdd 5

BS2OsdpStandardDeviceUpdate 6

BS2OsdpStandardDeviceCapability 6

BS2OsdpStandardDeviceResult 8

BS2OsdpStandardDeviceSecurityKey 9

BS2OsdpStandardDeviceAvailable 9

Slave Control API

RS485

- [BS2_GetSlaveDevice](#): RS485
- [BS2_SetSlaveDevice](#): 가/ /
- [BS2_GetSlaveExDevice](#): CoreStation RS485
- [BS2_SetSlaveExDevice](#): CoreStation 가/ /
- [BS2_SearchDevicesCoreStation](#): CoreStation
- [BS2_SearchDevicesCoreStationEx](#): [+ 2.6.3] CoreStation
(host ip)
- [BS2_GetDevicesCoreStation](#): CoreStation
- [BS2_AddOsdpStandardDevice](#): [+ 2.8.4] OSDP 가
- [BS2_GetOsdpStandardDevice](#): [+ 2.8.4] OSDP 가
- [BS2_GetAvailableOsdpStandardDevice](#): [+ 2.8.4] OSDP 가
- [BS2_UpdateOsdpStandardDevice](#): [+ 2.8.4] OSDP
- [BS2_RemoveOsdpStandardDevice](#): [+ 2.8.4] OSDP
- [BS2_GetOsdpStandardDeviceCapability](#): [+ 2.8.4] OSDP 가
- [BS2_SetOsdpStandardDeviceSecurityKey](#): [+ 2.8.4] OSDP

SDK가 가

BS2Rs485SlaveDevice

```
typedef struct {
    uint32_t deviceID;
    uint16_t deviceType;
    uint8_t enableOSDP;
    uint8_t connected;
} BS2Rs485SlaveDevice;
```

1. *deviceID*
2. *deviceType*
3. *enableOSDP*

flag .

4. *connected*

가 flag .

BS2Rs485SlaveDeviceEX

```

typedef struct {
    uint32_t deviceID;
    uint16_t deviceType;
    uint8_t enableOSDP;
    uint8_t connected;
    uint8_t channelInfo;
    uint8_t reserved[3];
} BS2Rs485SlaveDeviceEX;

```

1. *deviceID*

2. *deviceType*

3. *enableOSDP*

flag .

4. *connected*

가 flag .

5. *channelInfo*

Channel .

6. *reserved*

BS2OsdpStandardDevice

```

typedef struct {
    BS2_DEVICE_ID    deviceID;           ///< 4 bytes
    BS2_DEVICE_TYPE  deviceType;        ///< 2 bytes
    BS2_B00L         enableOSDP;        ///< 1 byte
    BS2_B00L         connected;         ///< 1 byte

    uint8_t          channelInfo;       ///< 1 byte
    uint8_t          osdpID;            ///< 1 byte
    BS2_B00L         supremaSearch;     ///< 1 byte
    BS2_B00L         activate;          ///< 1 byte

    BS2_B00L         useSecure;         ///< 1 byte

```

```

uint8_t      vendorCode[3];    ///< 3 bytes
BS2_VERSION  fwVersion;       ///< 4 bytes

uint8_t      modelName;       ///< 1 byte
uint8_t      modelVersion;    ///< 1 byte
BS2_B00L     readInfo;        ///< 1 byte
uint8_t      reserved[25];    ///< 25 byte (packing)
} BS20sdpStandardDevice;      ///< 48 bytes

```

1. *deviceID*

OSDP

2. *deviceType*

BS2_DEVICE_TYPE_3RD_OSDP_DEVICE

3. *enableOSDP*

true

4. *connected*

true OSDP 가

5. *channelInfo*

. CoreStation40 0~4 5 가 ,

6. *osdpID*

OSDP

7. *supremaSearch*

OSDP RS485 , false

8. *activate*9. *useSecure*[BS2_SetOsdpStandardDeviceSecurityKey](#)10. *vendorCode*

Vendor

11. *fwVersion*

OSDP FW

12. *modelName*

OSDP

13. *modelVersion*

OSDP

14. *readInfo*

vendorCode fwVersion, model , OSDP
가 master

15. *reserved***BS2OsdpStandardNotify**

```
typedef struct {
    BS2_DEVICE_ID    deviceID;           ///< 4 bytes
    BS2_DEVICE_TYPE  deviceType;        ///< 2 bytes
    BS2_B00L         enableOSDP;        ///< 1 byte
    BS2_B00L         connected;         ///< 1 byte

    uint8_t          channelInfo;       ///< 1 byte
    uint8_t          osdpID;           ///< 1 byte
    BS2_B00L         supremaSearch;    ///< 1 byte
    BS2_B00L         activate;         ///< 1 byte

    BS2_B00L         useSecure;        ///< 1 byte
    uint8_t          vendorCode[3];    ///< 3 bytes

    BS2_VERSION     fwVersion;         ///< 4 bytes

    uint8_t          modelNumber;      ///< 1 byte
    uint8_t          modelVersion;     ///< 1 byte
    BS2_B00L         readInfo;        ///< 1 byte
    uint8_t          reserved[5];     ///< 5 bytes (packing)
} BS2OsdpStandardNotify;           ///< 48 bytes
```

1. *deviceID*

OSDP

2. *deviceType*

BS2_DEVICE_TYPE_3RD_OSDP_DEVICE

3. *enableOSDP*

true

4. *connected*

true OSDP 가

5. *channelInfo*

. CoreStation40 0~4 5 가 ,

6. *osdpID*

OSDP

7. *supremaSearch*

OSDP RS485 , false .

8. *activate*

9. *useSecure*

[BS2_SetOsdpStandardDeviceSecurityKey](#)

10. *vendorCode*

Vendor .

11. *fwVersion*

OSDP FW .

12. *modelName*

OSDP .

13. *modelVersion*

OSDP .

14. *readInfo*

vendorCode fwVersion, model , OSDP
가 master .

15. *reserved*

BS2OsdpStandardDeviceAdd

```

typedef struct {
    uint8_t      osdpID;           ///< 1 byte
    uint8_t      activate;        ///< 1 byte
    uint8_t      useSecureSession; ///< 1 byte
    uint8_t      deviceType;      ///< 1 byte
    BS2_DEVICE_ID deviceID;       ///< 4 bytes
} BS2osdpStandardDeviceAdd;      ///< 8 bytes

```

1. *osdpID*

OSDP 가 0~126 .

가 .

2. *activate*

, false .

3. useSecureSession

[BS2_SetOsdpStandardDeviceSecurityKey](#)

4. deviceType

. BS2_DEVICE_TYPE_3RD_OSDP_DEVICE

5. deviceID

. 0 master 가

BS2OsdpStandardDeviceUpdate

```
typedef struct {
    uint8_t      osdpID;           ///< 1 byte
    uint8_t      activate;        ///< 1 byte
    uint8_t      useSecureSession; ///< 1 byte
    uint8_t      deviceType;      ///< 1 byte
    BS2_DEVICE_ID deviceID;       ///< 4 bytes
} BS2OsdpStandardDeviceUpdate;   ///< 8 bytes
```

1. osdpID

OSDP 가 0~126

가

2. activate

, false

3. useSecureSession

[BS2_SetOsdpStandardDeviceSecurityKey](#)

4. deviceType

. BS2_DEVICE_TYPE_3RD_OSDP_DEVICE

5. deviceID

BS2OsdpStandardDeviceCapability

```
typedef struct {
    uint8_t      compliance;
    uint8_t      count;
} BS2OsdpStandardDeviceCapabilityItem;
```

```

typedef struct {
    BS20sdpStandardDeviceCapabilityItem input;           ///< 2 bytes
    BS20sdpStandardDeviceCapabilityItem output;         ///< 2 bytes
    BS20sdpStandardDeviceCapabilityItem led;           ///< 2 bytes
    BS20sdpStandardDeviceCapabilityItem audio;         ///< 2 bytes
    BS20sdpStandardDeviceCapabilityItem textOutput;    ///< 2 bytes
    BS20sdpStandardDeviceCapabilityItem reader;        ///< 2 bytes

    uint16_t recvBufferSize;                            ///< 2 bytes
    uint16_t largeMsgSize;                              ///< 2 bytes

    uint8_t osdpVersion;                                ///< 1 byte
    uint8_t cardFormat;                                 ///< 1 byte
    uint8_t timeKeeping;                                ///< 1 byte
    uint8_t canCommSecure;                              ///< 1 byte

    BS2_B00L crcSupport;                                ///< 1 byte
    BS2_B00L smartCardSupport;                          ///< 1 byte
    BS2_B00L biometricSupport;                          ///< 1 byte
    BS2_B00L securePinEntrySupport;                     ///< 1 byte

    uint8_t reserved[4];                                ///< 4 bytes
} BS20sdpStandardDeviceCapability;                    ///< 28 bytes
    
```

1. *compliance*

PD (function) (compliance level) , OSDP
 input, output, led, audio, textOutput

2. *count*

PD (function) (number of objects) , 가
 OSDP

3. *input*

()

4. *output*

5. *led*

LED

6. *audio*

Buzzer

7. *textOutput*

8. *reader*

, count

9. *recvBufferSize*

PD가

10. *largeMsgSize*

PD가

11. *osdpVersion*

OSDP

12. *cardFormat*

compliance level , 01, 02, 03 . OSDP

13. *timeKeeping*

PD . OSDP 2.2

14. *canCommSecure*

15. *crcSupport*

16. *smartCardSupport*

17. *biometricSupport*

가 가

18. *securePinEntrySupport*

SPE(Secure PIN Entry)

19. *reserved*

BS2OsdpStandardDeviceResult

```
typedef struct {
    BS2_DEVICE_ID    deviceID;
    BS2_OSDP_RESULT result;
} BS2osdpStandardDeviceResult;
```

1. *deviceID*

2. *result*

OSDP 가

0	Success
1	Fail

CoreStation40 Suprema , OSDP
 OSDP 가 가 2
 channelType 3 가

0	Normal
1	Suprema
2	OSDP
3	OSDP FULL

3. *maxOsdpDevice*
 가
 channelType 1 32 , 2 3 2

4. *numOsdpAvailableDevice*
 가

5. *deviceIDs*
 ()

6. *numOfChannel*
 . CoreStation40 5

7. *reserved*

8. *channels*
 OSDP
 8 가 , CoreStation40 5 가 0~4

9. *reserved1*

From:
<http://kb.supremainc.com/bs2sdk/> - **BioStar 2 Device SDK**
 Permanent link:
http://kb.supremainc.com/bs2sdk./doku.php?id=ko:slave_control_api&rev=1675929639
 Last update: **2023/02/09 17:00**