

SPM Software Reference for Windows 2000

Copyright © 1999 - 2000 Seoul Commtech Corporation

Table of Contents

1. SPM Introduction	3
1.1.	3
1.2.	4
1.3.	4
1.4.	4
1.5.	4
1.6.	5
2. SPM Library Function	6
2.1. Library Function	6
2.1.1	
2.1.2	
2.1.3	
2.1.4 Device	
2.1.5 Station	
2.2.	7
2.3.	7
2.4. Include Files	8
3. SPM Function Reference	9
3.1. Documentation Conventions	9
4. SPM Application	86
4.1.	86
4.1.1 Symbolic Define	
4.1.2 Header File Include	
4.1.3 Check Return Codes	
4.2.	88
4.2.1 H/W	
4.2.2 Set event mask on SPM stations	
4.2.3	
4.3. Compiling and Linking	89
4.4	89
Appendix A	90

1. SPM Introduction

SPM Product Overview

SPM	Interface	VPM	SCbus
SPM	Inbound/Outbound	Call Center	
SPM	8 (CT-S08A)	16 (CT-S16A)	가
SPM	Power Module		BLT Circuit
Configuration	Shared RAM	Interface	

1.1

SPM Application program SCbus device
가 .

~~SC~~ Inbound/Outbound telemarketing

~~SC~~

~~SC~~

~~SC~~ (ACD)

~~SC~~ Dictation/transcription

~~SC~~ Local information services

SCbus

~~SC~~ 8 6 Group

~~SC~~ SCbus Resource Conference

~~SC~~ Coach

~~SC~~ Monitoring

Station Interface

~~SC~~ Ring Generator

~~SC~~ Hook ON/OFF detection

~~SC~~ DTMF (4)

~~SC~~ Tone Generator

1.2

SPM Board VPM SCbus
- 16가 Ring Cadency
- 16가 Direct Tone Generator가 가 .

1.3

SPM .

Monitor mode Monitoring .

Coach 가
.

Pupil Conference
. Pupil coach .

NOTE: Coach Pupil 가 .

1.4

Coach가 .

NOTES: 1. nr_scroute() .
2. Ap . SPM S/W

1.5

SPM B'd 8 6Group Conference Resource가
가 resource 가 .

spm_estconf() Group .
spm_addtoconf() Group party가 .
spm_remfromconf() party .
spm_delconf() Group .

spm_monconf()	party가	.
spm_unmonconf ()	party가	.

NOTES: 1. Party Channel resource .

2. Monitor party .

3. Coach & Pupil Conference Group

Group .

1.6

SPM SCbus Interface Module, ISA BUS Interface Module Conference Tone
Generator/Tone Detection ASIC .

2. SPM Library Function

2.1 Library Function

SPM Windows 2000 Library SPM Boards .
Library Function 3 .

SPM library function .

: device-specific 가 .
:
device : open/close device
: Interface

NOTE: SPM library function Sync/Async mode가 가 .

SPM SCbus Routing Function SCbus Routing Function Reference

2.1.1.

spm_confgrpcount()	: 가 Group
--------------------	-----------

2.1.2.

spm_addtoconf()	: party 가
spm_delconf()	:
spm_estconf()	:
spm_getcde()	: party table
spm_getcnflist()	: 가 list
spm_monconf()	: monitoring 가
spm_remfromconf()	: party
spm_setcde()	: party table
spm_unmonconf()	: monitoring

2.1.3.

spm_getbrdparm()	: board parameter
spm_getevt()	: event
spm_getevtmsk()	: station event mask
spm_setbrdparm()	: board parameter
spm_setevtmsk()	: station event mask

2.1.4. device

spm_close()	: closes spm device
spm_open()	: opens spm device
spm_stopfn()	: multitasking function

2.1.5. Station

spm_genring()	: station	ring
spm_gentone()	: station	tone
spm_stoptone()	: station	tone
spm_getdtmf()	: station	DTMF
spm_stopdtmf()	: station	DTMF

2.2.

SPMX_DNLDVER()	: firmware
SPMX_STATINFO()	: board
SPMX_TSSGBIT()	: station Hook

2.3.

SPM library function

0

NOTES:	1. <code>spm_open()</code>	error	device handle
		<code>AT_FAILURE</code>	.
	2. <code>ATDV_LASTERR()</code>	<code>ATDV_ERRMSGP()</code>	error
			.

```
Function prototype      smartsdk.h      .
                        Include          .
```

error include 가 .

8

3. SPM Function Reference

SPM board Interface library function .
 Standard Attribute function Appendix A . SCbus Routing Function
 SCbus Routing Function Reference for Windows 2000 .

3.1 Documentation Conventions

function 가 .

Function Header function header , syntax,
 parameter, return, includes, category, mode(Syncs/Async)가
 .
 parameter .
 .
 coding .
 function error .
 function .

SPMX_DNLDVER()

returns the SPM firmware version

Name : long SPMX_DNLDVER (devh)

Inputs :

int devh: SPM board device handle

Returns : MSI/SC firmware version

AT_FAILURE on failure

Includes: smartsdk.h

Category:

Mode: synchronous



SPMX_DNLDVER()	devh	open	SPM	F/W	version	.
----------------	------	------	-----	-----	---------	---

Parameter	.
-----------	---

devh: spm_open



device handle	.
---------------	---

Example

```
#include <windows.h>          /* For Windows 2000 applications only */
```

```
#include <errno.h>
```

```
#include "smartsdk.h"
```

```
/* Basic error handler */
```

```
do_error( devh, funcname )
```

```
int devh;
```

```
char *funcname;
```

```
{
```

```
int errorval = ATDV_LASTERR( devh );
```

```
printf( "Error while calling function %s.\n", funcname );
```

```

printf( "Error value = %d.", errorval );
    printf( "\n" );
}
main()
{
    int bddev;      /* Board device descriptor variable */
    long version;   /* Version number of firmware */
    /*
    * Open board 1 device
    */
    if ( ( bddev = ms_open( "spmB1", 0 ) ) == -1 ) {
        printf( "Cannot open board spmB1. errno = %d", errno );
        exit( 1 );
    }
    /*
    * Get the version number of the firmware
    */
    version = SPMX_DNLDVER( bddev );
    if ( version == AT_FAILURE ) {
        do_error( bddev, "ATMS_DNLDVER(?)" );
        exit( 1 );
    }
    /*
    * Display it
    */
    printf( "SPM Download version number is %d.%02x\n",
            (int)((version>>24L)&0x0F), ((version >>16L)&0xFF) );
    /*
    * Continue processing
    *
    *
    *
    */

    /* Done processing - close device. */
    if ( spm_close( bddev ) == -1 ) {

```

```
    printf( "Cannot close board msiB1. errno = %d", errno );  
  }  
}
```

□ **Error**

가 , AT_FAILURE ATDV_LASTERR() error
.

SPMX_ STATINFO()

returns information about the SPM board

Name : long SPMX_ STATINFO(devh)

Inputs :

int devh: SPM board device handle

Returns : station information

AT_FAILURE on failure

Includes: smartsdk.h

Category:

Mode: synchronous

□

SPMX_ STATINFO() SPM board .

Parameter .

devh: spm_open

□

device handle .

□ **Example**

```
#include <windows.h>                    /* For Windows 2000 applications only */
```

```
#include <errno.h>
```

```
#include "smartsdk.h"
```

```
int i;
```

```
int devh;                                /* Board device handle */
```

```
int chcnt;
```

```
/* Open board 1, device */
```

```
if ((devh = spm_open("spmB1",0)) == -1) {
```

```
    printf( "Cannot open SPM B1, errno=%d", errno);
```

```
    exit(1);
```

```

}
/*
 * Continue processing
 */
/* Get board Ids and number of stations */
if ((chcnt = SPMX_ STATINFO(devh)==-1){
    printf("Error getting station info\n");
    /* Close device and exit */
}

printf("Number of stations = %d\n", chcnt );

/* Done processing - close device */
if (spm_close(devh) == -1) {
    printf("Cannot close device spmB1. errno = %d", errno);
    exit(1);
}

```

□ Error

가 , AT_FAILURE ATDV_LASTERR() error

SPMX_TSSGBIT()

retrieves the current station hook status

Name : long SPMX_TSSGBIT(devh)

Inputs :

int devh: SPM board device handle

Returns : state of channel

AT_FAILURE on failure

Includes: smartsdk.h

Category:

Mode: synchronous

□

SPMX_TSSGBIT()

hook

.

Parameter

.

devh: spm_open

bitmask

.

MS_ONHOOK

SPM station

on-hook

.

MS_OFFHOOK

SPM station

off-hook

.

□

device handle

.

□ Example

```
#include <windows.h>          /* For Windows 2000 applications only */
```

```
#include <errno.h>
```

```
#include "smartsdk.h"
```

```
/* Basic error handler */
```

```
do_error( devh, funcname )
```

```
int devh;
```

```
char *funcname;
```

```

{
    int errorval = ATDV_LASTERR( devh );

    printf( "Error while calling function %s.\n", funcname );
    printf( "Error value = %d.", errorval );
    printf( "\n" );
}

main()
{
    int tsdev;          /* Station device descriptor variable */
    long tsbits;        /* Time slot signaling bits */

    /*
     * Open board 1 channel 1 device
     */
    if ( ( tsdev = spm_open( "spmB1C1", 0 ) ) == -1 ) {
        printf( "Cannot open station spmB1C1. errno = %d", errno );
        exit( 1 );
    }

    /*
     * Get station signaling bits
     */
    tsbits = SPMX_TSSGBIT( tsdev );
    if ( tsbits == -1 ) {
        do_error( tsdev, " SPMX_TSSGBIT(?) );
        exit( 1 );
    }

    switch( tsbits ) {
        case MS_ONHOOK:
            /* continue processing (on-hook) */
            break;
        case MS_OFFHOOK:
            /* continue processing (off-hook) */

```



```

        break;
default:
    printf("undefined parameter value = %d\n", tsbits);
    break;
}

/*
 * Continue processing
 *
 *
 *
 */

/* Done processing - close device. */
if ( spm_close( tsdev ) == -1 ) {
    printf( "Cannot close station spmB1C1. errno = %d", errno );
}
}

```

□ Error

가 , AT_FAILURE ATDV_LASTERR() error

spm_addtoconf()

adds one party to an existing conference

Name : int spm_addtoconf(devh,confID,cdt)

Inputs :

int devh: SPM board device handle

int confID: conference identifier

CONFDESC *cdt: pointer to conference descriptor table

Returns : 0

AT_FAILURE on failure

Includes: smartsdk.h

Category: Conference Management

Mode: synchronous

□

spm_addtoconf() conference group party 가 .
Conference ID conference group .

Parameter .

devh: ~~spm_open~~

confID: conference group ID

cdt: Pointer to the conference descriptor table

NOTES: 1. party 가 .

2. 가 가 conference party가 .

CONFDESC format 가 .

typedef struct {

int chan_num; /* channel/time slot number */

int chan_sel; /* channel/time slot number */

int chan_attr; /* */

int chan_lts /* time slot number */

```
} CONFDESC;
```

```
chan_num    chan_sel        가 .
```

```
// MSPN_STATION    SPM        board        port
```

```
// MSPN_TS        SCbus time slot number
```

```
Chan_attr    party        bitmask        가 .
```

```
// MSPA_NULL        No special attributes for party
```

```
// MSPA_RO        receive-only Party
```

```
// MSPA_COACH        Party is a coach. Coach heard by pupil only
```

```
// MSPA_PUPIL        Party is a pupil. Pupil hears everyone including coach
```

NOTES: 1. coach pupil Group 가 .

2. MSPA_NULL channel .

3. SCbus time slot conference time slot chan_Its
. chan_Its NSPN_STA .

□

```
// device handle
```

```
// Conference group 8party
```

```
// Conference group ID가
```

□ Example

```
#include <windows.h> /* For Windows 2000 applications only */
```

```
#include <errno.h>
```

```
#include "smartsdk.h"
```

```
#define NUM_PARTIES 2
```

```
int dev1; /* Board dev descriptor variables */
```

```
int chdev2; /* Channel dev descriptor */
```

```
int tsdev1, tsdev2; /* Time slot dev desc */
```

```
CONFDESC cdt[NUM_PARTIES]; /* Conf. desc. table */
```

```

int  confID;                /* Conf. ID */
SC_TSINFO  tsinfo;          /* Time slot info */
int  ts1, ts2;              /* SCbus time slots */
int  station;               /* Station number */

/* Open board 1 device */
if ((dev1 = spm_open("spmB1",0)) == -1) {
    printf( "Cannot open SPMB1: errno=%d", errno);
    exit(1);
}

/* Open board 1, channel 2 device */

if ((chdev2 = ms_open("spmB1C2",0) == -1) {
    printf("Cannot open spmB1C2. errno = %d", errno);
    exit(1);
}

/* VPM channel          open          */
/* vpm_getxmitslot()      SCbus time slots  */
/*          71          */

/* Set up CDT structure */
cdt[0].chan_num = station ; /* station is a valid station number */
cdt[0].chan_sel = MSPN_STATION;
cdt[0].chan_attr = MSPA_NULL;

/* SCbus time slot to be conferenced */
cdt[1].chan_num = ts1 ;      /* ts1 should be a valid time slot */
cdt[1].chan_sel = MSPN_TS;
cdt[1].chan_attr = MSPA_NULL;

/* Establish conference */
if (spm_estconf(dev1, cdt, NUM_PARTIES, MSCA_ND, &confID) == -1) {
    printf("Error Message = %s",ATDV_ERRMSGP(dev1));
}

```

```

        exit(1);
    }

    /* Do a listen for the TS */
    tsinfo.sc_numts = 1;
    tsinfo.sc_tsarray = &cdt[1].chan_lts;

    if (vag_listen(tsdev1, &tsinfo) == -1){
        printf("Error Message = %s",ATDV_ERRMSGP(tsdev1));
        exit(1);
    }

    /* Continue processing */
    /* Add another party to conference */
    cdt[0].chan_num = ts2;      /* ts2 should be a valid time slot */
    cdt[0].chan_sel = MSPN_TS;
    cdt[0].chan_attr = MSPA_RO|MSPA_TARIFF;

    if (spm_addtoconf(dev1, confID,&cdt[0]) == -1) {
        printf("Error Message = %s",ATDV_ERRMSGP(dev1));
        exit(1);
    }

    /* Do a listen for the TS */

    tsinfo.sc_numts = 1;
    tsinfo.sc_tsarray = &cdt[0].chan_lts;

    if (vag_listen(tsdev2, &tsinfo) == -1){
        printf("Error Message = %s",ATDV_ERRMSGP(tsdev2));
        exit(1);
    }

    /* Continue processing */

```

□ Error

가 , AT_FAILURE ATDV_LASTERR() error

□

~~del~~ ~~del~~ spm_delconf()

~~del~~ ~~del~~ spm_estconf()

~~del~~ ~~del~~ spm_monconf()

~~del~~ ~~del~~ spm_remfromconf()

~~del~~ ~~del~~ spm_unmonconf()

closes the MSI/SC device

Inputs :

Returns : 0

Includes: smartsdk.h

Mode: synchronous

```
spm_close()      spm_open      open      device      close      . Device      board
channel          ,              call process device link      .
```

■

spm_open

1. device handle .
2. event notification sr_dishdlr() call .
3. SPM configuration .
4. device close() close .

```
#include <windows.h>           /* For Windows 2000 applications only */
#include <errno.h>
#include "smartsdk.h"
```

{

```
/* Board device descriptor variable */
```

```

/* Open board 1 device */
if ( ( bddev = spm_open( " spmB1", 0 ) ) == -1 ) {
    printf( "Cannot open board spmB1. errno = %d\n", errno );
    exit( 1 );
}

/*
 *   Continue processing
 *       .
 *       .
 *       .
 */

/* Done processing - close device */
if ( spm_close( bddev ) == -1 ) {
    printf( "Cannot close board spmB1. errno = %d", errno );
}
}

```

□ Error

가 , AT_FAILURE ATDV_LASTERR() error

□

~~스스~~ spm_open()

spm_delconf()

deletes a conference

Name : int spm_delconf(devh,confID)

Inputs :

int devh: SPM board device handle

int confID: conference identifier

Returns : 0

AT_FAILURE on failure

Includes: smartsdk.h

Category: Conference Management

Mode: synchronous

□

spm_delconf()	conference group	. Conference ID
---------------	------------------	-----------------

spm_estconf()	.
---------------	---

Parameter	.
-----------	---

devh: spm_open

confID: spm conference identifier

NOTES: 1. conference group party resource free .

□ **Example**

```
#include <windows.h> /* For Windows 2000 applications only */
```

```
#include <errno.h>
```

```
#include "smartsdk.h"
```

```
#define NUM_PARTIES 3
```

```
int dev1; /* Board dev descriptor variables */
```

```
CONFDESC cdt[NUM_PARTIES]; /* Conf. desc. table */
```

```
int confID; /* Conf. ID */
```

```

/* Open board 1 device */

if ((dev1 = spm_open("spmB1",0)) == -1) {
    printf( "Cannot open SPM B1: errno=%d", errno);
    exit(1);
}

/*
 * Continue processing
 */

/* Set up CDT structure */
/* station 2, 4 and 7 are used to establish a conference */
cdt[0].chan_num = 2;
cdt[0].chan_sel = MSPN_STATION;
cdt[0].chan_attr = MSPA_NULL;

cdt[1].chan_num = 4;
cdt[1].chan_sel = MSPN_STATION;
cdt[1].chan_attr = MSPA_PUPIL;

cdt[2].chan_num = 7;
cdt[2].chan_sel = MSPN_STATION;
cdt[2].chan_attr = MSPA_COACH;

/* Establish conference */
if (spm_estconf(dev1, cdt, NUM_PARTIES, MSCA_ND, &confID) != 0) {
    printf("Error Message = %s",ATDV_ERRMSGP(dev1));
    exit(1);
}

/*
 * Continue processing
 *
 */
if (spm_delconf(dev1, confID) == -1){
    printf("Error Message = %s",ATDV_ERRMSGP(dev1));

```

```

        exit(1);
    }
    /* Continue processing */

```

Error

가 , AT_FAILURE ATDV_LASTERR() error

 .

~~스피어~~ spm_addtoconf()

~~스피어~~ spm_estconf()

~~스피어~~ spm_monconf()

~~스피어~~ spm_remfromconf()

~~스피어~~ spm_unmonconf()

spm_confgrpcount()

returns the available DSP resource count

Name : int spm_confgrpcount(devh,valuep)

Inputs :

int devh: SPM board device handle

int *valuep: pointer to the memory location to receive the free conference group

Returns : 0

AT_FAILURE on failure

Includes: smartsdk.h

Category: Attribute

Mode: synchronous

□

spm_confgrpcount() 가 conference group ..

Parameter

devh: ~~spm_open~~

valuep: Pointer to the free resource count

SPM Board 6 conference group .

□

device handle .

□ Example

```
#include <windows.h>                      /* For Windows 2000 applications only */
```

```
#include <errno.h>
```

```
#include "smartsdk.h"
```

```
int dev1;                      /* Board dev descriptor variables */
```

```
int valuep;                      /* Resource count */
```

```

/* Open board 1 device */
if ((dev1 = spm_open("msiB1",0)) == -1) {
    printf( "Cannot open SPM B1: errno=%d", errno);
    exit(1);
}

/* Get conference group resource count */
if (spm_confgrpcount(dev1, &valuep) == -1) {
    printf("Error Message = %s",ATDV_ERRMSGP(dev1));
    exit(1);
}

printf("Free group resource count = %d\n", valuep);
/*
 * Continue processing
 *
 */

if (spm_close(dev1)== -1){
    printf( "Cannot Close SPMB1: errno=%d", errno);
    exit(1);
}

```

Error

가 , AT_FAILURE ATDV_LASTERR() error

~~del~~ spm_delconf()
 ~~del~~ spm_estconf()

spm_estconf()

establishes a conference

Name : int spm_estconf(devh,cdt,numpty,confattr,confID)

Inputs :

int devh: SPM board device handle

CONFDESC *cdt: pointer to conference descriptor table

int numpty: number of parties in a conference

int confattr: conference attributes

int *confID: pointer to memory location to receive the conference identifier

Returns : 0

AT_FAILURE on failure

Includes: smartsdk.h

Category: Conference Management

Mode: synchronous

□

spm_estconf() conference party descriptor conference group

.

Parameter

.

devh: spm_open

cdt: The pointer to the conference descriptor table

numpty: cdt party

confattr:

confID: conference ID Pointer

conference descriptor CONFDESC Array .

CONFDESC format 가 .

typedef struct {

int chan_num; /* channel/time slot number */

int chan_sel; /* channel/time slot number */

int chan_attr; /* */

```

        int chan_lts          /*          time slot number */
    } CONFDESC;
    chan_num    chan_sel          가
    ✂✂ MSPN_STATION    SPM          board          port
    ✂✂ MSPN_TS          SCbus time slot number

    Chan_attr    party          bitmask          가
    ✂✂ MSPA_NULL          No special attributes for party
    ✂✂ MSPA_RO          receive-only Party
    ✂✂ MSPA_COACH          Party is a coach. Coach heard by pupil only
    ✂✂ MSPA_PUPIL          Party is a pupil. Pupil hears everyone including coach

```

NOTES: 1. station party chan_lts .

□

```

✂✂          device handle
✂✂ board          6          conference group

```

□ Example

```

#include <windows.h>
#include <errno.h>
#include "smartsdk.h"

#define          NUM_PARTIES          3

int  dev1;          /* Board dev descriptor variables */
int  chdev1,chdev2;          /* Channel dev descriptor */
CONFDESC cdt[NUM_PARTIES];          /* Conf. desc. table */
int  confID;          /* Conf. ID */
int  ts1, ts2;

/* Open board 1 device */
if ((dev1 = spm_open("spmB1",0)) == -1) {
    printf( "Cannot open SPM B1: errno=%d", errno);
}

```

```

        exit(1);
    }

    /* VPM channel          open          */
    /* vpm_getxmitslot()      SCbus time slots  */
    /*                      7|      */

    /*
    * Continue processing
    */

    /* Set up CDT structure */
    /* Include station 2 on MSI board in conference */
    cdt[0].chan_num = 2;
    cdt[0].chan_sel = MSPN_STATION;
    cdt[0].chan_attr = MSPA_NULL;

    cdt[1].chan_num = ts1;
    cdt[1].chan_sel = MSPN_TS;
    cdt[1].chan_attr = MSPA_PUPIL;

    cdt[2].chan_num = ts2;
    cdt[2].chan_sel = MSPN_TS;
    cdt[2].chan_attr = MSPA_COACH;

    /* Establish conference */
    if (spm_estconf(dev1, cdt, NUM_PARTIES, MSCA_ND, &confID) != 0) {
        printf("Error Message = %s", ATDV_ERRMSGP(dev1));
        exit(1);
    }

    tsinfo.sc_numts = 1;
    tsinfo.sc_tsarray = &cdt[1].chan_lts;

    /* Now, listen to TS */
    if (vag_listen(tsdev1, &tsinfo) == -1){

```



```

        printf("Error Message = %s",ATDV_ERRMSGP(tsdev1));
        exit(1);
    }

    /* Do a listen for cdt[2] */
    /* Set up SC_TSINFO structure for SCbus tslot */
    tsinfo.sc_tsarray = &cdt[2].chan_lts;

    /* Now, listen to TS */
    if (vag_listen(tsdev2,&tsinfo) == -1) {
        printf("Error Message = %s",ATDV_ERRMSGP(tsdev2));
        exit(1);
    }

    /*
     * Continue processing
     */

    if (spm_delconf(dev1, confID) == -1){
        printf("Error Message = %s",ATDV_ERRMSGP(dev1));
        exit(1);
    }

    /* Continue processing */

```

Error

가 , AT_FAILURE ATDV_LASTERR() error

~~스~~spm_addtoconf()
~~스~~spm_delconf()
~~스~~spm_remfromconf()

spm_genring()

generates ringing to a station

Name : int spm_genring(devh,len,ringtype,mode)

Inputs :

- int devh: SPM board device handle
- unsigned short len: length in cycles for ring
- unsigned short ringtype: ring cadency type
- unsigned short mode: asynchronous/synchronous

Returns : 0 for asynchronous
>0 for synchronous
AT_FAILURE on failure

Includes: smartsdk.h

Category: Station

Mode: synchronous/ asynchronous

spm_genring()

len

ring

ring

가 off-hook

Parameter

devh: spm_open

len: ring , 255

:100ms

Ringtype	On	Off	On	Off
0	10	20	10	20
1	10	30	10	30
2	2	2	2	2
3	4	2	4	40
4	4	2	4	2
5	4	2	4	20
6	10	10	10	10
7	30	10	30	10
8	4	2	4	30

9	2	2	2	30
10	4	2	4	2
11	10	40	10	40
12	6	2	2	40
13	3	4	3	20
14	3	4	3	20
15	3	4	3	30

mode: sync :
error/off-hook/ring count return .
async :
return event가 .

NOTES: 1. ring count 2 .
ring .
2. spm_genring off-hook E_MSBADRNGSTA error
return .
3. / return MSEV_NORING event가 .

Async AT_FAILURE error .
Ring MSEV_RING event .

~~MSMM_RNGOFFHK~~ offhook
~~MSMM_TERM~~ ring

Sync .

~~MSMM_RNGOFFHK~~ offhook
MSMM_TERM ring

□

~~device handle~~

□ Example

```
#include <windows.h>
#include <errno.h>
#include "smartsdk.h"

int dev1;          /* Station device descriptor */
int rc;            /* Return code */

/* Open board 1, station 1 device */
if ((dev1 = spm_open("spmB1C1",0)) == -1) {
    printf( "Cannot open SPMBC11, station 1,channel 1: errno=%d", errno);
    exit(1);
}

/*
 * Continue processing
 */
/* Generate ringing for 10 cycles in sync mode*/
if ((rc =spm_genring(dev1,10,0,EV_SYNC)) == -1) {
    /* process error */
}
/* If timeout, process the condition */
if (rc=MSMM_TERM) {
    printf("Station not responding");
}
/*
 * Continue Processing
 */

/* Done processing - close device */
if (spm_close(dev1) == -1) {
    printf("Cannot close device spmB1C1. errno = %d", errno);
    exit(1);
}
```

Asynchronous mode:

```
#include <windows.h>          /* For Windows 2000 applications only */
#include <errno.h>
#include "smartsdk.h"

int  dev1;                    /* Station dev descriptor */
int  srlmode;                 /* SRL mode indicator */

/* Open board 1, station 1 device */
if ((dev1 = spm_open("spmB1C1",0)) == -1) {
    printf( "Cannot open SPMB1C1, station 1,channel 1: errno=%d", errno);
    exit(1);
}

/* Set up handler function to handle play completion */
if (sr_enbhdr(dev1,MSEV_RING,sig_hdr) == -1) {
    /* process error */
}

/*
 * Continue processing
 */
/* Generate ringing */
if (spm_genring(dev1,10,0,EV_ASYNC) == -1) {
    printf("Error could not set up ringing. Errno = %d", errno);
    exit(1);
}

/* On receiving the completion event, MSEV_RING, control is
   transferred to the handler function previously established
   using sr_enbhdr(?).
 */

/*
```

```

*   Continue Processing
*/

/* Done processing - close device */
if (spm_close(dev1) == -1) {
    printf("Cannot close device spmB1C1. errno = %d", errno);
    exit(1);
}
/*
* Continue processing
*/
int sig_hdlr()
{
    int dev = sr_getevtdev();
    unsigned short *sigtype = (unsigned short *)sr_getevtdatap();

    if (sigtype != NULL) {
        switch (*sigtype) {
            case MSMM_TERM:
                printf("Station does not answer");
                return 0;

            case MSMM_RNGOFFHK:
                printf("Station offhook detected\n");
                return 0;

            default:
                return 1;
        }
    }
}

/*
* Continue processing
*/
}

```

❑ Error

가 , AT_FAILURE ATDV_LASTERR() error

.

❑

~~스피~~ spm_setevtmsk()

spm_gentone()

generates a zip tone

Name : int spm_gentone(devh,tonetype)

Inputs :

int devh: SPM board device handle

int tonetype: generation tone type

Returns : 0

 AT_FAILURE on failure

Includes: smartsdk.h

Category: Station

Mode: synchronous

□

spm_gentone() tone frequency cadency tone type .

Parameter .

devh: spm_open

tone type: tone type

Ringtype	Cadency(50ms)				Frequency
0	20	5	20	5	350+440
1	10	2	10	2	440+480
2	10	10	10	10	480+620
3	0	0	0	0	350+440
4	2	2	2	30	440+480
5	2	98	2	98	440+480
6	4	4	4	68	350+440
7	2	2	2	2	480+620
8	4	4	4	28	350+440
9	20	40	20	40	350+440
10	0	0	0	0	-

□

~~스스~~ device handle

~~스스~~ tone type

□ Example

```
#include <windows.h>
#include <errno.h>
#include "smartsdk.h"

int chdev1; /*Station dev descriptor variable */
/* Open station 1 device */
if ((chdev1 = spm_open("spmB1C1",0)) == -1) {

    printf( "Cannot open SPMB1C1: errno=%d", errno);
    exit(1);
}

/* Generate Ziptone */
if (spm_gentone(chdev1,0) == -1){
    printf("Error Message = %s",ATDV_ERRMSGP(chdev1));
    exit(1);
}

/* Close station 1 */
if ( spm_close(chdev1)) == -1) {
    printf( "Cannot Close SPMB1C1: errno=%d", errno);
    exit(1);
}
```

□ Error

가 , AT_FAILURE ATDV_LASTERR() error

spm_getbrdparm()

returns board parameters

Name : int spm_getbrdparm(devh,param,valuep)

Inputs :

int devh: SPM board device handle

unsigned long param: device parameter defined name

void *valuep: pointer to variable where the parameter value will be placed

Returns : 0

AT_FAILURE on failure

Includes: smartsdk.h

Category: Configuration

Mode: synchronous



spm_getbrdparm() board parameter . parameter

"smartsdk.h" . parameter spm_setbrdparm .


Parameter .


devh: spm_open

param: The parameter to be examined

valuep: Pointer to the variable where the parameter value will be returned



 device handle

 parameter가

Example

```
#include <windows.h>
```

```
#include <errno.h>
```

```
#include "smartsdk.h"
```

```

main()
{
    int    devh;           /* MSI/SC board device descriptor */
    int    value;          /* Parameter value */
    int    cadence[8];     /* Ring cadence length and pattern */
    int    cadence_len;    /* Cadence active period length (in bytes) */

    if ((devh = spm_open("spmB1", 0)) == -1) {
        printf("Error opening spmB1 : errno = %d\n", errno);
        exit(1);
    }

    /* Determine board type : MAX Hook Flash detection Time */
    if (spm_getbrdparm(devh, SPM_MAXFLASH, (void *)&value) == -1) {
        printf("Error retrieving board parameter : %s\n ",
            ATDV_ERRMSGP(devh));
        exit(1);
    }

    printf("MAX Hook Flash Time %d\n", value);

    if (ms_close(devh) == -1) {
        printf("Error Closing msiB1 : errno = %d\n", errno);
        exit(1);
    }
    return;
}

```

□ Error

가 , AT_FAILURE ATDV_LASTERR() error

.

□

~~스스~~ spm_setbrdparm()

spm_getcde()

retrieves the attributes of a participant

Name : int spm_getcde(devh,confID,cdt)

Inputs :

int devh: SPM board device handle

int confID: conference identifier

CONFDESC *cdt: pointer to CONFDESC structure

Returns : 0

AT_FAILURE on failure

Includes: smartsdk.h

Category: Conference Management

Mode: synchronous

□

spm_getbrdparm() party .

Parameter .

devh: spm_open

confID: The conference identifier

cdt: Pointer to an CONFDESC structure

chan_num chan_sel input parameter chan_attr

.

CONFDESC format 가 .

typedef struct {

int chan_num; /* channel/time slot number */

int chan_sel; /* channel/time slot number */

int chan_attr; /* */

int chan_lts /* time slot number */

} CONFDESC;

chan_num	chan_sel	가	.
/del> MSPN_STATION	SPM	board	port
/del> MSPN_TS	SCbus time slot number		

Table Possible Returns for Channel Attribute

Channel Attribute	Description
MSPA_NULL	
MSPA_RO	receive-only mode
MSPA_COACH	Party is a coach - coach is heard by pupil only
MSPA_PUPIL	Party is a pupil - pupil hears everyone including coach

NOTE: party spm_getcde()

□

/del>	device handle
/del>	group ID
/del>	party

□ Example

```
#include <windows.h>
#include <errno.h>
#include "smartsdk.h"

#define NUM_PARTIES 2

int dev1=1; /* Board dev descriptor variables */
CONFDESC cdt[NUM_PARTIES]; /* Conf. desc. table */
int confID; /* Conf. ID */
int attr; /* Channel attribute */
int station, ts;

/* Start the system */
```

```

/* Set up CDT structure */
cdt[0].chan_num = station ; /* station is a valid station number */
cdt[0].chan_sel = MSPN_STATION;
cdt[0].chan_attr = MSPA_NULL;

/* SCbus time slot to be conferenced */
cdt[1].chan_num = ts ; /* ts should be a valid time slot */
cdt[1].chan_sel = MSPN_TS;
cdt[1].chan_attr = MSPA_NULL;

/* Establish conference */
if (spm_estconf(dev1, cdt, NUM_PARTIES, MSCA_ND, &confID) == -1) {
    printf("Error Message = %s",ATDV_ERRMSGP(dev1));
    exit(1);
}

/*
 *
 * Continue processing
 *
 */

/* Now get the attribute of MSI Station */
cdt[0].chan_num = station; /* Station in the conference */
cdt[0].chan_sel = MSPN_STATION;

if( spm_getcde(dev1, confID, &cdt[0])== -1){
    printf("Error Message = %s",ATDV_ERRMSGP(dev1));
    exit(1);
}

attr = cdt[0].chan_attr;
/*
 * Continue Processing
 *
 */

```

□ Error

가 , AT_FAILURE ATDV_LASTERR() error

.

□

~~스피커~~ spm_setcde()

Name : int spm_getdtmf(chdev,tptp,digitp,mode)

Inputs : int chdev : SPM channel device handle
DV_TPT *tptp : Termination Parameter Structure Pointer
DV_DIGIT *digitp : User Digit Buffer Structure Pointer
unsigned short mode : /

Returns :
0 : initial
digit : digit 1
-1 :

Includes: srllib.h, smartsdk.h

Category: I/O

Mode: synchronous/asynchronous

spm_getdtmf() open channel digit buffer digit
가 digit ASCIIZ format DV_DIGIT
structure local buffer .

termination DV_TPT structure , structure
tptp parameter .

Asynchronous Operation

mode field EV_ASYNC .
가 0
, digit termination event
termination event SRL event Management Function
Event Management Function Appendix A
digit TDX_GETDIG Event .

Synchronous Operation

	digit	0
Parameter		
<hr/>		
chdev : spm_open()	SPM channel device handle	
tptp :	Termination	Termination Parameter Table
Structure pointer		
Termination		
DX_MAXDTMF :	Maximum number of digits received	
DX_IDDTIME :	Inter-digit delay	
DX_MAXTIME :	Function Time	
DX_DIGMASK :	Digit mask termination	
digitp :	digit digit type	digit buffer structure
pointer		
digits	digit type array	
mode : spm_getdig()		
	가	
EV_ASYNC :		
EV_SYNC :		
digit buffer 31	FIFO	digit
buffer digit	overwrite	
digit	.digit buffer clear	vpm_clrdigbuf
NOTE :	spm_getdigbuf()	31 digit
digit		
□		
1.	가	, digit buffer duration
2.	가	, buffer digit , return
value 1	, digit buffer가	null termination
3. spm dtmf receiver resource가	4	4 dtmf

4. dtmf gathering 가 spm_stopdtmf() dtmf receiver resource

□ **Example 1 : Using spm_getdtmf() in synchronous mode**

```
#include <stdio.h>
#include <srllib.h>
#include <smartsdk.h>
#include <windows.h>
main()
{
    DV_TPT tpt[2];
    DV_DIGIT digp;
    int chdev, numdigs, cnt;
    /* open the channel with spm_open( ). Obtain channel device descriptor
    * in chdev
    */
    if ((chdev = spm_open("spmB1C1",NULL)) == -1) {
        /* process error */
    }
    /* initiate the call */
    .
    .
    /* Set up the DV_TPT and get the digits */
    vpm_clrtpt(tpt,2);
    tpt[0].tp_type = IO_CONT;
    tpt[0].tp_termno = DX_MAXDTMF; /* Maximum number of digits */
    tpt[0].tp_length = 4; /* terminate on 4 digits */
    tpt[0].tp_flags = TF_MAXDTMF; /* terminate if already in buf. */
    tpt[1].tp_type = IO_EOT;
    tpt[1].tp_termno = DX_MAXTIME; /* Function Time */
    tpt[1].tp_length = 100; /* 10 seconds (100 ms resolution
    * timer) */
    tpt[1].tp_flags = TF_MAXTIME; /* Edge-triggered */
```

```

/* clear previously entered digits */
if (vpm_clrdigbuf(chdev) == -1) {
/* process error */
}
if ((numdigs = spm_getdtmf(chdev,tpt, &digp, EV_SYNC)) == -1) {
/* process error */
}
for (cnt=0; cnt < numdigs; cnt++) {
    printf("\nDigit received = %c, digit type = %d",
        digp.dg_value[cnt], digp.dg_type[cnt]);
}
spm_stopdtmf(chdev);
/* go to next state */
.
.
}

```

□ **Example 2 : Using spm_getdtmf() in asynchronous mode**

```

#include <stdio.h>
#include <srllib.h>
#include <smartsdk.h>
#include <windows.h>
#define MAXCHAN 24
int digit_handler();
DV_TPT stpt[3];
DV_DIGIT digp[256];
main()
{
    int i, chdev[MAXCHAN];
    char *chnamep;
    for (i=0; i<MAXCHAN; i++) {
        /* Set chnamep to the channel name - e.g., spmB1C1 */
        /* open the channel with vpm_open( ). Obtain channel device
        * descriptor in chdev[i]
        */
        if ((chdev[i] = spm_open(chnamep,NULL)) == -1) {

```

```

/* process error */
}
/* Using sr_enbhdr(), set up handler function to handle vpm_getdig()
* completion events on this channel.
*/
if (sr_enbhdr(chdev[i], TDX_GETDIG, digit_handler) == -1) {
/* process error */
}
/* initiate the call */
.
.
/* Set up the DV_TPT and get the digits */
vpm_clrtpt(tpt,2);
tpt[0].tp_type = IO_CONT;
tpt[0].tp_termno = DX_MAXDTMF; /* Maximum number of digits */
tpt[0].tp_length = 4; /* terminate on 4 digits */
tpt[0].tp_flags = TF_MAXDTMF; /* terminate if already in buf*/
tpt[1].tp_type = IO_EOT;
tpt[1].tp_termno = DX_MAXTIME; /* Function Time */
tpt[1].tp_length = 100; /* 10 seconds (100 ms resolution
* timer) */
tpt[1].tp_flags = TF_MAXTIME; /* Edge triggered */
/* clear previously entered digits */
if (vpm_clrdigbuf(chdev[i]) == -1) {
/* process error */
}
if (spm_getdtmf(chdev[i], tpt, &digp[chdev[i]], EV_ASYNC) == -1) {
/* process error */
}
}
/* Use sr_waitvt() to wait for the completion of spm_getdtmf().
* On receiving the completion event, TDX_GETDIG, control is transferred
* to the handler function previously established using sr_enbhdr().
*/
.
.

```

```

}
int digit_handler()
{
int chfd;
int cnt, numdigs;
chfd = sr_getevtdev();
numdigs = strlen(digp[chfd].dg_value);
for(cnt=0; cnt < numdigs; cnt++) {
    printf("\nDigit received = %c, digit type = %d",
        digp[chfd].dg_value[cnt], digp[chfd].dg_type[cnt]);
}

spm_stopdtmf(chdev[chfd])
/* Kick off next function in the state machine model. */
.
.
return 0;
}

```

□ Errors

가 -1, ATDV_LASTERR() ATDV_ERRMSGP()
Error .

EDX_BADPARAM : Invalid Parameter

EDX_BADTPT : Invalid DV_TPT entry

EDX_BUSY : Channel busy

EDX_SYSTEM : Windows 2000 System error – check errno

□

? DV_DIGIT

? vpm_getdig()

? spm_stopdtmf()

? vpm_getdig()

? vpm_clrdigbuf()

spm_getevt()

blocks and returns control to the application

Name : int spm_getevt (devh,ebkp,timeout)

Inputs :

int devh : SPM device handle

EV_EBLK *ebkp : pointer to event block

int timeout

Returns :

0 :

AT_FAILURE :

Includes: srllib.h

spmlib.h

Category: Configuration

Mode: Synchronous

□

spm_getevttt()

spm_setevtmsk()

가

timeout

Parameter

devh : spm_open

evtblkp : Blocking 가

timeout : 가 -1 , AT_FAILURE

timeout 0 , EV_EBLK

eblk.ev_dev : 가

eblk.ev_event : MSEV_SIG EVT

eblk.ev_data : ev_data[0] ev_data[1]

structure

```

typedef struct ev_eblk {
    int ev_dev; /* Device on which event occurred */
    unsigned long ev_event; /* Event type */
    int ev_len; /* Length of data associated with event */
    unsigned char ev_data[8]; /* 8 byte data buffer */
    void ev_datap; /* variable pointer if more than 8 bytes of
    data */
} EV_EBLK;

```

□

```

:
?
? event field가 .

```

□ Example

```

#include <windows.h> /* For Windows 2000 applications only */
#include <errno.h>
#include "smartsdk.h"
EV_EBLK eblk;
main()
{
    int devh; /* Board device handle */
    unsigned short sigmsk = MSMM_ONHOOK | MSMM_OFFHOOK |
    MSMM_HOOKFLASH;
    short sig;
    /*
    * Open station 1 device
    */
    if ((devh = spm_open("spmB1C1",0)) == -1) {
        printf("Error: Cannot open board 1 station 1. errno =
        0x%x\n",errno);
        exit(1);
    }
    if (spm_setevtmsk(devh, MSEV_SIG, sigmsk, DTA_SETMSK) == -1) {
        printf("%s: spm_setevtmsk MSEV_SIGMSK DTA_SETMSK

```

```

                                ERROR %d: %s:Mask = 0x%x\n",
    ATDV_NAMEP(devh), ATDV_LASTERR(devh), ATDV_ERRMSGP(devh), si
    gmsk);
    spm_close(devh);
    exit(1);
}
/*
 * Wait for events on this time slot
 */
while(1) {
    spm_getevt ( devh, &eblk, -1 ); /* Wait forever */
    if (eblk.ev_event == MSEV_SIG EVT) {
        sig = ebld.ev_data[0] | (short) ebld.ev_data[1] <<
                                                8 ;

        if ((sig & MSMM_ONHOOK) == MSMM_ONHOOK)
            printf("Onhook signal received\n");
        if ((sig & MSMM_OFFHOOK) == MSMM_OFFHOOK)
            printf("Offhook signal received\n");
        if ((sig & MSMM_HOOKFLASH) ==
                                MSMM_HOOKFLASH)
            printf("Hook flash signal
                    received\n");
    }
} /* end of while statement */
}

```

□ Error

가 -1	, error code	SRL Standard Attribute
ATDV_LASTERR()	.	

□

? spm_getevtmsk()

spm_getevtmsk()

returns station event mask

Name : int spm_getevtmsk (devh,event,bitmaskp)

Inputs :

int devh	: SPM device handle
int event	: event to retrieve
unsinged short *bitmaskp	: pointer to bitmask variable

Returns :

0 :

AT_FAILURE :

Includes: srllib.h

spmlib.h

Category: Configuration

Mode: Synchronous

□

spm_getevtmsk()

Parameter

devh : spm_open

event :

? MSEV_SIGMSK

On-hook transition event

Off-hook transition event

Hookflash event

bitmaskp : bitmask

□

:

?

? event field가

□ Example

```
#include <windows.h> /* For Windows 2000 applications only */
#include <errno.h>
#include "smartsdk.h"
/* Basic error handler */
do_error( devh, funcname )
int devh;
char *funcname;
{
    int errorval = ATDV_LASTERR( devh );
    printf( "Error while calling function %s.\n", funcname );
    printf( "Error value = %d.", errorval );
    printf( "\n" );
}
main()
{
    int tsdev; /* Station device descriptor variable */
    unsigned short bitmask; /* Bitmask variable */
    /* Open board 1 device */
    if ( ( tsdev = spm_open( "spmB1C1", 0 ) ) == -1 ) {
        printf( "Cannot open board spmB1C1. errno = %d", errno );
        exit( 1 );
    }
    /* Get signaling event mask*/
    if ( spm_getevtsk( tsdev, MSEV_SIGMSK, &bitmask ) == -1 ) {
        do_error( tsdev, "spm_getevtsk( )");
    }
    if ( bitmask & MS_ONHOOK ) {
        /* continue processing (ON-HOOK event is set) */
        printf("ON-HOOK event is set\n");
    }
    if ( bitmask & MS_OFFHOOK ) {
        /* continue processing (OFF-HOOK event is set) */
        printf("OFF-HOOK event is set\n");
    }
}
```

```

if ( bitmask & MS_HOOKFLASH) {
/* continue processing (HOOK FLASH event is set) */
    printf("HOOK FLASH event is set\n");
}
/*
* Continue processing
* .
* .
* .
*/
/* Done processing - close device */
if ( spm_close( tsdev ) == -1 ) {
    printf( "Cannot close board spmB1C1. errno = %d",
            errno );
}
}

```

❑ Error

가 -1 , error code SRL Standard Attribute
 ATDV_LASTERR() .

❑

? spm_setevtmask()

spm_monconf()

adds a monitor to a conference

Name : int spm_monconf (devh,confID,lts)

Inputs :

int devh : SPM device handle

int confID : conference identifier

long *lts : pointer to listen SCbus time slot

Returns :

0 :

AT_FAILURE :

Includes: srllib.h

spmlib.h

Category: Conference Management

Mode: Synchronous

□

spm_monconf() conference monitor 가 . monitoring SCbus time slot 가 , receive only mode conference 가 . receive-only mode conference 가 .

Monitor part가 SCbus time slot parties가 가 .

Parameter .

devh : spm_open

confID : Conferencd identifier

lts : monitored time slot

NOTE : 1. conference group monitor port 가 .
2. 가 가 part가 .

□

? device handle

? conference party가 8

? conference group ID가

□ Example

```
#include <windows.h> /* For Windows 2000 applications only */
#include <errno.h>
#include "smartsdk.h"

#define NUM_PARTIES 2

int dev1; /* Board dev descriptor variables */
int tsdev1; /* DTI time slot device handle */
MS_CDT cdt[NUM_PARTIES]; /* Conf. desc. table */
int confID; /* Conf. ID */
long lts; /* listen time slot */
SC_TSINFO tsinfo; /* Time slot information structure */
int ts1;

/* Open board 1 device */
if ((dev1 = spm_open("spmB1",0)) == -1) {
    printf( "Cannot open MSI B1: errno=%d", errno);
    exit(1);
}

/* Assume that there is a DTI in the system.
 * Assume the device handle for a time slot on the DTI
 * is tsdev1 and time slot it is assigned to is ts1
 */

/* Set up CDT structure */
cdt[0].chan_num = station ; /* Valid MSI Station */
cdt[0].chan_sel = MSPN_STATION;
cdt[0].chan_attr =MSPA_NULL;
cdt[1].chan_num = ts1; /* ts1 is a valid DTI time slot */
```

```

cdt[1].chan_sel = MSPN_TS;
cdt[1].chan_attr =MSPA_TARIFF;
/* Establish conference */
if (spm_estconf(dev1, cdt, NUM_PARTIES, MSCA_ND, &confID) == -1)
{
    printf("Error Message = %s",ATDV_ERRMSGP(dev1));
    exit(1);
}
/*
* Continue Processing
*/
/* Now monitor the conference */
if (spm_monconf(dev1, confID,&ltts) == -1){
    printf("Error Message = %s",ATDV_ERRMSGP(dev1));
    exit(1);
}
/* Assume that a DTI device tsdev1 is available */
tsinfo.sc_numts = 1;
tsinfo.sc_tsarray = &ltts;
if (dt_listen(tsdev1,&tsinfo) == -1){
    printf("Error Message = %s",ATDV_ERRMSGP(tsdev1));
    exit(1);
}
/*
* Continue Processing
*/

```

□ Error

가 -1 , error code SRL Standard Attribute
 ATDV_LASTERR() .

□

? spm_unmonconf ()

spm_open() opens an SPM device

Name : int spm_open (name,oflags)

Inputs :

char *name : SPM station or board device name

int oflags : open attribute flags

Returns :

device handle :

AT_FAILURE :

Includes: srllib.h

spmlib.h

Category: Device Management

Mode: Synchronous

□

spm_open() SPM device open handle . open
device handle .

NOTE : handle system unique system process
가 .

Parameter

name : board channel spmBbCc spmBb 가 b 1
- n board id, c 1-16 channel ID가 .

oflags : Reserved

□

device os open open .

? device name

□ **Example**

```

#include <windows.h> /* For Windows 2000 applications only */
#include <errno.h>
#include "smartsdk.h"
main()
{
    int bddev; /* Board device descriptor variable */
    /* Open board 1 device */
    if ( ( bddev = spm_open( "spmB1", 0 ) ) == -1 ) {
        printf( "Cannot open board spmB1. errno = %d\n", errno );
        exit( 1 );
    }
    /*
    * Continue processing
    * .
    * .
    * .
    */
    /* Done processing - close device */
    if ( spm_close( bddev ) == -1 ) {
        printf( "Cannot close board spmB1. errno = %d", errno );
    }
}

```

□ Error

spm_open () AT_FAILURE .

□

? spm_close()

spm_remfromconf()

removes a party a conference

Name : int spm_remfromconf (devh,confID,cdt)

Inputs :

int devh : SPM device handle

int confID : conference identifier

CONFDESC *cdt

Returns :

0 :

AT_FAILURE :

Includes: SmartSdk.h

Category: Conference Management

Mode: Synchronous

□

spm_remfromconf()	conference	party	. Conference ID
spm_estconf()	return	, party	cdt

Parameter

devh : spm_open

confID : Conferencd identifier

cdt : pointer to an CONFDESC structure

NOTE : 1. SCbus time slot unlisten

2. 가 party 가 .

□

party error return spm_delconf()

? handle

? conference group ID가

? party

□ Example

```
#include <windows.h> /* For Windows 2000 applications only */
#include <errno.h>
#include "smartsdk.h"
#define NUM_PARTIES 3
int dev1; /* Board dev descriptor variables */
CONFDESC cdt[NUM_PARTIES]; /* Conf. desc. table */
int confID; /* Conf. ID */
/* Open board 1 device */
if ((dev1 = spm_open("spmB1",0)) == -1) {
    printf( "Cannot open MSI B1: errno=%d", errno);
    exit(1);
}
/*
 * Continue processing
 */
/* Set up CDT structure */
/* Assume MSI stations 2, 4 and 7 are in the conference */
cdt[0].chan_num = 2;
cdt[0].chan_sel = MSPN_STATION;
cdt[0].chan_attr = MSPA_NULL;
cdt[1].chan_num = 4
cdt[1].chan_sel = MSPN_STATION;
cdt[1].chan_attr = MSPA_PUPIL;
cdt[2].chan_num = 7;
cdt[2].chan_sel = MSPN_STATION;
cdt[2].chan_attr = MSPA_COACH;
/* Establish conference */
if (spm_estconf(dev1, cdt, NUM_PARTIES, MSCA_ND, &confID) != 0) {
    printf("Error Message = %s",ATDV_ERRMSGP(dev1));
    exit(1);
}
/*
 * Continue processing
 */
```

```

*
*/
cdt[0].chan_num = 2;
cdt[0].chan_sel = MSPN_STATION;
if (spm_remfromconf(dev1, confID, &cdt[0]) == -1){
    printf("Error Message = %s", ATDV_ERRMSGP(dev1));
    exit(1);
}
if (spm_delconf(dev1, confID) == -1){
    printf("Error Message = %s", ATDV_ERRMSGP(dev1));
    exit(1);
}
/* Continue processing */

```

□ Error

가 -1 , error code SRL Standard Attribute
 ATDV_LASTERR() .

□

? spm_addtoconf ()
 ? spm_delconf ()
 ? spm_estconf ()

Name : int spm_setbrdparm(devh,param,valuep)

Inputs :

int devh: SPM board device handle

unsigned long param: device parameter defined name

void *valuep: pointer to variable where the parameter value will be placed

Returns : 0

AT_FAILURE on failure

Includes: smartsdk.h

Category: Configuration

Mode: synchronous

□

spm_setbrdparm() board parameter .

Parameter .

devh: spm_open

param: parameter

valuep: parameter pointer

Typically, the default value for each SPM parameter is adequate for operation.
However, the user may need to change the following conditions:

- ✂✂ SPM_DBONTM - Debounce on time
- ✂✂ SPM_DBOFFTM - Debounce off time
- ✂✂ SPM_MINFLASH - Minimum hook flash time
- ✂✂ SPM_MAXFLASH - Maximum hook flash time

SPM device parameters .

SPM Board/Device parameters Table

Parameter ID	Description			
MSG_DBOFFTM	off-hook	(50ms)	off-hook event

MSG_DBONTM on-hook (50ms) on-hook 7f

event

NOTE: SPM_MAXFLASH .

: 15H

MSG_MINFLASH	Hook-flash	on-hook
	2 - 14CH,	6H

parameter integer .

Example

```
main()
{
    int          devh;          /* Board device descriptor variable */
    int          ontime;

    /* Hook-On Time setting */

    if (ms_setbrdparm(devh,SPM_BDONTM,(void *)&ontime[0])) == -1){
```

```

        printf("Error setting board parameter : %s\n",
               ATDV_ERRMSGP(devh));
        exit(1);
    }

    if ((devh = spm_open("msiB1", 0)) == -1) {
        printf("Error opening spmB1 : errno = %d\n", errno);
        exit(1);
    }

    if (spm_close(devh) == -1) {
        printf("Error Closing spmB1 : errno = %d\n", errno);
        exit(1);
    }
}

```

□ Error

가 , AT_FAILURE ATDV_LASTERR() error

□

~~스스~~ spm_getbrdparm()

spm_setcde()

changes the attributes of a party

Name : int spm_setcde(devh,confID,cdt)

Inputs :

int devh: SPM board device handle

int confID: conference identifier

CONFDESC *cdt: pointer to CONFDESC structure

Returns : 0

AT_FAILURE on failure

Includes: smartsdk.h

Category: Conference Management

Mode: synchronous



spm_setbrdparm() conference party .

Parameter .

devh: spm_open

confID: The conference identifier

cdt: Pointer to an CONFDESC structure

CONFDESC format 가 .

```
typedef struct {  
    int chan_num;                      /* channel/time slot number */  
    int chan_sel;                      /* channel/time slot number                      */  
    int chan_attr;                      /*                      */  
    int chan_Its                      /*                      time slot number */  
} CONFDESC;
```

bitmask .

~~MSPA_NULL~~

~~MSPA_RO~~ receive-only mode

~~MSPA_COACH~~ Party is a coach - coach is heard by pupil only

~~MSPA_PUPIL~~ Party is a pupil - pupil hears everyone including coach

□



device handle

□ Example

```
#include <windows.h>
#include <errno.h>
#include "smartsdk.h"

#define    NUM_PARTIES    2

int dev1;                                /* Board dev descriptor variables */
int chdev2;                              /* Channel dev descriptor */
CONFDESC cdt[NUM_PARTIES];              /* Conf. desc. table */
int confID;                              /* Conf. ID */

/* Open board 1 device */
if ((dev1 = spm_open("spmB1",0)) == -1) {
    printf( "Cannot open SPM B1: errno=%d", errno);
    exit(1);
}

/* Open board 1, channel 2 device */
if ((chdev2 = spm_open("spmB1C2",0) == -1) {
    printf("Cannot open SPM B1, C2. errno = %d", errno);
    exit(1);
}

//Continue processing

/* Set up CDT structure */
cdt[0].chan_num = 2;
cdt[0].chan_sel = MSPN_STATION;
```



```

cdt[0].chan_attr = MSPA_COACH;

cdt[1].chan_num = 1;
cdt[1].chan_sel = MSPN_TS;
cdt[1].chan_attr = MSPA_PUPIL;

/* Establish conference */
if (spm_estconf(dev1, cdt, NUM_PARTIES, MSCA_ND, &confID) != 0) {
    printf("Error Message = %s", ATDV_ERRMSGP(dev1));
    exit(1);
}
//Continue processing

/* Now change the attribute of MSI Station 2 */
cdt[0].chan_num = 2;
cdt[0].chan_sel = MSPN_STATION;
cdt[0].chan_attr = MSPA_NULL;

if((ms_setcde(dev1, confID, cdt)) == -1) {
    printf("Error Message = %s", ATDV_ERRMSGP(dev1));
    exit(1);
}
// Continue Processing

```

□ Error

가 , AT_FAILURE ATDV_LASTERR() error

□

~~스~~ spm_addtoconf()

~~스~~ spm_estconf()

~~스~~ spm_getcde()

spm_setevtmask ()

changes transition event masks

Name : spm_setevtmask (devh,event,bitmask,action)

Inputs :

int devh: SPM board device handle

int event: event to be enabled/disabled

unsigned short bitmask: bitmask for events

int action: set, add, or subtract bitmask

Returns : 0 on success

AT_FAILURE on failure

Includes: smartsdk.h

Category: Configuration

Mode: synchronous



SPM_SETEVTMSK() transition , 가, .

Parameter .

devh: spm_open

event: transition

bitmask: bitmask

MSMM_OFFHOOK – off_hook 가

MSMM_ONHOOK – on_hook 가

MSMM_HOOKFLASH – hook flash 가



Device handle invalid

Event 가 invalid

Action invalid

Example

```
#include <windows.h> /* For Windows 2000 applications only */
```

```

#include <errno.h>
#include "smartsdk.h"

/* Basic error handler */

do_error( devh, funcname )
    int devh;
    char *funcname;
{
    int errorval = ATDV_LASTERR( devh );

    printf( "Error while calling function %s.\n", funcname );

    printf( "Error value = %d.", errorval );
    printf( "\n" );
}

main()

{
    int tsdev;          /* Channel device descriptor variable */

    /* Open board device */

    if ((dev1 = spm_open("spmB1",0)) == -1) {
        printf( "Cannot open SPM B1: errno=%d", errno);
        exit(1);
    }

    /* Enable signaling transition events (off-hook event) */

    if ( spm_setevtmsk( tsdev, MSEV_SIGMSK, MSMM_OFFHOOK, DTA_SETMSK ) == -1 ){
        do_error( tsdev, "spm_setevtmsk()" );
        exit( 1 );
    }
}

```

```

}

/*

*   Continue processing
*       .
*       .
*       .
*/

/* Done processing - close device */

if (spm_close(dev1)== -1){
    printf( "Cannot Close SPMB1: errno=%d", errno);
    exit(1);
}
}

```

□ **Error**

가 , AT_FAILURE ATDV_LASTERR() error .

□

spm_getevtmsk()

spm_stopdtmf()

stops dtmf gathering

Name : int spm_stopdtmf (devh)

Inputs :

Unsigned int devh: SPM channel device handle

Returns : 0 on success

AT_FAILURE on failure

Includes: smartsdk.h

Category: Device Management

Mode: synchronous

□

spm_stopdtmf()

dtmf resource

.

Parameter

.

devh: spm_open

□ **Example**

```
#include <stdio.h>
```

```
#include <srllib.h>
```

```
#include <smartsdk.h>
```

```
#include <windows.h>
```

```
main()
```

```
{
```

```
DV_TPT tpt[2];
```

```
DV_DIGIT digp;
```

```
int chdev, numdigs, cnt;
```

```
/* open the channel with spm_open( ). Obtain channel device descriptor
```

```
* in chdev
```

```
*/
```

```
if ((chdev = spm_open("spmB1C1",NULL)) == -1) {
```

```
/* process error */
```

```
}
```

```
/* initiate the call */
```

```

.
/* Set up the DV_TPT and get the digits */
vpm_clrtpt(tpt,2);
tpt[0].tp_type = IO_CONT;
tpt[0].tp_termno = DX_MAXDTMF; /* Maximum number of digits */
tpt[0].tp_length = 4; /* terminate on 4 digits */
tpt[0].tp_flags = TF_MAXDTMF; /* terminate if already in buf. */
tpt[1].tp_type = IO_EOT;
tpt[1].tp_termno = DX_MAXTIME; /* Function Time */
tpt[1].tp_length = 100; /* 10 seconds (100 ms resolution
* timer) */
tpt[1].tp_flags = TF_MAXTIME; /* Edge-triggered */
/* clear previously entered digits */
if (vpm_clrdigbuf(chdev) == -1) {
/* process error */
}
if ((numdigs = spm_getdtmf(chdev,tpt, &digp, EV_SYNC)) == -1) {
/* process error */
}
for (cnt=0; cnt < numdigs; cnt++) {
    printf("\nDigit received = %c, digit type = %d",
        digp.dg_value[cnt], digp.dg_type[cnt]);
}
spm_stopdtmf(chdev);
/* go to next state */
.
.
}

```

□ Error

가 , AT_FAILURE ATDV_LASTERR() error

□

spm_getdtmf ()

spm_stopfn()

stops a multitasking function

Name : int SPM_ STOPFN (devh,funcid)

Inputs :

Unsigned int devh: SPM channel device handle

Unsigned int funcid : ID of multitasking function

Returns : 0 on success

AT_FAILURE on failure

Includes: smartsdk.h

Category: Device Management

Mode: synchronous

□

SPM_ STOPFN()

Ring

.

Parameter

.

devh: spm_open

funcid: Ring ID

MTF_RING: Ring

.

: Ring 가

□ Example

```
#include <windows.h> /* For Windows 2000 applications only */
```

```
#include <errno.h>
```

```
#include " smartsdk.h "
```

```
int chdev1 ;
```

```
/* Open board 1 device */
```

```
if ((dev1 = spm_open("spmB1C1",0)) == -1) {
```

```
    printf( "Cannot open SPM B1C1: errno=%d", errno);
```

```
    exit(1);
```

```
}
```

```
/* ring the station 2 five times */
```

```
if (spm_genring(chdev1, 5, EV_ASYNC)== -1){  
    printf("Error Message = %s",ATDV_ERRMSGP(chdev1));  
    exit(1);  
}
```

```
/* 2 seconds later, ringing has not completed and station 2
```

```
* has not gone off-hook. However, there is a need to abort the  
* ringing on station 2. Issue the abort command  
*/
```

```
if (spm_stopfn(chdev1,MTF_RING)== -1) {  
  
    printf("Error Message = %s",ATDV_ERRMSGP(chdev1));  
    exit(1);  
}
```

□ Error

가 , AT_FAILURE ATDV_LASTERR() error

.

□

spm_genring()

spm_stoptone()

stops tone sending

Name : int spm_stoptone (devh)

Inputs :

Unsigned int devh: SPM channel device handle

Returns : 0 on success

AT_FAILURE on failure

Includes: smartsdk.h

Category: Device Management

Mode: synchronous



spm_stoptone()

tone

.

Parameter

.

devh: spm_open

: tone

가

Example

```
#include <windows.h>          /* For Windows 2000 applications only */
```

```
#include <errno.h>
```

```
#include " smartsdk.h "
```

```
int chdev1 ;
```

```
/* Open board 1 channel device */
```

```
if ((dev1 = spm_open("spmB1C1",0)) == -1) {  
    printf( "Cannot open SPM B1C1: errno=%d", errno);  
    exit(1);  
}
```

```
/* ring the station 2 five times */
```

```

if (spm_gentone(chdev1, 5)== -1){
    printf("Error Message = %s",ATDV_ERRMSGP(chdev1));
    exit(1);
}

```

```

if (spm_stoptone(chdev1)== -1) {

    printf("Error Message = %s",ATDV_ERRMSGP(chdev1));
    exit(1);
}

```

□ **Error**

가 , AT_FAILURE ATDV_LASTERR() error

.

□

spm_gentone()

spm_unmonconf() removes a monitor from a conference

Name : int spm_unmonconf (devh,confID)

Inputs :

int devh : SPM device handle
int confID : conference identifier

Returns :

0 :
AT_FAILURE :

Includes: SmartSdk.h

Category: Conference Management

Mode: Synchronous



spm_unmonconf() conference monitor .

NOTE : 가 party가 .

Parameter .

devh : spm_open

confID : Conferencd identifier



? handle

? conference group ID가

? monitor가

□ Example

```
#include <windows.h> /* For Windows 2000 applications only */  
#include <errno.h>  
#include "smartsdk.h"
```

```

#define NUM_PARTIES 2
int dev1; /* Board dev descriptor variables */
int tsdev1; /* DTI time slot device handle */
CONFDESC cdt[NUM_PARTIES]; /* conference descriptor table */
int confID; /* conference ID */
long lts; /* listen time slot */
SC_TSINFO tsinfo; /* time slot information structure */
/* Open board 1 device */
if ((dev1 = spm_open("spmB1",0)) == -1) {
    printf( "Cannot open MSI B1: errno=%d", errno);
    exit(1);
}
/* Assume that there is a DTI in the system.
* Assume the device handle for a time slot on the DTI
* is tsdev1 and time slot it is assigned to is ts1
*/
/* Set up CDT structure */
cdt[0].chan_num = station ; /* Valid MSI Station */
cdt[0].chan_sel = MSPN_STATION;
cdt[0].chan_attr = MSPA_NULL;
cdt[1].chan_num = ts1 ; /* ts1 is the DTI time slot */
cdt[1].chan_sel = MSPN_TS;
cdt[1].chan_attr = MSPA_RO;
/* Establish conference */
if (spm_estconf(dev1, cdt, NUM_PARTIES, MSCA_ND, &confID) == -1) {
    printf("Error Message = %s",ATDV_ERRMSGP(dev1));
    exit(1);
}
/*
* Continue Processing
*/
/* Now monitor the conference */
if (spm_monconf(devh,confID,&lts)== -1) {
    printf("Error Message = %s",ATDV_ERRMSGP(dev1));
    exit(1);
}

```

```

/* Assume that a DTI time slot tsdev1 is the available */
tsinfo.sc_numts = 1;
tsinfo.sc_tsarray = &ltts;
if (dt_listen(tsdev1,&tsinfo) == -1){
    printf("Error Message = %s",ATDV_ERRMSGP(tsdev1));
    exit(1);
}
/*
* Continue processing
*/
/* Unlisten to the monitor's time slot first */
if (dt_unlisten(tsdev1) == -1) {
    printf("Error message = %s\n", ATDV_ERRMSGP(tsdev1));
    exit(1);
}
/* Now unmonitor the conference */
if (spm_unmonconf(devh, confID) == -1) {
    printf("Error message = %s\n", ATDV_ERRMSGP(devh));
    exit(1);
}
/* Continue processing */

```

□ Error

가 -1 , error code SRL Standard Attribute
 ATDV_LASTERR() .

□

? spm_estconf ()
 ? spm_monconf ()

4. SPM Application

4.1

SPM Application

.
session
.
?
?
? Compiling & Linking
?
section
? Symbolic defines
? Head include
? return
.

4.1.1 Symbolic define

define SDK
.
, symbolic define
. symbolic defines "smartsdk.h".

4.1.2 Header File Include

error check SDK library function event handling,
function 가 header file include ,
SDK "smartsdk.h" include file header file
include .

4.1.3 Check Return Codes

SPM function AT_FAILURE
library function check .
/* call to SPM library function */
if (spm_xxx(arguments) == -1) {

```

/* error handling routine */
}
/* successful function call -
continue processing ... */

```

		function	0
error	check	.	

```

/* error handling routine */
void do_error( devh, funcname )
int devh;
char *funcname;
{
    int errorval = ATDV_LASTERR( devh );
    printf( "Error while calling function %s on device %s. \n", funcname,
    ATDV_NAMEP( devh ) );
    if ( errorval == E_MSSYSTEM ) {
        printf( "errno = %d\n", errno );
        perror("");
    } else {
        printf( "Error value = %d\n Error message = %s\n",
        errorval, ATDV_ERRMSGP( devh ) );
    }
    return;
}
main( )
{
    .
    .
    .

    if (spm_setevtmask( devh, MSEV_SIGMSK, 0, DTA_SETMSK ) != 0) {
        do_error( devh, "spm_setevtmask()" );
    }
}
/* successful function call -
continue processing ... */
.

```

```

.
.
}

```

NOTE: `spm_open` 0 가 , handle
 -1 error .

4.2

SPM application device F/W
 system Application parameter .

? H/W
 ? Event Mask
 ?

? `spm_setevtmask()`
 ? `spm_setbrdparm()`

4.2.1 H/W

H/W `spm_setbrdparm()` parameter

? **SPM_DBONTM** - Debounce on time
 ? **SPM_DBOFFTM** - Debounce off time
 ? **SPM_MINFLASH** - Minimum hook flash time
 ? **SPM_MAXFLASH** - Maximum hook flash time

SPM Application

4.2.2 Set event mask on SPM stations

`spm_setevtmask()` function hook on/off, hook flash event
 mask clear set . MS_OFFHOOK, MS_ONHOOK, MS_HOOKFL setting
 clear .

4.2.3

device가

,

? event disabling

? time slots reset

? device close

spm_setevtmsk() event notification disable .

NOTE: SRL function sr_dishdlr() event handler

.

4.3 Compiling and Linking

Compile Link libr.l.lib, libspm.lib link list .

Application scroute.lib, libvpm.lib, libfpm.lib .

4.4

Application interrupt thread , event pending

4.2 .

Appendix A

Standard Runtime Library : SPM Entries and Returns

Standard Runtime Library

Event Management Functions

spm_setevtmsk()

TABLE. SPM Inputs for Event Management Functions

Event Management Function	SPM specific Input	Value
sr_enbhdlr() Enable event handler	event type	MSEV_NORING MSEV_RING MSEV_SIGEV DTEV_COMRSP DTEV_DATRSP
sr_dishdlr() Disable event handler	event type	MSEV_NORING MSEV_RING MSEV_SIGEV DTEV_COMRSP DTEV_DATRSP
sr_waitevt() Wait for next event	N/A	N/A
sr_waitevtEX() Externed wait event	list of device handlers	

TABLE. SPM Returns for Event Management Functions

Event Management Function	SPM specific Input	Value
sr_getevtdev() Get SCT device handle	device	SPM device handle
sr_getevttype() Get event type	event type	MSEV_NORING MSEV_RING MSEV_SIGEV DTEV_COMRSP DTEV_DATRSP
sr_getevtlen() Get event length	event length	Number of bytes in the data returned

sr_getevtlidatap()	event data	Pointer to variable containing the value of a selected bitmask
--------------------	------------	--

Standard Attribute Functions

TABLE. Standard Attribute Functions

Standard Attribute Function	Information Returned for SPM
ATDV_ERRMSGP()	.
ATDV_LASTERR()	.
ATDV_NAMEP()	.(spmB1C1)
ATDV_SUBDEVS()	.