Table of Contents

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.

> **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.