

SCbus Routing Guide

Copyright © 2001 Dialogic Corporation

05-0289-007

COPYRIGHT NOTICE

Copyright © 2001 Dialogic Corporation. All Rights Reserved.

All contents of this document are subject to change without notice and do not represent a commitment on the part of Dialogic Corporation. Every effort is made to ensure the accuracy of this information. However, due to ongoing product improvements and revisions, Dialogic Corporation cannot guarantee the accuracy of this material, nor can it accept responsibility for errors or omissions. No warranties of any nature are extended by the information contained in these copyrighted materials. Use or implementation of any one of the concepts, applications, or ideas described in this document or on Web pages maintained by Dialogic may infringe one or more patents or other intellectual property rights owned by third parties. Dialogic does not condone or encourage such infringement. Dialogic makes no warranty with respect to such infringement, nor does Dialogic waive any of its own intellectual property rights which may cover systems implementing one or more of the ideas contained herein. Procurement of appropriate intellectual property rights and licenses is solely the responsibility of the system implementer. The software referred to in this document is provided under a Software License Agreement. Refer to the Software License Agreement for complete details governing the use of the software.

All names, products, and services mentioned herein are the trademarks or registered trademarks of their respective organizations and are the sole property of their respective owners. DIALOGIC (including the Dialogic logo), DTI/124, and SpringBoard are registered trademarks of Dialogic Corporation. A detailed trademark listing can be found at: <http://www.dialogic.com/legal.htm>.

Publication Date: September, 2001

Part Number: 05-0289-007

Dialogic, an Intel Company
1515 Route 10
Parsippany NJ 07054
U.S.A.

For **Technical Support**, visit the Dialogic support website at:
<http://support.dialogic.com>

For **Sales Offices** and other contact information, visit the main Dialogic website at:
<http://www.dialogic.com>

Table of Contents

1. SCbus Introduction	1
SCbus Product Overview	1
2. SCbus Routing of Resources.....	5
SCbus Routing Overview	5
2.1. Network/Resource SCbus Convenience Functions	6
2.2. Individual SCbus Routing Functions	6
2.3. Using SCbus Convenience Functions	7
2.4. Using Individual SCbus Routing Functions.....	7
3. SCbus Resource Routing Overview	11
Examples of SCbus Routing Resources	11
Connecting Devices using Individual SCbus Routing Functions.....	12
3.1. SCbus System Initialization	12
4. SCbus Analog Inbound/Outbound Call Routing Example	15
SCbus Routing Example Overview	15
4.1. Routing Using SCbus Convenience Functions	15
4.1.1. Windows Sample Code for SCbus Convenience Functions	19
4.1.2. Linux Sample Code for SCbus Convenience Functions.....	20
4.1.3. MS-DOS Sample Code for SCbus Convenience Functions	21
4.1.4. OS/2 Sample Code for SCbus Convenience Functions	22
4.2. Routing Using Individual SCbus Routing Functions	22
4.2.1. Windows Sample Code for Individual SCbus Routing Functions.....	23
4.2.2. Linux Sample Code for Individual SCbus Routing Functions	24
4.2.3. MS-DOS Sample Code for Individual SCbus Routing Functions.....	27
4.2.4. OS/2 Sample Code for Individual SCbus Routing Functions.....	29
5. SCbus Digital Service Inbound/Outbound Call Routing Example.....	31
SCbus Call Routing Example Overview	31
5.1. Routing Using SCbus Convenience Functions	31
5.1.1. Windows Sample Code for SCbus Convenience Functions	34
5.1.2. Linux Sample Code for SCbus Convenience Functions.....	35
5.1.3. MS-DOS Sample Code for SCbus Convenience Functions	36
5.1.4. OS/2 Sample Code for SCbus Convenience Functions	38
5.2. Routing Using Individual SCbus Routing Functions	38
5.2.1. Windows Sample Code for Individual SCbus Routing Functions.....	39
5.2.2. Linux Sample Code for Individual SCbus Routing Functions	40

5.2.3. MS-DOS Sample Code for Individual SCbus Routing Functions.....	42
5.2.4. OS/2 Sample Code for Individual SCbus Routing Functions.....	43
6. SCbus Fax Sharing Example	47
Fax Sharing Example Overview	47
6.1. Routing Using SCbus Convenience Functions	48
6.1.1. Windows Sample Code for SCbus Convenience Functions.....	50
6.1.2. Linux Sample Code for SCbus Convenience Functions.....	51
6.1.3. MS-DOS Sample Code for SCbus Convenience Functions.....	52
6.1.4. OS/2 Sample Code for SCbus Convenience Functions.....	55
6.2. Routing Using Individual SCbus Routing Functions	57
6.2.1. Win NT Sample Code for Individual SCbus Routing Functions.....	58
6.2.2. Linux Sample Code for Individual SCbus Routing Functions	60
6.2.3. MS-DOS Sample Code for Individual SCbus Routing Functions.....	62
6.2.4. OS/2 Sample Code for Individual SCbus Routing Functions.....	65
7. SCbus Special Considerations	69
SCbus Transmit Channel Assignment Overview	69
7.1. Voice Boards with On-board Analog Loop-Start Interfaces.....	69
7.2. Connecting SCbus Devices.....	70
7.3. SCbus Time Slot Information	70
SCbus Routing Function Listing.....	71
Appendix A - SCbus Routing Function Summary	71
Glossary	75
Index	79

List of Figures

Figure 1. System Initialization SCbus Time Slot Assignments	14
Figure 2. Initial Analog & Voice Device Routing Example	16
Figure 3. Analog Inbound/Outbound Routing Example	17
Figure 4. Initial Digital Service Routing Example.....	32
Figure 5. Digital Service Inbound/Outbound Routing Example	33
Figure 6. Initial Fax & Digital Channel Routing Example.....	47
Figure 7. Fax Sharing Example	49

1. SCbus Introduction

SCbus Product Overview

The SCbus is a real-time, high-speed, time division multiplexed (TDM) communications bus that provides 1024 time slots for transmission of digital information between SCbus products. The SCbus allows high-density systems to efficiently share resources so that multiple technologies can be connected to each port as needed.

Each SCbus product consists of several devices. Each of these devices can communicate via the SCbus with any other device connected to the SCbus. For example:

- a D/41ESC board provides 4 on-board analog loop start interface devices and 4 voice devices, for a total of 8 devices communicating over the SCbus.
- a D/160SC-LS board provides 16 on-board analog loop start interface devices and 16 voice devices, for a total of 32 devices communicating over the SCbus.
- a D/240SC-T1 board provides a T-1/DSX-1 interface for processing 24 T-1 time slots/calls and 24 voice devices, for a total of 48 devices communicating over the SCbus.
- a D/300SC-E1 board provides an E-1 interface for processing 30 E-1 time slots/calls and 30 voice devices, for a total of 60 devices communicating over the SCbus.
- a VFX/40ESC board provides 4 on-board analog loop start interface devices and 4 fax/voice devices, for a total of 8 devices communicating over the SCbus.
- a MSI/240SC board provides 24 on board station devices, for a total of 24 devices communicating over the SCbus.

All devices connected to the SCbus have a transmit (TX) channel and a receive (RX) (listen) channel. At system initialization, each transmit channel is assigned to a specific and unique SCbus time slot. This transmit channel assignment cannot be changed by the application.

Since all transmit channels are pre-assigned, routing an SCbus device only requires connecting the receive (listen) channel of the device to an SCbus time

SCbus Routing Guide

slot. The connected device then listens to all data transmitted over that SCbus time slot. This receive channel can be moved (disconnected and connected) to a different SCbus time slot at any time by the application.

NOTE: The SCbus routing functions only support SCbus configurations. This guide provides an overview of SCbus routing and defines the nomenclature used to identify devices, channels and time slots in various SCbus configurations. These configurations may include combinations of resources such as voice devices, analog loop-start network interfaces (analog devices), digital network interfaces (T-1 or E-1 devices) and facsimile devices (fax devices). Chapter 2 describes SCbus routing concepts for communicating with resources via the SCbus and provides a summary listing of the SCbus routing functions. See the companion volumes listed below for detailed information about each SCbus routing function. Chapters 3 through 6 contain examples of routing resources using SCbus routing functions.

- Chapter 3 provides an overview of SCbus routing assignments after system initialization and routing resources using SCbus convenience functions or individual SCbus routing functions.
- Chapter 4 provides an example of SCbus routing and sample code for an analog inbound/outbound call application.
- Chapter 5 provides an example of SCbus routing and sample code for a digital service inbound/outbound call application.
- Chapter 6 provides an example of SCbus routing and sample code for sharing a fax resource.
- Chapter 7 describes special considerations to keep in mind when routing SCbus devices and when programming voice and analog loop-start device configurations.

The following companion volumes describe the network/resource (nr_) SCbus convenience functions, the individual SCbus routing functions and SCbus routing demo programs.

- *SCbus Routing Software Reference for MS-DOS*
- *SCbus Routing Software Reference for OS/2*
- *SCbus Routing Software Reference for Linux and Windows*

1. *SCbus Introduction*

Appendices contain a summary list of the SCbus routing functions, a list of related publications and a glossary.

SCbus Routing Guide

2. SCbus Routing of Resources

SCbus Routing Overview

SCbus routing concepts for connecting resources via the SCbus are presented in this chapter. SCbus routing functions provide the flexibility to connect together any two devices attached to the SCbus and to allow any number of SCbus devices to listen to a single transmitting device. This flexibility enables:

- communications between voice devices and analog, digital network, or station interface devices;
- rerouting (switching) a voice device from one network interface device to another;
- moving (rerouting) shared resources, such as fax devices, from one network interface device to another;
- connecting together two network interface devices;
- connecting any number of incoming calls on analog or digital network interface devices to a single SCbus device;
- connecting a station interface device to a voice device, analog, or digital interface device.

Conceptually, think of all SCbus time slots as transmit time slots and that at system initialization, the transmit channel of each device connected to the SCbus is assigned to a unique and separate SCbus time slot. Then routing is merely connecting the receive (listen) channel of any SCbus device to the transmit channel of another SCbus device. In this manner, any number of devices can listen to the transmissions of another SCbus device.

The application can also disconnect (unlisten) the receive channel from the SCbus. When disconnected, no data is received by the device from the SCbus.

Two types of SCbus routing functions are provided:

- network/resource (nr_) SCbus convenience functions and
- individual SCbus routing functions.

SCbus Routing Guide

2.1. Network/Resource SCbus Convenience Functions

For most routing applications, the following SCbus convenience functions should suffice. The SCbus convenience functions include all of the functionality of the individual SCbus routing functions.

NOTE: The network/resource SCbus convenience functions are not part of the *voice* library. Their C source code is provided in a separate file called *sctools.c* located in the <install drive>:\<install directory>\<dialogic>\<sctools> directory.

nr_scroute()	makes half or full duplex connection between two SCbus devices
nr_scunroute()	breaks half or full duplex connection between two SCbus devices

2.2. Individual SCbus Routing Functions

The individual SCbus routing functions provide the ability to program each phase of connecting or disconnecting the receive channel of a device to the transmit channel of another device or to build your own convenience functions. These individual SCbus routing functions can be characterized by their prefix and by their suffix. The prefix of the individual SCbus routing functions identify the device to which the function applies:

ag_	analog device (loop-start interface)
dl_	voice device (MS-DOS)
dx_	voice device (Linux, OS/2, Windows)
dt_	digital device (T-1 or E-1 digital service interface)
fx_	fax device
ms_	MSI device

2. SCbus Routing of Resources

The suffix of the individual SCbus routing functions identify the operation or task performed by the function:

_getctinfo()	returns information about the device (analog device, voice device, digital device, or other technology device); this function is not used for routing SCbus time slots.
_getxmitslot()	returns SCbus time slot information into an SC_TSINFO structure that includes the number of the SCbus time slot connected to the transmit channel of the specified device
_listen()	connects the listen (receive) channel of the specified device to an SCbus time slot
_unlisten()	disconnects the listen (receive) channel of the specified device from an SCbus time slot

2.3. Using SCbus Convenience Functions

To route channels using the SCbus convenience functions, you need only perform the following:

- Issue a **nr_scroute()** call. This convenience function automatically:
 - connects (routes) the listen channel of the first device to the SCbus time slot connected to the transmit channel of the second device and then
 - connects the listen channel of this second device to the SCbus time slot connected to the transmit channel of the first device.
 - When the function returns, the devices will be listening to each other; thus full duplex communications are established.

To disconnect devices, you need only disconnect the listen channel of each device from the SCbus by issuing a **nr_scunroute()** call. When the function returns, the listen channel of each device will be disconnected from the SCbus.

2.4. Using Individual SCbus Routing Functions

To route channels using individual SCbus routing functions, perform the following (**xx** = the prefix that identifies the type of device: analog, digital, fax, voice, or MSI):

1. issue a **xx_getxmitslot()** call for the first device. This function returns the SCbus time slot information contained in a SC_TSINFO structure

SCbus Routing Guide

that includes the number of the SCbus time slot connected to the transmit channel of the first device; for example, for the transmit channel of voice device 14.

2. issue a **xx_listen()** call for the second device. This function connects the listen channel of the second device to the transmit channel of the first device by using the information contained in the SC_TSINFO structure; for example, the listen channel of digital T-1 device 12 to the transmit channel of voice device 14.
3. issue a **xx_getxmitslot()** call for the second device. This function returns the SCbus time slot information contained in a SC_TSINFO structure that includes the number of the SCbus time slot connected to the transmit channel of the second device; for example, the transmit channel of digital T-1 device 12.
4. issue a **xx_listen()** call for the first device. This function connects the listen channel of the first device to the transmit channel of the second device by using the information contained in the SC_TSINFO structure; for example, the listen channel of voice device 14 to the transmit channel of digital T-1 device 12.

When these functions return, full duplex communications between the devices will be established. Throughout this process, **the actual SCbus time slot number is never needed to code the application.**

To disconnect devices, the listen channel of each device must be disconnected from the SCbus by issuing a **xx_unlisten()** call for each device. When the function returns, the listen channel of the device will be disconnected from the SCbus and no data will be received by the device.

NOTE: When moving the receive (listen) channel of a device to a different SCbus time slot, the **xx_listen()** function automatically disconnects the device from the existing SCbus time slot connection thus eliminating the need to issue a **xx_unlisten()** function call.

2. *SCbus Routing of Resources*

3. SCbus Resource Routing Overview

Examples of SCbus Routing Resources

Examples of routing resources using both SCbus convenience functions and individual SCbus routing functions are described in the chapters that follow. These examples use the initial configuration and the routing concepts described in this chapter.

Although the configurations and applications that the examples illustrate may differ, the SCbus routing process is the same from configuration to configuration. The examples illustrate routing between resource devices located on the following SCbus products, see *Figure 1*:

- a D/160SC-LS board with 16 analog loop-start interface devices and 16 voice devices,
- a VFX/40ESC board with 4 analog loop start interface devices and 4 fax/voice devices,
- two D/240SC-T1 boards; each D/240SC-T1 board provides 24 digital channels (T-1 time slots) and 24 voice devices.

Although we will refer to SCbus time slot numbers for illustrative purposes, **actual SCbus time slot numbers are never needed to write your application.**

When connecting devices using SCbus convenience functions, the general procedure is to issue a **nr_scroute()** call. When the function returns, communications between the devices will be established.

To disconnect devices, issue a **nr_scunroute()** call. When the function returns, the listen channel of each device will be disconnected from the SCbus and no data will be received by the devices.

Connecting Devices using Individual SCbus Routing Functions

When connecting devices using **individual** SCbus routing functions, the general procedure is:

1. Get the SC_TSINFO structure that contains the number of the SCbus time slot connected to the transmit channel of the first device by issuing a **xx_getxmitslot()** call for the first device (**xx** = the prefix that identifies the type of device: analog, digital, fax, voice, or MSI).
2. Connect the listen channel of the second device to the transmit channel of the first device by issuing a **xx_listen()** call for the second device. This function uses as input the information contained in the SC_TSINFO structure returned by the **xx_getxmitslot()** call for the first device (step 1).
3. Get the SC_TSINFO structure that contains the number of the SCbus time slot connected to the transmit channel of the second device; by issuing a **xx_getxmitslot()** call for the second device.
4. Connect the listen channel of the first device to the transmit channel of the second device by issuing a **xx_listen()** call for the first device. This function uses as input the information contained in the SC_TSINFO structure returned by the **xx_getxmitslot()** call for the second device (step 3).

When these functions return, full duplex communications between the devices will be established.

To disconnect devices, the listen channel of each device must be disconnected from the SCbus by issuing a **xx_unlisten()** call for each device. When the function returns, the listen channel of the device will be disconnected from the SCbus and no data will be received by the device.

3.1. SCbus System Initialization

At system initialization, the number of devices (analog interface, digital interface, voice, facsimile, etc.) on each board and the number of SCbus time slots required to service these devices are identified. Only one device can transmit to an SCbus time slot at a time. To assure this, the transmit channel of each device is assigned to a specific and unique SCbus time slot at system initialization. These transmit assignments cannot be changed by the application.

Assume that after system initiation, the SCbus time slots are assigned to the transmit channel of each device as shown in *Figure 1*. As this figure shows, the following transmit channel assignments were made.

3. SCbus Resource Routing Overview

- For the D/160SC-LS board:
 - analog channel 6 transmits to SCbus time slot 13.
 - voice channel 6 transmits to SCbus time slot 29.
- For the VFX/40ESC board:
 - analog channel 2 transmits to SCbus time slot 1.
 - voice channel 2 transmits to SCbus time slot 5.
 - the fax channels use the same SCbus time slots as the corresponding voice channel, thus fax channel 2 will transmit on SCbus time slot 5 when this resource is connected.
- For the first D/240SC-T1 board:
 - digital channel 12 transmits to SCbus time slot 51.
 - voice channel 14 transmits to SCbus time slot 77.
- For the second D/240SC-T1 board:
 - digital channel 23 transmits to SCbus time slot 110.
 - voice channel 5 transmits SCbus time slot 116.

These transmit channel assignments cannot be changed by the application.

Each device transmits on their assigned SCbus time slots and listens to other devices via a Switch Handler that manages all communications across the SCbus.

SCbus Routing Guide

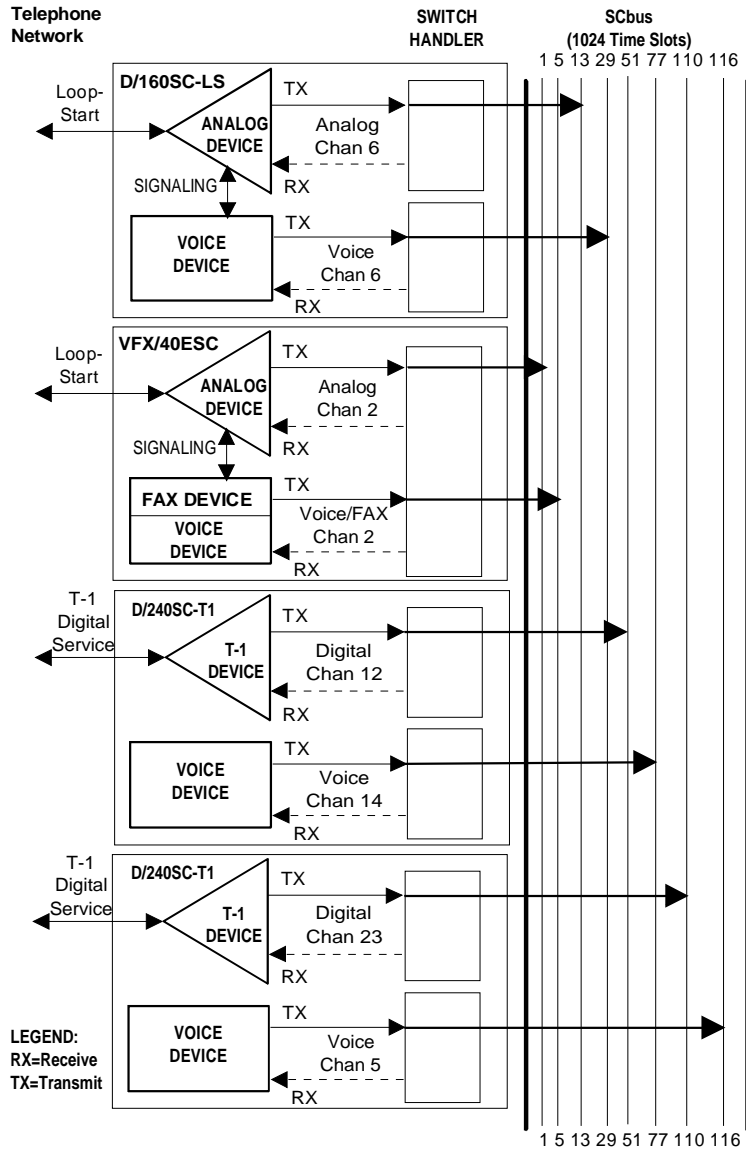


Figure 1. System Initialization SCbus Time Slot Assignments

4. SCbus Analog Inbound/Outbound Call Routing Example

SCbus Routing Example Overview

In this example, an analog call answered by a D/160SC-LS analog device is connected to a loop-start line interfaced to a VFX/40ESC analog device. To make this connection, full duplex communications are established between analog channel 6 on the D/160SC-LS board and analog channel 2 on the VFX/40ESC board. See *Figure 2* for the SCbus routing connections as established at system initialization. Since these boards interface to the telephone network via analog devices, signaling events are handled by the on-board voice device and are not transmitted over the SCbus. Thus at system initialization, default routing established full-duplex communications between each analog device and its associated voice device (on a one-to-one basis) as shown in *Figure 2*.

4.1. Routing Using SCbus Convenience Functions

To route these channels using SCbus convenience functions, perform the following:

- Make the analog device on the D/160SC-LS board and the analog device on the VFX/40ESC board listen to each other by issuing a **nr_scroute()** call. As illustrated in *Figure 2* and *Figure 3*, this function automatically:
 - moves analog receive channel 2 on the VFX/40ESC board from voice transmit channel 2 (SCbus time slot 5) to analog transmit channel 6 (SCbus time slot 13) on the D/160SC-LS board and
 - moves analog receive channel 6 on the D/160SC-LS board from voice transmit channel 6 (SCbus time slot 29) to analog transmit channel 2 (SCbus time slot 1) on the VFX/40ESC board.
- When the function returns, analog channel 6 on the D/160SC-LS board and analog channel 2 on the VFX/40ESC board will be listening to each other. Thus full duplex communications are established.

SCbus Routing Guide

- The corresponding voice receive channels were not disconnected from the SCbus, thus one or both of these voice channels could continue to monitor or record the information being transmitted.

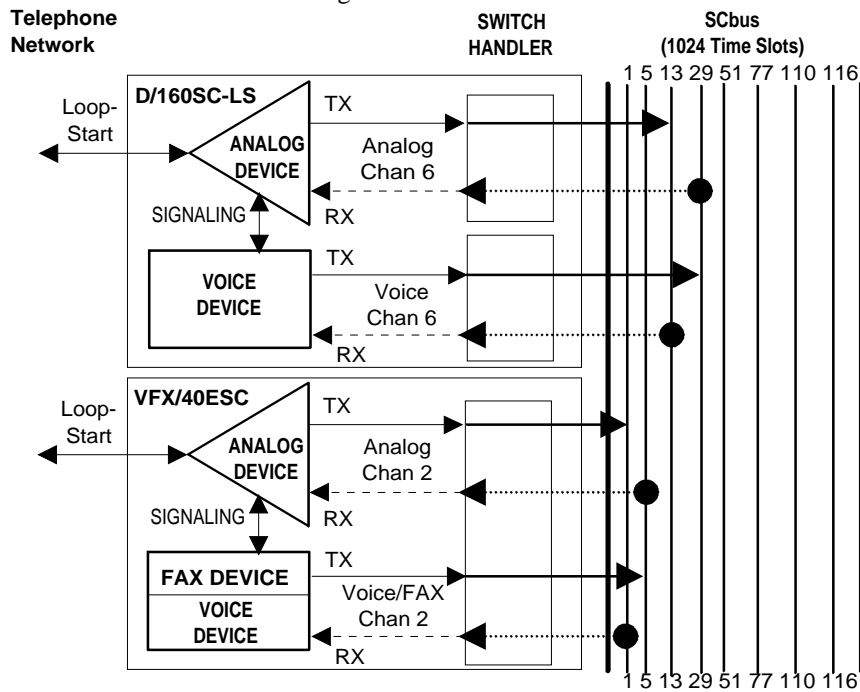


Figure 2. Initial Analog & Voice Device Routing Example

4. SCbus Analog Inbound/Outbound Call Routing Example

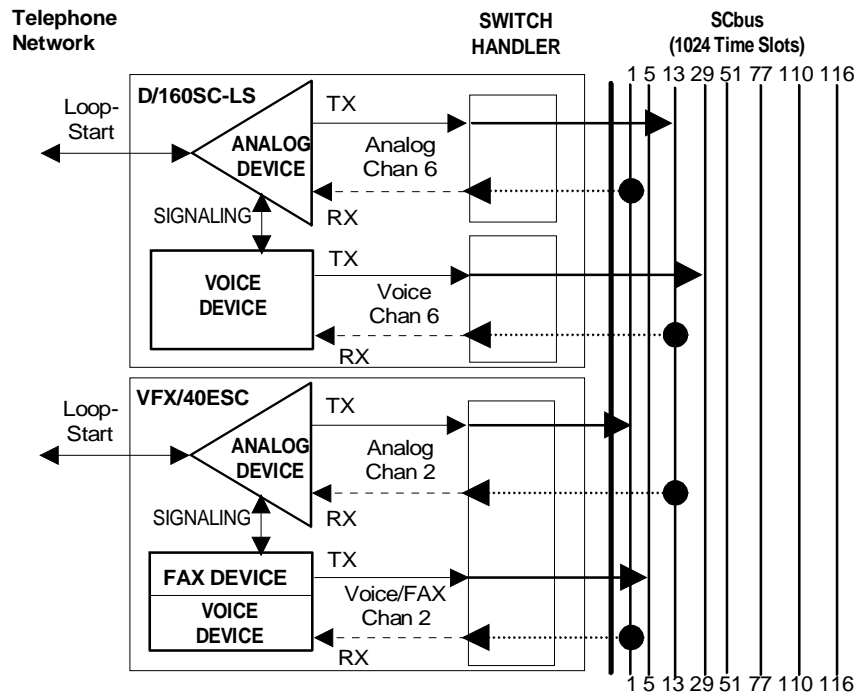


Figure 3. Analog Inbound/Outbound Routing Example

After the call completes, these analog channels can be easily rerouted to their associated voice device, *Figure 2*, by performing the following:

- Make analog channel 6 and voice channel 6 on the D/160SC-LS board listen to each other by issuing a **nr_scroute()** call. This function automatically:
 - moves analog receive channel 6 on the D/160SC-LS board to voice transmit channel 6 (SCbus time slot 29) and
 - moves voice receive channel 6 on the D/160SC-LS board to analog transmit channel 6 (SCbus time slot 13).
- Make analog channel 2 and voice channel 2 on the VFX/40ESC board listen to each other by issuing a **nr_scroute()** call. This function automatically:
 - moves analog receive channel 2 on the VFX/40ESC board to voice transmit channel 2 (SCbus time slot 5) and

SCbus Routing Guide

- moves voice receive channel 2 on the VFX/40ESC board to analog transmit channel 2 (SCbus time slot 1).
- When these functions return, analog channel 6 and voice channel 6 on the D/160SC-LS board and analog channel 2 and voice channel 2 on the VFX/40ESC board will be listening to each other. Thus full duplex communications are established.

Note that in this example, since the voice channels were never disconnected, the **nr_scroute()** calls could be issued in half-duplex mode so that only the analog channels were rerouted.

Sample code for each operating system illustrating SCbus routing for this example using the SCbus convenience functions and also using the individual SCbus routing functions is contained in the following paragraphs.

4. SCbus Analog Inbound/Outbound Call Routing Example

4.1.1. Windows Sample Code for SCbus Convenience Functions

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

/*
 * For nr_ function error printing, make sure PRINTON is defined when
 * compiling the sctools.c file.
 */
#include "sctools.h"

main()
{
    int          chdev1,          /* D/160SC-LS voice channel handle */
               chdev2;          /* VFX/40ESC voice channel handle */

    /* Open voice channel 2 on board 2 (D/160SC-LS voice channel 6). */
    if ((chdev1 = dx_open("dxxxB2C2", 0)) == -1) {
        printf("Cannot open channel dxxxB2C2.  errno = %d", errno);
        exit(1);
    }

    /* Open voice channel 2 on board 5 (VFX/40ESC voice channel 2). */
    if ((chdev2 = dx_open("dxxxB5C2", 0)) == -1) {
        printf("Cannot open channel dxxxB5C2.  errno = %d", errno);
        exit(1);
    }

    .
    . Continue processing.
    .

    /***** SCbus routing example. *****/
    /*
     * Make a full duplex connection between analog channel 6 on the
     * D/160SC-LS board and analog channel 2 on the VFX/40ESC board.
     */
    if (nr_scroute(chdev1, SC_LSI, chdev2, SC_LSI, SC_FULLDUP) == -1) {
        printf("Could not route the two analog channels.");
        exit(1);
    }

    .
    . Continue processing.
    .
}
```

SCbus Routing Guide

4.1.2. Linux Sample Code for SCbus Convenience Functions

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

/*
 * For nr_ function error printing, make sure PRINTON is defined when
 * compiling the sctools.c file.
 */
#include "sctools.h"

main()
{
    int          chdev1,          /* D/160SC-LS voice channel handle */
               chdev2;          /* VFX/40ESC voice channel handle */

    /* Open voice channel 2 on board 2 (D/160SC-LS voice channel 6). */
    if ((chdev1 = dx_open("dxxxB2C2", 0)) == -1) {
        printf("Cannot open channel dxxxB2C2.  errno = %d", errno);
        exit(1);
    }

    /* Open voice channel 2 on board 5 (VFX/40ESC voice channel 2). */
    if ((chdev2 = dx_open("dxxxB5C2", 0)) == -1) {
        printf("Cannot open channel dxxxB5C2.  errno = %d", errno);
        exit(1);
    }

    .
    . Continue processing.
    .

    /***** SCbus routing example. *****/
    /*
     * Make a full duplex connection between analog channel 6 on the
     * D/160SC-LS board and analog channel 2 on the VFX/40ESC board.
     */
    if (nr_scroute(chdev1, SC_LSI, chdev2, SC_LSI, SC_FULLDUP) == -1) {
        printf("Could not route the two analog channels.");
        exit(1);
    }

    .
    . Continue processing.
    .
}
```

4. SCbus Analog Inbound/Outbound Call Routing Example

4.1.3. MS-DOS Sample Code for SCbus Convenience Functions

```
#include <stdlib.h>
#include <stdio.h>
#include "d40.h"
#include "d40lib.h"
#include "vfcns.h"

#include "sctools.h"

int main()
{
    int          chdev1 = 6, /* Sixth voice channel on the D/160SC-LS. */
               chdev2 = 18, /* Second voice channel on the VFX/40ESC. */
               rcode; /* To hold function error return codes. */
    unsigned int sc_IRQ = 10, /* SCbus board interrupt (from DIALOGIC.CFG) */
               num_chans; /* Location for number of voice channels */
                       /* returned by startsys(). */

    /***** Initialize system. *****/
    /* Make sure the voice driver is loaded. */
    if (getvctr() == 0) {
        printf("Voice driver is not loaded.");
        exit(1);
    }

    /* Make sure the voice driver is stopped before starting. */
    stopsys();

    /* Start the voice driver. */
    if ((rcode = startsys(sc_IRQ, SM_EVENT, 0, 0, &num_chans)) != E_SUCC) {
        printf("Could not start voice driver: Error message = %s", d4xerr(rcode));
        exit(1);
    }

    .
    .   Continue processing.
    .

    /***** SCbus routing example. *****/
    /*
     * Make a full duplex connection between the D/160SC-LS 6th analog
     * channel and the VFX/40ESC 2nd analog channel.
     */
    if ((rcode = nr_scroute(chdev1, SC_LSI, chdev2, SC_LSI, SC_FULLDUP)) != 0) {
        printf("Could not route the analog channels: Error = %d", rcode);
        exit(1);
    }

    .
    .   Continue processing.
    .
}
```

SCbus Routing Guide

4.1.4. OS/2 Sample Code for SCbus Convenience Functions

```
#define INCL_DOSERRORS
#include <os2.h>
#include <stdlib.h>
#include <stdio.h>
#include <fcntl.h>
#include <voxlib.h>

/*
 * For nr_ function error printing, make sure PRINTON is defined when
 * compiling the sctools.c file.
 */
#include <sctools.h>

main()
{
    int          chdev1,          /* D/160SC-LS voice channel handle */
              chdev2;           /* VFX/40ESC voice channel handle */

    /* Open voice channel 2 on board 2 (D/160SC-LS voice channel 6). */
    if ((chdev1 = dl_open("VOXB1C1", O_RDWR)) == -1) {
        printf("Cannot open channel VOXB1C1. dl_errno = %d", dl_errno);
        exit(1);
    }

    /* Open voice channel 2 on board 5 (VFX/40ESC voice channel 2). */
    if ((chdev2 = dl_open("VOXB4C1", O_RDWR)) == -1) {
        printf("Cannot open channel VOXB4C1. dl_errno = %d", dl_errno);
        exit(1);
    }

    .
    . Continue processing.
    .

    /***** SCbus routing example. *****/
    /*
     * Make a full duplex connection between the D/160SC-LS 6th analog
     * channel and the VFX/40ESC 2nd analog channel.
     */
    if (nr_scroute(chdev1, SC_LSI, chdev2, SC_LSI, SC_FULLDUP) == -1) {
        printf("Could not route the two analog channels.");
        exit(1);
    }

    .
    . Continue processing.
    .
}
```

4.2. Routing Using Individual SCbus Routing Functions

To route the channels using individual SCbus routing functions, perform the following, see *Figure 3*:

1. Initialize the SC_TSINFO structure with the required information.

4. SCbus Analog Inbound/Outbound Call Routing Example

2. Get the SC_TSINFO structure that contains the number of the SCbus time slot connected to analog transmit channel 6 of the D/160SC-LS board by issuing an **ag_getxmitslot()** call.
3. Connect analog listen channel 2 of the VFX/40ESC board to analog transmit channel 6 of the D/160SC-LS board by issuing an **ag_listen()** call. This function uses as input the information contained in the SC_TSINFO structure returned by the **ag_getxmitslot()** call of step 2.
4. Get the SC_TSINFO structure that contains the number of the SCbus time slot connected to analog transmit channel 2 of the VFX/40ESC board by issuing an **ag_getxmitslot()** call.
5. Connect analog receive channel 6 of the D/160SC-LS board to analog transmit channel 2 of the VFX/40ESC board by issuing an **ag_listen()** call. This function uses as input the information contained in the SC_TSINFO structure returned by the **ag_getxmitslot()** call of step 4.

When these functions return, full duplex communications between the two analog channels will be established.

4.2.1. Windows Sample Code for Individual SCbus Routing Functions

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

main()
{
    int          chdev1,          /* D/160SC-LS voice channel handle */
               chdev2;          /* VFX/40ESC voice channel handle */
    SC_TSINFO    sc_tsinfo;      /* time slot information structure */
    long         scts;           /* SCbus time slot */

    /* Open voice channel 2 on board 2 (D/160SC-LS voice channel 6). */
    if ((chdev1 = dx_open("dxxxB2C2", 0)) == -1) {
        printf("Cannot open channel dxxxB2C2.  errno = %d", errno);
        exit(1);
    }

    /* Open voice channel 2 on board 5 (VFX/40ESC voice channel 2). */
    if ((chdev2 = dx_open("dxxxB5C2", 0)) == -1) {
        printf("Cannot open channel dxxxB5C2.  errno = %d", errno);
        exit(1);
    }

    .
    . Continue processing.
    .

    /***** SCbus routing example. *****/
}
```

SCbus Routing Guide

```
/* Initialize the SC_TSINFO structure with the necessary information. */
sc_tsinfo.sc_numts = 1;
sc_tsinfo.sc_tsarrayp = &scts;

/*
 * Get the SCbus time slot on which the D/160SC-LS sixth analog
 * channel is transmitting.
 */
if (ag_getxmitslot(chdev1, &sc_tsinfo) == -1) {
    printf("ag_getxmitslot() failed: Error message = %s", ATDV_ERRMSGP(chdev1));
    exit(1);
}

/*
 * Make analog channel 2 (of the VFX/40ESC board) listen to the
 * SCbus time slot that the D/160SC-LS sixth analog channel is
 * transmitting on.
 */
if (ag_listen(chdev2, &sc_tsinfo) == -1) {
    printf("ag_listen() failed: Error message = %s", ATDV_ERRMSGP(chdev2));
    exit(1);
}

/*
 * Get the SCbus time slot on which the VFX/40ESC second analog
 * channel is transmitting.
 */
if (ag_getxmitslot(chdev2, &sc_tsinfo) == -1) {
    printf("ag_getxmitslot() failed: Error message = %s", ATDV_ERRMSGP(chdev2));
    exit(1);
}

/*
 * Make analog channel 6 (of the D/160SC-LS board) listen to the
 * SCbus time slot that the VFX/40ESC second analog channel is
 * transmitting on.
 */
if (ag_listen(chdev1, &sc_tsinfo) == -1) {
    printf("ag_listen() failed: Error message = %s", ATDV_ERRMSGP(chdev1));
    exit(1);
}

.
. Continue processing.
.
}
```

4.2.2. Linux Sample Code for Individual SCbus Routing Functions

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

main()
{
    int      chdev1,      /* D/160SC-LS voice channel handle */
            chdev2;      /* VFX/40ESC voice channel handle */
    SC_TSINFO sc_tsinfo; /* time slot information structure */
    long      scts;       /* SCbus time slot */

    /* Open voice channel 2 on board 2 (D/160SC-LS voice channel 6). */
```


4. SCbus Analog Inbound/Outbound Call Routing Example

```
if ((chdev1 = dx_open("dxxxB2C2", 0)) == -1) {
    printf("Cannot open channel dxxxB2C2.  errno = %d", errno);
    exit(1);
}

/* Open voice channel 2 on board 5 (VFX/40ESC voice channel 2). */
if ((chdev2 = dx_open("dxxxB5C2", 0)) == -1) {
    printf("Cannot open channel dxxxB5C2.  errno = %d", errno);
    exit(1);
}

.
. Continue processing.
.

/***** SCbus routing example. *****/
/* Initialize the SC_TSINFO structure with the necessary information. */
sc_tsinfo.sc_numts = 1;
sc_tsinfo.sc_tsarrayp = &scts;

/*
 * Get the SCbus time slot on which the D/160SC-LS sixth analog
 * channel is transmitting.
 */
if (ag_getxmitslot(chdev1, &sc_tsinfo) == -1) {
    printf("ag_getxmitslot() failed: Error message = %s", ATDV_ERRMSGP(chdev1));
    exit(1);
}

/*
 * Make analog channel 2 (of the VFX/40ESC board) listen to the
 * SCbus time slot that the D/160SC-LS sixth analog channel is
 * transmitting on.
 */
if (ag_listen(chdev2, &sc_tsinfo) == -1) {
    printf("ag_listen() failed: Error message = %s", ATDV_ERRMSGP(chdev2));
    exit(1);
}

/*
 * Get the SCbus time slot on which the VFX/40ESC second analog
 * channel is transmitting.
 */
if (ag_getxmitslot(chdev2, &sc_tsinfo) == -1) {
    printf("ag_getxmitslot() failed: Error message = %s", ATDV_ERRMSGP(chdev2));
    exit(1);
}

/*
 * Make analog channel 6 (of the D/160SC-LS board) listen to the
 * SCbus time slot that the VFX/40ESC second analog channel is
 * transmitting on.
 */
if (ag_listen(chdev1, &sc_tsinfo) == -1) {
    printf("ag_listen() failed: Error message = %s", ATDV_ERRMSGP(chdev1));
    exit(1);
}

.
. Continue processing.
.
}
```

SCbus Routing Guide

4. SCbus Analog Inbound/Outbound Call Routing Example

4.2.3. MS-DOS Sample Code for Individual SCbus Routing Functions

```
#include <stdlib.h>
#include <stdio.h>
#include "d40.h"
#include "d40lib.h"
#include "vfcns.h"

int main()
{
    int          chdev1 = 6, /* Sixth voice channel on the D/160SC-LS */
               chdev2 = 18, /* Second voice channel on the VFX/40ESC */
               rcode; /* To hold function error return codes. */
    unsigned int sc_IRQ = 10, /* SCbus board interrupt (from DIALOGIC.CFG) */
               num_chans; /* Location for number of voice channels */
               /* returned by startsys(). */
    SC_TSINFO    sc_tsinfo; /* time slot information structure */
    long         scts; /* SCbus time slot */

    /***** Initialize system. *****/
    /* Make sure the voice driver is loaded. */
    if (getvctr() == 0) {
        printf("Voice driver is not loaded.");
        exit(1);
    }

    /* Make sure the voice driver is stopped before starting. */
    stopsys();

    /* Start the voice driver. */
    if ((rcode = startsys(sc_IRQ, SM_EVENT, 0, 0, &num_chans)) != E_SUCC) {
        printf("Could not start voice driver: Error message = %s", d4xerr(rcode));
        exit(1);
    }

    .
    .   Continue processing.
    .

    /***** SCbus routing example. *****/
    /* Initialize the SC_TSINFO structure with the necessary information. */
    sc_tsinfo.sc_numts = 1;
    sc_tsinfo.sc_tsarrayp = &scts;

    /*
     * Get the SCbus time slot on which the D/160SC-LS sixth analog
     * channel is transmitting.
     */
    if ((rcode = ag_getxmitslot(chdev1, &sc_tsinfo)) != E_SUCC) {
        printf("ag_getxmitslot() failed: Error message = %s", d4xerr(rcode));
        exit(1);
    }

    /*
     * Make analog channel 2 (of the VFX/40ESC board) listen to the
     * SCbus time slot that the D/160SC-LS sixth analog channel is
     * transmitting on.
     */
    if ((rcode = ag_listen(chdev2, &sc_tsinfo)) != E_SUCC) {
        printf("ag_listen() failed: Error message = %s", d4xerr(rcode));
        exit(1);
    }
}
```

SCbus Routing Guide

```
    }

    /*
     * Get the SCbus time slot on which the VFX/40ESC second analog
     * channel is transmitting.
     */
    if ((rcode = ag_getxmitslot(chdev2, &sc_tsinfo)) != E_SUCC) {
        printf("ag_getxmitslot() failed: Error message = %s", d4xerr(rcode));
        exit(1);
    }

    /*
     * Make analog channel 6 (of the D/160SC-LS board) listen to the
     * SCbus time slot that the VFX/40ESC second analog channel is
     * transmitting on.
     */
    if ((rcode = ag_listen(chdev1, &sc_tsinfo)) != E_SUCC) {
        printf("ag_listen() failed: Error message = %s", d4xerr(rcode));
        exit(1);
    }

    .
    .   Continue processing.
    .
}
```

4. SCbus Analog Inbound/Outbound Call Routing Example

4.2.4. OS/2 Sample Code for Individual SCbus Routing Functions

```
#define INCL_DOSERRORS
#include <os2.h>
#include <stdlib.h>
#include <stdio.h>
#include <fcntl.h>
#include <voxlib.h>

main()
{
    int      chdev1,      /* D/160SC-LS voice channel handle */
            chdev2;      /* VFX/40ESC voice channel handle */
    SC_TSINFO sc_tsinfo;  /* time slot information structure */
    long      scts;       /* SCbus time slot

    /* Open voice channel 2 on board 2 (D/160SC-LS voice channel 6). */
    if ((chdev1 = dl_open("VOXB1C1", O_RDWR)) == -1) {
        printf("Cannot open channel VOXB1C1.  dl_errno = %d", dl_errno);
        exit(1);
    }

    /* Open voice channel 2 on board 5 (VFX/40ESC voice channel 2). */
    if ((chdev2 = dl_open("VOXB4C1", O_RDWR)) == -1) {
        printf("Cannot open channel VOXB4C1.  dl_errno = %d", dl_errno);
        exit(1);
    }

    .
    . Continue processing.
    .

    /***** SCbus routing example. *****/
    /* Initialize the SC_TSINFO structure with the necessary information. */
    sc_tsinfo.sc_numts = 1;
    sc_tsinfo.sc_tsarray = &scts;

    /*
     * Get the SCbus time slot on which the D/160SC-LS sixth analog
     * channel is transmitting.
     */
    if (ag_getxmitslot(chdev1, &sc_tsinfo) == -1) {
        printf("ag_getxmitslot() failed: dl_errno = %d", dl_errno);
        exit(1);
    }

    /*
     * Make analog channel 2 (of the VFX/40ESC board) listen to the
     * SCbus time slot that the D/160SC-LS sixth analog channel is
     * transmitting on.
     */
    if (ag_listen(chdev2, &sc_tsinfo) == -1) {
        printf("ag_listen() failed: dl_errno = %d", dl_errno);
        exit(1);
    }

    /*
     * Get the SCbus time slot on which the VFX/40ESC second analog
     * channel is transmitting.
     */
    if (ag_getxmitslot(chdev2, &sc_tsinfo) == -1) {
```

SCbus Routing Guide

```
        printf("ag_getxmitslot() failed: dl_errno = %d", dl_errno);
        exit(1);
    }

    /*
     * Make analog channel 6 (of the D/160SC-LS board) listen to the
     * SCbus time slot that the VFX/40ESC second analog channel is
     * transmitting on.
     */
    if (ag_listen(chdev1, &sc_tsinfo) == -1) {
        printf("ag_listen() failed: dl_errno = %d", dl_errno);
        exit(1);
    }

    .
    . Continue processing.
    .
}
```

5. SCbus Digital Service Inbound/Outbound Call Routing Example

SCbus Call Routing Example Overview

In this example, an incoming call on a T-1 time slot received by the T-1 device on the first D/240SC-T1 board is passed to the T-1 device on the second D/240SC-T1 board, see *Figure 4* and *Figure 5*. To make this connection, full duplex communications must be established between digital channel 12 on the first D/240SC-T1 board and digital channel 23 on the second D/240SC-T1 board. *Figure 4* shows the SCbus time slot connections established at system initialization to the digital transmit channels used in this example.

5.1. Routing Using SCbus Convenience Functions

To route these channels using SCbus convenience functions, perform the following:

- Make the digital channels on each D/240SC-T1 board listen to each other by issuing a **nr_scroute()** call. This function automatically, see *Figure 4* and *Figure 5*:
 - connects digital listen channel 23 on the second D/240SC-T1 board to digital transmit channel 12 on the first D/240SC-T1 board (via SCbus time slot 51) and
 - connects digital listen channel 12 on the first D/240SC-T1 board to digital transmit channel 23 on the second D/240SC-T1 board (via SCbus time slot 110)
- When the function returns, digital channel 12 on the first D/240SC-T1 board and digital channel 23 on the second D/240SC-T1 board will be listening to each other. Thus full duplex communications are established.

After the call completes, these digital channels are easily rerouted to a voice channel, another network interface channel or to a complementary technology channel to handle the next call.

SCbus Routing Guide

Sample code for each operating system illustrating SCbus routing for this example using the SCbus convenience functions and also using the individual SCbus routing functions is contained in the following paragraphs.

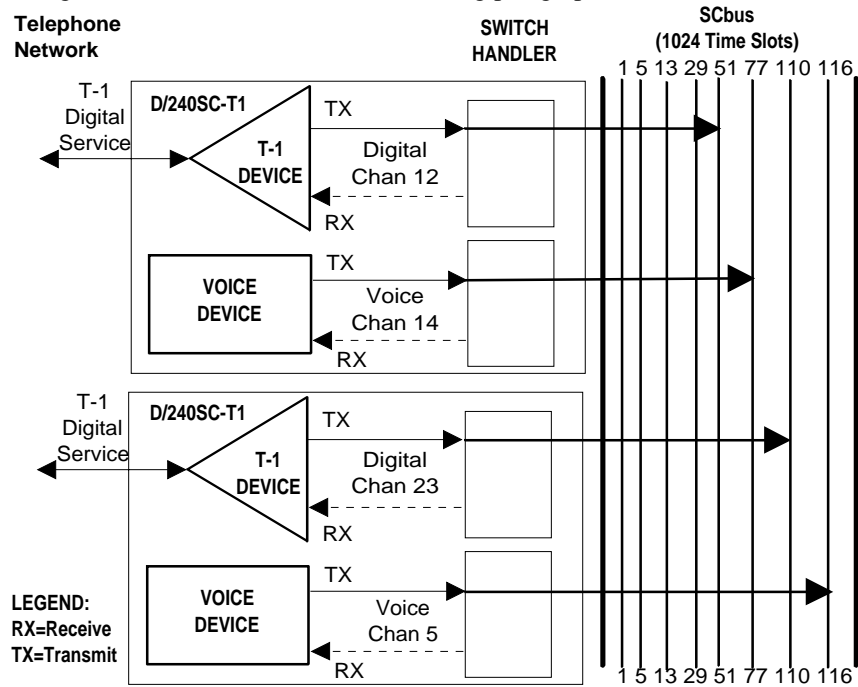


Figure 4. Initial Digital Service Routing Example

5. SCbus Digital Service Inbound/Outbound Call Routing Example

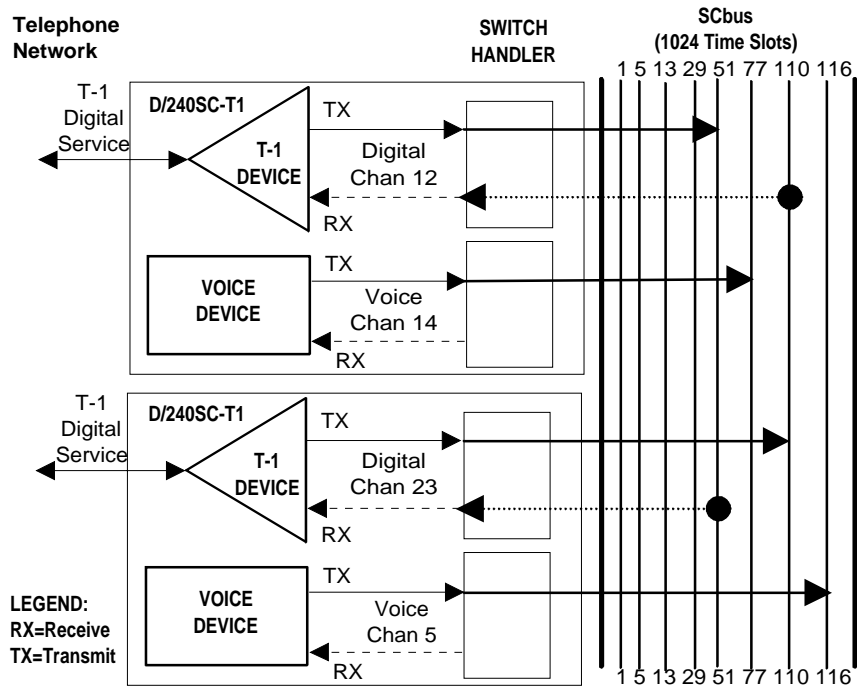


Figure 5. Digital Service Inbound/Outbound Routing Example

SCbus Routing Guide

5.1.1. Windows Sample Code for SCbus Convenience Functions

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

/*
 * For nr_ function error printing and DTI functionality, make sure
 * PRINTON and DTISC are defined when compiling the sctools.c file.
 */
#include "sctools.h"

main()
{
    int          dtih1,          /* D/240SC-T1 #1  T-1 time slot handle */
               dtih2;          /* D/240SC-T1 #2  T-1 time slot handle */

    /* Open T-1 time slot 12 on D/240SC-T1 #1. */
    if ((dtih1 = dt_open("dtiB1T12", 0)) == -1) {
        printf("Cannot open T-1 time slot dtiB1T12.  errno = %d", errno);
        exit(1);
    }

    /* Open T-1 time slot 23 on D/240SC-T1 #2. */
    if ((dtih2 = dt_open("dtiB2T23", 0)) == -1) {
        printf("Cannot open T-1 time slot dtiB2T23.  errno = %d", errno);
        exit(1);
    }

    .
    . Continue processing.
    .

    /***** SCbus routing example. *****/
    /*
     * Make a full duplex connection between T-1 time slot 12 (on the
     * first D/240SC-T1) and T-1 time slot 23 (on the second D/240SC-T1).
     */
    if (nr_scroute(dtih1, SC_DTI, dtih2, SC_DTI, SC_FULLDUP) == -1) {
        printf("Could not route the two T-1 time slots.");
        exit(1);
    }

    .
    . Continue processing.
    .
}
```

5. SCbus Digital Service Inbound/Outbound Call Routing Example

5.1.2. Linux Sample Code for SCbus Convenience Functions

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

/*
 * For nr_ function error printing and DTI functionality, make sure
 * PRINTON and DTISC are defined when compiling the sctools.c file.
 */
#include "sctools.h"

main()
{
    int          dtih1,          /* D/240SC-T1 #1 T-1 time slot handle */
                dtih2;          /* D/240SC-T1 #2 T-1 time slot handle */

    /* Open T-1 time slot 12 on D/240SC-T1 #1. */
    if ((dtih1 = dt_open("dtiB1T12", 0)) == -1) {
        printf("Cannot open T-1 time slot dtiB1T12.  errno = %d", errno);
        exit(1);
    }

    /* Open T-1 time slot 23 on D/240SC-T1 #2. */
    if ((dtih2 = dt_open("dtiB2T23", 0)) == -1) {
        printf("Cannot open T-1 time slot dtiB2T23.  errno = %d", errno);
        exit(1);
    }

    .
    . Continue processing.
    .

    /***** SCbus routing example. *****/
    /*
     * Make a full duplex connection between T-1 time slot 12 (on the
     * first D/240SC-T1) and T-1 time slot 23 (on the second D/240SC-T1).
     */
    if (nr_scroute(dtih1, SC_DTI, dtih2, SC_DTI, SC_FULLDUP) == -1) {
        printf("Could not route the two T-1 time slots.");
        exit(1);
    }

    .
    . Continue processing.
    .
}
```

SCbus Routing Guide

5.1.3. MS-DOS Sample Code for SCbus Convenience Functions

```
#include <stdlib.h>
#include <stdio.h>
#include "d40.h"
#include "dti.h"
#include "d40lib.h"
#include "dtilib.h"
#include "dtierr.h"
#include "vfcons.h"

/*
 * In order for the nr_ functions to have DTI functionality, DTISC must
 * be defined when compiling the sctools.c file.
 */
#include "sctools.h"

int main()
{
    int          dti_bdl = 1, /* D/240SC-T1 #1 network brd device handle. */
               dti_ts1 = 12, /* T-1 time slot number. */
               dti_bd2 = 2, /* D/240SC-T1 #2 network brd device handle. */
               dti_ts2 = 23, /* T-1 time slot number. */
               rcode;        /* To hold function error return codes. */
    unsigned int sc_IRQ = 10, /* SCbus board interrupt (from DIALOGIC.CFG) */
               num_chans,     /* Location for number of voice channels */
               num_dti_brd,   /* Location for number of DTI boards */
               num_dti_ts;    /* Location for number of DTI time slots */

    /* returned by startsys(). */
    /* returned by startdti(). */
    /* returned by startdti(). */

    /***** Initialize system. *****/
    /* Make sure the voice driver is loaded. */
    if (getvctr() == 0) {
        printf("Voice driver is not loaded.");
        exit(1);
    }

    /* Make sure the DTI driver is stopped before starting. */
    stopdti();

    /* Make sure the voice driver is stopped before starting. */
    stopsys();

    /* Start the voice driver. */
    if ((rcode = startsys(sc_IRQ, SM_EVENT, 0, 0, &num_chans)) != E_SUCC) {
        printf("Could not start voice driver: Error message = %s", d4xerr(rcode));
        exit(1);
    }

    /* Start the DTI driver. */
    if ((rcode = startdti(&num_dti_brd, &num_dti_ts)) != E_DTSUCC) {
        printf("Could not start DTI driver: Error = %d", rcode);
        exit(1);
    }

    .
    .
    .
    Continue processing.
```

5. SCbus Digital Service Inbound/Outbound Call Routing Example

```
/*
***** SCbus routing example. *****/
/*
 * Make a full duplex connection between T-1 time slot 12 (on the
 * first D/240SC-T1) and T-1 time slot 23 (on the second D/240SC-T1).
 */
if ((rcode = nr_scroute(DT_MKTSLOT(dti_bd1, dti_ts1) , SC_DTI,
                        DT_MKTSLOT(dti_bd2, dti_ts2) , SC_DTI,
                        SC_FULLDUP)) != 0) {
    printf("Could not route the T-1 time slots: Error = %d", rcode);
    exit(1);
}

.
. Continue processing.
.
}
```

SCbus Routing Guide

5.1.4. OS/2 Sample Code for SCbus Convenience Functions

```
#define INCL_DOSERRORS
#include <os2.h>
#include <stdlib.h>
#include <stdio.h>
#include <voxlib.h>
#include <dtilib.h>
#include <dti2lib.h>

/*
 * For nr_ function error printing and DTI functionality, make sure
 * PRINTON and DTISC are defined when compiling the sctools.c file.
 */
#include <sctools.h>

main()
{
    int          dtih1,          /* D/240SC-T1 #1  T-1 time slot handle */
               dtih2;          /* D/240SC-T1 #2  T-1 time slot handle */

    /* Open T-1 time slot 12 on D/240SC-T1 #1. */
    if ((dtih1 = dt_open("DTIB0T11", 0)) == -1) {
        printf("Cannot open T-1 time slot DTIB0T11.  dl_errno = %d", dl_errno);
        exit(1);
    }

    /* Open T-1 time slot 23 on D/240SC-T1 #2. */
    if ((dtih2 = dt_open("DTIB1T22", 0)) == -1) {
        printf("Cannot open T-1 time slot DTIB1T22.  dl_errno = %d", dl_errno);
        exit(1);
    }

    .
    . Continue processing.
    .

    /***** SCbus routing example. *****/
    /*
     * Make a full duplex connection between T-1 time slot 12 (on the
     * first D/240SC-T1) and T-1 time slot 23 (on the second D/240SC-T1).
     */
    if (nr_scroute(dtih1, SC_DTI, dtih2, SC_DTI, SC_FULLDUP) == -1) {
        printf("Could not route the two T-1 time slots.");
        exit(1);
    }

    .
    . Continue processing.
    .
}
```

5.2. Routing Using Individual SCbus Routing Functions

To route the channels using individual SCbus routing functions, perform the following, see *Figure 5*:

5. SCbus Digital Service Inbound/Outbound Call Routing Example

1. Initialize the SC_TSINFO structure with the required information.
2. Get the SC_TSINFO structure that contains the number of the SCbus time slot connected to digital transmit channel 23 of the second D/240SC-T1 board by issuing a **dt_getxmitslot()** call.
3. Connect digital listen channel 12 of the first D/240SC-T1 board to digital transmit channel 23 of the second D/240SC-T1 board by issuing a **dt_listen()** call. This function uses as input the information contained in the SC_TSINFO structure returned by the **dt_getxmitslot()** call of step 2.
4. Get the SC_TSINFO structure that contains the number of the SCbus time slot connected to digital transmit channel 12 of the first D/240SC-T1 board by issuing a **dt_getxmitslot()** call.
5. Connect digital receive channel 23 of the second D/240SC-T1 board to digital transmit channel 12 of the first D/240SC-T1 board by issuing a **dt_listen()** call. This function uses as input the information contained in the SC_TSINFO structure returned by the **dt_getxmitslot()** call of step 4.

When these functions return, full duplex communications between the two digital channels will be established.

5.2.1. Windows Sample Code for Individual SCbus Routing Functions

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

main()
{
    int          dtih1,          /* D/240SC-T1 #1 T-1 time slot handle */
                dtih2;          /* D/240SC-T1 #2 T-1 time slot handle */
    SC_TSINFO    sc_tsinfo;      /* time slot information structure */
    long         scts;           /* SCbus time slot */

    /* Open T-1 time slot 12 on D/240SC-T1 #1. */
    if ((dtih1 = dt_open("dtiB1T12", 0)) == -1) {
        printf("Cannot open T-1 time slot dtiB1T12.   errno = %d", errno);
        exit(1);
    }

    /* Open T-1 time slot 23 on D/240SC-T1 #2. */
    if ((dtih2 = dt_open("dtiB2T23", 0)) == -1) {
        printf("Cannot open T-1 time slot dtiB2T23.   errno = %d", errno);
        exit(1);
    }

    .
    .
    . Continue processing.
```

SCbus Routing Guide

```
.
.
/***** SCbus routing example. *****/
/* Initialize the SC_TSINFO structure with the necessary information. */
sc_tsinfo.sc_numts = 1;
sc_tsinfo.sc_tsarrayp = &scts;

/*
 * Get the SCbus time slot on which the 12th T-1 time slot (of the first
 * D/240SC-T1) is transmitting.
 */
if (dt_getxmitslot(dtih1, &sc_tsinfo) == -1) {
    printf("dt_getxmitslot() failed: Error message = %s", ATDV_ERRMSGP(dtih1));
    exit(1);
}

/*
 * Make T-1 time slot 23 (of the second D/240SC-T1) listen to the
 * SCbus time slot that T-1 time slot 12 (on the first D/240SC-T1)
 * is transmitting on.
 */
if (dt_listen(dtih2, &sc_tsinfo) == -1) {
    printf("dt_listen() failed: Error message = %s", ATDV_ERRMSGP(dtih2));
    exit(1);
}

/*
 * Get the SCbus time slot on which the 23rd T-1 time slot (of the
 * second D/240SC-T1) is transmitting.
 */
if (dt_getxmitslot(dtih2, &sc_tsinfo) == -1) {
    printf("dt_getxmitslot() failed: Error message = %s", ATDV_ERRMSGP(dtih2));
    exit(1);
}

/*
 * Make T-1 time slot 12 (of the first D/240SC-T1) listen to the
 * SCbus time slot that T-1 time slot 23 (on the second D/240SC-T1)
 * is transmitting on.
 */
if (dt_listen(dtih1, &sc_tsinfo) == -1) {
    printf("dt_listen() failed: Error message = %s", ATDV_ERRMSGP(dtih1));
    exit(1);
}

.
.
. Continue processing.
.
}
```

5.2.2. Linux Sample Code for Individual SCbus Routing Functions

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

main()
{
    int          dtih1,          /* D/240SC-T1 #1 T-1 time slot handle */
                dtih2;          /* D/240SC-T1 #2 T-1 time slot handle */
    SC_TSINFO    sc_tsinfo;      /* time slot information structure */
```


5. SCbus Digital Service Inbound/Outbound Call Routing Example

```

long      scts;          /* SCbus time slot          */

/* Open T-1 time slot 12 on D/240SC-T1 #1. */
if ((dtih1 = dt_open("dtiB1T12", 0)) == -1) {
    printf("Cannot open T-1 time slot dtiB1T12.  errno = %d", errno);
    exit(1);
}

/* Open T-1 time slot 23 on D/240SC-T1 #2. */
if ((dtih2 = dt_open("dtiB2T23", 0)) == -1) {
    printf("Cannot open T-1 time slot dtiB2T23.  errno = %d", errno);
    exit(1);
}

.
. Continue processing.
.

/***** SCbus routing example. *****/
/* Initialize the SC_TSINFO structure with the necessary information. */
sc_tsinfo.sc_nmnts = 1;
sc_tsinfo.sc_tsarrayp = &scts;

/*
 * Get the SCbus time slot on which the 12th T-1 time slot (of the first
 * D/240SC-T1) is transmitting.
 */
if (dt_getxmitslot(dtih1, &sc_tsinfo) == -1) {
    printf("dt_getxmitslot() failed: Error message = %s", ATDV_ERRMSGP(dtih1));
    exit(1);
}

/*
 * Make T-1 time slot 23 (of the second D/240SC-T1) listen to the
 * SCbus time slot that T-1 time slot 12 (on the first D/240SC-T1)
 * is transmitting on.
 */
if (dt_listen(dtih2, &sc_tsinfo) == -1) {
    printf("dt_listen() failed: Error message = %s", ATDV_ERRMSGP(dtih2));
    exit(1);
}

/*
 * Get the SCbus time slot on which the 23rd T-1 time slot (of the
 * second D/240SC-T1) is transmitting.
 */
if (dt_getxmitslot(dtih2, &sc_tsinfo) == -1) {
    printf("dt_getxmitslot() failed: Error message = %s", ATDV_ERRMSGP(dtih2));
    exit(1);
}

/*
 * Make T-1 time slot 12 (of the first D/240SC-T1) listen to the
 * SCbus time slot that T-1 time slot 23 (on the second D/240SC-T1)
 * is transmitting on.
 */
if (dt_listen(dtih1, &sc_tsinfo) == -1) {
    printf("dt_listen() failed: Error message = %s", ATDV_ERRMSGP(dtih1));
    exit(1);
}

.

```

SCbus Routing Guide

```
    . Continue processing.  
    .  
}
```

5.2.3. MS-DOS Sample Code for Individual SCbus Routing Functions

```
#include <stdlib.h>  
#include <stdio.h>  
#include "d40.h"  
#include "dti.h"  
#include "d40lib.h"  
#include "dtilib.h"  
#include "dtierr.h"  
#include "vfcns.h"  
  
int main()  
{  
    int          dti_bdl = 1, /* D/240SC-T1 #1 network brd device handle. */  
                dti_tsl = 12, /* T-1 time slot number. */  
                dti_bdl2 = 2, /* D/240SC-T1 #2 network brd device handle. */  
                dti_tsl2 = 23, /* T-1 time slot number. */  
                rcode; /* To hold function error return codes. */  
    unsigned int sc_irq = 10, /* SCbus board interrupt (from DIALOGIC.CFG) */  
                num_chans, /* Location for number of voice channels */  
                /* returned by startsys(). */  
                num_dti_brd, /* Location for number of DTI boards */  
                /* returned by startdti(). */  
                num_dti_ts; /* Location for number of DTI time slots */  
                /* returned by startdti(). */  
    SC_TSINFO    sc_tsinfo; /* time slot information structure */  
    long          scts; /* SCbus time slot */  
  
    /***** Initialize system. *****/  
    /* Make sure the voice driver is loaded. */  
    if (getvctr() == 0) {  
        printf("Voice driver is not loaded.");  
        exit(1);  
    }  
  
    /* Make sure the DTI driver is stopped before starting. */  
    stopdti();  
  
    /* Make sure the voice driver is stopped before starting. */  
    stopsys();  
  
    /* Start the voice driver. */  
    if ((rcode = startsys(sc_irq, SM_EVENT, 0, 0, &num_chans)) != E_SUCC) {  
        printf("Could not start voice driver: Error message = %s", d4xerr(rcode));  
        exit(1);  
    }  
  
    /* Start the DTI driver. */  
    if ((rcode = startdti(&num_dti_brd, &num_dti_ts)) != E_DTSUCC) {  
        printf("Could not start DTI driver: Error = %d", rcode);  
        exit(1);  
    }  
  
    .  
    . Continue processing.  
    .  
}
```

5. SCbus Digital Service Inbound/Outbound Call Routing Example

```
/****** SCbus routing example. *****/
/* Initialize the SC_TSINFO structure with the necessary information. */
sc_tsinfo.sc_numts = 1;
sc_tsinfo.sc_tsarrayp = &scts;

/*
 * Get the SCbus time slot on which the 12th T-1 time slot (of the first
 * D/240SC-T1) is transmitting.
 */
if ((rcode = dt_getxmitslot(DT_MKTSLOT(dti_bd1, dti_ts1), &sc_tsinfo)) != E_DTSUCC) {
    printf("dt_getxmitslot() failed: Error = %d", rcode);
    exit(1);
}

/*
 * Make T-1 time slot 23 (of the second D/240SC-T1) listen to the
 * SCbus time slot that T-1 time slot 12 (on the first D/240SC-T1)
 * is transmitting on.
 */
if ((rcode = dt_listen(DT_MKTSLOT(dti_bd2, dti_ts2), &sc_tsinfo)) != E_DTSUCC) {
    printf("dt_listen() failed: Error = %d", rcode);
    exit(1);
}

/*
 * Get the SCbus time slot on which the 23rd T-1 time slot (of the
 * second D/240SC-T1) is transmitting.
 */
if ((rcode = dt_getxmitslot(DT_MKTSLOT(dti_bd2, dti_ts2), &sc_tsinfo)) != E_DTSUCC) {
    printf("dt_getxmitslot() failed: Error = %d", rcode);
    exit(1);
}

/*
 * Make T-1 time slot 12 (of the first D/240SC-T1) listen to the
 * SCbus time slot that T-1 time slot 23 (on the second D/240SC-T1)
 * is transmitting on.
 */
if ((rcode = dt_listen(DT_MKTSLOT(dti_bd1, dti_ts1), &sc_tsinfo)) != E_DTSUCC) {
    printf("dt_listen() failed: Error = %d", rcode);
    exit(1);
}

.
.   Continue processing.
.
}
```

5.2.4. OS/2 Sample Code for Individual SCbus Routing Functions

```
#define INCL_DOSERRORS
#include <os2.h>
#include <stdlib.h>
#include <stdio.h>
#include <voxlib.h>
#include <dtlib.h>
#include <dti2lib.h>

main()
{
    int          dtih1,          /* D/240SC-T1 #1 T-1 time slot handle */

```

SCbus Routing Guide

```
        dtih2;          /* D/240SC-T1 #2 T-1 time slot handle */
SC_TSINFO sc_tsinfo;    /* time slot information structure */
long      scts;         /* SCbus time slot */

/* Open T-1 time slot 12 on D/240SC-T1 #1. */
if ((dtih1 = dt_open("DTIB0T11", 0)) == -1) {
    printf("Cannot open T-1 time slot DTIB0T11. dl_errno = %d", dl_errno);
    exit(1);
}

/* Open T-1 time slot 23 on D/240SC-T1 #2. */
if ((dtih2 = dt_open("DTIB1T22", 0)) == -1) {
    printf("Cannot open T-1 time slot DTIB1T22. dl_errno = %d", dl_errno);
    exit(1);
}

.
. Continue processing.
.

/***** SCbus routing example. *****/
/* Initialize the SC_TSINFO structure with the necessary information. */
sc_tsinfo.sc_numts = 1;
sc_tsinfo.sc_tsarrayp = &scts;

/*
 * Get the SCbus time slot on which the 12th T-1 time slot (of the first
 * D/240SC-T1) is transmitting.
 */
if (dt_getxmitslot(dtih1, &sc_tsinfo) == -1) {
    printf("dt_getxmitslot() failed: Error message = %s", ATDT_ERRMSGP(dtih1));
    exit(1);
}

/*
 * Make T-1 time slot 23 (of the second D/240SC-T1) listen to the
 * SCbus time slot that T-1 time slot 12 (on the first D/240SC-T1)
 * is transmitting on.
 */
if (dt_listen(dtih2, &sc_tsinfo) == -1) {
    printf("dt_listen() failed: Error message = %s", ATDT_ERRMSGP(dtih2));
    exit(1);
}

/*
 * Get the SCbus time slot on which the 23rd T-1 time slot (of the
 * second D/240SC-T1) is transmitting.
 */
if (dt_getxmitslot(dtih2, &sc_tsinfo) == -1) {
    printf("dt_getxmitslot() failed: Error message = %s", ATDT_ERRMSGP(dtih2));
    exit(1);
}

/*
 * Make T-1 time slot 12 (of the first D/240SC-T1) listen to the
 * SCbus time slot that T-1 time slot 23 (on the second D/240SC-T1)
 * is transmitting on.
 */
if (dt_listen(dtih1, &sc_tsinfo) == -1) {
    printf("dt_listen() failed: Error message = %s", ATDT_ERRMSGP(dtih1));
    exit(1);
}
```

5. SCbus Digital Service Inbound/Outbound Call Routing Example

```
.  
. Continue processing.  
.  
}
```

SCbus Routing Guide

6. SCbus Fax Sharing Example

Fax Sharing Example Overview

In this example, a fax call received by the T-1 device on the first D/240SC-T1 board is connected to the fax device on the VFX/40ESC board. To make this connection, full duplex communications must be established between digital channel 12 on the first D/240SC-T1 board and fax channel 2 on the VFX/40ESC board. See *Figure 6* for the SCbus routing connections established at system initialization.

For fax sharing applications, the VFX/40ESC board is setup in resource mode. In this mode, the analog devices on this board are disabled.

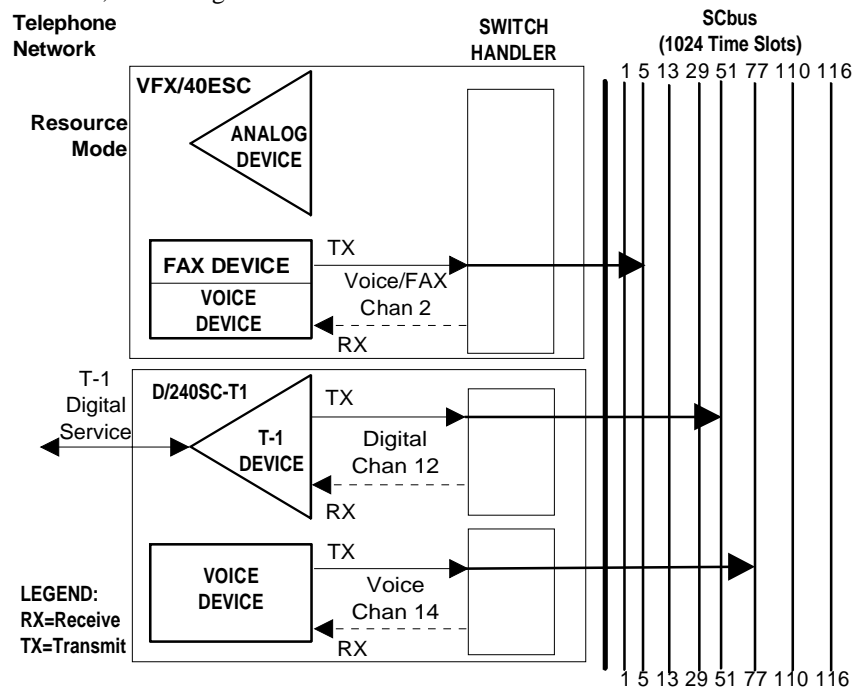


Figure 6. Initial Fax & Digital Channel Routing Example

SCbus Routing Guide

6.1. Routing Using SCbus Convenience Functions

In this example, we will:

- connect digital channel 12 to fax channel 2,
- process the fax call, disconnect the fax channel and then
- connect digital channel 12 to voice channel 14 for call processing.

To route these channels using SCbus convenience functions, perform the following:

- Make the fax channel and the digital channel listen to each other by issuing a **nr_scroute()** call, see *Figure 6* and *Figure 7*. This function automatically:
 - connects receive fax channel 2 on the VFX/40ESC board to digital transmit channel 12 (via SCbus time slot 109) of the D/240SC-T1 board and
 - connects digital receive channel 12 on the D/240SC-T1 board to fax transmit channel 2 (via SCbus time slot 5) on the VFX/40ESC board.
- When the function returns, digital channel 12 and fax channel 2 will be listening to each other. Thus full duplex communications are established.

NOTE: Any resource that was previously listening to digital channel 12 will continue to listen to this channel unless explicitly disconnected.

To disconnect digital channel 12 from fax channel 2, disconnect the listen channel of each device from the SCbus by issuing **nr_scunroute()** calls. When the functions return, digital channel 12 and fax channel 2 will be disconnected from the SCbus.

To connect voice channel 14 on the D/240SC-T1 board to digital channel 12, perform the following:

- Connect voice listen channel 14 to digital transmit channel 12 by issuing a **nr_scroute()** call. When the function returns, the voice channel and the digital channel will be listening to each other and full duplex communications will be established.
- Sample code for each operating system illustrating SCbus routing for this example using the SCbus convenience functions and also using the individual SCbus routing functions is contained in the following paragraphs.

6. SCbus Fax Sharing Example

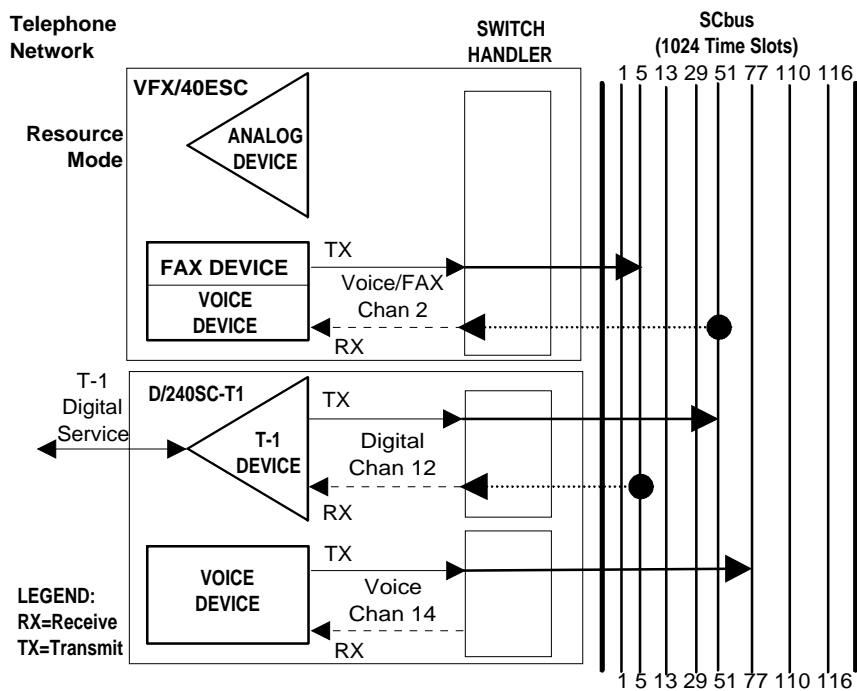


Figure 7. Fax Sharing Example

SCbus Routing Guide

6.1.1. Windows Sample Code for SCbus Convenience Functions

```
#include <windows.h>
#include <errno.h>
#include <srllib.h>
#include <dxlib.h>
#include <dtilib.h>
#include <faxlib.h>

/*
 * For nr_ function error printing, DTI and fax functionality, make sure
 * PRINTON, DTISC and FAXSC are defined when compiling the sctools.c file.
 */
#include "sctools.h"

main()
{
    int          dtih,          /* T-1 time slot handle */
              faxh,            /* Fax channel handle */
              chdev;           /* Voice channel handle */

    /* Open T-1 time slot 12 on the first D/240SC-T1. */
    if ((dtih = dt_open("dtiB1T12", 0)) == -1) {
        printf("Cannot open T-1 time slot dtiB1T12.  errno = %d", errno);
        exit(1);
    }

    /* Open fax channel 2 on board 5 (fax channel 2 on the VFX/40ESC). */
    if ((faxh = fx_open("dxxxB5C2", 0)) == -1) {
        printf("Cannot open fax channel dxxxB5C2.  errno = %d", errno);
        exit(1);
    }

    /* Open voice channel 2 on board 9 (voice channel 14 on the first D/240SC-T1). */
    if ((chdev = dx_open("dxxxB9C2", NULL)) == -1) {
        printf("Cannot open voice channel dxxxB9C2.  errno = %d", errno);
        exit(1);
    }

    .
    . Continue processing.
    .

    /***** SCbus routing example. *****/
    /*
     * Make a full duplex connection between T-1 time slot 12
     * and fax channel 2.
     */
    if (nr_scroute(dtih, SC_DTI, faxh, SC_FAX, SC_FULLDUP) == -1) {
        printf("Could not route the T-1 time slot to the fax channel");
        exit(1);
    }

    .
    . Process fax call.
    .

    /***** Disconnect fax channel. *****/
    /*
     * Break the full duplex connection between T-1 time slot 12
     * and fax channel 2.
     */
}
```

6. SCbus Fax Sharing Example

```
    */
    if (nr_scuroute(dtih, SC_DTI, faxh, SC_FAX, SC_FULLDUP) == -1) {
        printf("Could not disconnect the T-1 time slot from the fax channel");
        exit(1);
    }

    /***** Re-connect T-1 time slot 12 for voice processing *****/
    /***** using voice channel 14. *****/
    /*
    * Make a full duplex connection between T-1 time slot 12
    * and voice channel 14.
    */
    if (nr_scroute(dtih, SC_DTI, chdev, SC_VOX, SC_FULLDUP) == -1) {
        printf("Could not route the T-1 time slot to the voice channel");
        exit(1);
    }

    .
    . Continue processing.
    .
}
```

6.1.2. Linux Sample Code for SCbus Convenience Functions

```
#include <srllib.h>
#include <dxlib.h>
#include <dtilib.h>
#include <faxlib.h>
#include <errno.h>

/*
 * For nr_ function error printing, DTI and fax functionality, make sure
 * PRINTON, DTISC and FAXSC are defined when compiling the sctools.c file.
 */
#include "sctools.h"

main()
{
    int          dtih,          /* T-1 time slot handle */
               faxh,          /* Fax channel handle */
               chdev;         /* Voice channel handle */

    /* Open T-1 time slot 12 on the first D/240SC-T1. */
    if ((dtih = dt_open("dtiB1T12", 0)) == -1) {
        printf("Cannot open T-1 time slot dtiB1T12.  errno = %d", errno);
        exit(1);
    }

    /* Open fax channel 2 on board 5 (fax channel 2 on the VFX/40ESC). */
    if ((faxh = fx_open("dxxxB5C2", 0)) == -1) {
        printf("Cannot open fax channel dxxxB5C2.  errno = %d", errno);
        exit(1);
    }

    /* Open voice channel 2 on board 9 (voice channel 14 on the first D/240SC-T1). */
    if ((chdev = dx_open("dxxxB9C2", NULL)) == -1) {
        printf("Cannot open voice channel dxxxB9C2.  errno = %d", errno);
        exit(1);
    }

    .
}
```

SCbus Routing Guide

```
. Continue processing.
.

/***** SCbus routing example. *****/
/*
 * Make a full duplex connection between T-1 time slot 12
 * and fax channel 2.
 */
if (nr_scroute(dtih, SC_DTI, faxh, SC_FAX, SC_FULLDUP) == -1) {
    printf("Could not route the T-1 time slot to the fax channel");
    exit(1);
}

.
. Process fax call.
.

/***** Disconnect fax channel. *****/
/*
 * Break the full duplex connection between T-1 time slot 12
 * and fax channel 2.
 */
if (nr_scroute(dtih, SC_DTI, faxh, SC_FAX, SC_FULLDUP) == -1) {
    printf("Could not disconnect the T-1 time slot from the fax channel");
    exit(1);
}

/***** Re-connect T-1 time slot 12 for voice processing *****/
/***** using voice channel 14. *****/
/*
 * Make a full duplex connection between T-1 time slot 12
 * and voice channel 14.
 */
if (nr_scroute(dtih, SC_DTI, chdev, SC_VOX, SC_FULLDUP) == -1) {
    printf("Could not route the T-1 time slot to the voice channel");
    exit(1);
}

.
. Continue processing.
.
}
```

6.1.3. MS-DOS Sample Code for SCbus Convenience Functions

```
#include <stdlib.h>
#include <stdio.h>
#include "d40.h"
#include "dti.h"
#include "fax.h"
#include "d40lib.h"
#include "dtilib.h"
#include "dtierr.h"
#include "faxlib.h"
#include "vfcns.h"

/*
 * In order for the nr_ functions to have DTI and fax functionality,
 * DTISC and FAXSC must be defined when compiling the sctools.c file.
 */
#include "sctools.h"
```

6. SCbus Fax Sharing Example

```
int main()
{
    int          dti_bd = 1, /* D/240SC-T1 network board device handle. */
                dti_ts = 12, /* T-1 time slot number. */
                fax_ch = 2, /* Fax channel. */
                chdev = 14, /* Voice channel. */
                rcode; /* To hold function error return codes. */
    unsigned int sc_IRQ = 10, /* SCbus board interrupt (from DIALOGIC.CFG) */
                num_chans, /* Location for number of voice channels */
                /* returned by startsys(). */
                num_dti_brd, /* Location for number of DTI boards */
                /* returned by startdti(). */
                num_dti_ts, /* Location for number of DTI time slots */
                /* returned by startdti(). */
                num_fax_brd, /* Location for number of fax boards */
                /* returned by fx_startsys(). */
                num_fax_chan; /* Location for number of fax channels */
                                /* returned by fx_startsys(). */

    /***** Initialize system. *****/
    /* Make sure the voice driver is loaded. */
    if (getvctr() == 0) {
        printf("Voice driver is not loaded.");
        exit(1);
    }

    /* Make sure the fax driver is loaded. */
    if (fx_isdrvact() == 0) {
        printf("Fax driver is not loaded.");
        exit(1);
    }

    /* Make sure the fax driver is stopped before starting. */
    fx_stopsys();

    /* Make sure the DTI driver is stopped before starting. */
    stopdti();

    /* Make sure the voice driver is stopped before starting. */
    stopsys();

    /* Start the voice driver. */
    if ((rcode = startsys(sc_IRQ, SM_EVENT, 0, 0, &num_chans)) != E_SUCC) {
        printf("Could not start voice driver: Error message = %s", d4xerr(rcode));
        exit(1);
    }

    /* Start the DTI driver. */
    if ((rcode = startdti(&num_dti_brd, &num_dti_ts)) != E_DTSUCC) {
        printf("Could not start DTI driver: Error = %d", rcode);
        exit(1);
    }

    /* Start the fax driver. */
    if ((rcode = fx_startsys(&num_fax_brd, &num_fax_chan)) != EFX_SUCC) {
        printf("Could not start fax driver: Error message = %s", fx_faxerr(rcode));
        exit(1);
    }

    .
    .
    . Continue processing.
}
```

SCbus Routing Guide

```
.
/***** SCbus routing example. *****/
/*
 * Make a full duplex connection between T-1 time slot 12
 * and fax channel 2.
 */
if ((rcode = nr_scroute(DT_MKTSLOT(dti_bd, dti_ts), SC_DTI,
                        fax_ch, SC_FAX,
                        SC_FULLDUP)) != 0) {
    printf("Could not route the T-1 time slot to the fax channel: Error = %d", rcode);
    exit(1);
}

.
. Process fax call.
.

/***** Disconnect fax channel. *****/
/*
 * Break the full duplex connection between T-1 time slot 12
 * and fax channel 2.
 */
if ((rcode = nr_scunroute(DT_MKTSLOT(dti_bd, dti_ts), SC_DTI,
                           fax_ch, SC_FAX,
                           SC_FULLDUP)) != 0) {
    printf("Could not disconnect T-1 time slot from fax channel: Error = %d", rcode);
    exit(1);
}

/***** Re-connect T-1 time slot 12 for voice processing *****/
/***** using voice channel 14. *****/
/*
 * Make a full duplex connection between T-1 time slot 12
 * and voice channel 14.
 */
if ((rcode = nr_scroute(DT_MKTSLOT(dti_bd, dti_ts), SC_DTI,
                           chdev, SC_VOX,
                           SC_FULLDUP)) != 0) {
    printf("Could not route the T-1 time slot to the voice channel: Error = %d",
rcode);
    exit(1);
}

.
. Continue processing.
.
}
```

6. SCbus Fax Sharing Example

6.1.4. OS/2 Sample Code for SCbus Convenience Functions

```
#define INCL_DOSERRORS
#include <os2.h>
#include <stdlib.h>
#include <stdio.h>
#include <fcntl.h>
#include <voxlib.h>
#include <dtilib.h>
#include <dti2lib.h>
#include <faxlib.h>

/*
 * For nr_ function error printing, DTI and fax functionality, make sure
 * PRINTON, DTISC and FAXSC are defined when compiling the sctools.c file.
 */
#include <sctools.h>

main()
{
    int          dtih,          /* T-1 time slot handle */
              faxh,          /* Fax channel handle */
              chdev;          /* Voice channel handle */

    /* Open T-1 time slot 12 on the first D/240SC-T1. */
    if ((dtih = dt_open("DTIB0T11", 0)) == -1) {
        printf("Cannot open T-1 time slot DTIB0T11. dl_errno = %d", dl_errno);
        exit(1);
    }

    /* Open fax channel 2 on board 5 (fax channel 2 on the VFX/40ESC). */
    if ((faxh = fx_open("VOXB4C1", O_RDWR)) == -1) {
        printf("Cannot open fax channel VOXB4C1. dl_errno = %d", dl_errno);
        exit(1);
    }

    /* Open voice channel 2 on board 9 (voice channel 14 on the first D/240SC-T1). */
    if ((chdev = dl_open("VOXB8C1", O_RDWR)) == -1) {
        printf("Cannot open voice channel VOXB8C1. dl_errno = %d", dl_errno);
        exit(1);
    }

    .
    . Continue processing.
    .

    /***** SCbus routing example. *****/
    /*
     * Make a full duplex connection between T-1 time slot 12
     * and fax channel 2.
     */
    if (nr_scroute(dtih, SC_DTI, faxh, SC_FAX, SC_FULLDUP) == -1) {
        printf("Could not route the T-1 time slot to the fax channel");
        exit(1);
    }

    .
    . Process fax call.
    .

    /***** Disconnect fax channel. *****/
}
```

SCbus Routing Guide

```
/*
 * Break the full duplex connection between T-1 time slot 12
 * and fax channel 2.
 */
if (nr_scuroute(dti, SC_DTI, faxh, SC_FAX, SC_FULLDUP) == -1) {
    printf("Could not disconnect T-1 time slot from fax channel");
    exit(1);
}

/***** Connect T-1 time slot 12 for voice processing *****/
/***** using voice channel 14. *****/
/*
 * Make a full duplex connection between T-1 time slot 12
 * and voice channel 14.
 */
if (nr_scroute(dti, SC_DTI, chdev, SC_VOX, SC_FULLDUP) == -1) {
    printf("Could not route the T-1 time slot to the voice channel");
    exit(1);
}

.
. Continue processing.
.
}
```


6. SCbus Fax Sharing Example

6.2. Routing Using Individual SCbus Routing Functions

To route the channels using individual SCbus routing functions, perform the following, see *Figure 7*:

1. Initialize the SC_TSINFO structure with the required information.
2. Get the SC_TSINFO structure that contains the number of the SCbus time slot connected to digital transmit channel 12 on the D/240SC-T1 board by issuing a **dt_getxmitslot()** call.
3. Connect fax receive channel 2 of the VFX/40ESC board to digital transmit channel 12 on the D/240SC-T1 board by issuing a **fx_listen()** call. This function uses as input the information contained in the SC_TSINFO structure returned by the **dt_getxmitslot()** call of step 2.
4. Get the SC_TSINFO structure that contains the number of the SCbus time slot connected to fax transmit channel 2 of the VFX/40ESC board by issuing a **fx_getxmitslot()** call.
5. Connect digital receive channel 12 on the D/240SC-T1 board to fax transmit channel 2 of the VFX/40ESC board by issuing an **dt_listen()** call. This function uses as input the information contained in the SC_TSINFO structure returned by the **fx_getxmitslot()** call of step 5.

When these functions return, full duplex communications between the fax and digital channels will be established.

SCbus Routing Guide

6.2.1. Win NT Sample Code for Individual SCbus Routing Functions

```
#include <windows.h>
#include <errno.h>
#include <srllib.h>
#include <dxlib.h>
#include <dtilib.h>
#include <faxlib.h>

main()
{
    int          dtih,          /* T-1 time slot handle          */
              faxh,          /* Fax channel handle            */
              chdev;          /* Voice channel handle          */
    SC_TSINFO    sc_tsinfo;    /* time slot information structure */
    long         scts;         /* SCbus time slot               */

    /* Open T-1 time slot 12 on the first D/240SC-T1. */
    if ((dtih = dt_open("dtiB1T12", 0)) == -1) {
        printf("Cannot open T-1 time slot dtiB1T12.  errno = %d", errno);
        exit(1);
    }

    /* Open fax channel 2 on board 5 (fax channel 2 on the VFX/40ESC). */
    if ((faxh = fx_open("dxxxB5C2", 0)) == -1) {
        printf("Cannot open fax channel dxxxB5C2.  errno = %d", errno);
        exit(1);
    }

    /* Open voice channel 2 on board 9 (voice channel 14 on the first D/240SC-T1). */
    if ((chdev = dx_open("dxxxB9C2", NULL)) == -1) {
        printf("Cannot open voice channel dxxxB9C2.  errno = %d", errno);
        exit(1);
    }

    .
    . Continue processing.
    .

    /***** SCbus routing example. *****/
    /* Initialize the SC_TSINFO structure with the necessary information. */
    sc_tsinfo.sc_numts = 1;
    sc_tsinfo.sc_tsarrayp = &scts;

    /*
     * Get the SCbus time slot on which T-1 time slot 12 is transmitting.
     */
    if (dt_getxmitslot(dtih, &sc_tsinfo) == -1) {
        printf("dt_getxmitslot() failed: Error message = %s", AITDV_ERRMSGP(dtih));
        exit(1);
    }

    /*
     * Make fax channel 2 listen to the SCbus time slot that T-1 time slot
     * 12 is transmitting on.
     */
    if (fx_listen(faxh, &sc_tsinfo) == -1) {
        printf("fx_listen() failed: Error message = %s", AITDV_ERRMSGP(faxh));
        exit(1);
    }
}
```

6. SCbus Fax Sharing Example

```
/*
 * Get the SCbus time slot on which fax channel 2 is transmitting.
 */
if (fx_getxmitslot(faxh, &sc_tsinfo) == -1) {
    printf("fx_getxmitslot() failed: Error message = %s", ATDV_ERRMSGP(faxh));
    exit(1);
}

/*
 * Make T-1 time slot 12 listen to the SCbus time slot that
 * fax channel 2 is transmitting on.
 */
if (dt_listen(dtih, &sc_tsinfo) == -1) {
    printf("dt_listen() failed: Error message = %s", ATDV_ERRMSGP(dtih));
    exit(1);
}

.
. Process fax call.
.

/***** Disconnect fax channel. *****/
/*
 * Make fax channel 2 stop listening to the SCbus time slot on which
 * T-1 time slot 12 is transmitting.
 */
if (fx_unlisten(faxh) == -1) {
    printf("fx_unlisten() failed: Error message = %s", ATDV_ERRMSGP(faxh));
    exit(1);
}

/***** Connect T-1 time slot 12 for voice processing *****/
/***** using voice channel 14. *****/
/* Initialize the SC_TSINFO structure with the necessary information. */
sc_tsinfo.sc_numts = 1;
sc_tsinfo.sc_tsarray = &scts;

/*
 * Get the SCbus time slot on which T-1 time slot 12 is transmitting.
 */
if (dt_getxmitslot(dtih, &sc_tsinfo) == -1) {
    printf("dt_getxmitslot() failed: Error message = %s", ATDV_ERRMSGP(dtih));
    exit(1);
}

/*
 * Make voice channel 14 listen to the SCbus time slot that T-1 time slot
 * 12 is transmitting on.
 */
if (dx_listen(chdev, &sc_tsinfo) == -1) {
    printf("dx_listen() failed: Error message = %s", ATDV_ERRMSGP(chdev));
    exit(1);
}

/*
 * Get the SCbus time slot on which voice channel 14 is transmitting.
 */
if (dx_getxmitslot(chdev, &sc_tsinfo) == -1) {
    printf("dx_getxmitslot() failed: Error message = %s", ATDV_ERRMSGP(chdev));
    exit(1);
}
```

SCbus Routing Guide

```
/*
 * Make T-1 time slot 12 listen to the SCbus time slot that
 * voice channel 14 is transmitting on.
 */
if (dt_listen(dtih, &sc_tsinfo) == -1) {
    printf("dt_listen() failed: Error message = %s", ATDV_ERRMSGP(dtih));
    exit(1);
}

.
. Continue processing.
.
}
```

6.2.2. Linux Sample Code for Individual SCbus Routing Functions

```
#include <srllib.h>
#include <dxxlib.h>
#include <dtilib.h>
#include <faxlib.h>
#include <errno.h>

main()
{
    int          dtih,          /* T-1 time slot handle          */
              faxh,          /* Fax channel handle            */
              chdev;         /* Voice channel handle          */
    SC_TSINFO    sc_tsinfo;    /* time slot information structure */
    long         scts;         /* SCbus time slot               */

    /* Open T-1 time slot 12 on the first D/240SC-T1. */
    if ((dtih = dt_open("dtiB1T12", 0)) == -1) {
        printf("Cannot open T-1 time slot dtiB1T12.  errno = %d", errno);
        exit(1);
    }

    /* Open fax channel 2 on board 5 (fax channel 2 on the VFX/40ESC). */
    if ((faxh = fx_open("dxxxB5C2", 0)) == -1) {
        printf("Cannot open fax channel dxxxB5C2.  errno = %d", errno);
        exit(1);
    }

    /* Open voice channel 2 on board 9 (voice channel 14 on the first D/240SC-T1). */
    if ((chdev = dx_open("dxxxB9C2", NULL)) == -1) {
        printf("Cannot open voice channel dxxxB9C2.  errno = %d", errno);
        exit(1);
    }

    .
    . Continue processing.
    .

    /***** SCbus routing example. *****/
    /* Initialize the SC_TSINFO structure with the necessary information. */
    sc_tsinfo.sc_numts = 1;
    sc_tsinfo.sc_tsarrayp = &scts;

    /*
     * Get the SCbus time slot on which T-1 time slot 12 is transmitting.
     */
    if (dt_getxmitslot(dtih, &sc_tsinfo) == -1) {
```

6. SCbus Fax Sharing Example

```
    printf("dt_getxmitslot() failed: Error message = %s", ATDV_ERRMSGP(dtih));
    exit(1);
}

/*
 * Make fax channel 2 listen to the SCbus time slot that T-1 time slot
 * 12 is transmitting on.
 */
if (fx_listen(faxh, &sc_tsinfo) == -1) {
    printf("fx_listen() failed: Error message = %s", ATDV_ERRMSGP(faxh));
    exit(1);
}

/*
 * Get the SCbus time slot on which fax channel 2 is transmitting.
 */
if (fx_getxmitslot(faxh, &sc_tsinfo) == -1) {
    printf("fx_getxmitslot() failed: Error message = %s", ATDV_ERRMSGP(faxh));
    exit(1);
}

/*
 * Make T-1 time slot 12 listen to the SCbus time slot that
 * fax channel 2 is transmitting on.
 */
if (dt_listen(dtih, &sc_tsinfo) == -1) {
    printf("dt_listen() failed: Error message = %s", ATDV_ERRMSGP(dtih));
    exit(1);
}

.
.   Process fax call.
.

/***** Disconnect fax channel. *****/
/*
 * Make fax channel 2 stop listening to the SCbus time slot on which
 * T-1 time slot 12 is transmitting.
 */
if (fx_unlisten(faxh) == -1) {
    printf("fx_unlisten() failed: Error message = %s", ATDV_ERRMSGP(faxh));
    exit(1);
}

/***** Connect T-1 time slot 12 for voice processing *****/
/***** using voice channel 14. *****/
/* Initialize the SC_TSINFO structure with the necessary information. */
sc_tsinfo.sc_numts = 1;
sc_tsinfo.sc_tsarrayp = &scts;

/*
 * Get the SCbus time slot on which T-1 time slot 12 is transmitting.
 */
if (dt_getxmitslot(dtih, &sc_tsinfo) == -1) {
    printf("dt_getxmitslot() failed: Error message = %s", ATDV_ERRMSGP(dtih));
    exit(1);
}

/*
 * Make voice channel 14 listen to the SCbus time slot that T-1 time slot
 * 12 is transmitting on.
 */
```

SCbus Routing Guide

```
if (dx_listen(chdev, &sc_tsinfo) == -1) {
    printf("dx_listen() failed: Error message = %s", ATDV_ERRMSGP(chdev));
    exit(1);
}

/*
 * Get the SCbus time slot on which voice channel 14 is transmitting.
 */
if (dx_getxmitslot(chdev, &sc_tsinfo) == -1) {
    printf("dx_getxmitslot() failed: Error message = %s", ATDV_ERRMSGP(chdev));
    exit(1);
}

/*
 * Make T-1 time slot 12 listen to the SCbus time slot that
 * voice channel 14 is transmitting on.
 */
if (dt_listen(dtih, &sc_tsinfo) == -1) {
    printf("dt_listen() failed: Error message = %s", ATDV_ERRMSGP(dtih));
    exit(1);
}

.
. Continue processing.
.
}
```

6.2.3. MS-DOS Sample Code for Individual SCbus Routing Functions

```
#include <stdlib.h>
#include <stdio.h>
#include "d40.h"
#include "dti.h"
#include "fax.h"
#include "d40lib.h"
#include "dtilib.h"
#include "dtierr.h"
#include "faxlib.h"
#include "vfcns.h"

int main()
{
    int          dti_bd = 1, /* D/240SC-T1 network board device handle. */
               dti_ts = 12, /* T-1 time slot number. */
               fax_ch = 2,  /* Fax channel. */
               chdev = 14,  /* Voice channel. */
               rcode;        /* To hold function error return codes. */
    unsigned int sc_IRQ = 10, /* SCbus board interrupt (from DIALOGIC.CFG) */
               num_chans,    /* Location for number of voice channels */
               /* returned by startsys(). */
               num_dti_brd,  /* Location for number of DTI boards */
               /* returned by startdti(). */
               num_dti_ts,  /* Location for number of DTI time slots */
               /* returned by startdti(). */
               num_fax_brd, /* Location for number of fax boards */
               /* returned by fx_startsys(). */
               num_fax_chan; /* Location for number of fax channels */
               /* returned by fx_startsys(). */
    SC_TSINFO    sc_tsinfo; /* Time slot information structure. */
    long         scts;      /* SCbus time slot */
}
```

6. SCbus Fax Sharing Example

```
/***** Initialize system. *****/
/* Make sure the voice driver is loaded. */
if (getvctr() == 0) {
    printf("Voice driver is not loaded.");
    exit(1);
}

/* Make sure the fax driver is loaded. */
if (fx_isdrvact() == 0) {
    printf("fax driver is not loaded.");
    exit(1);
}

/* Make sure the fax driver is stopped before starting. */
fx_stopsys();

/* Make sure the DTI driver is stopped before starting. */
stopdti();

/* Make sure the voice driver is stopped before starting. */
stopsys();

/* Start the voice driver. */
if ((rcode = startsys(sc_IRQ, SM_EVENT, 0, 0, &num_chans)) != E_SUCC) {
    printf("Could not start voice driver: Error message = %s", d4xerr(rcode));
    exit(1);
}

/* Start the DTI driver. */
if ((rcode = startdti(&num_dti_brd, &num_dti_ts)) != E_DTSUCC) {
    printf("Could not start DTI driver: Error = %d", rcode);
    exit(1);
}

/* Start the fax driver. */
if ((rcode = fx_startsys(&num_fax_brd, &num_fax_chan)) != EFX_SUCC) {
    printf("Could not start fax driver: Error message = %s", fx_faxerr(rcode));
    exit(1);
}

.
. Continue processing.
.

/***** SCbus routing example. *****/
/* Initialize the SC_TSINFO structure with the necessary information. */
sc_tsinfo.sc_numts = 1;
sc_tsinfo.sc_tsarrayp = &scts;

/*
 * Get the SCbus time slot on which T-1 time slot 12 is transmitting.
 */
if ((rcode = dt_getxmitslot(DT_MKTSLOT(dti_bd, dti_ts), &sc_tsinfo)) != E_DTSUCC) {
    printf("dt_getxmitslot() failed: Error = %d", rcode);
    exit(1);
}

/*
 * Make fax channel 2 listen to the SCbus time slot that T-1 time slot
 * 12 is transmitting on.
 */
if ((rcode = fx_listen(fax_ch, &sc_tsinfo)) != EFX_SUCC) {
```

SCbus Routing Guide

```
    printf("fx_listen() failed: Error message = %s", fx_faxerr(rcode));
    exit(1);
}

/*
 * Get the SCbus time slot on which fax channel 2 is transmitting.
 */
if ((rcode = fx_getxmitslot(fax_ch, &sc_tsinfo)) != EFX_SUCC) {
    printf("fx_getxmitslot() failed: Error message = %s", fx_faxerr(rcode));
    exit(1);
}

/*
 * Make T-1 time slot 12 listen to the SCbus time slot that
 * fax channel 2 is transmitting on.
 */
if ((rcode = dt_listen(DT_MKTSLOT(dti_bd, dti_ts), &sc_tsinfo)) != E_DTSUCC) {
    printf("dt_listen() failed: Error = %d", rcode);
    exit(1);
}

.
. Process fax call.
.

/***** Disconnect fax channel. *****/
/*
 * Make fax channel 2 stop listening to the SCbus time slot on which
 * T-1 time slot 12 is transmitting.
 */
if ((rcode = fx_unlisten(fax_ch)) != EFX_SUCC) {
    printf("fx_unlisten() failed: Error message = %s", fx_faxerr(rcode));
    exit(1);
}

/***** Connect T-1 time slot 12 for voice processing *****/
/***** using voice channel 14. *****/
/* Initialize the SC_TSINFO structure with the necessary information. */
sc_tsinfo.sc_numts = 1;
sc_tsinfo.sc_tsarrayp = &scts;

/*
 * Get the SCbus time slot on which T-1 time slot 12 is transmitting.
 */
if ((rcode = dt_getxmitslot(DT_MKTSLOT(dti_bd, dti_ts), &sc_tsinfo)) != E_DTSUCC) {
    printf("dt_getxmitslot() failed: Error = %d", rcode);
    exit(1);
}

/*
 * Make voice channel 14 listen to the SCbus time slot that T-1 time slot
 * 12 is transmitting on.
 */
if ((rcode = dl_listen(chdev, &sc_tsinfo)) != E_SUCC) {
    printf("dl_listen() failed: Error message = %s", d4xerr(rcode));
    exit(1);
}

/*
 * Get the SCbus time slot on which voice channel 14 is transmitting.
 */
if ((rcode = dl_getxmitslot(chdev, &sc_tsinfo)) != E_SUCC) {
```


6. SCbus Fax Sharing Example

```
        printf("dl_getxmitslot() failed: Error message = %s", d4xerr(rcode));
        exit(1);
    }

    /*
     * Make T-1 time slot 12 listen to the SCbus time slot that
     * voice channel 14 is transmitting on.
     */
    if ((rcode = dt_listen(DT_MKTSLOT(dti_bd, dti_ts), &sc_tsinfo)) != E_DTSUCC) {
        printf("dt_listen() failed: Error = %d", rcode);
        exit(1);
    }

    .
    .   Continue processing.
    .
}
```

6.2.4. OS/2 Sample Code for Individual SCbus Routing Functions

```
#define INCL_DOSERRORS
#include <os2.h>
#include <stdlib.h>
#include <stdio.h>
#include <fcntl.h>
#include <voxlib.h>
#include <dtlib.h>
#include <dti2lib.h>
#include <faxlib.h>

main()
{
    int          dtih,          /* T-1 time slot handle          */
               faxh,          /* fax channel handle           */
               chdev;         /* Voice channel handle         */
    SC_TSINFO    sc_tsinfo;    /* time slot information structure */
    long         scts;         /* SCbus time slot              */

    /* Open T-1 time slot 12 on the first D/240SC-T1. */
    if ((dtih = dt_open("DTIB0T11", 0)) == -1) {
        printf("Cannot open T-1 time slot DTIB0T11. dl_errno = %d", dl_errno);
        exit(1);
    }

    /* Open fax channel 2 on board 5 (fax channel 2 on the VFX/40ESC). */
    if ((faxh = fx_open("VOXB4C1", O_RDWR)) == -1) {
        printf("Cannot open fax channel VOXB4C1. dl_errno = %d", dl_errno);
        exit(1);
    }

    /* Open voice channel 2 on board 9 (voice channel 14 on the first D/240SC-T1). */
    if ((chdev = dl_open("VOXB8C1", O_RDWR)) == -1) {
        printf("Cannot open voice channel VOXB8C1. dl_errno = %d", dl_errno);
        exit(1);
    }

    .
    .   Continue processing.
    .

    /***** SCbus routing example. *****/
}
```

SCbus Routing Guide

```
/* Initialize the SC_TSINFO structure with the necessary information. */
sc_tsinfo.sc_numts = 1;
sc_tsinfo.sc_tsarrayp = &scts;

/*
 * Get the SCbus time slot on which T-1 time slot 12 is transmitting.
 */
if (dt_getxmitslot(dtih, &sc_tsinfo) == -1) {
    printf("dt_getxmitslot() failed: Error message = %s", ATDT_ERRMSGP(dtih));
    exit(1);
}

/*
 * Make fax channel 2 listen to the SCbus time slot that T-1 time slot
 * 12 is transmitting on.
 */
if (fx_listen(faxh, &sc_tsinfo) == -1) {
    printf("fx_listen() failed: dl_errno = %d", dl_errno);
    exit(1);
}

/*
 * Get the SCbus time slot on which fax channel 2 is transmitting.
 */
if (fx_getxmitslot(faxh, &sc_tsinfo) == -1) {
    printf("fx_getxmitslot() failed: dl_errno = %d", dl_errno);
    exit(1);
}

/*
 * Make T-1 time slot 12 listen to the SCbus time slot that
 * fax channel 2 is transmitting on.
 */
if (dt_listen(dtih, &sc_tsinfo) == -1) {
    printf("dt_listen() failed: Error message = %s", ATDT_ERRMSGP(dtih));
    exit(1);
}

.
. Process fax call.
.

/***** Disconnect fax channel. *****/
/*
 * Make fax channel 2 stop listening to the SCbus time slot on which
 * T-1 time slot 12 is transmitting.
 */
if (fx_unlisten(faxh) == -1) {
    printf("fx_unlisten() failed: Error = %d", dl_errno);
    exit(1);
}

/***** Connect T-1 time slot 12 for voice processing *****/
/***** using voice channel 14. *****/
/* Initialize the SC_TSINFO structure with the necessary information. */
sc_tsinfo.sc_numts = 1;
sc_tsinfo.sc_tsarrayp = &scts;

/*
 * Get the SCbus time slot on which T-1 time slot 12 is transmitting.
 */
if (dt_getxmitslot(dtih, &sc_tsinfo) == -1) {
```

6. SCbus Fax Sharing Example

```
        printf("dt_getxmitslot() failed: Error message = %s", ATDT_ERRMSGP(dtih));
        exit(1);
    }

    /*
     * Make voice channel 14 listen to the SCbus time slot that T-1 time slot
     * 12 is transmitting on.
     */
    if (dx_listen(chdev, &sc_tsinfo) == -1) {
        printf("dx_listen() failed: dl_errno = %d", dl_errno);
        exit(1);
    }

    /*
     * Get the SCbus time slot on which voice channel 14 is transmitting.
     */
    if (dx_getxmitslot(chdev, &sc_tsinfo) == -1) {
        printf("dx_getxmitslot() failed: dl_errno = %d", dl_errno);
        exit(1);
    }

    /*
     * Make T-1 time slot 12 listen to the SCbus time slot that
     * voice channel 14 is transmitting on.
     */
    if (dt_listen(dtih, &sc_tsinfo) == -1) {
        printf("dt_listen() failed: Error message = %s", ATDT_ERRMSGP(dtih));
        exit(1);
    }

    .
    . Continue processing.
    .
}
```


7. SCbus Special Considerations

Special considerations to keep in mind when routing SCbus devices are described in this chapter.

SCbus Transmit Channel Assignment Overview

At system initialization, each transmit channel is assigned to a specific and unique SCbus time slot. This transmit channel assignment cannot be changed by the application. This ensures that only one device transmits on an SCbus time slot at a time.

7.1. Voice Boards with On-board Analog Loop-Start Interfaces

For voice boards with on-board analog devices (D/41ESC, D/160SC-LS and VFX/40ESC boards), a voice device and an analog device comprise a single channel. Although these devices can be addressed separately, all analog signaling is processed by the associated voice device; analog signaling (ring detection and loop current detection) events are not transmitted over the SCbus.

For example, a call received by an analog channel on a D/160SC-LS board can be routed to a fax channel on a VFX/40ESC board or to another resource device.

However, the fax channel or other resource device will not receive any analog signaling events. Signaling events will only be received by the D/160SC-LS voice channel associated with that analog channel. Voice devices on analog loop-start interface boards and their associated analog devices can be used separately, but should be used mutually exclusively.

Because of the analog signaling between on-board voice and analog devices, at system initialization default SCbus routing connects corresponding analog and voice device pairs (analog device 1 to voice device 1, etc.) in full duplex communications. These receive (listen) channel assignments can be changed by the application at any time.

When a D/41ESC or a VFX/40ESC board is configured as a resource device, the analog devices on that board will be disabled. This configuration is typically used for fax or voice sharing applications.

7.2. Connecting SCbus Devices

Devices communicate via the SCbus by listening to each other. Devices receive data by listening to an SCbus time slot on which another SCbus device transmits. The receive channel of each device can only be connected to one SCbus time slot at a time. However, any number of devices can listen to the same SCbus time slot.

7.3. SCbus Time Slot Information

Although SCbus time slot numbers are used in this Guide to illustrate SCbus routing, **actual SCbus time slot numbers are never used nor needed when coding SCbus routing applications**. An SC_TSINFO structure contains the SCbus time slot information needed. The SCbus routing functions use board and channel designations (or time slot handles for digital devices) to identify the transmit channel and return the SCbus time slot information in an SC_TSINFO structure. This information is then used to connect a receive (listen) channel to this SCbus time slot.

Appendix A

SCbus Routing Function Summary

SCbus Routing Function Listing

ag_getctinfo()	returns information about analog device
ag_getxmitslot()	returns SCbus time slot information into an SC_TSINFO structure that includes the number of the SCbus time slot connected to the analog transmit channel
ag_listen()	connects analog receive (listen) channel to an SCbus time slot
ag_unlisten()	disconnects analog receive (listen) channel from SCbus time slot
dl_getctinfo()	returns information about voice device
dl_getxmitslot()	returns SCbus time slot information into an SC_TSINFO structure that includes the number of the SCbus time slot connected to the voice transmit channel
dl_listen()	connects voice receive (listen) channel to an SCbus time slot
dl_unlisten()	disconnects voice receive (listen) channel from SCbus time slot
dt_getctinfo()	returns information about digital interface device
dt_getxmitslot()	returns SCbus time slot information into an SC_TSINFO structure that includes the number of the SCbus time slot connected to the digital transmit channel
dt_listen()	connects digital receive (listen) channel to an SCbus time slot
dt_unlisten()	disconnects digital receive (listen) channel from SCbus time slot
dx_getctinfo()	returns information about voice device
dx_getxmitslot()	returns SCbus time slot information into an SC_TSINFO structure that includes the number of the SCbus time slot connected to the voice transmit channel
dx_listen()	connects voice receive (listen) channel to an SCbus time slot
dx_unlisten()	disconnects voice receive (listen) channel from SCbus time

SCbus Routing Guide

	slot
fx_getxmitslot()	returns SCbus time slot information into an SC_TSINFO structure that includes the number of the SCbus time slot connected to the fax transmit channel
fx_listen()	connects fax receive (listen) channel to an SCbus time slot
fx_unlisten()	disconnects fax receive (listen) channel from SCbus time slot
ms_getctinfo()	returns information about a MSI station
ms_getxmitslot()	returns SCbus time slot information into an SC_TSINFO structure that includes the number of the SCbus time slot connected to the MSI station
ms_listen()	connects a MSI station to an SCbus time slot
ms_unlisten()	disconnects a MSI station from SCbus time slot
nr_scroute()	makes half or full duplex connection between two SCbus devices
nr_scunroute()	breaks half or full duplex connection between two SCbus devices

Appendix B

Related Publications

This appendix lists publications you should refer to for additional information on Dialogic products and SCbus technology.

- *Digital Network Interface Software Reference* for your operating system
- *Fax Software Reference* for your operating system
- *ISDN Software Reference*
- *MSI/SC Software Reference*
- *SCbus Configuration Planning Guide* for your operating system
- *SCbus Routing Software Reference* for your operating system
- *SCbus Routing Software Reference* for your operating system
- *Voice Software Reference – Features Guide* for your operating system
- *Voice Software Reference – Programmer's Guide* for your operating system
- *Voice Software Reference – Standard Runtime Library* for your operating system
- *Voice Software Installation Reference* for your operating system

SCbus Routing Guide

Glossary

The following terms are used in the context of defining relationships between devices and processes communicating via the SCbus:

Analog channel Designates a bi-directional transfer of data for a single analog loop-start call between an analog device that received the call and the SCbus. Digitized voice or data from the analog device is sent to the SCbus over the analog transmit channel. The response to this call is sent from the SCbus to the analog device over the analog receive (listen) channel.

Analog device Interfaces to an analog loop-start line from the telephone network, receives and converts an incoming call into digitized voice or data, sends this digitized information over the analog transmit channel to an SCbus time slot for transmission to a voice device or other resource for processing. Also receives the processed data via the SCbus, converts this data into an analog voice or analog data signal and then sends the response to the telephone network.

Digital channel Designates a bi-directional transfer of data for a single time slot of a T-1 or an E-1 digital frame between a T-1/E-1 device that connects to the digital service and the SCbus. Digitized information from the T-1/E-1 device is sent to the SCbus over the digital transmit channel. The response to this call is sent from the SCbus to the T-1/E-1 device over the digital receive (listen) channel.

Fax channel Designates a bi-directional transfer of data for a single fax between a fax device processing that fax and the SCbus. Digitized fax from an analog or T-1/E-1 interface device is transmitted over the SCbus to the fax receive (listen) channel for processing by the fax device. The fax device sends signaling or faxes over the fax transmit channel to an SCbus time slot that transmits this information to the analog or T-1/E-1 interface device.

Fax device Processes faxes received or to be transmitted. Incoming fax data is sent from the SCbus over the fax receive channel to the fax device. Fax signaling or fax transmissions are sent by the fax device over the fax transmit channel to the SCbus, then to an analog or T-1/E-1 device that interfaces to the telephone network.

SCbus Routing Guide

- Listen** Accepting or taking digitized information transmitted by another device. Indicates that a device is monitoring an SCbus time slot.
- Loop-start** Indicates a form of signaling associated with the transmission of analog voice signals from the telephone network.
- Receive** Accepting or taking digitized information transmitted by another device.
- SCbus** Signal Computing bus. A hardwired connection between Switch Handlers (SC2000 chips) on SCbus-based products for transmitting information over 1024 time slots to all devices connected to the SCbus.
- SCbus routing functions** Setup communications between devices connected to the SCbus. These functions enable an application to connect or disconnect (make or break) the receive (listen) channel of a device to or from an SCbus time slot.
- SCbus time slot** Indicates one segment of time out of 1024 segments during which data is broadcast by an SCbus Switch Handler to all other Switch Handlers connected to the SCbus. Functionally, each segment or time slot can be thought of as a connection between the channel transmitting on an SCbus time slot and all channels listening to that SCbus time slot.
- Switch Handler** The hardware (SC2000 chip) and firmware that: (1) takes digitized information from on-board devices (transmit channels) and broadcasts this information on their assigned SCbus time slots, (2) takes information received (listen) from an SCbus time slot and sends it to the assigned receive (listen) channel of the on-board device, and (3) disconnects the receive (listen) channel of an on-board device from one SCbus time slot and connects it to another SCbus time slot under application control.
- T-1/E-1 device** Interfaces to the T-1 (24 time slots/24 voice channels) or E-1 (32 time slots/30 voice channels) digital service provided by the telephone network, converts the incoming digitized data into separate digital channels (digital time slots), sends this digitized data over the transmit digital channel to an SCbus time slot for transmission to a voice device or other resource for processing. Also receives processed data via the SCbus, converts and injects this data into a T-1 or E-1 time slot and sends the entire frame to the telephone network.

Glossary

Transmit Sending or broadcasting of digitized information by a device.

Voice channel Designates a bi-directional transfer of data for a single call between a voice device processing that call and the SCbus. Digitized voice from the analog or T-1/E-1 interface device is transmitted over the SCbus to the voice receive (listen) channel for processing by the voice device. The voice device sends the response to the call over the voice transmit channel to an SCbus time slot that transmits this response to the analog or T-1/E-1 interface device.

Voice device Performs call processing on voice calls received by either an analog interface device or a T-1/E-1 interface device and transmitted to the voice device via the SCbus. Incoming data is sent from the SCbus over the voice receive channel to the voice device. Responses are sent by the voice device over the voice transmit channel to the SCbus, then to the analog or T-1/E-1 device that interfaces to the telephone network.

SCbus Routing Guide

Index

A

- analog call, 15
- analog channel, 69
- analog device, 6, 15
- analog devices, 69
- analog signaling, 69
- assigned SCbus time slot, 13
- associated voice device, 15

C

- communications, 5
- communications bus, 1
- connected device, 2
- connecting, 5
- connecting devices using individual SCbus routing functions, 12
- connecting devices using SCbus convenience functions, 11
- convenience functions
 - analog routing using, 15
 - digital routing using, 31
 - fax routing using, 48
 - SCbus, 5
 - using, 7

D

- D/160SC-LS board, 1, 13, 69
 - example, 11
- D/240SC-T1 board, 1, 13, 47
 - example, 11

- D/300SC-E1 board, 1
- D/41ESC board, 1
- default routing, 15
- devices
 - connecting using individual SCbus routing functions, 12
- digital channel, 31
- digital device, 6
- digital transmit channel, 31
- disconnect (unlisten), 5
- disconnect devices, 7, 8, 11, 12

E

- E-1 digital service interface, 6
- examples
 - fax sharing, 47
 - SCbus call routing, 31
 - SCbus routing, 11

F

- fax call, 47
- fax channel, 48, 69
- fax channels, 13
- fax device, 6, 47
- fax sharing, 47

I

- individual SCbus routing functions, 6, 22, 38, 57
- initialization, 15
 - digital device example, 31

- fax sharing example, 47
- SCbus system, 12

L

- listen channel, 1, 8
- listening, 70
- loop current detection, 69

M

- monitor, 16
- moving (rerouting), 5
- moving the receive (listen) channel, 8

N

- network/resource (nr_) SCbus
convenience functions, 5
- number of devices, 12

O

- on-board analog devices, 69

P

- pre-assigned, 1
- prefix of the individual SCbus routing
functions, 6

R

- receive (RX)(listen) channel., 1
- receive channel, 70
- receive data, 70
- record, 16
- rerouting (switching), 5
- resource device, 69
- resource mode, 47

- ring detection, 69
- route channels, 7
- route channels using individual SCbus
routing functions, 7, 38, 57
- route channels using SCbus
convenience functions, 48
- routing, 1, 5
- routing concepts, 5
- routing connections, 47
- routing functions
 - individual, 6
 - overview, 5
 - using, 22
 - using individual, 38, 57
- routing SCbus devices, 69
- RX (listen) channel, 1

S

- sample code
 - SCbus call routing, 32
 - SCbus routing, convenience
functions, 18
 - SCbus routing, fax sharing, 48
- SC_TSINFO structure, 12, 70
 - initializing, 22, 39, 57
- SCbus, 1
- SCbus convenience functions, 5
 - analog routing using, 15
 - digital routing using, 31
 - fax routing using, 48
 - using, 7
- SCbus routing concepts, 5
- SCbus routing connections, 47
- SCbus routing functions
 - individual, 6

- overview, 5
- using, 22
- using individual, 38, 57

SCbus time slot, 69

SCbus time slot connections, 31

SCbus time slot information, 70

SCbus time slot numbers, 11, 70

special considerations, 69

suffix of the individual SCbus routing functions, 7

Switch Handler, 13

system initialization, 15

- digital device example, 31
- fax sharing example, 47
- SCbus, 12

T

T-1 device, 31

T-1 digital service interface, 6

T-1 time slot, 31

time slot, 69

time slot connections, 31

time slot information, 70

time slot numbers, 11, 70

transmit (TX) channel, 1

transmit assignments, 12

transmit channel, 12, 69, 70

transmit channel assignment, 69

transmit channel assignments, 13

TX channel, 1

U

unlisten, 5

using convenience functions, 31

using individual SCbus routing functions, 38, 57

using SCbus convenience functions, 31, 48

V

VFX/40ESC board, 1, 13, 47, 69

- example, 11

voice boards with on-board analog devices, 69

voice device, 6, 15

voice receive channel, 16

