

Table of Contents

- Slave Control API** 1
- Structure** 1
- BS2Rs485SlaveDevice 1
- BS2Rs485SlaveDeviceEX 2
- BS2OsdpStandardDevice 2
- BS2OsdpStandardDeviceAvailable 4
- BS2OsdpStandardNotify 5
- BS2OsdpStandardDeviceAdd 7
- BS2OsdpStandardDeviceUpdate 7
- BS2OsdpStandardDeviceCapability 8
- BS2OsdpStandardDeviceResult 10
- BS2OsdpStandardDeviceSecurityKey 10

Slave Control API

API that connects the master and slave device by using the RS-485 network. The v2 devices will now work as a dummy reader when set as a slave device. It will not store any kind of information for user and all will be stored inside the master device. The slave device will send the information scanned, and the matching and access rule check will be done from the master device. The slave device will only receive the result.

- [BS2_GetSlaveDevice](#): Searches a slave device from the RS-485 network.
- [BS2_SetSlaveDevice](#): Add/Modify/Delete a slave device from the master device.
- [BS2_GetSlaveExDevice](#): In case of CoreStation, searches a slave device from the RS-485 network.
- [BS2_SetSlaveExDevice](#): In case of CoreStation, Add/Modify/Delete a slave device from the master device.
- [BS2_SearchDevicesCoreStation](#): Searches CoreStation devices from the current network.
- [BS2_SearchDevicesCoreStationEx](#): [+ 2.6.3] Searches CoreStation devices from the current network with host IP.
- [BS2_GetDevicesCoreStation](#): Gets searched CoreStation devices.

CAUTION

When the SDK is initialized, there are no information about the slave devices. Therefore, a slave device must be searched or added before controlling.

Structure

BS2Rs485SlaveDevice

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

1. *deviceID*

Device ID.

2. *deviceType*

Device type.

3. *enableOSDP*

Decides whether to use a slave device.

4. *connected*

Displays whether a slave device is connected to the master device.

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*

Device ID.

2. *deviceType*

Device type.

3. *enableOSDP*

Decides whether to use a slave device.

4. *connected*

Displays whether a slave device is connected to the master device.

5. *channelInfo*

Channel value of slave device.

6. *reserved*

Reserved space.

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          modelNumber;  ///< 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 Device Identifier.

2. *deviceType*

Device type. Usually gets BS2_DEVICE_TYPE_3RD_OSDP_DEVICE.

3. *enableOSDP*

Always set to true.

4. *connected*

If true, an OSDP device is currently connected.

5. *channelInfo*

Connected channel information. CoreStation40 has 5 channels from 0 to 4, it has channel values within this range.

6. *osdpID*

OSDP Identifier.

7. *supremaSearch*

Information that is distinct from existing RS485 that does not support OSDP, and always set to false.

8. *activate*

Regardless of the connection status of the device, you can set whether the connected device is operating, which indicates the activation status of the operation.

9. *useSecure*

Indicates whether encrypted communication is enabled.

If a separate key is not set, encryption communication is used as the default key, and can be changed with [BS2_SetOsdpStandardDeviceSecurityKey](#).

10. *vendorCode*

Unique value of Vendor.

11. *fwVersion*

FW version information of OSDP device.

12. *modelName*

Model number of the OSDP device.

13. *modelVersion*

Model version of the OSDP device.

14. readInfo

Indicates whether OSDP device information such as vendorCode, fwVersion, or model has. If it has the OSDP device information, it means that the device has ever been connected to the master device.

15. reserved

Reserved Space.

BS2OsdpStandardDeviceAvailable

```
typedef struct {
    uint8_t          channelIndex;          ///< 1 byte
    BS2_OSDP_CHANNEL_TYPE channelType;    ///< 1 byte
    uint8_t          maxOsdpDevice;       ///< 1 byte
    uint8_t          numOsdpAvailableDevice; ///< 1 byte
    BS2_DEVICE_ID   deviceIDs[8];        ///< 4 x 8 = 32
bytes
} BS2OsdpStandardChannelInfo;          ///< 36 bytes

typedef struct {
    uint8_t          numOfChannel;         ///< 1 byte
    uint8_t          reserved[3];         ///< 3 bytes
    BS2OsdpStandardChannelInfo channels[BS2_RS485_MAX_CHANNELS_EX]; ///< 36
x 8 = 288 bytes
    uint8_t          reserved1[32];      ///< 32 bytes
} BS2OsdpStandardDeviceAvailable;      ///< 288 bytes + 36
```

1. channelIndex

OSDP 장치가 연결되어 있는 통신 채널 번호입니다.

2. channelType

RS485 통신을 하는 장치가 어떠한 방식으로 연결되어 있는지를 나타냅니다.

CoreStation40 기준으로, 할당 가능한 채널은 0~4까지 5개인데, 각 채널내에서는 슈프리마 장치와 OSDP 장치가 혼용되어 운용될 수 없습니다.

특정 채널에 연결된 장치가 없는 경우, Suprema 장치든, OSDP 장치이든 연결될 수 있음을 나타내는 0을 갖습니다.

만일 특정 채널로 Suprema 장치가 연결되면, 이후 해당 채널은 Suprema 장치들의 연결만 허용되며, channelType은 1을 갖습니다. OSDP 장치가 연결되어 있더라도 무시되어집니다.

만일 특정 채널로 OSDP 장치가 연결되면, 이후 해당 채널은 OSDP 장치 연결만 허용되며, channelType은 2를 갖습니다. Suprema 장치가 연결되어 있더라도 무시되어집니다.

CoreStation40 각각의 채널들은 Suprema 장치 채널, OSDP 장치 채널들로 혼용되어 운용될 수 있습니다. 채널에 OSDP로 장치가 연결이되면 허용이 가능한 최대 장치수는 2대로 제한되며, 이미 해당 채널이 최대치에 도달했으면 channelType은 3을 갖으며 더이상 연결이 불가능함을 나타냅니다.

값	설명
0	Normal
1	Suprema 장치
2	OSDP 장치

값	설명
3	OSDP 장치 FULL

3. *maxOsdpDevice*

해당 채널에서 연결 가능한 장치의 최대치를 나타냅니다.
channelType이 1이면 32를, 2나 3이면 2를 갖습니다.

4. *numOsdpAvailableDevice*

해당 채널에서 현재 연결 가능한 장치수를 나타냅니다.

5. *deviceIDs*

해당 채널에 연결된(또는 연결될 수 있는) 장치 식별자 리스트입니다.

6. *numOfChannel*

채널 수입니다. CoreStation40 기준 5를 갖습니다.

7. *reserved*

예약된 공간입니다.

8. *channels*

채널별 OSDP 장치정보입니다.

최대 8개까지 채널정보를 가질 수 있으나, CoreStation40이 5개의 채널을 가지므로 0~4번까지만 유의미합니다.

9. *reserved1*

예약된 공간입니다.

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)
} BS20sdpStandardNotify; ///< 48 bytes
```

1. *deviceID*

OSDP Device Identifier.

2. *deviceType*

Device type. Usually gets BS2_DEVICE_TYPE_3RD_OSDP_DEVICE.

3. *enableOSDP*

Always set to true.

4. *connected*

If true, an OSDP device is currently connected.

5. *channelInfo*

Connected channel information. CoreStation40 has 5 channels from 0 to 4, it has channel values within this range.

6. *osdpID*

OSDP Identifier.

7. *supremaSearch*

Information that is distinct from existing RS485 that does not support OSDP, and always set to false.

8. *activate*

Regardless of the connection status of the device, you can set whether the connected device is operating, which indicates the activation status of the operation.

9. *useSecure*

Indicates whether encrypted communication is enabled.

If a separate key is not set, encryption communication is used as the default key, and can be changed with [BS2_SetOsdpStandardDeviceSecurityKey](#).

10. *vendorCode*

Unique value of Vendor.

11. *fwVersion*

FW version information of OSDP device.

12. *modelName*

Model number of the OSDP device.

13. *modelVersion*

Model version of the OSDP device.

14. *readInfo*

Indicates whether OSDP device information such as vendorCode, fwVersion, or modelName has. If it has the OSDP device information, it means that the device has ever been connected to the master device.

15. *reserved*

Reserved Space.

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 Identifier. The identifier must be set to a random value between 0 and 126 by the user.

This value does not allow duplication within the same channel and may raise an error if duplicated or out-of-range values are set.

If the channels are different within the master device, you can add devices by setting them to the same identifier.

2. *activate*

Specifies the device activation state.

Regardless of the connection state of the device, if set to false, the operation of the device is ignored even if it is successfully connected.

3. *useSecureSession*

Specifies whether to encrypt communication.

If a separate key is not set, encryption communication is used as the default key, and can be changed with [BS2_SetOsdpStandardDeviceSecurityKey](#).

4. *deviceType*

Device type. Should be set to BS2_DEVICE_TYPE_3RD_OSDP_DEVICE.

5. *deviceID*

Device Identifier. When set to 0, the master device automatically allocates.

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 Identifier. The identifier must be set to a random value between 0 and 126 by the user. This value does not allow duplication within the same channel and may raise an error if duplicated or out-of-range values are set.

If the channels are different within the master device, you can add devices by setting them to the same identifier.

2. *activate*

Specifies the device activation state.

Regardless of the connection state of the device, if set to false, the operation of the device is ignored even if it is successfully connected.

3. *useSecureSession*

Specifies whether to encrypt communication.

If a separate key is not set, encryption communication is used as the default key, and can be changed with [BS2_SetOsdpStandardDeviceSecurityKey](#).

4. *deviceType*

Device type. Should be set to BS2_DEVICE_TYPE_3RD_OSDP_DEVICE.

5. *deviceID*

Device Identifier.

BS2OsdpStandardDeviceCapability

```
typedef struct {
    uint8_t          compliance;
    uint8_t          count;
} BS2osdpStandardDeviceCapabilityItem;

typedef struct {
    BS2osdpStandardDeviceCapabilityItem input;    ///< 2 bytes
    BS2osdpStandardDeviceCapabilityItem output;   ///< 2 bytes
    BS2osdpStandardDeviceCapabilityItem led;      ///< 2 bytes
    BS2osdpStandardDeviceCapabilityItem audio;    ///< 2 bytes
    BS2osdpStandardDeviceCapabilityItem textOutput; ///< 2 bytes
    BS2osdpStandardDeviceCapabilityItem 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*

PD의 기능(function)에 따른 준수수준(compliance level)을 나타냅니다.

기능은 input, output, led, audio, textOutput 등이 있으며, 기능 별로 준수수준이 상이하므로 관련 OSDP 문서를 참고바랍니다.

2. *count*

PD의 기능(function)에 따른 갯수(number of objects)를 의미하는데, 기능 별로 갯수에 부여되는 의미가 다르므로 관련 OSDP 문서를 참고바랍니다.

3. *input*

입력(모니터링) 기능을 정의합니다.

4. *output*

출력 기능을 정의합니다.

5. *led*

LED 기능을 정의합니다.

6. *audio*

Buzzer 기능을 정의합니다.

7. *textOutput*

문자 출력 기능을 정의합니다.

8. *reader*

지원되는 장치 수를 나타내며, count 정보만 의미를 갖습니다.

9. *recvBufferSize*

PD가 수신할 수 있는 단문 메시지 크기를 나타냅니다.

10. *largeMsgSize*

PD가 처리할 수 있는 장문 메시지의 최대 크기입니다.

11. *osdpVersion*

OSDP 버전입니다.

12. *cardFormat*

카드 데이터 포맷 기능을 정의하며, 01, 02, 03 중 하나의 값을 갖습니다. OSDP 문서의 카드 데이터 포맷 관련 compliance level을 참고바랍니다.

13. *timeKeeping*

PD의 날짜 및 시간 정보 유형과 이를 유지하는 것에 대해 나타냅니다. OSDP 2.2에서 이 기능은 사용되지 않습니다.

14. *canCommSecure*

보안통신 지원 여부를 나타냅니다.

15. *crcSupport*

체크섬 지원 여부를 나타냅니다.

16. *smartCardSupport*

스마트 카드 지원 여부를 나타냅니다.

17. *biometricSupport*

생체정보 처리가 가능한지를 나타냅니다.

18. *securePinEntrySupport*

SPE(Secure PIN Entry) 지원 여부를 나타냅니다.

19. *reserved*

예약된 공간입니다.

BS2OsdpStandardDeviceResult

```
typedef struct {
    BS2_DEVICE_ID    deviceID;
    BS2 OSDP_RESULT  result;
} BS2OsdpStandardDeviceResult;
```

1. *deviceID*

장치 식별자입니다.

2. *result*

OSDP 장치 명령 결과 값을 가집니다.

값	설명
0	Success
1	Fail
2	Not available

BS2OsdpStandardDeviceSecurityKey

```
typedef struct {
    uint8_t    key[BS2 OSDP_STANDARD_KEY_SIZE];
    uint8_t    reserved[32];
} BS2OsdpStandardDeviceSecurityKey;
```

1. *key*

OSDP 장치에서 사용할 16byte 크기의 보안키입니다.

2. *reserved*

예약된 공간입니다.

From:

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

Permanent link:

https://kb.supremainc.com/bs2sdk/doku.php?id=en:slave_control_api&rev=1677564172

Last update: **2023/02/28 15:02**