

SCbus Routing Function Reference for Windows 2000

Copyright ©2000 Seoul Comm Tec Corporation

Table of Contents

1. Introduction to SCbus Functions.....	3
SCbus Overview.....	3
1.1. SCbus Routing Function Overview.....	
2. SCbus Convenience Functions	5
SCbus Convenience Function Overview.....	5
nr_scroute() – connects two devices	5
nr_scunroute() – breaks the connection between two devices	13
3. SCbus Function Reference	19
SCbus Function Reference Overview.....	19
vag_getxmitslot() – provides SCbus time slot number.....	20
vag_listen() – connects analog receive channel to SCbus time slot.....	23
vag_unlisten() – disconnects analog receive channel form SCbus.....	27
vpm_getxmitslot() – provides SCbus time slot number.....	30
vpm_listen() – connects voice receive channel to SCbus time slot.....	33
vpm_unlisten() – disconnects voice receive channel from SCbus.....	36
fpm_getxmitslot() – returns the SCbus time slot information.....	39
fpm_listen() – connects the receive of a station.....	42
fpm_unlisten() – disconnects the receive of a station.....	47
spm_getxmitslot() – returns the SCbus time slot information.....	50
spm_listen() – connects the receive of a station.....	53
spm_unlisten() – disconnects the receive of a station.....	56
4. SCbus Data Structure Reference	58
SCbus Data Structure Overview.....	58
4.1. SCbus Time Slot INfOrmation(SC_TSINFO)	58
Appendix A – SCbus Routing Fuction Summary.....	59
vag_functions	
vpm_functions	
fpm_functions	
spm_functions	
nr_functions	
Appendix B– SCbus-related Publications.....	61

1. Instruction to SCbus Functions

SCbus Overview

가 SCbus device
SCbus Window NT demo program 가
SCbus routing the SCbus Routing Guide

NOTE: The SCbus routing function SCbus hardware

SCbus SCbus product digital 1024 time slot
SCbus multiple
port
SCbus time slot
application
device SCbus time slot

NOTE: SCbus routing function SCbus

1.1 SCbus Routing Function Overview

SCbus Routing Function network/resource(nr_) SCbus convenience function
Scbus route function routing function
convenience function

nr_scroute()	device
nr_scunroute()	device

SCbus routing function device transmit channel
device receive channel phase
program build

SCbus routing function device

vag_	Analog device (loop-start interface)
vpm_	Voice device
fpm_	FAX device
spm_	SPM device

SCbus routing function

operation

task

_getxmitslot()	SCbus time slot information device transmit channel SCbus time slot SC_TSINFO structure
_listen()	device listen(receive) channel SCbus time slot
_unlisten()	device listen(receive) channel SCbus time slot

SCbus routing function

vag_getxmitslot()	SCbus time slot information analog transmit channel SCbus time slot SC_TSINFO structure
vag_listen()	SCbus time slot analog receive(listen) channel
vag_unlisten()	SCbus time slot analog receive(listen) channel
vpm_getxmitslot()	SCbus time slot SCbus time slot SC_TSINFO structure
vpm_listen()	SCbus time slot receive(listen)
vpm_unlisten()	SCbus time slot receive(listen)
fpm_getxmitslot()	SCbus time slot Fax SCbus time slot SC_TSINFO structure
fpm_listen()	Fax receive(listen) SCbus time slot
fpm_unlisten()	Fax receive(listen) SCbus time slot
spm_getxmitslot()	SCbus time slot SPM station SCbus time slot SC_TSINFO structure

spm_listen()	SCbus time slot SPM receive station
spm_unlisten()	SCbus time slot SPM station

2. SCbus Convenience Functions

SCbus Convenience Function Overview

School.h 2 function call SCbus

 application

 routing process SCbus .

nr_scroute() nr_scunroute() application

device handle device type , application

 . SCbus .

nr_scroute()

 connects two devices

Name : int nr_scroute(devh1, devtype1, devh2, devtype2, mode)

Inputs :

```
int devh1 :
unsigned short devtype1 : devh1    device type
int devh2 :
unsigned short devtype2 : devh2    device type
unsigned char mode :
```

Returns :

$$\begin{array}{l} 0 : \\ -1 : \end{array}$$

Includes: vpmlib.h, spmlib.h, fpmlib.h

Category: routing Convenience

Mode: Synchronous

□

nr_scroute() device . device

nr_sc SCbus

access network(analog and digital) device resource(voice and fax) device .

Parameter .

Devh1 :

Devtype1 : devh1 device type

SC_CODEC

SC_DSP

SC_SPM

SC_FAX

Devh2:

Devtype2: devh2 device type.

Mode: . parameter

sctools.h

SC_FULLDUP

SC_HALFDUP

SC_FULLDUP . SC_HALFDUP가

device

SCbus time slot

listening

device

□

devtype1 devtype2 devh1 devh2 device type match

□ Source Code

/*

* Include files.

*/

#include <windows.h>

#include <stdio.h>

#include <srllib.h>

#include <dxxplib.h>

#ifdef DTISC

#include <dtilib.h>

#include <spmlib.h>

#endif

#ifdef FAXSC

#include <faxlib.h>

```

#endif
#include "sctools.h"
/*
 * Function prototypes
 */
static void nr_scerror(char *,...);
#if ( defined( __STDC__ ) || defined( __cplusplus ) )
int nr_scroute( int devh1, unsigned short devtype1,
                int devh2, unsigned short devtype2, unsigned char mode )

#else
int nr_scroute( devh1, devtype1, devh2, devtype2, mode )
    int          devh1;
    unsigned short devtype1;
    int          devh2;
    unsigned short devtype2;
    unsigned char mode;
#endif
{
    SC_TSINFO sc_tsinfo;      /* SCbus Timeslots information structure */
    long      scts;           /* SCbus Timeslot */
    /*
     * Setup the SCbus Timeslots information structure.
     */
    sc_tsinfo.sc_numts = 1;
    sc_tsinfo.sc_tsarrayp = &scts;
    /*

     * Get the SCbus timeslot connected to the transmit of the first device.
     */
    switch (devtype1) {
    case SC_DSP:
        if (vpm_getxmitslot(devh1, &sc_tsinfo) == -1) {
            nr_scerror("nr_scroute: %s: vpm_getxmitslot ERROR: %s\n",
                      ATDV_NAMEP(devh1), ATDV_ERRMSGP(devh1));
            return -1;
        }
    }
}

```



```

    }
    break;
case SC_CODEC:
    if (vag_getxmitslot(devh1, &sc_tsinfo) == -1) {
        nr_sccerror("nr_scroute: %s: vag_getxmitslot ERROR: %s\n",
                    ATDV_NAMEP(devh1), ATDV_ERRMSGP(devh1));
        return -1;
    }
    break;
case SC_SPM:
    if (spm_getxmitslot(devh1, &sc_tsinfo) == -1) {
        nr_sccerror("nr_scroute: %s: spm_getxmitslot ERROR: %s\n",
                    ATDV_NAMEP(devh1), ATDV_ERRMSGP(devh1));
        return -1;
    }
    break;
#endif
#ifdef FAXSC
case SC_FAX:
    if (fpm_getxmitslot(devh1, &sc_tsinfo) == -1) {
        nr_sccerror("nr_scroute: %s: fpm_getxmitslot ERROR: %s\n",
                    ATDV_NAMEP(devh1), ATDV_ERRMSGP(devh1));
        return -1;
    }
    break;
#endif
default:
    nr_sccerror("nr_scroute: %s: ERROR: Invalid 1st device type\n",
                ATDV_NAMEP(devh1));
    return -1;
}
/*
* Make the second device type listen to the timeslot that the first
* device is transmitting on. If a half duplex connection is desired,
* then return. Otherwise, get the SCbus timeslot connected to the
* transmit of the second device.

```

```

*/
switch (devtype2) {
case SC_DSP:
    if (vpm_listen(devh2, &sc_tinfo) == -1) {
        nr_scerror("nr_scroute: %s: Cannot vpm_listen %d ERROR: %s\n",
            ATDV_NAMEP(devh2),scts,ATDV_ERRMSGP(devh2));
        return -1;
    }
    if (mode == SC_HALFDUP) {
        return 0;
    }
    if (vpm_getxmitslot(devh2, &sc_tinfo) == -1) {
        nr_scerror("nr_scroute: %s: vpm_getxmitslot ERROR: %s\n",
            ATDV_NAMEP(devh2),scts,ATDV_ERRMSGP(devh2));
        return -1;
    }
    break;
case SC_CODEC:
    if (vag_listen(devh2, &sc_tinfo) == -1) {
        nr_scerror("nr_scroute: %s: Cannot vag_listen %d ERROR: %s\n",
            ATDV_NAMEP(devh2),scts,ATDV_ERRMSGP(devh2));
        return -1;
    }
    if (mode == SC_HALFDUP) {
        return 0;
    }
    if (vag_getxmitslot(devh2, &sc_tinfo) == -1) {
        nr_scerror("nr_scroute: %s: vag_getxmitslot ERROR: %s\n",
            ATDV_NAMEP(devh2),scts,ATDV_ERRMSGP(devh2));
        return -1;
    }
    break;
case SC_SPM:
    if (spm_listen(devh2, &sc_tinfo) == -1) {
        nr_scerror("nr_scroute: %s: Cannot spm_listen %d ERROR: %s\n",

```

```

        ATDV_NAMEP(devh2),scts,ATDV_ERRMSGP(devh2));
    return -1;
}
if (mode == SC_HALFDUP) {
    return 0;
}
if (spm_getxmitslot(devh2, &sc_tsinfo) == -1) {
    nr_sccerror("nr_scroute: %s: spm_getxmitslot ERROR: %s\n",
        ATDV_NAMEP(devh2),scts,ATDV_ERRMSGP(devh2));
    return -1;
}
break;
#endif
#ifdef FAXSC
case SC_FAX:
    if (fpm_listen(devh2, &sc_tsinfo) == -1) {
        nr_sccerror("nr_scroute: %s: Cannot fpm_listen %d ERROR: %s\n",
            ATDV_NAMEP(devh2),scts,ATDV_ERRMSGP(devh2));
        return -1;
    }
    if (mode == SC_HALFDUP) {
        return 0;
    }
    if (fpm_getxmitslot(devh2, &sc_tsinfo) == -1) {
        nr_sccerror("nr_scroute: %s: fpm_getxmitslot ERROR: %s\n",
            ATDV_NAMEP(devh2),scts,ATDV_ERRMSGP(devh2));
        return -1;
    }
    break;
#endif
default:
    nr_sccerror("nr_scroute: %s: ERROR: Invalid 2nd device type\n",
        ATDV_NAMEP(devh2));
    return -1;
}
/*

```

```

* Now make the first device listen to the SCbus timeslot that the
* second device is transmitting on.

*/
switch (devtype1) {
case SC_DSP:
    if (vpm_listen(devh1, &sc_tsinfo) == -1) {
        nr_scerrror("nr_scroute: %s: Cannot vpm_listen %d ERROR: %s\n",
            ATDV_NAMEP(devh1),scts,ATDV_ERRMSGP(devh1));
        return -1;
    }
    break;
case SC_CODEC:
    if (vag_listen(devh1, &sc_tsinfo) == -1) {
        nr_scerrror("nr_scroute: %s: Cannot vag_listen %d ERROR: %s\n",
            ATDV_NAMEP(devh1),scts,ATDV_ERRMSGP(devh1));
        return -1;
    }
    break;
#ifdef FAXSC
case SC_FAX:
    if (fpm_listen(devh1, &sc_tsinfo) == -1) {
        nr_scerrror("nr_scroute: %s: Cannot fpm_listen %d ERROR: %s\n",
            ATDV_NAMEP(devh1),scts,ATDV_ERRMSGP(devh1));
        return -1;
    }
    break;
#endif
}
return 0;
}

void nr_scerrror(char *fmt,  )
#ifdef PRINTON
    va_list args;
    /*
    * Make args point to the 1st unnamed argument and then print to stderr.

```

```

    */
    va_start(args, fmt);
    fmt = va_arg(args, char *);
    vfprintf(stderr, fmt, args);
    va_end(args);
#endif
}

```

□

• nr_scunroute ()

nr_scunroute() breaks the connection between two devices

Name : int nr_scunroute(devh1, devtype1, devh2, devtype2, mode)

Inputs :

int devh1 :
 unsigned short devtype1 : devh1 device type
 int devh2 :
 unsigned short devtype2 : devh2 device type
 unsigned char mode :

Returns :

0 :
 -1 :

Includes: vpmlib.h, spmlib.h, fpmlib.h

Category: routing Convenience

Mode: Synchronous

□

```

nr_scunroute()          device          .
    device              .               sctools.c
    source file
sctools.lib          . nr_sc          SCbu          access 7|          network device
    resource device          .

```

NOTE: SPM FAX makefile
 DTISC FAXSC
 . DTI FAX
 DTI FAX .

Parameter

Devh1 :

Devtype1 : devh1 device type
 SC_CODEEC
 SC_DSP
 SC_SPM
 SC_FPM

Devh2:

Devtype2: devh2 device type.

Mode:

 sctools.h
 SC_FULLDUP .
 SC_HALFDUP .
 SC_FULLDUP . SC_HALFDUP가
 device SCbus time slot listening
 device .

□

1. devtype1 devtype2 devh1 devh2 device
 type match .
2. sctools.c DTISC FAXSC
 가 digital time slot SPM station, fax
 route .

□ **Source Code**

```
/*
 * Include files.
 */
#include <windows.h>
```

```

#include <stdio.h>
#include <srllib.h>
#include <vpmlib.h>
#ifdef DTISC
#include <spmlib.h>
#endif
#ifdef FAXSC
#include <faxlib.h>
#endif
#include "sctools.h"
/*
 * Function prototypes
 */
static void nr_scerrror(char *, ...);
#if ( defined( __STDC__ ) || defined( __cplusplus ) )
int nr_scuroute( int devh1, unsigned short devtype1,
                 int devh2, unsigned short devtype2, unsigned char mode )

#else
int nr_scuroute( devh1, devtype1, devh2, devtype2, mode )
    int      devh1;
    unsigned short devtype1;
    int      devh2;
    unsigned short devtype2;
    unsigned char mode;
#endif
{
    short rc = 0;          /* Return code from the function */
    /*
     * Disconnect the receive of the second device from the SCbus timeslot.
     */
    switch (devtype2) {
    case SC_DSP:
        if (vpm_unlisten(devh2) == -1) {
            nr_scerrror("nr_scuroute: %s: vpm_unlisten ERROR: %s\n",
                        ATDV_NAMEP(devh2), ATDV_ERRMSGP(devh2));

```

```

        rc = -1;
    }
    break;
case SC_CODEC:
    if (vag_unlisten(devh2) == -1) {
        nr_scerror("nr_scunroute: %s: vag_unlisten ERROR: %s\n",
            ATDV_NAMEP(devh2), ATDV_ERRMSGP(devh2));
        rc = -1;
    }
    break;
case SC_SPM:
    if (spm_unlisten(devh2) == -1) {
        nr_scerror("nr_scunroute: %s: spm_unlisten ERROR: %s\n",
            ATDV_NAMEP(devh2), ATDV_ERRMSGP(devh2));
        rc = -1;
    }
    break;
#endif
#ifdef FAXSC
case SC_FAX:
    if (fpm_unlisten(devh2) == -1) {
        nr_scerror("nr_scunroute: %s: fpm_unlisten ERROR: %s\n",
            ATDV_NAMEP(devh2), ATDV_ERRMSGP(devh2));
        rc = -1;
    }
    break;
#endif
default:
    nr_scerror("nr_scunroute: %s: ERROR: Invalid 2nd device type\n",
        ATDV_NAMEP(devh2));
    rc = -1;
}
/*
 * A half duplex connection has already been broken. If this is all that
 * is required, then return now.
 */

```



```

if (mode == SC_HALFDUP) {
    return rc;
}
/*
 * Disconnect the receive of the first device from the SCbus timeslot.
 */
switch (devtype1) {
case SC_DSP:
    if (vpm_unlisten(devh1) == -1) {
        nr_scerror("nr_scunroute: %s: vpm_unlisten ERROR: %s\n",
                    ATDV_NAMEP(devh1), ATDV_ERRMSGP(devh1));
        rc = -1;
    }
    break;
case SC_CODEC:
    if (vag_unlisten(devh1) == -1) {
        nr_scerror("nr_scunroute: %s: vag_unlisten ERROR: %s\n",
                    ATDV_NAMEP(devh1), ATDV_ERRMSGP(devh1));
        rc = -1;
    }
    break;
case SC_SPM:
    if (spm_unlisten(devh1) == -1) {
        nr_scerror("nr_scunroute: %s: spm_unlisten ERROR: %s\n",
                    ATDV_NAMEP(devh1), ATDV_ERRMSGP(devh1));
        rc = -1;
    }
    break;
#endif
#ifdef FAXSC
case SC_FAX:
    if (fpm_unlisten(devh1) == -1) {
        nr_scerror("nr_scunroute: %s: fpm_unlisten ERROR: %s\n",
                    ATDV_NAMEP(devh1), ATDV_ERRMSGP(devh1));
        rc = -1;
    }
}

```

```

        break;
#endif
    default:
        nr_scerror("nr_scunroute: %s: ERROR: Invalid 1st device type\n",
                  ATDV_NAMEP(devh1));
        rc = -1;
    }
    return rc;
}

void nr_scerror(char *fmt, ...)
{
#ifdef PRINTON
    va_list args;
    char    *fmt;
    /*
     * Make args point to the 1st unnamed argument and then print to stderr.
     */
    va_start(args, fmt);
    fmt = va_arg(args, char *);
    vfprintf(stderr, fmt, args);
    va_end(args);
#endif
}

□
• nr_scunroute ( )

```

3. SCbus Function Reference

SCbus Function Reference Overview

Scbus	device	device	receive
	library function	reference	
.			

SCbus Function Description

vag_getxmislots()

provides Scbus time slot number

Name : int vag_getxmislots (chdev, sc_tsinfo)

Inputs :

int chdev : analog channel handle

SC_TSINFO *sc_tsinfo : pointer to SCbus time slot information structure

Returns :

0 :

-1 :

Includes: vpmllib.h

Category: SCbus Routing

Mode: Synchronous

□

vag_getxmislots()

SCbus time slot number

Scbus

Scbus time slot

SC_TSINFO

NOTE: Scbus nr_scroute() vag_getxmislots()

Convenience

Scbus Routing Guide

Parameter

Chdev : vpm_open

sc_tsinfo: SC_TSINFO data structure pointer.

SC_TSINFO

```
typedef struct {  
    unsigned long  sc_numts;  
    long  *sc_tsarrayp;  
} SC_TSINFO;
```

```

SC_TSINFO          sc_numts          1  Scbus time slots
가                  . SC_TSINFO      sc_tsarrayp      long
.
Scbus time slot    (0  1023  )
.

```

□

:

~~SC~~

□ Example

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

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

```
#include <vpmlib.h>
```

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

```
main
```

```
{
```

```
    int chdev;          /* Channel device handle */
```

```
    SC_TSINFO sc_tsinfo; /* Time slot information structure */
```

```
    long scts;          /* SCbus time slot */
```

```
/* Open board 1 channel 1 devices */
```

```
if ((chdev = vpm_open("vpmB1C1", 0)) == -1) {
```

```
    printf("Cannot open channel vpmB1C1.  errno = %d", errno);
```

```
    exit(1);
```

```
}
```

```
/* Fill in the SCbus time slot information */
```

```
sc_tsinfo.sc_numts = 1;
```

```
sc_tsinfo.sc_tsarrayp = &scts;
```

```
/* Get SCbus time slot connected to transmit of analog channel 1 on board ...1 */
```

```
if (vag_getxmitslot(chdev, &sc_tsinfo) == -1) {
```

```
    printf("Error message = %s", ATDV_ERRMSGP(chdev));
```

```
    exit(1);
```

```
}
```

```
printf("%s is transmitting on SCbus time slot %d", ATDV_NAMEP(chdev), scts);
```

}

□ Error

가 -1, error code SRL Standard Attribute
 ATDV_LASTERR() descriptive error message
 ATDV_ERRMSGP() .
 error code :

Equate	Returned When
EDX_BADPARAM	Parameter error
EDX_SH_BADCMD	Command is not supported in current bus configuration
EDX_SH_BADINDX	Invalid Switch Handler library index number
EDX_SH_BADLCTS	Invalid channel number
EDX_SH_BADMODE	Function not supported in current bus configuration
EDX_SH_BADTYPE	Invalid channel type (voice, analog, etc.)number
EDX_SH_CMDBLOCK	Blocking command is in progress
EDX_SH_LCLDSCNCT	Channel is already disconnected from SCbus time slot
EDX_SH_LIBBSY	Switch Handler library busy
EDX_SH_LIBNOTINIT	Switch Handler library uninitialized
EDX_SH_MISSING	Switch Handler is not present
EDX_SH_NOCLK	Switch Handler clock fallback failed
EDX_SYSTEM	Windows 2000 system error

□

~~del~~ vpm_listen()
~~del~~ fpm_listen()
~~del~~ spm_listen()

vag_listen() connects analog receive channel to Scbus time slot

Name : int vag_listen (chdev, sc_tsinfo)

Inputs :

int chdev : analog channel handle

SC_TSINFO *sc_tsinfo : pointer to Scbus time slot information structure

Returns :

0 :

-1 :

Includes: vpmllib.h

Category: Scbus Routing

Mode: Synchronous

□

vag_listen () SCbus time slot
SCbus time slot SC_TSINFO
on-board 가 single
default SCbus routing
SCbus Routing Guide

NOTE: Scbus **nr_scroute()** **vag_listen()**
Convenience SCbus Routing
Guide
Parameter

Chdev : vpm_open

sc_tsinfo: SC_TSINFO data structure pointer.

SC_TSINFO

```
typedef struct {
    unsigned long  sc_numts;
    long  *sc_tsarrayp;
} SC_TSINFO;
```

```
SC_TSINFO          sc_numts          1    Scbus time slots
가                  . SC_TSINFO      sc_tsarrayp
                  Scbus time slot
                  .                  ,          SCbus
time slot          .  sc_tsarray
SCbus time slot    xx_getxmitslot()          vag_listen()
```

```
vpm400, vpm800 or vpm1600          SCbus time slot
SCbus time slot
```

□

:

~~SCbus~~

~~time~~

SCbus time slot

□ Example

```
#include <windows.h>
#include <srllib.h>
#include <vpmlib.h>
#include <errno.h>
```

```
main
```

```
{
    int chdev;          /* Channel device handle */
    SC_TSINFO sc_tsinfo; /* Time slot information structure */
    long scts;          /* SCbus time slot */
    /* Open board 1 channel 1 devices */
    if ((chdev = vpm_open("vpmB1C1", 0)) == -1) {
```



```

        printf("Cannot open channel vpmB1C1.  errno = %d", errno);
        exit(1);
    }
/* Fill in the SCbus time slot information */
    sc_tsinfo.sc_numts = 1;
    sc_tsinfo.sc_tsarrayp = &scts;

/* Get SCbus time slot connected to transmit of voice channel 1 on board 1 */
    if (vag_getxmitslot(chdev, &sc_tsinfo) == -1) {
        printf("Error message = %s", ATDV_ERRMSGP(chdev));
        exit(1);
    }
/* Connect the receive of analog channel 1 on board 1 to SCbus
time slot of voice channel 1 */
    if (vag_listen(chdev, &sc_tsinfo) == -1) {
        printf("Error message = %s", ATDV_ERRMSGP(chdev));
        exit(1);
    }
}

```

□ Error

가 -1 , error code SRL Standard Attribute
 ATDV_LASTERR() descriptive error message
 ATDV_ERRMSGP() .
 error code :

Equate	Returned When
EDX_BADPARM	Parameter error
EDX_SH_BADCMD	Command is not supported in current bus configuration
EDX_SH_BADEXTTS	SCbus time slot is not supported at current clock rate
EDX_SH_BADINDX	Invalid Switch Handler index number
EDX_SH_BADLCLTS	Invalid channel number
EDX_SH_BADMODE	Function not supported in current bus configuration
EDX_SH_BADTYPE	Invalid channel local time slot type (voice, analog, etc.)
EDX_SH_CMDBLOCK	Blocking command is in progress
EDX_SH_LCLTSCNCT	Channel is already connected to SCbus time slot

EDX_SH_LIBBSY	Switch Handler library busy
EDX_SH_LIBNOTINIT	Switch Handler library uninitialized
EDX_SH_NOCLK	Switch Handler clock fallback failed
EDX_SYSTEM	Windows 2000 system error

□

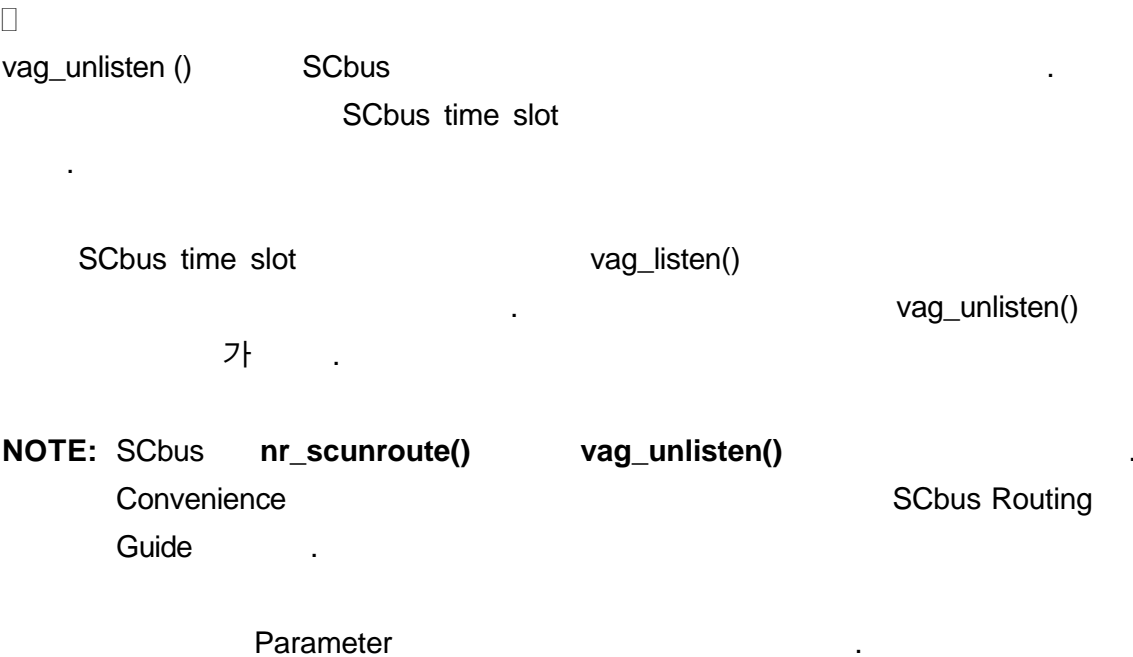
~~✂~~ vpm_getxmitslot()

~~✂~~ fpm_getxmitslot ()

~~✂~~ vag_unlisten()

vag_unlisten() disconnects analog receive channel from SCbus

Name : int vag_unlisten (chdev)
Inputs :
int chdev : analog channel handle
Returns :
0 :
-1 :
Includes: vpmplib.h
Category: SCbus Routing
Mode: Synchronous



Chdev : vpm_open

□

:

SS

□ Example

```
#include <windows.h>
#include <srllib.h>
#include <vpmlib.h>
#include <errno.h>

main
{
    int chdev;                                /* Voice Channel handle */
    /* Open board 1 channel 1 device */
    if ((chdev = vpm_open("vpmB1C1", 0)) == -1) {
        printf("Cannot open channel vpmB1C1.  errno = %d", errno);
        exit(1);
    }
    /* Disconnect receive of board 1, channel 1 from SCbus time slots */
    if (vag_unlisten(chdev) == -1) {
        printf("Error message = %s", ATDV_ERRMSGP(chdev));
        exit(1);
    }
}
```


□ Error

가 -1, error code SRL Standard Attribute
 ATDV_LASTERR() descriptive error message
 ATDV_ERRMSGP() .
 error code :

Equate	Returned When
EDX_BADPARAM	Parameter error
EDX_SH_BADCMD	Command is not supported in current bus configuration
EDX_SH_BADINDX	Invalid Switch Handler index number
EDX_SH_BADLCLTS	Invalid channel number
EDX_SH_BADMODE	Function not supported in current bus configuration
EDX_SH_BADTYPE	Invalid channel local time slot type (voice, analog, etc.)
EDX_SH_CMDBLOCK	Blocking command is in progress
EDX_SH_LCLDSCNCT	Channel already disconnected from SCbus time slot

EDX_SH_LIBBSY	Switch Handler library busy
EDX_SH_LIBNOTINIT	Switch Handler library uninitialized
EDX_SH_MISSING	Switch Handler is not present
EDX_SH_NOCLK	Switch Handler clock fallback failed
EDX_SYSTEM	Windows 2000 system error

□

 vag_listen()

vpm_getxmislot() provides SCbus time slot number

Name : int vpm_getxmislot (chdev, sc_tsinfo)
Inputs :
 int chdev : voice channel handle
 SC_TSINFO *sc_tsinfo : pointer to SCbus time slot information structure
Returns :
 0 :
 -1 :
Includes: vpmllib.h
Category: Scbus Routing
Mode: Synchronous

□
vag_getxmislot() SCbus time slot number ..
SCbus time slot Scbus time slot
SC_TSINFO .

NOTE: Scbus nr_scroute() vpm_getxmislot()
.
Convenience SCbus Routing
Guide .

Parameter .

Chdev : vpm_open
sc_tsinfo: SC_TSINFO data structure pointer.

SC_TSINFO .

```
typedef struct {  
    unsigned long  sc_numts;  
    long  *sc_tsarrayp;  
} SC_TSINFO;
```

```

SC_TSINFO          sc_numts          SCbus time slots
                    (1 for a voice channel). SC_TSINFO          sc_tsarrayp
long
                    .
                    SCbus time slot          (0
1023          )
                    .
                    SCbus time slot          .

```

□

:

~~SC~~

□ Example

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

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

```
#include <vpmlib.h>
```

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

```
main
```

```
{
```

```
    int chdev;          /* Channel device handle */
```

```
    SC_TSINFO sc_tsinfo; /* Time slot information structure */
```

```
    long scts;          /* SCbus time slot */
```

```
/* Open board 1 channel 1 devices */
```

```
if ((chdev = vpm_open("vpmB1C1", 0)) == -1) {
```

```
    printf("Cannot open channel vpmB1C1.  errno = %d", errno);
```

```
    exit(1);
```

```
}
```

```
/* Fill in the SCbus time slot information */
```

```
sc_tsinfo.sc_numts = 1;
```

```
sc_tsinfo.sc_tsarrayp = &scts;
```

```
/* Get SCbus time slot connected to transmit of analog channel 1 on board ...1 */
```

```
if (vpm_getxmitslot(chdev, &sc_tsinfo) == -1) {
```

```
    printf("Error message = %s", ATDV_ERRMSGP(chdev));
```

```
    exit(1);
```

```

    }
    printf("%s is transmitting on SCbus time slot %d", ATDV_NAMEP(chdev), ...scts);
}

```

❑ **Error**

가 -1 , error code SRL Standard Attribute
 ATDV_LASTERR() descriptive error message
 ATDV_ERRMSGP() .
 error code :

Equate	Returned When
EDX_BADPARM	Parameter error
EDX_SH_BADCMD	Command is not supported in current bus configuration
EDX_SH_BADINDX	Invalid Switch Handler library index number
EDX_SH_BADLCTS	Invalid channel number
EDX_SH_BADMODE	Function not supported in current bus configuration
EDX_SH_BADTYPE	Invalid channel type (voice, analog, etc.)number
EDX_SH_CMDBLOCK	Blocking command is in progress
EDX_SH_LCLDSCNCT	Channel is already disconnected from SCbus time slot
EDX_SH_LIBBSY	Switch Handler library busy
EDX_SH_LIBNOTINIT	Switch Handler library uninitialized
EDX_SH_MISSING	Switch Handler is not present
EDX_SH_NOCLK	Switch Handler clock fallback failed
EDX_SYSTEM	Windows 2000 system error

❑

~~del~~ vag_listen()

~~del~~ fpm_listen()

vpm_listen() connects voice receive channel to Scbus time slot

Name : int vpm_listen (chdev, sc_tsinfo)

Inputs :

int chdev : voice channel handle

SC_TSINFO *sc_tsinfo : pointer to SCbus time slot information structure

Returns :

0 :

-1 :

Includes: vpmllib.h

Category: Scbus Routing

Mode: Synchronous

□

vpm_listen ()	SCbus time slot	.
	SCbus time slot	SC_TSINFO

NOTE: Scbus **nr_scroute()** **vpm_listen()** .
 Convenience Scbus Routing Guide

Parameter

Chdev : vpm_open

sc_tsinfo: SC_TSINFO data structure pointer.

SC_TSINFO

typedef struct {

unsigned long sc_numts;

long *sc_tsarrayp;

} SC_TSINFO;

SC_TSINFO	sc_numts	1	. SC_TSINFO
-----------	----------	---	-------------

```

sc_tsarrayp
    xx_getxmitslot()                                0    1023
SCbus time slot                                     . vpm_listen()
    SCbus time slot                                ..
    SCbus time slot
    SCbus time slot
    SCbus time slot

```

□

:

~~SCbus time slot~~

~~SCbus time slot~~

□ Example

```

#include <windows.h>
#include <srllib.h>
#include <vpmlib.h>
#include <errno.h>

main
{
    int chdev;                                /* Channel device handle */
    SC_TSINFO sc_tsinfo;                      /* Time slot information structure */
    long scts;                                /* SCbus time slot */
    /* Open board 1 channel 1 devices */
    if ((chdev = vpm_open("vpmB1C1", 0)) == -1) {
        printf("Cannot open channel vpmB1C1.  errno = %d", errno);
        exit(1);
    }
    /* Fill in the SCbus time slot information */
    sc_tsinfo.sc_numts = 1;
    sc_tsinfo.sc_tsarrayp = &scts;

    /* Get SCbus time slot connected to transmit of voice channel 1 on board 1 */
    if (vpm_getxmitslot(chdev, &sc_tsinfo) == -1) {
        printf("Error message = %s", ATDV_ERRMSGP(chdev));
    }
}

```

```

        exit(1);
    }
/* Connect the receive of analog channel 1 on board 1 to SCbus
time slot of voice channel 1 */
if (vpm_listen(chdev, &sc_tsinfo) == -1) {
    printf("Error message = %s", ATDV_ERRMSGP(chdev));
    exit(1);
}
}

```

□ Error

가 -1, error code SRL Standard Attribute
 ATDV_LASTERR() descriptive error message
 ATDV_ERRMSGP() .
 error code :

Equate	Returned When
EDX_BADPARM	Parameter error
EDX_SH_BADCMD	Command is not supported in current bus configuration
EDX_SH_BADEXTTS	SCbus time slot is not supported at current clock rate
EDX_SH_BADINDX	Invalid Switch Handler index number
EDX_SH_BADLCLTS	Invalid channel number
EDX_SH_BADMODE	Function not supported in current bus configuration
EDX_SH_BADTYPE	Invalid channel local time slot type (voice, analog, etc.)
EDX_SH_CMDBLOCK	Blocking command is in progress
EDX_SH_LCLTSCNCT	Channel is already connected to SCbus time slot
EDX_SH_LIBBSY	Switch Handler library busy
EDX_SH_LIBNOTINIT	Switch Handler library uninitialized
EDX_SH_NOCLK	Switch Handler clock fallback failed
EDX_SYSTEM	Windows 2000 system error

□

~~///~~ vag_getxmitslot()
~~///~~ fpm_getxmitslot ()
~~///~~ vpm_unlisten()

vpm_unlisten() disconnects voice receive channel from SCbus

Name : int vpm_unlisten (chdev)

Inputs :

int chdev : voice channel handle

Returns :

0 :

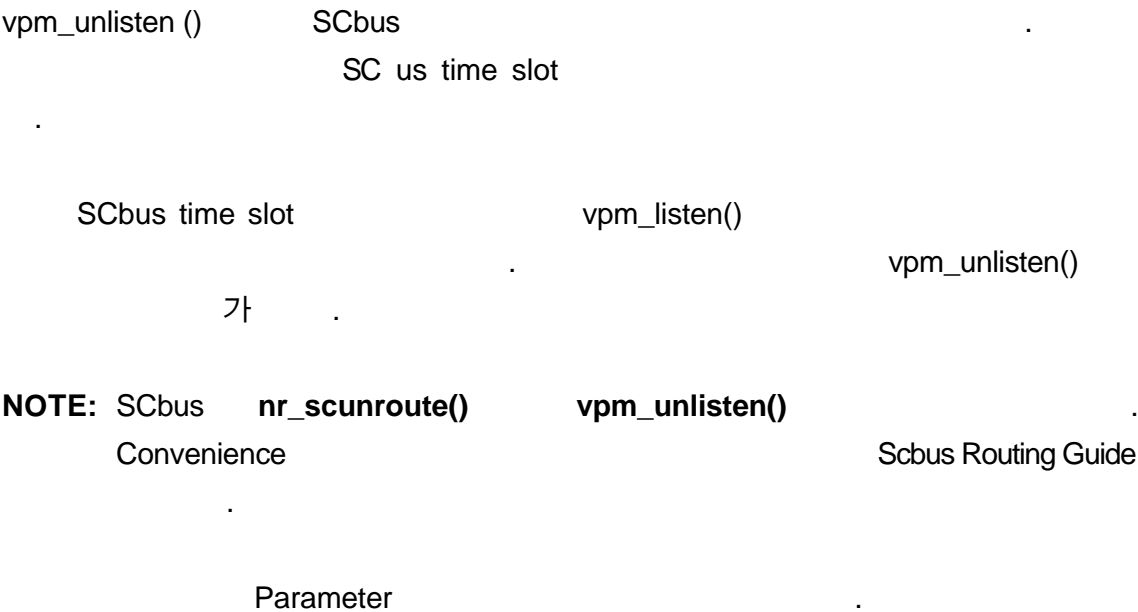
-1 :

Includes: vpmllib.h

Category: SCbus Routing

Mode: Synchronous

□



Chdev : vpm_open

□

:

SS

□ Example

```
#include <windows.h>
#include <srllib.h>
#include <vpmlib.h>
#include <errno.h>

main
{
    int chdev;                                /* Voice Channel handle */
    /* Open board 1 channel 1 device */
    if ((chdev = vpm_open("vpmB1C1", 0)) == -1) {
        printf("Cannot open channel vpmB1C1.  errno = %d", errno);
        exit(1);
    }
    /* Disconnect receive of board 1, channel 1 from SCbus time slots */
    if (vpm_unlisten(chdev) == -1) {
        printf("Error message = %s", ATDV_ERRMSGP(chdev));
        exit(1);
    }
}
```

□ Error

가 -1, error code SRL Standard Attribute
 ATDV_LASTERR() descriptive error message
 ATDV_ERRMSGP() .
 error code :

Equate	Returned When
EDX_BADPARAM	Parameter error
EDX_SH_BADCMD	Command is not supported in current bus configuration
EDX_SH_BADINDX	Invalid Switch Handler index number
EDX_SH_BADLCLTS	Invalid channel number
EDX_SH_BADMODE	Function not supported in current bus configuration
EDX_SH_BADTYPE	Invalid channel local time slot type (voice, analog, etc.)
EDX_SH_CMDBLOCK	Blocking command is in progress
EDX_SH_LCLDSCNCT	Channel already disconnected from SCbus time slot

EDX_SH_LIBBSY	Switch Handler library busy
EDX_SH_LIBNOTINIT	Switch Handler library uninitialized
EDX_SH_MISSING	Switch Handler is not present
EDX_SH_NOCLK	Switch Handler clock fallback failed
EDX_SYSTEM	Windows 2000 system error

□

? vpm_listen()

fpm_getxmitslot() provides SCbus time slot number

Name : int fpm_getxmitslot (dev,sc_tsinfo)

Inputs :

int dev : FAX channel device handle

SC_TSINFO *sc_tsinfo : pointer to SCbus time slot information structure

Returns :

0 :

-1 :

Includes: srlib.h dxxlib.h faxlib.h

Category: SCbus Routing

Mode: synchronous

□

fpm_getxmitslot() FAX SCbus time slot number .
FAX SCbus time slot number
SC_TSINFO SCbus time slot .
NOTE : SCbus convenience nr_scroute() fpm_getxmitslot()

Convenience function SCbus Routing Guide

Parameter

dev : FAX device 가 fpm_open() open
channel device handle .

sc_tsinfo : data structure SC_TSINFO pointer .

sc_TSINFO structure .

```
typedef struct {  
    unsigned long    sc_numts;  
    long             *sc_tsarrayp;  
} SC_TSINFO;
```

```

SC_TSINFO structure    sc_numts member          SCbus time slot    number
        . SC_TSINFO structure    sc_tsarrayp    member    valid
        pointer    가
        fpm channel          SCbus time slot    0    1023
number
        .

```

□

```

        fail
? invalid FAX channel device handle

```

□ Example

```

#include <windows.h>
#include <srllib.h>
#include <vpmlib.h>
#include <fpmlib.h>
#include <errno.h>
main( )
{
    int dev; /* Fax channel device handle. */
    SC_TSINFO sc_tsinfo; /* Timeslot information structure. */
    long scts; /* SCbus time slots. */
    .
    .
    /* Open the FAX channel resource device. */
    if ((dev = fpm_open("vpmB7C1", NULL)) == -1) {
        /* Error opening device. */
        printf("Error opening channel, errno = %d \n", errno);
        exit(1);
    }
    /* Fill in the SC_TSINFO structure time slot information. */
    sc_tsinfo.sc_numts = 1;
    sc_tsinfo.sc_tsarray = &scts;
    /* Get FAX device channel SCbus transmit time slot. */
    if (fpm_getxmitslot(dev, &sc_tsinfo) == -1) {
        printf("Error message = %s", ATDV_ERRMSGP(dev));
    }
}

```



```

        exit(1);
    }
    printf("Fax channel is transmitting on SCbus time slot %d \n", scts);
    .
    .

```

□ Error

ATDV_LASTERR() ATDV_ERRMSGP() error
 voice software reference . Error message

Equate	Returned When
EDX_BADPARM	Parameter error
EDX_SH_BADCMD	Command is not supported in current bus configuration
EDX_SH_BADINDX	Invalid Switch Handler index number
EDX_SH_BADLCLTS	Invalid channel number
EDX_SH_BADMODE	Function not supported in current bus configuration
EDX_SH_BADTYPE	Invalid channel type (voice, analog, etc.)
EDX_SH_CMDBLOCK	Blocking command is in progress
EDX_SH_LCLDSCNCT	Channel is already disconnected from SCbus
EDX_SH_LIBBSY	Switch Handler library busy
EDX_SH_LIBNOTINIT	Switch Handler library uninitialized
EDX_SH_MISSING	Switch Handler is not present
EDX_SH_NOCLK	Switch Handler clock fallback failed
EDX_SYSTEM	Windows 2000 System Error

□

- vag_listen()
- vpm_listen()
- fpm_listen()

connects FAX listen channel to SCbus time slot

fpm_listen()

Name : int fpm_listen(dev,sc_tsinfop)

Inputs :

int dev : FAX channel device handle
SC_TSINFO *sc_tsinfop : pointer to SCbus time slot information structure

Returns :

0 :
-1 :

Includes: srllib.h vpmllib.h fpmllib.h

Category: SCbus Routing

Mode: synchronous

□

fpm_listen() FAX listen channel SCbus time slot
FAX receive SCbus time slot SC_TSINFO
device receive FAX
NOTE: SCbus convenience function nr_scroute() fpm_listen()
Convenience function SCbus
Routing Guide

Parameter

dev : channel fpm_open() open valid
FAX channel device handle
sc_tsinfop: SC_TSINFO data structure pointer

SC_TSINFO structure

```
typedef struct {
    unsigned long sc_numts;
    long *sc_tsarray;
} SC_TSINFO;
```

```

SC_TSINFO structure    sc_numts    member    1    . SC_TSINFO
structure    sc_tsarrayp field    valid array    pointer
.
.    xx_getxmitslot() function
.    0    1023    valid SCbus time slot number    .
fpm_listen()    FAX receive    time slot
.

```

```

multiple SCbus device channel    time slot    FAX
receive channel    SCbus time slot    .

```

□

```

fail    .
? invalid FAX channel device handle
? invalid SCbus time slot

```

□ Example

```

#include <windows.h>
#include <srllib.h>
#include <vpmlib.h>
#include <fpmlib.h>
#include <errno.h>
main( )
{
    int voxdev; /* Voice channel device handle. */
    int dev; /* Fax channel device handle. */
    SC_TSINFO sc_tsinfo; /* SCbus time slot information structure. */
    long scts; /* SCbus time slot. */
    .
    .
    /* Open the FAX channel device. */
    if ((dev = fpm_open("vpmB7C1", NULL)) == -1) {
        /* Error opening device. */
        printf("Error opening channel, errno = %d \n", errno);
        exit(1);
    }
    /* Open the VOICE channel device on the D/160SC-LS. */

```

```

if ((voxdev = vpm_open("vpmB1C1", NULL)) == -1) {
    /* Error opening device. */
    printf("Error opening channel, errno = %d \n", errno);
    exit(1);
}
.
.
/*
* Break the full-duplex connection between the Voice
* channel device and the Network analog device.
* Use the SCbus routing convenience function nr_scunroute( ).
*/
if (nr_scunroute(voxdev, SC_DSP, voxdev, SC_CODEC, SC_FULLDUP) == -1) {
    /* Error during SCbus unrout. */
    printf("Error unrouting channel \n");
    printf("Error - %s (error code %d) \n",
        ATDV_ERRMSGP(voxdev), ATDV_LASTERR(voxdev));
    if (ATDV_LASTERR(voxdev) == EDX_SYSTEM) {
        printf("errno = %d \n", errno);
    }
}
/*
* Set full-duplex connection between the FAX
* channel device and the Network analog device.
*/
/* Fill in the SC_TSINFO structure time slot information. */
sc_tsinfo.sc_numts = 1;
sc_tsinfo.sc_tsarray = &scts;
/* Get Network analog device's SCbus transmit time slot. */
if (vag_getxmitslot(voxdev, &sc_tsinfo) == -1) {
    printf("Error message = %s", ATDV_ERRMSGP(voxdev));
    exit(1);
}
/*
* Connect the FAX channel to "listen" to the Network
* channel's SCbus transmit time slot. Pass the time slot

```

```

* information in the SC_TSINFO structure to fpm_listen( ).
*/
if (fpm_listen(dev, &sc_tsinfo) == -1) {
    printf("Error message = %s", ATDV_ERRMSGP(dev));
    exit(1);
}
.
.
/* Complete full-duplex connection between the FAX channel device
* and the Network channel device using fpm_getxmitslot( )
* and vag_listen( ).
*/
.
.
/* Call FAX API functions for FAX transfers. */
.
.

```

Error

가 -1 error
ATDV_LASTERR() ATDV_ERRMSGP() . Error message
voice software reference .

Equate	Returned When
EDX_BADPARAM	Parameter error
EDX_SH_BADCMD	Command is not supported in current bus configuration
EDX_SH_BADEXTTS	SCbus time slot is not supported at current clock rate
EDX_SH_BADINDX	Invalid Switch Handler index number
EDX_SH_BADLCLTS	Invalid channel number
EDX_SH_BADMODE	Function not supported in current bus configuration
EDX_SH_CMDBLOCK	Blocking command is in progress

Equate	Returned When
EDX_SH_LCLTSCNCT	Channel is already connected to SCbus
EDX_SH_LIBBSY	Switch Handler library busy
EDX_SH_LIBNOTINIT	Switch Handler library uninitialized
EDX_SH_MISSING	Switch Handler is not present
EDX_SH_NOCLK	Switch Handler clock fallback failed
EDX_SYSTEM	Windows 2000 System Error

□

- **vag_getxmitslot()**
- **vpm_getxmitslot()**
- **fpm_unlisten()**

fpm_unlisten() disconnects FAX receive channel from SCbus

Name : int fpm_unlisten(dev)
Inputs :
int dev : FAX channel device handle
Returns :
0 :
-1 :
Includes: srllib.h vpmlib.h fpmplib.h
Category: SCbus Routing
Mode: synchronous

□
fpm_unlisten() FAX receive channel SCbus .
SCbus time slot fpm_listen()
existing . fpm_unlisten()
가 .
NOTE : SCbus convenience function nr_scroute() fpm_unlisten()
. Convenience function SCbus Routing
Guide .

Parameter .

dev : fpm_open() open FAX channel
device handle .

□
fail .
? invalid FAX channel device handle

□ **Example**
#include <windows.h>
#include <srllib.h>
#include <vpmlib.h>
#include <fpmplib.h>

```

#include <errno.h>
main( )
{
    int dev; /* Fax channel device handle. */
    .
    .
    /* Open the FAX channel resource. */
    if ((dev = fpm_open("vpmB7C1", NULL)) == -1) {
        /* Error opening device. */
        printf("Error opening channel, errno = %d \n", errno);
        exit(1);
    }
    .
    .
    /*
    * Disconnect the FAX channel device from "listening" to an
    * SCbus transmit time slot.
    */
    if (fpm_unlisten(dev) == -1) {
        printf("Error message = %s", ATDV_ERRMSGP(dev));
        exit(1);
    }
    .
    .

```

□ Error

ATDV_LASTERR() 가 -1 error . Error message
 ATDV_ERRMSGP() voice software reference .

Equate	Returned When
EDX_BADPARAM	Parameter error
EDX_SH_BADCMD	Command is not supported in current bus configuration
EDX_SH_BADEXTTS	SCbus time slot is not supported at current clock rate

Equate	Returned When
EDX_SH_BADINDX	Invalid Switch Handler index number
EDX_SH_BADLCLTS	Invalid channel number
EDX_SH_BADMODE	Function not supported in current bus configuration
EDX_SH_BADTYPE	Invalid channel type (voice, analog, etc.)
EDX_SH_CMDBLOCK	Blocking command is in progress
EDX_SH_LCLDSCNCT	Channel already disconnected from SCbus
EDX_SH_LIBBSY	Switch Handler library busy
EDX_SH_LIBNOTINIT	Switch Handler library uninitialized
EDX_SH_MISSING	Switch Handler is not present
EDX_SH_NOCLK	Switch Handler clock fallback failed
EDX_SYSTEM	Windows 2000 System Error

□

• **fpm_listen()**

spm_getxmitslot() returns the SCbus time slot information

Name : int spm_ getxmitslot (devh,tsinfop)

Inputs :

int devh : station handle
SC_TSINFO *tsinfop : pointer to SCbus time slot information structure

Returns :

0 :
-1 :

Includes: srllib.h dtilib.h spmlib.h

Category: SCbus Routing

Mode: asynchronous

station	SCbus time slot	number	SC_TSINFO
structure	SCbus time slot	.	
Parameter			
.			

devh :station device handle
tsinfop: SCbus time slot information structure pointer

SC_TSINFO structure station transmit
SCbus time slot number . 0 1023
SCbus time slot array point . SC_TSINFO structure
.

typedef struct {
 unsigned long sc_numts;
 long *sc_tsarrayp;
} SC_TSINFO;

SPM/SC station device SC_TSINFO sc_numts 1
 . Sc_tsarrayp long type point .

station data SCbus time slot

NOTE : SPM/SC station device external SCbus time slot

□

fail

? invalid station device handle

spm_getxmitslot() 가 station valid
 SCbus time slot data station valid
 SCbus time slot

□ Example

```
#include "srllib.h"
```

```
#include "dtilib.h"
```

```
#include "spmlib.h"
```

```
#include "errno.h"
```

```
int chdev; /* Station dev descriptor */
```

```
SC_TSINFO tsinfo; /* Time slot information structure */
```

```
long scts; /* SCbus time slot */
```

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

```
if ((chdev1 = spm_open("spmB1C1",0)) == -1) {
    printf( "Cannot open SPM B1, C1: errno=%d", errno);
    exit(1);
}
```

```
/* Set up SC_TSINFO structure */
```

```
tsinfo.sc_numts = 1;
```

```
tsinfo.sc_tsarray = &scts;
```

```
/* Get time slot on which SPM board 1, channel 1 is xmitting */
```

```
if (spm_getxmitslot(chdev1,&tsinfo) == -1) {
    printf("Error message = %s", ATDV_ERRMSGP(chdev));
}
```

```

        exit(1);
    }
    printf("spmiB1C1 is transmitting on SCbus time slot %d",scts);

```

□ Error

If error occurs, the function returns -1. The error code is defined by the SRL Standard. The error message is returned by the function `ATDV_LASTERR()`.

□

• `spm_listen()`

spm_listen()

Connects the receive of a station

Name : int spm_listen(devh,tsinfop)

Inputs :

int devh : station handle

SC_TSINFO *tsinfop : pointer to the SCbus time slot information structure

Returns :

0 :

-1 :

Includes: srlib.h spmlib.h

Category: SCbus Routing

Mode: asynchronous

□

spm_listen() station receive SCbus time slot .
digital receive channel SCbus time slot SC_TSINFO
structure .
device receive digital transit

Parameter

devh :board device handle

tsinfop : SCbus time slot information structure pointer

SC_TSINFO structure pointer pass .
field field SCbus time slot
station device receive
SCbus time slot pointer . SC_TSINFO structure

```
typedef struct {  
    unsigned long sc_numts;  
    long *sc_tsarrayp;  
} SC_TSINFO;
```

```

spm station device      SC_TSINFO      sc_numts      1
      . Sc_tsarrayp      type long      point      .
      appropriate      xx_getxmitslot()
0      1023      valid SCbus time slot number      .

```

NOTE : 1. SPM/SC station device time slot
, multiple device time slot
.

□ fail .
? invalid station handle
? SCbus time slot number 가 invalid

□ Example

```

#include "srllib.h"
#include "spmlib.h"
#include "errno.h"

int chdev1,vpmdev4; /* Chan dev descriptor variables */
SC_TSINFO tsinfo; /* Time slot information structure */
long scts; /* SCbus time slot */

/* Open board 1, channel 1 device */
if ((chdev1 = spm_open("spmB1C1",0)) == -1) {
    printf( "Cannot open SPM B1, C1: errno=%d", errno);
    exit(1);
}

/* Open board 1, time slot 4 device */
if ((vpmdev1 = vpm_open("vpmB1C4",0)) == -1) {
    exit(1);
}

/* Set up SC_TSINFO structure */
tsinfo.sc_numts = 1;
tsinfo.sc_tsarray = &scts;

```

```

/* Get time slot on which DTI board 1, time slot 4 is xmitting */
if (vpm_getxmitslot(vpmdev4,&tsinfo) == -1) {
    printf("Error message = %s", ATDV_ERRMSGP(chdev));
    exit(1);
}
/* Make SPM board 1, station 1 listen to transmit time slot
if (spm_listen(chdev1,&tsinfo) == -1) {
    printf("Error Message = %s",ATDV_ERRMSGP(tsdev4));
    exit(1);
}

```

Error

가	error	가	-1	error code	SRL Standard
attributed	ATDV_LASTERR()				.
	descriptive error message		function		

- **spm_getxmitslot()**
- **spm_unlisten()**

spm_unlisten()

disconnects the receive of a station

Name : int spm_unlisten(devh)

Inputs :

int devh : station handle

Returns :

0 :

-1 :

Includes: srllib.h dtilib.h spmlib.h

Category: SCbus Routing

Mode: asynchronous



spm_unlisten()	SCbus time slot	station	receive	.
----------------	-----------------	---------	---------	---

Parameter

devh :station device handle



fail .

? invalid station device handle

Example

```
#include "srllib.h"
```

```
#include "dxxxlib.h"
```

```
#include "dtilib.h"
```

```
int chdev /* Station device handle */
```

```
/* Open board 1, channel 1 */
```

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

```
    printf("Cannot open channel spmB1C1. errno = %d", errno);
```

```
    exit(1);
```

```
}
```

```
/* Disconnect receive of board 1, station 1 from all SCbus time slots */
```



```

if (spm_unlisten(chdev) == -1) {
    printf("Error message = %s",ATDV_ERRMSGP(chdev));
    exit(1);
}

```

Error

가	-1	error code	error code
SRL Standard attribute function		.	descriptive error
message ATDV_ERRMSGP()			.

· spm_listen()

4. SCbus Data Structure Reference

SCbus Data Structure Overview

Scbus routing function
 . structure operation data structure
 voxlbr.h file .

 SCbus data structure SCbus time slot number device
 pointer SCbus time slot SCbus time slot
 information structure .

 xx_getxmitslot() data structure . xx_listen()
 device .

4.1 SCbus Time Slot INfOrmation (SC_TSINFO)

SC_TSINFO structure SCbus time slot device
 quantity . pointer
 SCbus time slot number hold . SC_TSINFO structure suffix
 SCbus routing .

 ? _getxmitslot() : device SCbus time slot
 ? _listen() : time slot information device

SC_TSINFO structure .

```
typedef struct {
    unsigned long    sc_numts;
    long far        *sc_tsarrayp;
} SC_TSINFO;
```

SC_TSINFO structure sc_numts field ?1 SCbus time slot
 number . Sc_tsarrayp field long integer
 pointer .

Appendix A

SCbus Routing Function Summary

vag_functions

vag_getctinfo()	analog device		
vag_getxmitslot()	analog number	structure	SCbus time slot SCbus time slot
vag_listen()	analog receive	SCbus time slot	
vag_unlisten()	SCbus time slot	analog receive channel	

vpm_functions

vpm_getctinfo()	voice device		
vpm_getxmitslot()	voice structure	scbus time slot	number SCbus time slot
vpm_listen()	voice receive channel	SCbus time slot	
vpm_unlisten()	SCbus time slot	voice receive	

fpm_functions

fpm_getxmitslot()	FAX structure	SCbus time slot	number SCbus time slot
fpm_listen()	FAX receive channel	SCbus time slot	
fpm_unlisten()	SCbus time slot	FAX receive	

spm_functions

spm_getxmitslot()	SPM station structure	SCbus time slot	number SCbus time slot
spm_listen()	SPM station	SCbus time slot	
spm_unlisten()	SPM station	SCbus time slot	

nr_functions

nr_scroute()

device

.

nr_scunroute()

device

.

Appendix B

SCbus-related Publications

Documentation

.

Software References

- Voice Programmer's Guide for Windows 2000
- FAX Programmer's Guide for Windows 2000
- SPM Software Reference for Windows 2000