

SUNIX Device Port Library

HOWTO

For Linux

Table of Contents

1. Introduction	03
2. Requirements and Platform Support	04
3. Install	05
4. DP management function	06
5. DIO function	27
6. Serial function	65
7. Control unit function	86
8. Multi-function AI	93
9. How to use library	112
10. Return Code	113
11. Release Note	116

1. Introduction

The SUNIX Device Port library help user to develop program for controlling SUNIX hardware more easily.

SUNIX Device Port library functions categories.

- DP management function
- DIO function
- SERIAL function
- CTU function
- Multi-function AI

2. Requirements and Platform Support

Minimum working version

- Kernel : 3.x , 4.x

Platform Support

- x86 32 bit
- x86 64 bit

3. Install

- Install library

step 1- login as root.

step 2- copy libsdxio.so and libsdxio.so.9 to /usr/lib

- Install service

README file in DP_LINUX directory for reference.

4.1 DP management function

```
int dp_lib_init
void dp_lib_free
int dp_get_version
int dp_get_all_nic_info
int dp_refresh_nic
int dp_connect
int dp_disconnect
int dp_search_dp
int dp_get_specific_dp_detail_info
int dp_mapping
int dp_unmapping
int dp_get_all_dp_info_connected_by_host
int dp_set_disconnect_event_callback
int dp_get_all_dp_mapping_info
int dp_get_service_connection_status
```

4.1.1 dp_lib_init

-- Description --

SUNIX device port library initialization. Call this function before using any other library function.

-- Syntax --

C/C+
int dp_lib_init(void)

-- Arguments --

None

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

4.1.2 dp_lib_free

-- Description --

SUNIX device port library de-initialization. Call this function before exit program.

-- Syntax --

C/C+

```
void dp_lib_free(void);
```

-- Arguments --

None

-- Return Code --

Always return STATUS_SUCCESS

4.1.3 dp_get_version

-- Description --

Get service and library version.

-- Syntax --

C/C+

```
int dp_get_version(PVERSION_INFO pInfo)
```

-- Arguments --

pInfo

[out] Version information struct pointer (define in sdxmng.h)

-- Return Code --

Always return STATUS_SUCCESS

4.1.4 dp_get_all_nic_info

-- Description --

Get all NIC information.

-- Syntax --

C/C+

```
int dp_get_all_nic_info(PNIC_LIST pNicList)
```

-- Arguments --

pNicList

[out] NIC information struct pointer (define in sdxmng.h)

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	MNG_STATUS_SERVICE_NOT_CONNECT MNG_STATUS_NIC_NOT_FOUND

4.1.5 dp_refresh_nic

-- Description --

Refresh all NIC information.

-- Syntax --

C/C+

```
int dp_refresh_nic(PNIC_LIST pNicList)
```

-- Arguments --

pNicList

[out] NIC information struct pointer (define in sdxmng.h)

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	MNG_STATUS_SERVICE_NOT_CONNECT MNG_STATUS_NIC_NOT_FOUND

4.1.6 dp_search_dp

-- Description --

Get all device port information.

-- Syntax --

C/C+

```
int dp_search_dp(PDP_LIST pDpList)
```

-- Arguments --

pDpList

[out] Device port list information struct pointer (define in sdxmng.h)

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	MNG_STATUS_SERVICE_NOT_CONNECT MNG_STATUS_DP_NOT_FOUND

```

typedef struct _DP_INFO
{
    unsigned char          Mac[6];
    unsigned char          ModelName[20];
    unsigned char          DpMode;
    unsigned char          PasswordEnable;
    unsigned char          ConnectionStatus;
    unsigned char          ConnectedHostMac[6];
    unsigned char          MappingStatus;// (0x01 YES, 0x00 NO)
    unsigned char          Channels;
    unsigned char          Major;
    unsigned char          Minor;
    unsigned char          CtuTcpPort;
    unsigned int           SioTcpPort;
    DIB_
        DibInfo[DP_DIB_MAX];

}DP_INFO,*PDP_INFO;

```

Mac : Device port Mac address.

ModelName : Device port model name , ex: DPAS02H00.

DpMode: 0x00 : dock mode
 0x01 : Advance mode
 0x02 : IP mode

PasswordEnable : 0x00 : Disable
 0x01 : Enable

ConnectionStatus : 0x00 : Disconnect
 0x01 : Connected

ConnectedHostMac : Connected host mac address.

MappingStatus : Device port mapping status.

Channels : Number of available DIB channels.

Major : Major version

Minor : Minor version

CtuTcpPort : CTU Tcp port number.

SioTcpPort : SIO Tcp port number.

DibInfo : Device port DIB information struct.

4.1.7 dp_get_specific_dp_detail_info

-- Description --

Get specific device port detail information.

-- Syntax --

C/C+

```
int dp_get_specific_dp_detail_info(unsigned char * id, PDP_INFO pInfo);
```

-- Arguments --

id

[in] Device port Mac address.

pInfo

[out] SUNIX device port information struct pointer (define in sdxmng.h)

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	MNG_STATUS_SERVICE_NOT_CONNECT MNG_STATUS_DP_NOT_FOUND

```

typedef struct _DIB_INFO
{
    unsigned char          ChlNumber ;
    unsigned char          Type ;
    unsigned char          MappingMinorNum ;
    DIO_INFO               DioInfo ;
    SER_INFO               SerInfo ;
    STR_INFO               StrInfo;
}DIB_INFO, *PDIB_INFO;

```

ChlNumber : DIB channel number.

Type : Type info

0x00 : Control Unit
 0x01 : Serial
 0x03 : DIO
 0x80 : Virtual

```

typedef struct _DIO_INFO
{
    unsigned char          Version;
    unsigned char          NumberOfBanks;
    unsigned char          ShareDirectionControl;
    unsigned char          SamplingCapability;
    unsigned int           ClockFrequency;
    BANK_INFO              BankInfo[32];
}DIO_INFO, *PDIO_INFO;

```

Version: DIO channel version

NumberOfBanks : Number of banks in this DIO channel.

ShareDirectionControl : IO ports in same bank shares same direction control.
 (0x00 -> none support , 0x01-> support)

SamplingCapability : Sampling clock scale capability.

Support mode bit	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
Support Divide	50	100	2.5k	5k	50k	100k	2.5M	5M

ClockFrequency : System clock frequency.

```
typedef struct _BANK_INFO
{
    unsigned char BankIndex;
    unsigned char IoNumber;
    unsigned char WritingSettingsCapability;
    unsigned char DirectionCapability;
    unsigned char EdgeTriggerCapability;
    unsigned char OutputPinConfigCapability;
    unsigned char OutputPinResistorApplied;
    unsigned int  DirectionMode;
    unsigned int  InputInvertEnable;
    unsigned char EventMode;
    unsigned char SamplingClockControl;
    unsigned char OutputPinConfig;
}BANK_INFO, *PBANK_INFO;
```

BankIndex : In this DIO channel bank serial number.

IoNumber: Number of I/O in this bank. (0 ~31)

WritingSettingsCapability : Writing settings to flash capability
(0x00 -> NO, 0x01-> YES)

DirectionCapability : Bank support direction.
(0x01 -> input, 0x02 -> output, 0x03 -> both)

EdgeTriggerCapability : Edge trigger capability.
(0x01 -> rising, 0x02 -> falling, 0x03 -> both)

OutputPinConfigCapability : Out pin config support mode
(0x00 -> logic, 0x01 -> open-drain, 0x02 -> logic/open-drain configurable)

OutputPinResistorApplied : Out pin resistor support mode.
(0x00 -> none, 0x01 -> pull-up, 0x02 -> pull-down)

DirectionMode : Bank direction (0 -> input, 1 -> output)
(Each bit is one I/O direction)
I/O [31 ~ 24] [23 ~ 16] [15 ~ 8] [7 ~ 0]

InputInvertEnable : DI input invert control mode(0 -> disable, 1 -> enable)
(Control each I/O by bit.)
I/O [31 ~ 24] [23 ~ 16] [15 ~ 8] [7 ~ 0]

EventMode : Event support mode .

(0x00 -> disable, 0x01 -> rising edge, 0x02 -> falling edge, 0x03 -> both)

SamplingClockControl : Sampling clock control.

Clock divider	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08
System Divide	1	50	100	2.5k	5k	50k	100k	2.5M	5M

OutputPinConfig : Out pin configuration.

(0x00 -> logic, 0x01 -> open-drain)

```
typedef struct _SER_INFO
```

```
{
    unsigned char    TransmitCapability
    unsigned char    AHDCCapability
    unsigned char    CSCapability
    unsigned char    RS422TerminationCtlCapability;
    unsigned char    RS485TerminationCtlCapability;
    unsigned char    RIPowerIOCapability;
    unsigned char    DCDCPowerIOCapability;
}SER_INFO,*PSER_INFO;
```

TransmitCapability	0x01 RS232 , 0x02 RS422 , 0x03 RS485 0x04 RS/422/485
AHDCCapability	0x00 none , 0x01 Enable
CSCapability	0x00 none , 0x01 Enable
RS422TerminationCtlCapability	0x00 none , 0x01 Enable
RS485TerminationCtlCapability	0x00 none , 0x01 Enable
RIPowerIOCapability	0x00 none, 0x01 5v , 0x02 12v , 0x03 no distinguish
DCDCPowerIOCapability	0x00 none, 0x01 5v , 0x02 12v , 0x03 no distinguish

```
typedef struct _STR_INFO
```

```
{
    unsigned int    DeviceNameLen;
    unsigned char    DeviceName[64];
}
```

4.1.8 dp_connect

-- Description --

Connect a device port

-- Syntax --

C/C+

```
int dp_connect(unsigned char * id , int * SioTcpPort , int * CtuTcpPort , unsigned char * PwSetting,  
              unsigned char * PassWord, unsigned int PassWordLength); );
```

-- Arguments --

id

[in] Device port Mac address.

SioTcpPort

[out] SIO Tcp port number.

CtuTcpPort

[out] CTU Tcp port number.

PwSetting

[in] Passored request. (0x00 ->NO , 0X01 -> Yes)

PassWord

[in] PassWord.

PassWordLength

[in] Password Length.

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	MNG_STATUS_SERVICE_NOT_CONNECT MNG_STATUS_DP_NOT_FOUND MNG_STATUS_DP_CONNECT_ALREADY MNG_STATUS_DP_CONNECT_BUSY

4.1.9 dp_disconnect

-- Description --

Disconnect a device port.

-- Syntax --

C/C+

```
int dp_disconnect(unsigned char * id);
```

-- Arguments --

id

[in] Device port Mac address.

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	MNG_STATUS_SERVICE_NOT_CONNECT MNG_STATUS_DP_NOT_FOUND MNG_STATUS_DP_DISCONNECT_ALREADY

4.1.10 dp_get_all_dp_info_connected_by_host

-- Description --

Get all device port connected by host.

-- Syntax --

C/C+

```
int dp_get_all_dp_info_connected_by_host(PDP_LIST DpList);
```

-- Arguments --

DpList

[out] Device port list information struct pointer (define in sdxmng.h)

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	MNG_STATUS_SERVICE_NOT_CONNECT MNG_STATUS_DP_NOT_FOUND

4.1.11 dp_mapping

-- Description --

Device port mapping. (Call this function before using serial port)

-- Syntax --

C/C+

```
int dp_mapping(unsigned char * id);
```

-- Arguments --

id

[in] Device port Mac address.

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	MNG_STATUS_SERVICE_NOT_CONNECT MNG_STATUS_DP_NOT_FOUND

4.1.12 dp_unmapping

-- Description --

Device port unmapping.

-- Syntax --

C/C++

```
int dp_unmapping(unsigned char * id);
```

-- Arguments --

id

[in] Device port Mac address.

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	MNG_STATUS_SERVICE_NOT_CONNECT MNG_STATUS_DP_NOT_FOUND

4.1.13 dp_get_all_dp_mapping_info

-- Description --

Get Mapping Device port info.

-- Syntax --

C/C+

```
int dp_get_all_mapping_info(PDP_MAPPING_LIST pMappingList);
```

-- Arguments --

pMappingList

[out] Device Port mapping list information struct pointer (define in sdxmng.h)

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	MNG_STATUS_SERVICE_NOT_CONNECT MNG_STATUS_DP_NOT_FOUND

4.1.14 dp_set_disconnet_event_callback

-- Description --

Set device port disconnect event callback function , when disconnect event occur using this function.

-- Syntax --

C/C+

```
int dp_set_disconnet_event_callback(unsigned char * Mac ,  
                                     void (*managementcallback)(unsigned char * mac))
```

-- Arguments --

Mac

[in] Connect device port mac address.

void (*managementcallback)(unsigned char * mac)

[in] When device port disconnect event occur, library will call this function.

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

4.1.15 **dp_get_service_connection_status**

-- Description --

Get connection status.

-- Syntax --

C/C+

dp_get_service_connection_status(void)

-- Arguments --

Void

-- Return Code --

Status	Value
Establish	0x01
Disestablish	0x00

5.1 DIO function

sio_init
sio_free
dio_get_bank_state
dio_set_bank_state
dio_get_bank_input_invert
dio_set_bank_input_invert
dio_get_bank_direction
dio_set_bank_direction
dio_get_bank_input_event_mode
dio_set_bank_input_event_mode
dio_get_bank_sampling_clock
dio_set_bank_sampling_clock
dio_get_bank_output_pin_config
dio_set_bank_output_pin_config
dio_get_bank_output_initial_value
dio_set_bank_output_initial_value
dio_get_bank_output_safe_value
dio_set_bank_output_safe_value
dio_get_output_safe_value_timer
dio_set_output_safe_value_timer
dio_get_bank_input_latch_rising_edge
dio_set_bank_input_latch_rising_edge
dio_get_bank_input_latch_falling_edge
dio_set_bank_input_latch_falling_edge
dio_set_bank_input_counter_reset
dio_get_bank_input_counter_increment_rising_edge
dio_set_bank_input_counter_increment_rising_edge
dio_get_bank_input_counter_increment_falling_edge
dio_set_bank_input_counter_increment_falling_edge
dio_get_bank_io_input_counter_value
dio_set_bank_event_callback
dio_set_io_state
dio_set_io_event_callback
dio_get_io_connection_status

5.1.1 sio_init

-- Description --

Initialization DIO setting. Call this function before using DIO function.

-- Syntax --

C/C+

```
int sio_init(unsigned char * Mac, int TcpPort);
```

-- Arguments --

Mac

[in] Device port mac address

TcpPort

[in] Tcp port

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	SIO_STATUS_CONNECT_FAIL

5.1.2 sio_free

-- Description --

De-initialization SIO setting.

-- Syntax --

C/C+
int sio_free(unsigned char * Mac);

-- Arguments --

Mac
[in] device port mac address

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_SIO_NOT_CONNECT

5.1.3 dio_get_info

-- Description --

Get DIO port information.

-- Syntax --

C/C+

```
int dio_get_info(unsigned char * Mac, int DibChlNumber, PDIO_INFO pDioInfo);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

pDioInfo

[out] DIO information struct pointer

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE STATUS_DIO_NOT_FOUND

```
typedef struct _DIO_INFO
{
    unsigned char    Version;
    unsigned char    NumberOfBanks;
    unsigned char    ShareDirectionControl;
    unsigned char    SamplingCapability;
    unsigned int     ClockFrequency;
    BANK_INFO        BankInfo[32];

}DIO_INFO, *PDIO_INFO;
```

Version: DIO channel version

NumberOfBanks : Number of banks in this DIO channel.

ShareDirectionControl : IO ports in same bank shares same direction control.
(0x00 -> none support , 0x01-> support)

SamplingCapability : Sampling clock scale capability.

Support mode bit	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
Support Divide	50	100	2.5k	5k	50k	100k	2.5M	5M

ClockFrequency : System clock frequency.

```
typedef struct _BANK_INFO
{
    unsigned char BankIndex;
    unsigned char IoNumber;
    unsigned char WritingSettingsCapability;
    unsigned char DirectionCapability;
    unsigned char EdgeTriggerCapability;
    unsigned char OutputPinConfigCapability;
    unsigned char OutputPinResistorApplied;
    unsigned int  DirectionMode;
    unsigned int  InputInvertEnable;
    unsigned char EventMode;
    unsigned char SamplingClockControl;
    unsigned char OutputPinConfig;
}BANK_INFO, *PBANK_INFO;
```

BankIndex : In this DIO channel bank serial number.

IoNumber: Number of I/O in this bank. (0 ~31)

WritingSettingsCapability : Writing settings to flash capability
(0x00 -> NO, 0x01-> YES)

DirectionCapability : Bank support direction.
(0x01 -> input, 0x02 -> output, 0x03 -> both)

EdgeTriggerCapability : Edge trigger capability.
(0x01 -> rising, 0x02 -> falling, 0x03 -> both)

OutputPinConfigCapability : Out pin config support mode
(0x00 -> logic, 0x01 -> open-drain, 0x02 -> logic/open-drain configurable)

OutputPinResistorApplied : Out pin resistor support mode.
(0x00 -> none, 0x01 -> pull-up, 0x02 -> pull-down)

DirectionMode : Bank direction (0 -> input, 1 -> output)
(Each bit is one I/O direction)
I/O [31 ~ 24] [23 ~ 16] [15 ~ 8] [7 ~ 0]

InputInvertEnable : DI input invert control mode(0 -> disable, 1 -> enable)
(Control each I/O by bit.)
I/O [31 ~ 24] [23 ~ 16] [15 ~ 8] [7 ~ 0]

EventMode : Event support mode .
(0x00 -> disable, 0x01 -> rising edge, 0x02 -> falling edge, 0x03 -> both)

SamplingClockControl : Sampling clock control.

Clock divider	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08
System Divide	1	50	100	2.5k	5k	50k	100k	2.5M	5M

OutputPinConfig : Out pin configuration.

(0x00 -> logic, 0x01 -> open-drain)

5.1.4 dio_get_bank_state

-- Description --

Get DIO bank io state. (both direction input and output)

-- Syntax --

C/C+

```
int dio_get_bank_state(unsigned char * Mac,int DibChlNumber,int BankIndex,  
                      unsigned int * BankState);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

BankState

[out] Bank State , (0 -> low, 1 -> high)

(Each bit is one I/O state)

I/O [31 ~ 24] [23 ~ 16] [15 ~ 8] [7 ~ 0]

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_BANK_NOT_FOUND

5.1.5 dio_set_bank_state

-- Description --

Set DO output bank state.

-- Syntax --

C/C+

```
int dio_set_bank_state(unsigned char * Mac,int DibChlNumber,int BankIndex,  
                        unsigned int BankState);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

BankState

[in] Bank State , (0 -> low, 1 -> high)

(Control each I/O by bit)

I/O [31 ~ 24] [23 ~ 16] [15 ~ 8] [7 ~ 0]

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	SIO_STATUS_SIO_NOT_CONNECT STATUS_NO_OUTPUT_IO

5.1.6 dio_get_bank_input_invert

-- Description --

Get DI input invert is enable/disable.

-- Syntax --

C/C+

```
int dio_get_bank_input_invert(unsigned char * Mac,int DibChlNumber,int BankIndex,  
                             unsigned int * BankInputInvert);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

BankInputInvert

[out] Bank Input Invert state. (0 -> disable, 1 -> enable)

(Each bit is one I/O input invert.)

I/O [31 ~ 24] [23 ~ 16] [15 ~ 8][7 ~ 0]

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.7 dio_set_bank_input_invert

-- Description --

Set DI input invert enable/disable.

-- Syntax --

C/C+

```
int dio_set_bank_input_invert(unsigned char * Mac,int DibChlNumber,int BankIndex,  
                              unsigned int BankInputInvert);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

BankInputInvert

[in] (0 -> disable, 1 -> enable)

(Control each I/O by bit)

I/O [31 ~ 24] [23 ~ 16] [15 ~ 8] [7 ~ 0]

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.8 dio_get_bank_direction

-- Description --

Get bank direction .

-- Syntax --

C/C+

```
int dio_get_bank_direction(unsigned char * Mac,int DibChlNumber,int BankIndex,  
                           unsigned char * BankDirection);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

BankDirection

[out] Bank Direction. (0x00 -> input ,0x01 -> output)

(Each bit is one I/O direction)

I/O [31 ~ 24] [23 ~ 16] [15 ~ 8] [7 ~ 0]

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE STATUS_BANK_NOT_FOUND

5.1.9 dio_set_bank_direction

-- Description --

Set bank direction .

-- Syntax --

C/C+

```
int dio_set_bank_direction(unsigned char * Mac,int DibChlNumber,int BankIndex,  
                           unsigned char BankDirection);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

BankDirection

[in] Bank Direction .(0 -> input , 1 -> output)

(Control each I/O by bit)

I/O [31 ~ 24] [23 ~ 16] [15 ~ 8] [7 ~ 0]

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.10 dio_get_bank_input_event_mode

-- Description --

Get bank input event mode (disable/rising/falling/both).

-- Syntax --

C/C+

```
int dio_get_bank_input_event_mode(unsigned char * Mac,int DibChlNumber,int BankIndex,  
                                unsigned char * BankInputEventMode);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

BankInputEventMode

[out]

0x00 -> Disable input event

0x01 -> Rising edge trigger

0x02 -> Falling edge trigger

0x03 -> Both rising and falling edge trigger

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.11 dio_set_bank_input_event_mode

-- Description --

Set bank input event mode.

-- Syntax --

C/C++

```
int dio_set_bank_input_event_mode(unsigned char * Mac,int DibChlNumber,int BankIndex,  
                                unsigned char BankInputEventMode);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

BankInputEventMode

[in]

0x00 -> Disable input event

0x01 -> Rising edge trigger

0x02 -> Falling edge trigger

0x03 -> Both rising and falling edge trigger

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.12 dio_get_bank_sampling_clock

-- Description --

Get bank clock.

-- Syntax --

C/C++

```
int dio_get_bank_sampling_clock(unsigned char * Mac,int DibChlNumber, int BankIndex,  
                                unsigned char * BankSamplingClock)
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

BankSamplingClock

[out] Get bank sample rate

input divider value	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08
System Divide	1	50	100	2.5k	5k	50k	100k	2.5M	5M

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.13 dio_set_bank_sampling_clock

-- Description --

Set bank clock.

-- Syntax --

C/C++

```
int dio_set_bank_sampling_clock(unsigned char * Mac,int DibChlNumber, int BankIndex,  
                                unsigned char BankSamplingClock)
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

BankSamplingClock

[in] Set bank sample rate

input divider value	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08
System Divide	1	50	100	2.5k	5k	50k	100k	2.5M	5M

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.14 dio_get_bank_output_pin_config

-- Description --

Get bank output pin configuration.

-- Syntax --

C/C+

```
int dio_get_bank_output_pin_config(unsigned char * Mac,int DibChlNumber, int BankIndex,  
                                   unsigned char * BankOutputPinConfig);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

BankOutputPinConfig

[out] Output pin configuration

0x00 -> Logic output

0x01 -> Open-drain output

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.15 dio_set_bank_output_pin_config

-- Description --

Set bank output pin configuration.

-- Syntax --

C/C+

```
int dio_set_bank_output_pin_config(unsigned char * Mac ,int DibChlNumber,  
                                   int BankIndex , unsigned char BankOutputPinConfig);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

BankOutputPinConfig

[out] Output pin configuration

0x00 -> Logic output

0x01 -> Open-drain output

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.16 dio_get_bank_output_initial_value

-- Description --

Get Initial Power-ON value for DO.

-- Syntax --

C/C+

```
int dio_get_bank_output_initial_value(unsigned char * Mac,int DibChlNumber,  
                                     int BankIndex,unsigned int * InitialValue);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

InitialValue

[out] Initial value, (0 -> low, 1 -> high)

(Each bit is one I/O state)

I/O [31 ~ 24] [23 ~ 16] [15 ~ 8] [7 ~ 0]

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.17 dio_set_bank_output_initial_value

-- Description --

Set Initial Power-ON value for DO.

-- Syntax --

C/C+

```
int dio_set_bank_output_initial_value(unsigned char * Mac,int DibChlNumber,  
                                     int BankIndex,unsigned int  InitialValue);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

InitialValue

[out] Initial value, (0 -> low, 1 -> high)

(Control each I/O by bit)

I/O [31 ~ 24] [23 ~ 16] [15 ~ 8] [7 ~ 0]

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE STATUS_NO_OUTPUT_IO

5.1.18 dio_get_bank_output_safe_value

-- Description --

Get bank output safe value. (Safe value for DO when DP is disconnected unexpectedly)

-- Syntax --

C/C+

```
int dio_get_bank_output_safe_value(unsigned char * Mac,int DibChlNumber,int BankIndex,  
                                   unsigned int * SafeValue);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

SafeValue

[out] Safe value, (0 -> low, 1 -> high)

(Each bit is one I/O state)

I/O [31 ~ 24] [23 ~ 16] [15 ~ 8] [7 ~ 0]

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.19 dio_set_bank_output_safe_value

-- Description --

Set bank output safe value. (Safe value for DO when DP is disconnected unexpectedly)

-- Syntax --

C/C+

```
int dio_set_bank_output_safe_value(unsigned char * Mac,int DibChlNumber,int BankIndex,  
                                   unsigned int SafeValue);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

SafeValue

[out] Safe value, (0 -> low, 1 -> high)

(Control each I/O by bit)

I/O [31 ~ 24] [23 ~ 16] [15 ~ 8] [7 ~ 0]

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.20 dio_get_output_safe_value_timer

-- Description --

Get output safe value timer. Safe value will be loaded after this time unit (100ms, 25.5s~0s) when DP is disconnected unexpectedly

-- Syntax --

C/C++

```
int dio_get_output_safe_value_timer(unsigned char * Mac,int DibChlNumber,  
                                     unsigned char * Setting , unsigned char * Seconds);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

Setting

[out] Setting ,(0x00 -> disable, 0x01 -> enable)

Seconds

[out] (0x00 ~ 0xff, 1 unit -> 100ms)

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.21 dio_set_output_safe_value_timer

-- Description --

Set output safe value timer.

-- Syntax --

C/C+

```
int dio_set_output_safe_value_timer(unsigned char * Mac,int DibChlNumber,  
                                     unsigned char Setting , unsigned char Seconds);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

Setting

[in] Setting ,(0x00 -> disable, 0x01 -> enable)

Seconds

[in] (0x00 ~ 0xff, 1 unit -> 100ms)

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.22 dio_get_bank_input_latch_rising_edge

-- Description --

Get bank input latch rising edge.

-- Syntax --

C/C+

```
int dio_get_bank_input_latch_rising_edge(unsigned char * Mac,int DibChlNumber,  
                                         int BankIndex,unsigned int * LatchEdge);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

LatchEdge

[out] Value, (0 -> no changed, 1 -> changed since last reset)

(Each bit is one I/O state)

I/O [31 ~ 24] [23 ~ 16] [15 ~ 8] [7 ~ 0]

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.23 dio_set_bank_input_latch_rising_edge

-- Description --

Set bank input latch rising edge.

-- Syntax --

C/C+

```
int dio_set_bank_input_latch_rising_edge(unsigned char * Mac,int DibChlNumber,  
                                         int BankIndex,unsigned int LatchEdge);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

LatchEdge

[in] Value, (0 -> no changed, 1 -> changed since last reset)

(Control each I/O by bit)

I/O [31 ~ 24] [23 ~ 16] [15 ~ 8] [7 ~ 0]

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.24 dio_get_bank_input_latch_falling_edge

-- Description --

Get bank input latch falling edge.

-- Syntax --

C/C+

```
int dio_get_bank_input_latch_falling_edge(unsigned char * Mac,int DibChlNumber,  
int BankIndex,unsigned int * LatchEdge);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

LatchEdge

[out] Value, (0 -> no changed, 1 -> changed since last reset)

(Each bit is one I/O state)

I/O [31 ~ 24] [23 ~ 16] [15 ~ 8] [7 ~ 0]

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.25 dio_set_bank_input_latch_falling_edge

-- Description --

Set bank input latch falling edge.

-- Syntax --

C/C+

```
int dio_set_bank_input_latch_falling_edge(unsigned char * Mac,int DibChlNumber,  
                                           int BankIndex,unsigned int LatchEdge);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

LatchEdge

[in] Value, (0 -> no changed, 1 -> changed since last reset)

(Control each I/O by bit)

I/O [31 ~ 24] [23 ~ 16] [15 ~ 8] [7 ~ 0]

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.26 dio_set_bank_input_counter_reset

-- Description --

Set bank input counter reset.

-- Syntax --

C/C+

```
int dio_set_bank_input_counter_reset(unsigned char * Mac,int DibChlNumber,  
                                     int BankIndex,unsigned int Value);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

Value

[in] Value, (0 -> none, 1 -> reset corresponding counter)

(Control each I/O by bit)

I/O [31 ~ 24] [23 ~ 16] [15 ~ 8] [7 ~ 0]

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.27 dio_get_bank_input_counter_increment_rising_edge

-- Description --

Get bank input counter increment rising edge.

-- Syntax --

C/C+

```
int dio_get_bank_input_counter_increment_rising_edge(unsigned char * Mac,int DibChlNumber,  
int BankIndex,unsigned int * Value);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

Value

[out] Value, (0 -> not set, 1 -> set)

(Each bit is one I/O state)

I/O [31 ~ 24] [23 ~ 16] [15 ~ 8] [7 ~ 0]

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.28 dio_set_bank_input_counter_increment_rising_edge

-- Description --

Set bank input counter increment rising edge.

-- Syntax --

C/C+

```
int dio_set_bank_input_counter_increment_rising_edge(unsigned char * Mac,int DibChlNumber,  
int BankIndex,unsigned int Value);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

Value

[in] Value, (0 -> not set, 1 -> set)

(Control each I/O by bit)

I/O [31 ~ 24] [23 ~ 16] [15 ~ 8] [7 ~ 0]

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.29 dio_get_bank_input_counter_increment_falling_edge

-- Description --

Get bank input counter increment falling edge.

-- Syntax --

C/C+

```
int dio_get_bank_input_counter_increment_falling_edge(unsigned char * Mac,int DibChlNumber,  
int BankIndex,unsigned int * Value);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

Value

[out] Value, (0 -> not set, 1 -> set)

(Each bit is one I/O state)

I/O [31 ~ 24] [23 ~ 16] [15 ~ 8] [7 ~ 0]

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.30 dio_set_bank_input_counter_increment_falling_edge

-- Description --

Set bank input counter increment falling edge.

-- Syntax --

C/C+

```
int dio_set_bank_input_counter_increment_falling_edge(unsigned char * Mac,int DibChlNumber,  
int BankIndex,unsigned int Value);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

Value

[in] Value, (0 -> not set, 1 -> set)

(Control each I/O by bit)

I/O [31 ~ 24] [23 ~ 16] [15 ~ 8] [7 ~ 0]

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.31 dio_get_bank_io_input_counter_value

-- Description --

Get bank IO input counter value.

-- Syntax --

C/C+

```
int dio_get_bank_io_input_counter_value(unsigned char * pMac,int DibChlNumber,  
                                         int BankIndex,unsigned char BankIoIndex , unsigned int * Value);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

BankIoIndex

[in] Bank io index , range from 0 to io amount.

Value

[out] Value

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.32 dio_set_bank_event_callback

-- Description --

Set bank event callback function , when event occur using this function.

-- Syntax --

C/C+

```
int dio_set_bank_event_callback(unsigned char * Mac,int DibChlNumber,  
                                int BankIndex , void (*callback)( unsigned int hdr,  
                                unsigned int InputValue))
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

void (*callback)(unsigned int InputDelta, unsigned int InputValue)

[in] When bank event occur, library will call this function.

InputDelta (0 -> unchanged, 1 -> changed)

(Each bit is one Input status)

I/O [31 ~ 24] [23 ~ 16] [15 ~ 8] [7 ~ 0]

InputValue (0 -> low, 1 -> high)

(Each bit is one Input value)

I/O [31 ~ 24] [23 ~ 16] [15 ~ 8] [7 ~ 0]

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.33 dio_set_io_state

-- Description --

Set DO state.

-- Syntax --

C/C+

```
int dio_set_io_state(unsigned char * pMac,int DibChlNumber,int BankIndex,  
                    int IoIndex, unsigned char state);
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

IoIndex

[in] Io port index, range from 0 to bank amount.

BankState

[out] Bank State , (0 -> low, 1 -> high)

(Each bit is one I/O state)

I/O [31 ~ 24] [23 ~ 16] [15 ~ 8] [7 ~ 0]

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_BANK_NOT_FOUND

5.1.34 dio_set_io_event_callback

-- Description --

Set DI event callback function , when event occur using this function.

-- Syntax --

C/C+

```
int dio_set_io_event_callback(unsigned char * Mac,int DibChlNumber,  
int BankIndex, int IoIndex,  
void (*iocalback)(int IoIndex, unsigned char InputDelta,unsigned char InputValue));
```

-- Arguments --

Mac

[in] Connect device port mac address.

DibChlNumber

[in] DIO index , range from 1 to DIO amount.

BankIndex

[in] Bank index , range from 0 to bank amount.

IoIndex

[in] IO port index , range from 0 to IO amount.

void (*iocalback)(int IoIndex ,unsigned char InputDelta,unsigned char InputValue)

[in] When event occur, library will call this function.

IoIndex IO port index

InputDelta (0x00 -> unchanged, 0x01 -> changed)

InputValue (0x00 -> low, 0x01 -> high)

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_FALSE

5.1.35 **dio_get_io_connection_status**

-- Description --

Get connection status.

-- Syntax --

C/C+

dio_get_io_connection_status (unsigned char *pMac)

-- Arguments --

Mac

[in] Connect device port mac address.

-- Return Code --

Status	Value
Establish	0x01
Disestablish	0x00

6.1.1 ser_open

-- Description --

Open Serial port. Call this function before using any other serial function.

-- Syntax --

C/C+

```
int ser_open(unsigned char * pMac, int DibChlNumber, PSER_STATE_INFO pStateInfo);
```

-- Arguments --

pMac

[in] mac address

DibChlNumber

[in] DIB channel number

pStateInfo

[out] Serial port register state information struct pointer

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_OBJECT_NOT_FOUND STATUS_OPEN_ALREADY

6.1.2 ser_close

-- Description --

Close serial port.

-- Syntax --

C/C+

```
int ser_close(unsigned char * pMac, int DibChlNumber);
```

-- Arguments --

pMac

[in] mac address

DibChlNumber

[in] DIB channel number

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_NOT_OPEN

6.1.3 ser_setting

-- Description --

Set serial port parameters.

-- Syntax --

C/C+

```
int ser_setting(unsigned char * pMac, int DibChlNumber, unsigned char Baud,  
                unsigned char DataBits, unsigned char Parity, unsigned char StopBit,  
                unsigned char FlowControl);
```

-- Arguments --

Mac

[in] mac address

DibChlNumber

[in] DIB channel number

Baud

[in]

0x00 -> 50

0x01 -> 75

0x02 -> 110

0x03 -> 134

0x04 -> 150

0x05 -> 300

0x06 -> 600

0x07 -> 1200

0x08 -> 2400

0x09 -> 4800

0x0a -> 9600

0x0b -> 19200

0x0c -> 38400

0x0d -> 57600

0x0e -> 115200

0x0f -> 230400

0x10 -> 460800

0x11 -> 921600

DataBits

[in] 0x00 -> 5 bit

0x01 -> 6 bit

0x02 -> 7 bit

0x03 -> 8 bit

Parity

[in] 0x00 -> none

0x01 -> odd

0x02 -> even

0x03 -> mark

0x04 -> space

StopBit

[in] 0x00 -> 1

0x01 -> 1.5

0x02 -> 2

FlowControl

[in] 0x00 -> noen

0x01 -> HW(RTSCTS)

0x02 -> HW(DTRDSR)

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_NOT_OPEN

6.1.4 ser_set_fifo_ctrl

-- Description --

FIFO control (Tx & Rx trigger level).

-- Syntax --

C/C++

```
int ser_set_fifo_ctrl(unsigned char * pMac, int DibChlNumber, unsigned char  FIFO_CTRL);
```

-- Arguments --

Mac

[in] mac address

DibChlNumber

[in] DIB channel number

FIFO_CTRL

[in]

Support Uart Ver 1,2,3

Bit 0: TX Reset

Bit 1: RX Reset

Bit 2~3: Tx trigger level (4 level, default set to 00)

00 => 0, 01 => 16, 10 => 32, 11 => 64

Bit 4~5: Rx Trigger Level (4 level, default set to 11)

00 => 8, 01 => 90, 10 => 172. 11 => 255

Support Uart Ver 4,5

Bit 0: TX Reset

Bit 1: RX Reset

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_NOT_OPEN

6.1.5 ser_set_mcr

-- Description --

Set mcr.

-- Syntax --

C/C+

```
int ser_set_mcr(unsigned char * pMac, int DibChlNumber, unsigned char MCR);
```

-- Arguments --

Mac

[in] mac address

DibChlNumber

[in] DIB channel number

MCR

[in]

Bit0 -> RTS

Bit1 -> DTR

Bit2 -> OUT1

Bit3 -> OUT2

Bit4 -> Loop

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_NOT_OPEN

6.1.6 ser_set_event_ctrl

-- Description --

Set event mode.

-- Syntax --

C/C++

```
int ser_set_event_ctrl(unsigned char * pMac, int DibChlNumber, unsigned char EVENT_CTRL);
```

-- Arguments --

Mac

[in] mac address

DibChlNumber

[in] DIB channel number

EVENT_CTRL

[in]

Bit 0: Data event enable (Tx & Rx) – Set to 1 when ready to receive and transmit data.

Bit 1: MSR event enable

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_NOT_OPEN

6.1.7 ser_set_power_io

-- Description --

Set power I/O.

-- Syntax --

C/C++

```
int ser_set_power_io (unsigned char * pMac, int DibChlNumber, unsigned char MODE);
```

-- Arguments --

Mac

[in] mac address

DibChlNumber

[in] DIB channel number

MODE

[in]

Bit 0~1:

00 : RI Input

01 : RI 5V power output enable

10 : RI 12V power output enable

11 : RI power output enable (no distinguish 5V/12V select function)

Bit 2~3:

00 : DCD Input

01 : DCD 5V power output

10 : DCD 12V power output

11 : DCD power output enable (no distinguish 5V/12V select function)

Bit 4 ~ Bit 7 : Set 0

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_NOT_OPEN

6.1.8 ser_set_mode_ctrl

-- Description --

I/O Mode select.

-- Syntax --

C/C+

```
int ser_set_mode_ctrl(unsigned char * pMac, int DibChlNumber, unsigned char MODE);
```

-- Arguments --

Mac

[in] mac address

DibChlNumber

[in] DIB channel number

MODE

[in]

Bit 0: RS232 Enable

Bit 1: RS422 Enable

Bit 2: RS485 Enable

Bit 3: AHDC enable

Bit 4: CS enable

Bit 5: RS422 termination enable

Bit 6: RS485 termination enable

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_NOT_OPEN

6.1.9 ser_get_on_chip_hw_flow_ctrl

-- Description --

Get on chip hardware flow control

-- Syntax --

C/C+

```
int ser_get_on_chip_hw_flow_ctrl(unsigned char * Mac, int DibChlNumber,  
                                unsigned char * HwFlowCtrl);
```

-- Arguments --

Mac

[in] mac address

DibChlNumber

[in] DIB channel number

HwFlowCtrl

[out]

(Support Uart Ver 4,5)

0x00 : None

0x01 : HW rts/cts flow control

0x02 : HW dtr/dsr flow control

0x03 : Both

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_NOT_OPEN

6.1.10 ser_set_on_chip_hw_flow_ctrl

-- Description --

Set hardware flow control

-- Syntax --

C/C++

```
int ser_set_on_chip_hw_flow_ctrl(unsigned char * Mac, int DibChlNumber,  
                                unsigned char  HwFlowCtrl);
```

-- Arguments --

Mac

[in] mac address

DibChlNumber

[in] DIB channel number

HwFlowCtrl

[in]

(Support Uart Ver 4,5)

0x00 : None

0x01 : HW rts/cts flow control

0x02 : HW dtr/dsr flow control

0x03 : Both

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_NOT_OPEN STATUS_CONTROLLER_VERSION_UNSupport

6.1.11 ser_get_tx_trigger_level

-- Description --

Get Tx trigger level (Support Uart Ver 4,5)

-- Syntax --

C/C+

```
int ser_get_tx_trigger_level(unsigned char * Mac, int DibChlNumber,  
                             unsigned char * TxTriggerLevel);
```

-- Arguments --

Mac

[in] mac address

DibChlNumber

[in] DIB channel number

TxTriggerLevel

[out]

Tx FIFO	1KB	2KB	4KB	8KB
Data Count	0~255	(0~255)*2	(0~255)*4	(0~255)*8

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_NOT_OPEN

6.1.12 ser_set_tx_trigger_level

-- Description --

Set Tx trigger level(Support Uart Ver 4,5)

-- Syntax --

C/C+

```
int ser_set_tx_trigger_level(unsigned char * Mac, int DibChlNumber,  
                             unsigned char TxTriggerLevel);
```

-- Arguments --

Mac

[in] mac address

DibChlNumber

[in] DIB channel number

TxTriggerLevel

[in]

Tx FIFO	1KB	2KB	4KB	8KB
Data Count	0~255	(0~255)*2	(0~255)*4	(0~255)*8

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_NOT_OPEN

6.1.13 ser_get_rx_trigger_level

-- Description --

Get Rx trigger level(Support Uart Ver 4,5)

-- Syntax --

C/C+

```
int ser_get_rx_trigger_level(unsigned char * Mac, int DibChlNumber,  
                             unsigned char * RxTriggerLevel);
```

-- Arguments --

Mac

[in] mac address

DibChlNumber

[in] DIB channel number

RxTriggerLevel

[out]

Rx FIFO	1KB	2KB	4KB	8KB
Data Count	1~255	(1~255)*2	(1~255)*4	(1~255)*8

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_NOT_OPEN

6.1.14 ser_set_rx_trigger_level

-- Description --

Set Rx trigger level(Support Uart Ver 4,5)

-- Syntax --

C/C+

```
int ser_set_rx_trigger_level(unsigned char * pMac, int DibChlNumber,  
                             unsigned char RxTriggerLevel);
```

-- Arguments --

Mac

[in] mac address

DibChlNumber

[in] DIB channel number

RxTriggerLevel

[in]

Rx FIFO	1KB	2KB	4KB	8KB
Data Count	1~255	(1~255)*2	(1~255)*4	(1~255)*8

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_NOT_OPEN

6.1.15 ser_set_msr_event_callback

-- Description --

Set msr event callback function.

-- Syntax --

C/C+

```
int ser_set_msr_event_callback(unsigned char * Mac,int DibChlNumber,  
                               void (*SerSetMsrCallBack)  
                               (unsigned char * pMac, int DibChlNumber, unsigned char MsrEvent));
```

-- Arguments --

Mac

[in] mac address

DibChlNumber

[in] DIB channel number

void (*SerSetMsrCallBack)(unsigned char * pMac, int DibChlNumber, unsigned char MsrEvent)

[out] When msr event occur, library will call this function.

pMac

[out] mac address

DibChlNumber

[out] DIB channel number

MsrEvent : (1 byte) 0x??

Bit 0 -> CTS

Bit 1 -> DSR

Bit 2 -> RI

Bit 3 -> DCD

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_NOT_OPEN

6.1.16 ser_set_lsr_evnet_callback

-- Description --

Set LSR event callback function.

-- Syntax --

C/C+

```
int ser_set_lsr_event_callback(unsigned char * Mac,int DibChlNumber, void
(*SerSetLsrCallBack)( unsigned char * pMac, int DibChlNumber, unsigned char LsrEvent,
                        unsigned char CreditBit8, unsigned char Credit));
```

-- Arguments --

Mac

[in] mac address

DibChlNumber

[in] DIB channel number

```
(*SerSetLsrCallBack)( unsigned char * pMac, int DibChlNumber, unsigned char LsrEvent,
                        unsigned char CreditBit8, unsigned char Credit)
```

[out] When lsr event occur, library will call this function.

pMac

[out] mac address

DibChlNumber

[out] DIB channel number

LsrEvent :

[out] Lsr Event

0x??

Bit 0 -> Tx trigger

Bit 1 -> Rx error

Bit 2 -> Rx over flow

Bit 3 -> Tx empty

CreditBit8 :

[out]

(1 byte) 0x0?

Flow control credit bit 8

Credit :

[out]

(1 byte) 0x??

Flow control credit bit 0 ~ 7

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_NOT_OPEN

6.1.17 ser_set_data_event_callback

-- Description --

Set data event callback function.

-- Syntax --

C/C+

```
int ser_set_data_event_callback(unsigned char * Mac,int DibChlNumber, void
                                (*SerSetDataEventCallBack)( unsigned char * pMac, int DibChlNumber,
                                unsigned char * Data , int DataLength));
```

-- Arguments --

Mac

[in] mac address

DibChlNumber

[in] DIB channel number

(*SerSetDataEventCallBack)

(unsigned char * pMac, int DibChlNumber, unsigned char * Data , int DataLength)

[out] When data event occur, library will call this function.

(n byte)

0x?? Data

0x?? State

Bit 0: Frame err

Bit 1: parity err

Bit 2: Break err

Bit 3: data rdy

... continue

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_NOT_OPEN

6.1.18 ser_write

-- Description --

Write data.

-- Syntax --

C/C+

```
int ser_write (unsigned char * Mac, int DibChlNumber,  
              unsigned char * Data, int DataLength);
```

-- Arguments --

Mac

[in] mac address

DibChlNumber

[in] DIB channel number

Data

[in] Data

DataLength

[in] Data length

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_NOT_OPEN

6.1.19 **ser_set_break**

-- Description --

Force a break condition.

-- Syntax --

C/C+

```
int ser_set_break(unsigned char * pMac, int DibChlNumber, unsigned char State);
```

-- Arguments --

Mac

[in] mac address

DibChlNumber

[in] DIB channel number

State

[in]

0x00 : Disables the Break.

0x01 : Set break enable

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_NOT_OPEN

7.1.1 ctu_connect

-- Description --

Call this function before using dp configuration function.

-- Syntax --

C/C+

```
int ctu_connect(unsigned char * DpMac);
```

-- Arguments --

Mac

[in] mac address

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	MNG_STATUS_DP_NOT_FOUND CTU_STATUS_CTU_CONNECT_ALREADY

7.1.2 ctu_disconnect

-- Description --

Control unit deisconnect.

-- Syntax --

C/C+

```
int ctu_disconnect(unsigned char * DpMac);
```

-- Arguments --

Mac

[in] mac address

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	CTU_STATUS_NOT_CONNECT

7.1.3 ctu_set_device_name

-- Description --

Set device port name.

-- Syntax --

C/C+

```
int ctu_set_device_name(unsigned char * DpMac,unsigned char * DeviceName);
```

-- Arguments --

Mac

[in] mac address

DeviceName

[in] Device port Name.

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	CTU_STATUS_NOT_CONNECT

7.1.4 ctu_set_uart_setting

-- Description --

Store UART setting to flash.

-- Syntax --

C/C+

```
int ctu_set_uart_setting(unsigned char * DpMac);
```

-- Arguments --

Mac

[in] mac address

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	CTU_STATUS_NOT_CONNECT CTU_STATUS_NO_SUPPORTED_UART_VERSION_CHANNEL

7.1.5 ctu_set_dio_setting

-- Description --

Store DIO setting to flash.

-- Syntax --

C/C+

```
int ctu_set_dio_setting(unsigned char * DpMac);
```

-- Arguments --

Mac

[in] mac address

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	CTU_STATUS_NOT_CONNECT CTU_STATUS_NO_SUPPORTED_DIO_VERSION_CHANNEL

7.1.6 ctu_update_firmware

-- Description --

Update device port firmware.

-- Syntax --

C/C+

```
int ctu_update_firmware(unsigned char * DpMac , unsigned char * FilePath);
```

-- Arguments --

Mac

[in] mac address

FilePath

[in] Firmware file path.

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	CTU_STATUS_CTU_NOT_CONNECT CTU_STATUS_CTU_RX_FAIL CTU_STATUS_CTU_TX_FAIL

7.1.7 ctu_set_password

-- Description --

Update device port firmware.

-- Syntax --

C/C+

```
int ctu_set_password(unsigned char * DpMac , unsigned char pwsetting, unsigned char * pw);
```

-- Arguments --

Mac

[in] mac address

pwsetting

[in] Password setting

0x00->disable, 0x01->enable

Pw

[in] Password data(maximum 32 byte)

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	CTU_STATUS_CTU_NOT_CONNECT

8.1.0 mai_get_info

-- Description --

Get Multi-AI information.

-- Syntax --

C/C+

```
int mai_get_info(unsigned char * pMac, int DibChl, PMAI_INFO pMaiInfo);
```

-- Arguments --

pMac

[in] mac address

DibChlNumber

[in] DIB channel number

pMaiInfo

[out] Multi -AI information struct pointer

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	SIO_STATUS_OBJECT_NOT_FOUND SIO_STATUS_SIO_NOT_CONNECT

```

typedef struct _MAI_INFO
{
    unsigned char Version;
    unsigned int  FifoSize;
    unsigned char FifoType;
    unsigned char MAI_Type;
    unsigned char SupportInputSource;
    unsigned char NumInputPort; (Max 16-port per Ch.)
    unsigned char Resolution;
    unsigned char SupportInputType;
    unsigned char MaxPromGains;
    double      Verf;

    MAI_PORT_INFO MaiPortInfo[16];
}MAI_INFO,*PMAI_INFO;

```

FifoType : 0x00 dedicated buffer , 0x01 shared buffer

MAI_Type: 0x00 Type0 , 0x01 Type1

SupportInputSource :

Bit0 : Current

Bit1 : Resistor

Bit2 : RTD

Bit3 : Thermocouple

Bit4 : Thermistor

SupportInputType :

Bit0 : Unipolar (0V~Vref)

Bit1 : Bipolar (-Vref~+Vref)

Bit2 : Software Unipolar (-Vref ~ 0 ~ +Vref)

MaxPromGains :

0x00: x1 Gain(Vref)

0x01: x2 Gain(Vref)

0x02: x4 Gain(Vref)

0x03: x8 Gain(Vref)

0x04: x16 Gain(Vref)

0x05: x32 Gain(Vref)

0x06: x64 Gain(Vref)

0x07: x128 Gain(Vref)

Verf :

$V_{ref} = \text{Significand} * 10^{(-\text{exponent})}$

Byte0~2 : Significand of Vref

Byte3 : Exponent of Vref

```
typedef struct _MAI_PORT_INFO
{
    unsigned char PortIndex;
    unsigned char AiMode;
    unsigned char LogginMode;
    unsigned char TriggerMode;
    double UpperTriggerValue;
    double LowerTriggerValue;
    unsigned char Type1InputType;
    unsigned char Type1InputChannel;
    unsigned char Type1PgaEnabel;
    unsigned char Type1GainControl;
}MAI_PORT_INFO, *PMAI_PORT_INFO;
```

AiMode :

0x00 Voltage

0x01 Current

Type1InputType : 0x00 -> bipolar, 0x01 -> unipolar

Type1InputChannel : 0x00 -> AIN1, 0x01 -> AIN2

Type1PgaEnabel : 0x00 -> disable, 0x01 -> enable

Type1GainControl :

0x00: x1 Gain(Vref)

0x01: x2 Gain(Vref)

0x02: x4 Gain(Vref)

0x03: x8 Gain(Vref)

0x04: x16 Gain(Vref)

0x05: x32 Gain(Vref)

0x06: x64 Gain(Vref)

0x07: x128 Gain(Vref)

8.1.1 mai_set_sample_enable

-- Description --

Set Multi-AI sampling enable.
)

-- Syntax --

C/C+

```
int mai_set_sample_enable(unsigned char * pMac, int DibChlNumber, unsigned char State);
```

-- Arguments --

pMac

[in] mac address

DibChlNumber

[in] DIB channel number

State

[in] 0x00 -> disable

0x01 -> enable

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	SIO_STATUS_OBJECT_NOT_FOUND SIO_STATUS_DATA_INVALID SIO_STATUS_SIO_NOT_CONNECT

8.1.2 mai_set_data_enable

-- Description --

Set Multi -AI data enable.

-- Syntax --

C/C+

```
int mai_set_data_enable(unsigned char * pMac, int DibChlNumber, unsigned char State);
```

-- Arguments --

pMac

[in] mac address

DibChlNumber

[in] DIB channel number

State

[in] 0x00 -> disable

0x01 -> enable

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	SIO_STATUS_OBJECT_NOT_FOUND SIO_STATUS_DATA_INVALID SIO_STATUS_SIO_NOT_CONNECT

8.1.3 set_event_enable

-- Description --

Set Multi -AI event enable.

-- Syntax --

C/C+

```
int mai_set_event_enable(unsigned char * pMac, int DibChlNumber, unsigned char State);
```

-- Arguments --

pMac

[in] mac address

DibChlNumber

[in] DIB channel number

State

[in] 0x00 -> disable

0x01 -> enable

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	SIO_STATUS_OBJECT_NOT_FOUND SIO_STATUS_DATA_INVALID SIO_STATUS_SIO_NOT_CONNECT

8.1.4 mai_set_ports_enable

-- Description --

Set Multi -AI ports enable. (Enable ports status before get data)

-- Syntax --

C/C+

```
int mai_set_ports_enable(unsigned char * pMac, int DibChlNumber, unsigned short State);
```

-- Arguments --

pMac

[in] mac address

DibChlNumber

[in] DIB channel number

State

[in] 0 -> disable

1 -> enable

0x???? (Max 16 port per channel)

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	SIO_STATUS_OBJECT_NOT_FOUND SIO_STATUS_DATA_INVALID SIO_STATUS_SIO_NOT_CONNECT

8.1.5 mai_get_ports_enable

-- Description --

Get Multi -AI ports enable status.

-- Syntax --

C/C+

```
int mai_get_ports_enable(unsigned char * pMac, int DibChlNumber, unsigned short * State);
```

-- Arguments --

pMac

[in] mac address

DibChlNumber

[in] DIB channel number

State

[out]0 -> disable

1 -> enable

0x???? (Max 16 port per channel)

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	SIO_STATUS_OBJECT_NOT_FOUND SIO_STATUS_DATA_INVALID SIO_STATUS_SIO_NOT_CONNECT

8.1.6 mai_set_port_ai_mode

-- Description --

Set port ai mode .

-- Syntax --

C/C+

```
int mai_set_port_ai_mode(unsigned char * pMac, int DibChlNumber, int PortChl, unsigned char value);
```

-- Arguments --

pMac

[in] mac address

DibChlNumber

[in] DIB channel number

value

[in]

0x00 : Voltage(Default)

0x01 : Current

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	SIO_STATUS_OBJECT_NOT_FOUND SIO_STATUS_DATA_INVALID SIO_STATUS_SIO_NOT_CONNECT

8.1.7 mai_set_port_trigger_mode

-- Description --

Set port trigger mode .

-- Syntax --

C/C+

```
int mai_set_port_trigger_mode(unsigned char * pMac, int  DibChlNumber, int PortChl,  
                             unsigned char mode);
```

-- Arguments --

pMac

[in] mac address

DibChlNumber

[in] DIB channel number

PortChl

[in] Port channel number

mode

[in]

Bit 0 -> MFAI >= upper trigger value

Bit 1 -> MFAI <= lower trigger value

Bit 2 -> lower trigger value < MFAI < upper trigger value

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	SIO_STATUS_OBJECT_NOT_FOUND SIO_STATUS_DATA_INVALID SIO_STATUS_SIO_NOT_CONNECT

8.1.8 mai_set_port_upper_trigger_value

-- Description --

Set port upper trigger value .

-- Syntax --

C/C+

```
int mai_set_port_upper_trigger_value(unsigned char * pMac, int DibChlNumber, int PortChl,  
                                     double Value);
```

-- Arguments --

pMac

[in] mac address

DibChlNumber

[in] DIB channel number

PortChl

[in] Port channel number

Value

[in] Upper trigger value

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	SIO_STATUS_OBJECT_NOT_FOUND SIO_STATUS_DATA_INVALID SIO_STATUS_SIO_NOT_CONNECT

8.1.9 mai_set_port_lower_trigger_value

-- Description --

Set port lower trigger value.

-- Syntax --

C/C+

```
int mai_set_port_lower_trigger_value(unsigned char * pMac, int DibChlNumber, int PortChl,  
                                     double Value);
```

-- Arguments --

pMac

[in] mac address

DibChlNumber

[in] DIB channel number

PortChl

[in] Port channel number

Value

[in] Lower trigger value

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	SIO_STATUS_OBJECT_NOT_FOUND SIO_STATUS_DATA_INVALID SIO_STATUS_SIO_NOT_CONNECT

8.1.10 **mai_set_type1_input_type**

-- Description --

Set port input type.

-- Syntax --

C/C+

```
int mai_set_type1_input_type(unsigned char * pMac, int DibChlNumber, int PortChl,  
                             unsigned char type);
```

-- Arguments --

pMac

[in] mac address

DibChlNumber

[in] DIB channel number

PortChl

[in] Port channel number

type

[in]

0x00 -> Bipolar

0x01 -> Unipolar

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	SIO_STATUS_OBJECT_NOT_FOUND SIO_STATUS_DATA_INVALID SIO_STATUS_SIO_NOT_CONNECT

8.1.11 mai_set_type1_input_select

-- Description --

Set port input select.

-- Syntax --

C/C++

```
int mai_set_type1_input_select (unsigned char * pMac, int DibChlNumber, int PortChl,  
                                unsigned char value);
```

-- Arguments --

pMac

[in] mac address

DibChlNumber

[in] DIB channel number

PortChl

[in] Port channel number

type

[in]

0x00 : AIN1 = Vref (\pm Vref)

0x01 : AIN2 = 4Vref (\pm 4Vref)

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	SIO_STATUS_OBJECT_NOT_FOUND SIO_STATUS_DATA_INVALID SIO_STATUS_SIO_NOT_CONNECT

8.1.12 **mai_set_type1_pga_enable**

-- Description --

Set internal PGA enable state.

-- Syntax --

C/C+

```
int mai_set_type1_pga_enable(unsigned char * pMac, int DibChlNumber, int PortChl,  
                             unsigned char value);
```

-- Arguments --

pMac

[in] mac address

DibChlNumber

[in] DIB channel number

PortChl

[in] Port channel number

value

[in]

0x00 -> disable (default)

0x01 -> enable (when Gain Control > 0 ,set pga enable)

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	SIO_STATUS_OBJECT_NOT_FOUND SIO_STATUS_DATA_INVALID SIO_STATUS_SIO_NOT_CONNECT

8.1.13 **mai_set_event_callback**

-- Description --

Set event callback function.

-- Syntax --

C/C+

```
int mai_set_event_callback(unsigned char *Mac,int DibChlNumber,  
void(*MaiEventCallBack)(unsigned char *pMac, int DibChlNumber, int PortChl, unsigned char Event,  
double EventValue));
```

-- Arguments --

Mac

[in] mac address

DibChlNumber

[in] DIB channel number

void(*MaiEventCallBack)

(unsigned char *pMac, int DibChlNumber, int PortChl, unsigned char Event, double EventValue)

[in] When event occur, library will call this function.

(1 byte) 0x?? Event

0x01 -> MFAI >= upper trigger value

0x02 -> MFAI <= lower trigger value

0x04 -> lower trigger value < MFAI < upper trigger value

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	SIO_STATUS_OBJECT_NOT_FOUND SIO_STATUS_DATA_INVALID SIO_STATUS_SIO_NOT_CONNECT

8.1.14 mai_set_data_callback

-- Description --

Set data callback function.

-- Syntax --

C/C+

```
int mai_set_data_callback(unsigned char * Mac,int DibChlNumber,  
void (*MaiDataCallBack)(unsigned char *pMac, int DibChlNumber, int PortChl, double Data));
```

-- Arguments --

Mac

[in] mac address

DibChlNumber

[in] DIB channel number

void (*MaiDataCallBack)(unsigned char *pMac, int DibChlNumber, int PortChl, double Data)

[in] When receive data, library will call this function.

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	STATUS_OBJECT_NOT_FOUND STATUS_SERVICE_NOT_CONNECT

8.1.15 mai_get_logging_mode

-- Description --

Get port logging mode state.

-- Syntax --

C/C+

```
int mai_get_logging_mode(unsigned char * pMac, int DibChlNumber, int PortChl,  
                        unsigned char * pMode);
```

-- Arguments --

pMac

[in] mac address

DibChlNumber

[in] DIB channel number

PortChl

[in] Port channel number

pMode

[out]

0x00 -> disable

0x01 -> enable

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	SIO_STATUS_OBJECT_NOT_FOUND SIO_STATUS_DATA_INVALID SIO_STATUS_SIO_NOT_CONNECT

8.1.16 **mai_set_logging_mode**

-- Description --

Set port logging mode state.

-- Syntax --

C/C+

```
int mai_set_logging_mode(unsigned char * pMac, int DibChlNumber, int PortChl,  
                        unsigned char Mode);
```

-- Arguments --

pMac

[in] mac address

DibChlNumber

[in] DIB channel number

PortChl

[in] Port channel number

pMode

[in]

0x00 -> disable

0x01 -> enable

-- Return Code --

Status	Value
Succeed	STATUS_SUCCESS
Fail	SIO_STATUS_OBJECT_NOT_FOUND SIO_STATUS_DATA_INVALID SIO_STATUS_SIO_NOT_CONNECT

9. How to use library

Step 1.

Please login as root and install SUNIX Device Port Library to system, section 3 installation is the reference.

Step 2.

Copy `sdxmng.h` and `sdxsio.h` to your project and include in your source code.

Step 3.

In project Makefile, add ``-L -lsdxio``,

10. Return Code

MNG_STATUS_SUCCESS	0x00
MNG_STATUS_PARAMETER_INVALID	0x01
MNG_STATUS_NIC_NOT_FOUND	0x10
MNG_STATUS_NIC_REFRESH_FAIL	0x11
MNG_STATUS_NIC_TX_FAIL	0x12
MNG_STATUS_DP_NOT_FOUND	0x20
MNG_STATUS_DP_KEEPALIVE_START_FAIL	0x21
MNG_STATUS_DP_CTU_SERVER_START_FAIL	0x22
MNG_STATUS_DP_SIO_SERVER_START_FAIL	0x23
MNG_STATUS_DP_MAPPING_ALREADY	0x30
MNG_STATUS_DP_NOT_MAPPING	0x31
MNG_STATUS_DP_NO_IO_FOR_MAPPING	0x32
MNG_STATUS_DP_CREATE_DEVICE_NODE_FAIL	0x33
MNG_STATUS_DP_DELETE_DEVICE_NODE_FAIL	0x34
MNG_STATUS_DP_ALLOC_TTY_MINOR_NUMBER_FAIL	0x35
MNG_STATUS_DP_CONNECT_ALREADY	0x40
MNG_STATUS_DP_CONNECT_TIMEOUT	0x41
MNG_STATUS_DP_CONNECT_BUSY	0x42
MNG_STATUS_DP_CONNECT_UNSUPPORT_VERSION	0x43
MNG_STATUS_DP_CONNECT_FAIL	0x44
MNG_STATUS_DP_CONNECT_PASSWD_INCORRECT	0x45
MNG_STATUS_DP_DISCONNECT_ALREADY	0x46
MNG_STATUS_DP_DISCONNECT_TIMEOUT	0x47
MNG_STATUS_DP_DISCONNECT_BUSY	0x48
MNG_STATUS_DP_DISCONNECT_UNSUPPORT_VERSION	0x49
MNG_STATUS_DP_DISCONNECT_FAIL	0x4a
MNG_STATUS_DP_DISCONNECT_PASSWD_INCORRECT	0x4b
MNG_STATUS_DP_NOT_CONNECT	0x4c
MNG_STATUS_DP_LIB_INIT_ALREADY	0x50
MNG_STATUS_GET_SERVICE_STRING_FAIL	0x51
MNG_STATUS_WRONG_PROTOCOL_VERSION	0x52
MNG_STATUS_WRONG_PACKET_RECEIVED	0x53
MNG_STATUS_EXTRA_DATA_LENGTH_INVALID	0x54
MNG_STATUS_CONNECT_SERVICE_FAIL	0x62
MNG_STATUS_SOCKET_TX_FAIL	0x63

MNG_STATUS_SOCKET_RX_FAIL	0x64
MNG_STATUS_SOCKET_RX_TIMEOUT	0x65
STATUS_FUNCTION_UNSupport	0x0001
STATUS_LENGTH_INVALID	0x0002
STATUS_DATA_INVALID	0x0003
STATUS_CONTROLLER_VERSION_UNSupport	0x0004
STATUS_DIO_NOT_FOUND	0x0010
STATUS_BANK_NOT_FOUND	0x0011
STATUS_PORT_NOT_FOUND	0x0012
STATUS_WRONG_TCP_PORT_NUMBER	0x0013
STATUS_TX_CMD_FAIL	0x0020
STATUS_RX_RESPONSE_TIMEOUT	0x0021
STATUS_RX_RESPONSE_DATA_FORMAT_INVALID	0x0022
STATUS_RX_RESPONSE_WITH_NO_DATA	0x0023
STATUS_OPEN_ALREADY	0x0030
STATUS_NOT_OPEN	0x0031
STATUS_UNSupport_MODE_CAPABILITY	0x0032
STATUS_SIO_NOT_CONNECT	0x0050
STATUS_CONNECT_FAIL	0x0051
STATUS_NO_INPUT_IO	0x0070
STATUS_NO_OUTPUT_IO	0x0071
STATUS_SUPPORT_WHOLE_BANK_INPUT_OR_OUTPUT	0x0072
STATUS_UNSupport_INPUT_CAPABILITY	0x0073
STATUS_UNSupport_OUTPUT_CAPABILITY	0x0074
STATUS_DIRECTION_CAPABILITY_UNKNOW	0x0075
CTU_STATUS_OBJECT_NOT_FOUND	0x0010
CTU_STATUS_UNSupport_COMMAND	0x0030
CTU_STATUS_SET_PASSWORD_FAIL	0x0031
CTU_STATUS_NO_SUPPORTED_UART_VERSION_CHANNEL	0x0032
CTU_STATUS_NO_SUPPORTED_DIO_VERSION_CHANNEL	0x0033
CTU_STATUS_FIRMWARE_FILE_OPEN_FAIL	0x0040
CTU_STATUS_FIRMWARE_FILE_READ_FAIL	0x0041
CTU_STATUS_FIRMWARE_FILE_NO_CONTENT	0x0042
CTU_STATUS_FIRMWARE_MEMORY_MALLOC_FAIL	0x0043
CTU_STATUS_FIRMWARE_SIZE_INVALID	0x0044
CTU_STATUS_FIRMWARE_MODEL_NOT_MATCH	0x0045

CTU_STATUS_FIRMWARE_DP_TYPE_NOT_MATCH	0x0046
CTU_STATUS_FIRMWARE_CHECKSUM_NOT_MATCH	0x0047
CTU_STATUS_FIRMWARE_UNSUPPORT_DOWNGRADE	0x0048
CTU_STATUS_FIRMWARE_SET_TOTAL_LENGTH_FAIL	0x0049
CTU_STATUS_FIRMWARE_DOWNLOAD_OK	0x004a
CTU_STATUS_FIRMWARE_DOWNLOAD_FAIL	0x004b
CTU_STATUS_DISCONNECT_FAIL	0x0050
CTU_STATUS_NOT_CONNECT	0x0051

11. Release Note

2018/03/07

Ver 1.0.0.0

2018/03/20

Ver 1.0.1.0

1. Add get service connection status function.
2. Add Device Port Mapping/Unmapping function.
3. Change get Device Port detail info function.

2018/05/02

Ver 1.0.2.0

1. Add set io port state function.
2. Add set io port event callback function.

2018/06/19

Ver 1.0.3.0

1. IO support UART
2. Support Control unit function.

2018/09/03

Ver 1.0.4.0

1. Support Multi-function AI.

2019/09/11

Ver 1.0.5.0

1. Fix serial port config issue.