Table of Contents

_BS2_GetOsdpStandardDevice_ ....

Slave Control API > BS2_GetOsdpStandardDevice

BS2_GetOsdpStandardDevice

[+ 2.9.1] `CoreStation40` Gets OSDP device information.

Declaration

```
#include "BS_API.h"

int BS2_GetOsdpStandardDevice(void* context, uint32_t osdpDeviceId,
BS2OsdpStandardDevice* osdpDevice);
```

> See BS2OsdpStandardDevice Structure

Parameter

- [In] *context* : Context
- [In] *osdpDeviceId* : OSDP Device Identifier
- [Out] *osdpDevice* : Pointer to store OSDP device information

Return Value

If successfully done, BS_SDK_SUCCESS will be returned. If there is an error, the corresponding error code will be returned.

See Also

BS2_AddOsdpStandardDevice
BS2_GetOsdpStandardDevice
BS2_GetAvailableOsdpStandardDevice
BS2_UpdateOsdpStandardDevice
BS2_RemoveOsdpStandardDevice
BS2_GetOsdpStandardDeviceCapability
BS2_SetOsdpStandardDeviceSecurityKey

Sample Code(C++)

sample_bs2_getosdpstandarddevice.cpp

```cpp
BS2OsdpStandardDevice osdpDevice = { , };

BS2_DEVICE_ID osdpID = Utility::selectSlaveID();
int sdkResult = BS2_GetOsdpStandardDevice(context_, osdpID,
&osdpDevice);
if (BS_SDK_SUCCESS != sdkResult)
    printf("BS2_GetOsdpStandardDevice call failed: %d", sdkResult);

return sdkResult;
```

Sample Code (C#)

sample_bs2_getosdpstandarddevice.cs

```csharp
UInt32 id = Util.GetInputSlaveID();

BS2OsdpStandardDevice osdpDevice;
BS2ErrorCode result =
(BS2ErrorCode)API.BS2_GetOsdpStandardDevice(sdkContext, id, out
osdpDevice);
if (result != BS2ErrorCode.BS_SDK_SUCCESS)
    Console.WriteLine("Got error({0}).", result);
else
    Console.WriteLine("Call success.");

return result;
```